MECHANISMS OF REACTIONS AT SOLID-LIQUID INTERFACES

A thesis submitted for the Degree of

Doctor of Philosophy

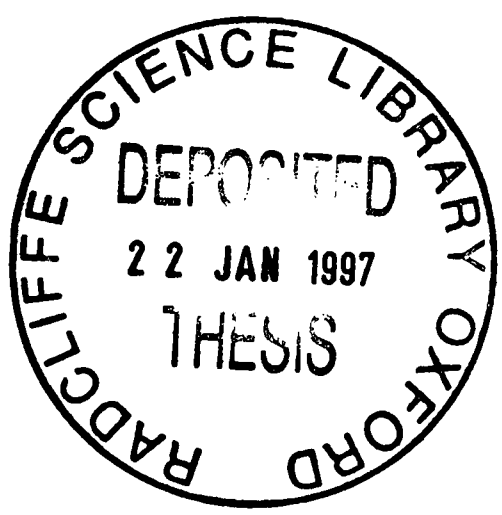
in the University of Oxford

by

Kin Yip Tam

St. John's College

Trinity Term 1996



MECHANISMS OF REACTIONS AT SOLID-LIQUID INTERFACES

Abstract of a thesis submitted for the degree of Doctor of Philosophy

Kin Yip Tam
St. John's College, Oxford
Trinity Term 1996

The work described in this thesis applies the channel flow cell methodology to the study of the mechanism of the hydrolytic dissolution of triphenylmethyl chloride (TPMCl), the kinetics of dimerisation of the methyl viologen radical cation ($MV^{+\cdot}$) and the reactive dyeing processes. New detection techniques based on AC impedance spectroscopy and ultraviolet-visible spectroscopy have been developed in order to facilitate investigation of these processes.

The hydrolytic dissolution of TPMCl in aqueous solution was shown to proceed via a direct heterogeneous process, whereby the reaction occurs at the solid-liquid interface rather than in solution after the prior dissolution of TPMCl. Potentiometric and AC impedance detection techniques were implemented to interrogate the dissolution of the particulate TPMCl samples under a wide range of salt concentrations. Kinetic parameters, including the effect of ionic strength, were derived. Moreover, the potentiometric technique was employed to examine the hydrolytic processes occurring at specific TPMCl crystal planes which had been indexed unambiguously using a genetic algorithm-based program. It was demonstrated that the rate of the interfacial reaction is dependent upon the availability of exposed chlorine atoms in the reacting crystal plane. In situ atomic force microscope (AFM) studies indicated that the TPMCl surface, once in contact with water, is covered rapidly by a product layer which, however, has negligible effect on the reaction rate because of its high porosity.

The development of a channel flow cell specifically for ultraviolet-visible spectroelectrochemical experiments is also described. The performance of this cell was characterised using the one-electron oxidation of N,N,N′,N′-tetramethyl-p-phenylenediamine in aqueous electrolytes. Excellent agreement between the evolved theory and the experimental results was noted. This method was then utilised to study the dimerisation rate of electrochemically generated $MV^{+\cdot}$ and the rate constant is quantified for the first time. The aforementioned spectroelectrochemical channel cell was then adopted to scrutinise the reactive dyeing kinetics on a cotton fabric. Kinetic results showed that the dye adsorption to the fabric is controlled by a solid-liquid interfacial reaction which is first order with respect to the surface concentration of the dye. However, the rate of this process is governed by the availability of the reaction sites for adsorption of dye molecules onto the fabric surface. It was demonstrated that the presence of supporting electrolyte in high pH media, and mercerisation pretreatment of the fabric, are essential to increase the dye uptake rate. Ex situ AFM studies suggested that mercerisation leads to a disordered fibre surface which may be responsible for the enhanced dye absorption rate.

Acknowledgements

I would like to thank Prof. R.G. Compton for his supervision of this project. Thanks are due to Dr. J.H. Atherton for particle size distribution and scanning electron microscopy measurements, Dr. C.M. Brennan for advice on dyeing kinetics studies, Dr. B.A. Coles for his help in SPICE programming, Dr. R. Docherty for the crystal plane modelling using CERIUS, Mr. A. Green for his excellent workmanship in constructing the spectroelectrochemical channel cell, Dr. C.K. Prout for help with x-ray diffraction experiments on triphenylmethyl chloride crystals and Mr. G.H.W. Sanders for his help on using the TopoMetrix liquid cell.

I wish to acknowledge the technical advice and helpful discussions provided by Drs. V.K.W. Cheng and X.Y. Liu on crystal growth, Drs. B. Cheung and K.F.C. Yiu on combinatorial optimisation and Dr. S.C. Tsang on surface chemistry. I am grateful to Prof. F.T. Chau for a copy of SCANGRAPH program (*Comput. Appl. Biosci.*, **1993**, *9*, 531) to digitise all the UV-Visible absorption spectra reported in this thesis.

Contents

Chapter 1

Introduction

1. Background 1
2. Chemical systems studied in this thesis 1
3. Reactions at solid-liquid interfaces 2
4. Techniques for the investigation of solid-liquid reactions 8
5. Summary of the work presented in this thesis 15
6. References 17

Chapter 2

Theory

1. Mass transport 21
2. Convective-diffusion within a channel flow cell 24
3. Solution of the convective-diffusion equations for a channel flow cell 27
4. Numerical simulation 38
5. References 43

Chapter 3

Experimental

1. The channel flow system 45
2. Chloride detection 47
3. Atomic force microscopy 48
4. Ultraviolet-visible spectroscopy 48
5. AC impedance spectroscopy 49
6. Voltammetry experiments 49
7. Computing 49
8. Chemicals used 49
9. References 50

Chapter 4

A Channel Flow Cell with Downstream AC Impedance Detection

1. Introduction 51

2. Flow cell design and impedance measurements 52

3. Background theory and the use of SPICE 55

4. Triphenylmethyl chloride (TPMCl) dissolution experiments: experimental 71

5. Triphenylmethyl chloride (TPMCl) dissolution experiments: modelling of experimental data 74

6. References 82

Chapter 5

GAMATCH - A Genetic Algorithm-Based Program for Indexing Crystal Planes

1. Introduction 84

2. Method of calculation 86

3. Results 94

4. Conclusion 102

5. References 104

Chapter 6

The Hydrolytic Dissolution of Solid Triphenylmethyl Chloride in Aqueous Solution

1. Introduction 106

2. Experimental 107

3. Results and discussion 112

4. Conclusions 138

5. References 139

Chapter 7

The Kinetics of Dimerisation of the Methyl Viologen Radical Cation

1. Introduction 141

2. Experimental 145

3. Modelling of transport within the flow cell 147

4. Results and discussion 153

5. Concluding remarks 161

6. References 161

Chapter 8

Applications of the Channel Flow Cell in Kinetic Studies of Reactive Dyeing Processes

1. Introduction 165
2. Experimental 169
3. Modelling of the dyeing process 172
4. Results and discussion 175
5. Concluding remarks 196
6. References 197

Appendices

Appendix One: The Thomas Algorithm 199
Appendix Two: Computer programs 202

Chapter 1

Introduction

1. Background

Reactions at the solid-liquid interface are of ubiquitous synthetic, industrial and environmental importance. They typically involve a complex sequence of mass transport, adsorption/desorption phenomena, surface diffusion, heterogeneous reaction/electron transfer as well as chemical transformations of intermediates. Recent work[1-4] has introduced the channel flow cell (CFC) method in which a well-defined hydrodynamic flow links the reacting interface to an appropriate detector so as to provide a sensitive and quantitative assessment of the relevant interfacial processes. Based on proposed reaction models, the detector response as a function of mass transport can be calculated precisely and compared with the experimental values to gain mechanistic insight. In conjunction with the CFC kinetic measurements, atomic force microscope (AFM) imaging of the reacting solid, either in situ or ex situ, enables the structure of the reacting surface to be scrutinised at a near-atomic level. The kinetic and structural information thus obtained provides a rigorous basis for the deduction of the mechanisms of reactions at the solid-liquid interface. This forms the general theme of the work presented in this thesis.

2. Chemical systems studied in this thesis

Three distinctive chemical systems have been interrogated. All of them are of fundamental

scientific and applied industrial interest and are listed as follows:

A. The hydrolytic dissolution of solid triphenylmethyl chloride in aqueous solution.

B. The kinetics of dimerisation of the methyl viologen radical cation.

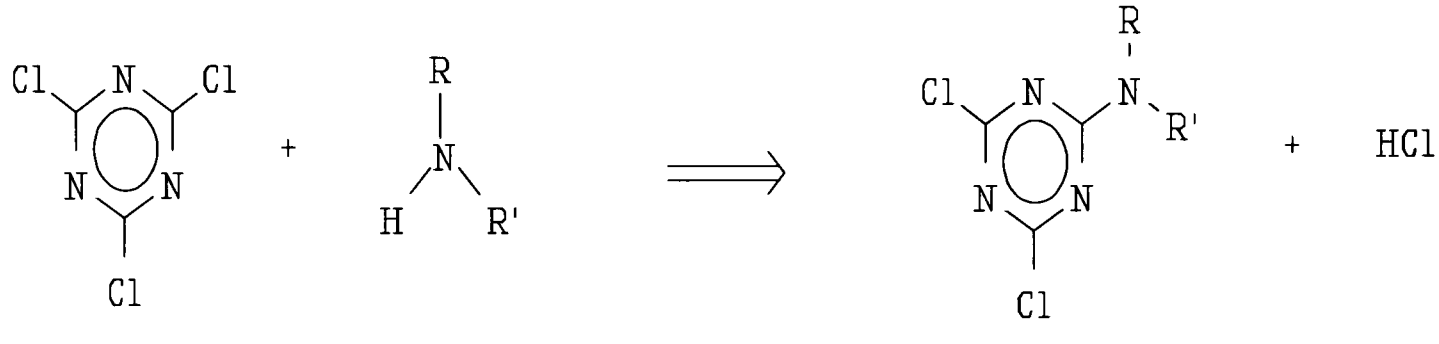
C. Kinetic studies of reactive dyeing processes.

In addition, new detection techniques based on ac impedance spectroscopy, ultraviolet-visible spectroelectrochemistry and in situ or ex situ AFM imaging have been established to facilitate investigation of these processes. In the subsequent discussion, an overview of the mechanism of reactions at solid-liquid interfaces will be given. Then a brief account of the experimental methods, including CFC and AFM techniques, for the study of interfacial reactions will be introduced. The final part of this chapter presents a summary of the work reported in this thesis.

3. Reactions at solid-liquid interfaces

The study of reactions between solids and liquids is a topic of huge industrial and environmental importance. Better control of waste emission and the need for more efficient use of raw materials in the chemical processing industry dictates a clear need for understanding of the related solid-liquid reaction mechanisms.[5] For instance, in the manufacturing of dyes, dye intermediates and fine chemicals, the involved solid-liquid reactions are usually not well understood and frequently cause problems on subsequent development and scale-up.[6-9] A proper description of the associated heterogeneous reaction mechanisms will undoubtedly lead to better control of the production process. Fig. 1.

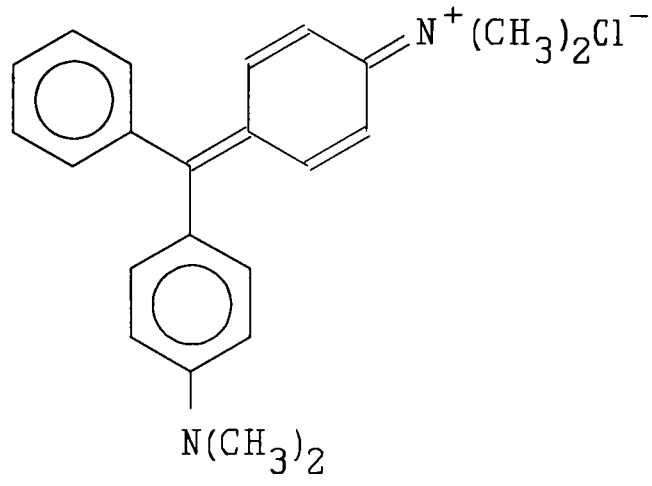
illustrates the synthetic formation of an important class of reactive dyes which are made by allowing a cyanuric chloride unit to react with an amine; R represents the chromophore. This unit also provides an active site for binding the dye molecule to the fabric surface. This forms the basis of reactive dyeing, which in this manner, produces more permanent coloration than the alternative dispersive process which relies on the physical solubility of the dye within the fabric.



Cyanuric Chloride Derivatised Amine Reactive Dye

Fig. 1 The reaction between cyanuric chloride and aqueous amine to form reactive dye.

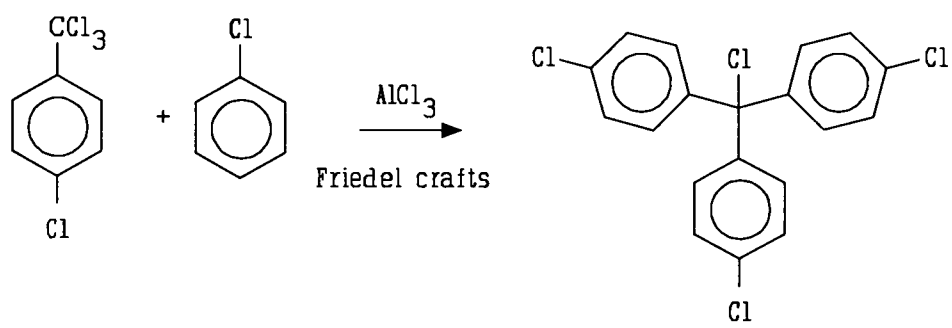
Despite of their deficient in fastness properties, interest in dispersive dyes, particularly of the triarylmethane type, was revived with the introduction of polyacrylonitrile fibres which readily adsorb dye molecules.[10] Moreover, triarylmethane dyes are of brilliant hue, relatively inexpensive and applicable to a wide range of substrates including leather, fur, anodised aluminum, glass, waxes, soap, plastics, drugs and cosmetics![10,11] A typical example of a triarylmethane dye is crystal violet as shown below:



Crystal violet

Fig. 2 A typical triarylmethane dye.

Triarylmethyl chloride and related classes of compounds are heavily utilised as intermediates in the benzotrichloride method[10-13] for the production of triarylmethane dye (see Fig. 3).



4,4',4''-trichlorotriphenylmethyl chloride

a. m-toluidine

b. o-chloroaniline

c. HCl, H_2O

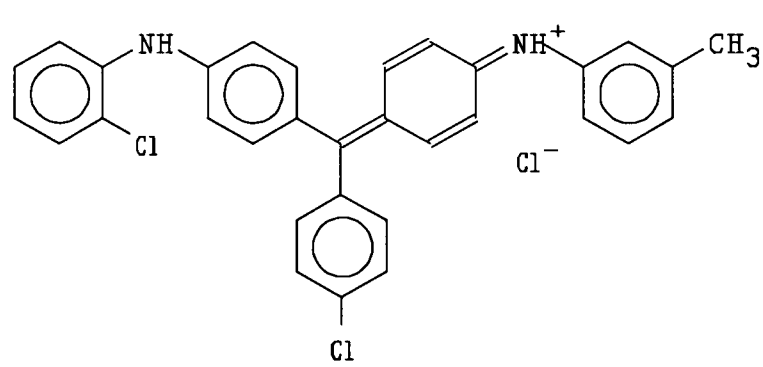


Fig. 3 Reactions showing the manufacture of triarylmethane dye via a triarylmethyl chloride intermediate in the benzotrichloride method.

An understanding of the interfacial reaction mechanisms of triphenylmethyl chloride as reported later may help to develop more efficient manufacturing processes in this area.

Another example of a solid-liquid reaction is in the dye fixation reaction shown in Fig. 4. This involves the formation of a covalent bond between a hydroxy group in the fabric and the dye molecule by the nucleophilic displacement of chloride ion.[14] In order to optimise and refine the process, the complicated heterogeneous mechanism involved will be established in chapter 8.

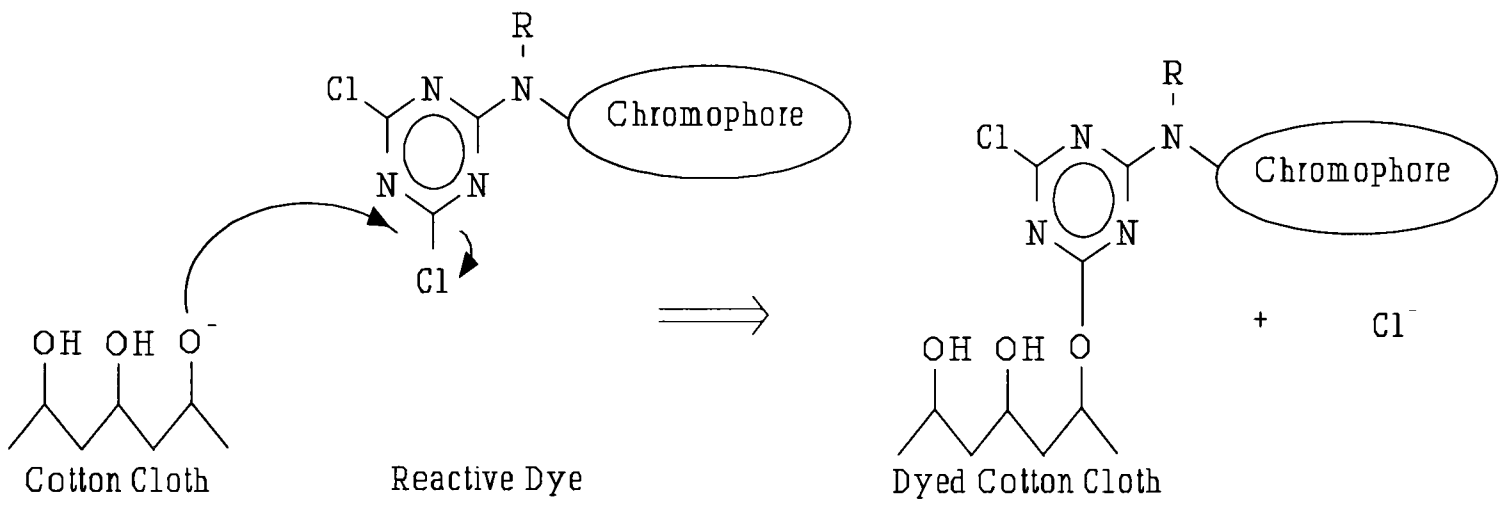


Fig. 4 The dyeing of cotton by a reactive dye.

In general, heterogeneous reactions can be described by a complex series of fundamental molecular processes. The following consecutive steps may be thought of as taking place during the solid-liquid interfacial reaction (see fig. 5):

1. Diffusion of reactants to the surface.
2. Adsorption of reactants onto the surface.
3. Formation of a surface complex.
4. Surface diffusion of the surface complex to a reaction site.

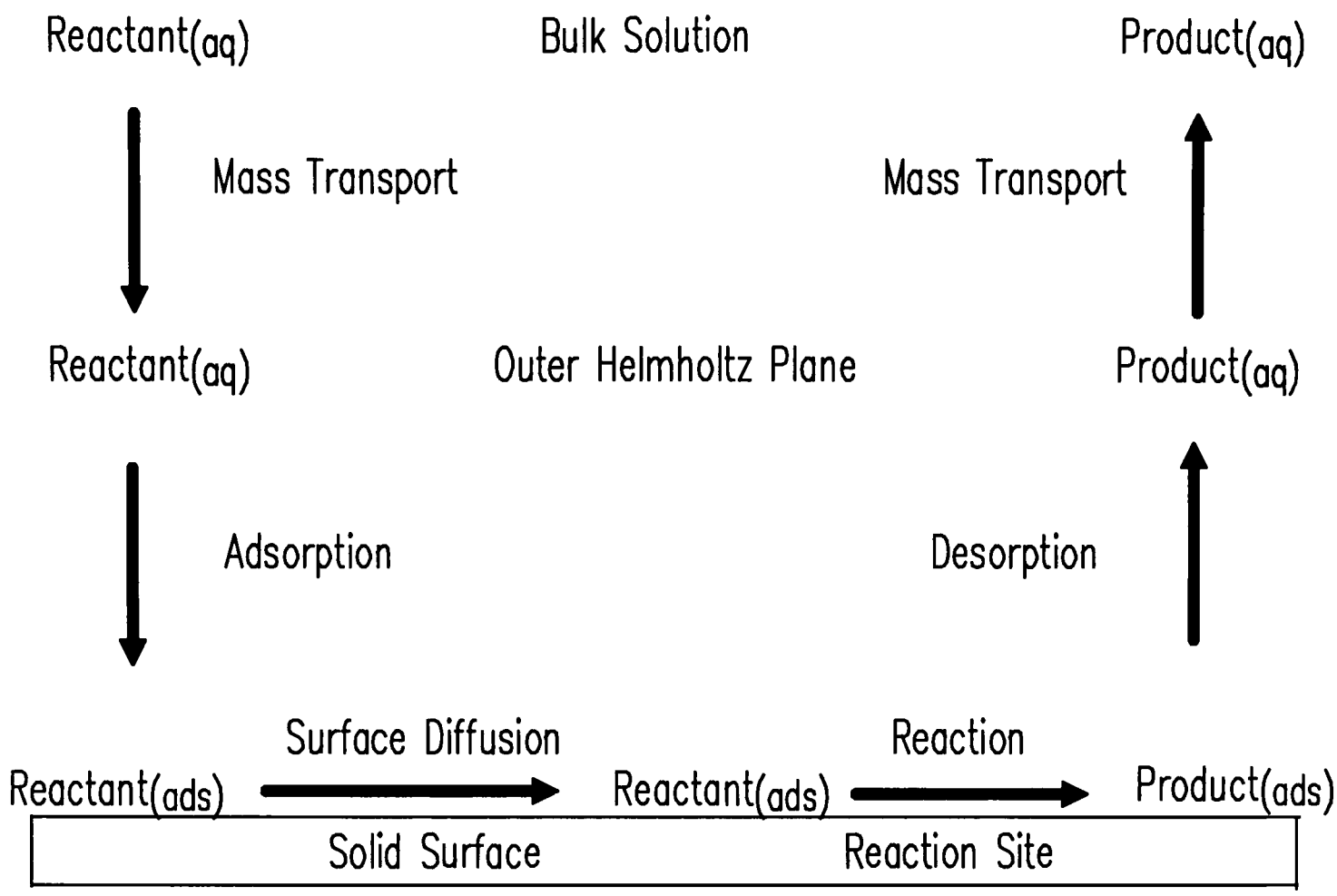


Fig. 5 Steps that may occur in a solid-liquid interface reaction.

5. Reaction of the surface complex to form reaction products.

6. Desorption of the reaction products from the surface.

7. Diffusion of the final reaction products into bulk solution.

The objectives of this study are to identify which of the above steps are significant and to quantify the rate of the interfacial chemical steps in various specific systems. It was noted that for precise determination of the interfacial kinetics, mass transport effects need to be unscrambled from the surface reaction. Previous work in Oxford group has introduced the CFC method to probe the mechanisms of solid-liquid interfacial reactions in which the mass transport to the reaction interface and nearby sensor is well defined and calculable.

In a recent study, the kinetics of dissolution of benzoic acid (BA) into aqueous solution were

studied using the CFC method.[15] It was found that the sample dissolved as a *constant flux* of uncharged BA molecules from the solid surface. In the case of a porous pressed pellet sample as depicted in fig. 6, the dissolution rate was intensified by the material diffusing into solution from surface pores saturated with BA.

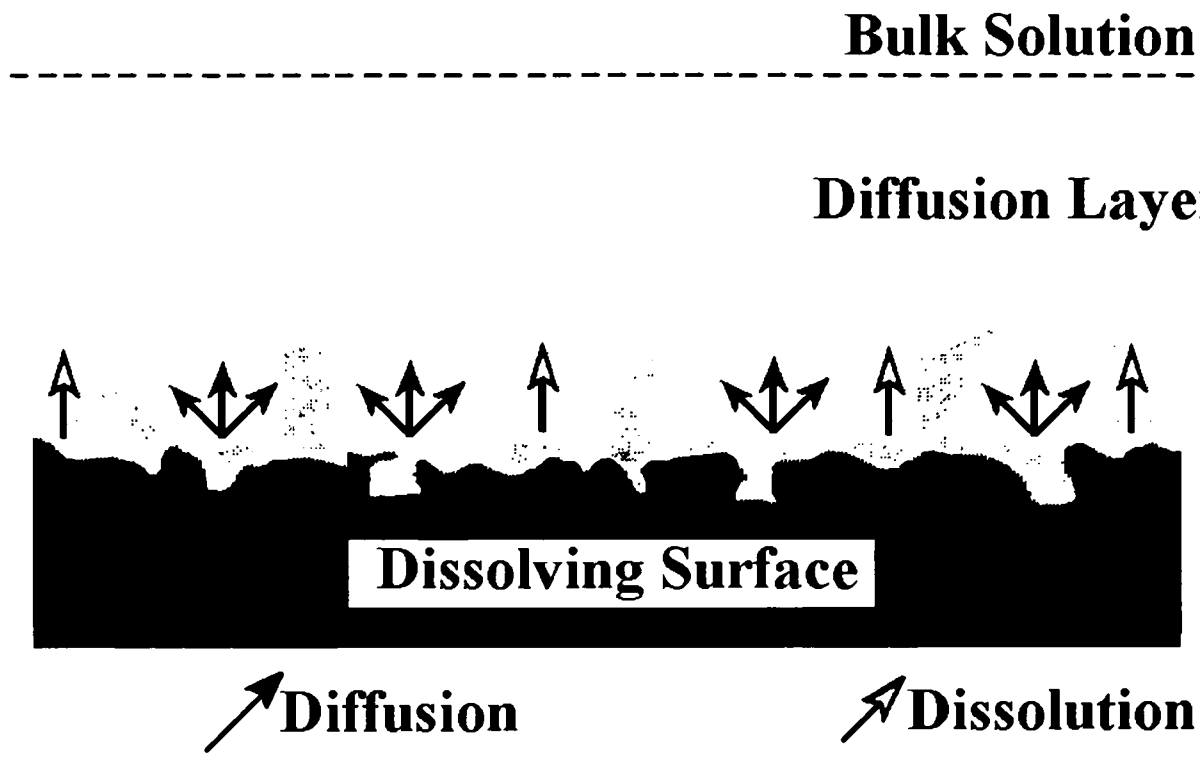


Fig. 6 Schematic diagram illustrating dissolution from the surface of a pressed pellet of BA. The arrows (solid) show the enhanced dissolution from pores emergent at the surface and saturated with BA. A smaller flux of BA (empty arrow) into solution arises from the direct dissolution of the solid BA located at the BA-water interface.

Next, the CFC approach was adopted to investigate the reaction between solid cyanuric chloride (CCl_3) and an aromatic amine in aqueous solution.[16] The reaction mechanism was proved to involve a *constant flux* of solid dissolving into solution then followed by homogeneous coupling to the amine, in parallel with the homogeneous hydrolysis of CCl_3. The porous pressed pellet provides pockets of saturated CCl_3 in the porous surface layer, in which hydrolysis produces chloride and hydrogen ions which are then released into solution as a constant flux of ions, thus mimicking a *heterogeneous reaction* of CCl_3 with water. Results using fused pellets with no porosity showed that, in this case, there was no authentic heterogeneous component to the reaction.

Based on the results of this previous work, we suggest three types of reaction patterns to characterise most organic solid-liquid reactions.

Type A

The rate determining step involves dissolution of the organic material followed by further slow reaction in the liquid phase. A typical example is the reaction of solid cyanuric chloride with aqueous aromatic amine solutions.[16]

Type B

This class is dedicated to reactions that proceed *at* the interface with the formation a of soluble product or a kinetically insignificant insoluble product layer. The hydrolytic dissolution of solid triphenylmethyl chloride in aqueous solution will be shown in chapter 6 to fall into this category.[17]

Type C

The reaction proceeds with the formation of insoluble products which accumulate at the interface. This results in the formation of an overlayer which can shut off the reaction leading to waste of solid reactant and subsequently a difficult purification procedure. A typical example is the reaction of p-chloranil with an aromatic amine.[18]

4. Techniques for the investigation of solid-liquid reactions

A number of different methods have traditionally been used to study solid-liquid reaction systems. Two commonly used well-established techniques, namely the dispersed powder

reactor and the rotating disc method will be discussed along with their limitations. Then the merits of the CFC technique will be established and shown to be the method of choice for most investigations. This section concludes with an overview on AFM for the imaging of reacting interfaces.

Quantification of the interfacial kinetics

Early empirical work on solid-liquid reactions used a dispersed powder reactor.[19] The solid is uniformly dispersed in a vessel by mechanical agitation in the liquid phase. However the detection of reaction products is remote to the interface and hence the sensitivity of this technique is less than satisfactory. In addition, it is difficult or impossible to disentangle mass transport and chemical kinetics in these systems and this precludes its use for rigorous mechanistic studies.[20]

The next development was the use of rotating disc (RD) techniques.[21] In this method, the solid substrate in the form of a disc is mounted in a non-conducting sheath. The face of substrate is polished flush with the surface of the sheath and exposed to the liquid phase for experimentation. By varying the rotation speed of the disc, the thickness of the diffusion layer above the substrate of interested can be varied and hence the individual contribution of chemical kinetics and mass transport to the overall reaction rate can be dissected. The RD has found wide applications both in electrochemical studies[22] and in the study of reaction rates at solid-liquid interfaces.[23,24] However, the RD method does suffer from a number of shortfall which make this a far from optimal technique to examine interfacial kinetics. In particular, the RD experiment is usually performed in a batch reaction vessel so that the concentration of the reaction product in the solution medium becomes progressively higher as the

experiment continues. This limits the time scale of the experiment and can promote back reactions at the solid surface. Moreover, the range of rotation speeds in RD experiments is very restricted, often being confined to less than 20 Hz.[25,26] More importantly, the amount of reaction product released is usually monitored in bulk solution. Detection remote from the site of reaction greatly degrades the sensitivity of the experiment. Finally, a critical issue is that the RD method has a much greater tendency to saturate with dissolved material in the solid-liquid interface since in the RD geometry convection is in a direction normal and towards the reactive surface. The dissolved material is therefore not effectively vented from the reactive surface[15] and the reaction rate may often be masked or restricted by the solubility of the solid substrate.

The channel flow cell (CFC) methodology has recently developed into a powerful method for the general elucidation of solid-liquid reaction mechanisms. The technique has been successfully applied to study of a wide range of solid-liquid reactions of industrial and environmental importance. These include the kinetics of the reaction between dye molecules and cotton fabric,[14] the dissolution of benzoic acid,[15] the reaction of protons with calcite[27] and the interfacial reaction of cyanuric chloride with an aromatic amine.[16]

The CFC approach has found considerable advantages in the study of interfacial reaction mechanisms.[28-31] It has been demonstrated elsewhere[28] that the CFC is superior in terms of mechanistic discriminating power as compared to the RD method. Moreover, the CFC can provide a much wider range of rates of mass transport to the reacting interface and also a much faster rate of mass transport. Hence it is an ideal method to probe rates of solid-liquid reactions that would be too fast to measure with other experimental techniques. Furthermore,

the technique is very useful in the study of heterogeneous reaction mechanisms which involve non-conducting solid substrate, where conventional electrochemical detection methods fail. In the work reported here, the CFC approach will be adopted to study the solid-liquid kinetics but used in conjunction with AFM imaging at the solid surface so as to visualise the reaction and the features of the surface structure vital to the kinetic processes.

A schematic diagram of the CFC is given in Fig. 7. The cell is composed of a rectangular duct cut in a perspex block and closed by a cover plate. The solid substrate is embedded into the cover plate together with a downstream detector system. The detector system (potentiometric, amperometric or spectroscopic) can monitor either the amount of reactant which survives passage over the substrate or the amount of product released from the reacting interface.

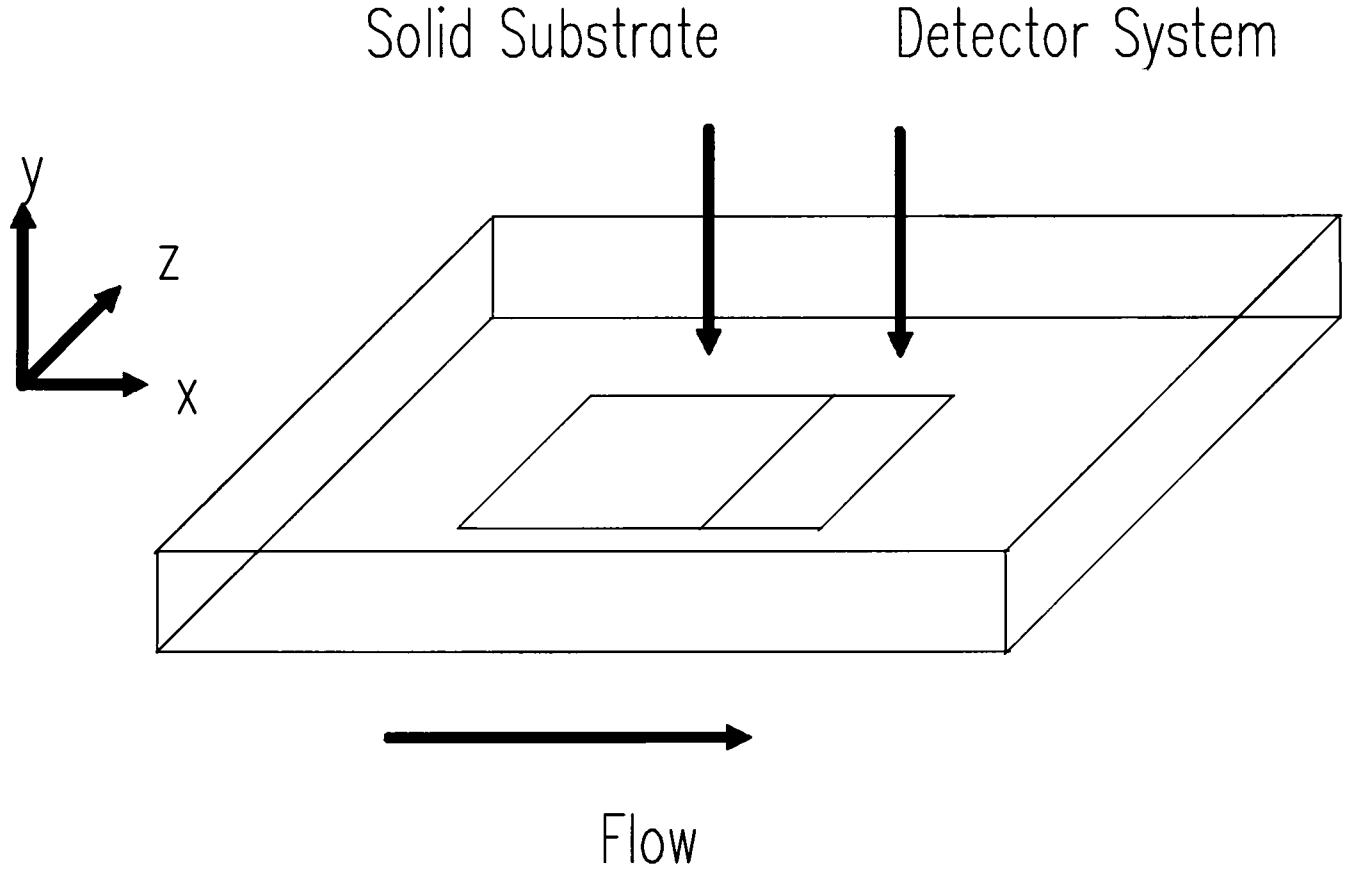


Fig. 7 Schematic diagram of the channel flow cell.

The liquid phase is allowed to contact with the solid substrate by flowing through the CFC under steady state laminar conditions. Thus, the transport of the reactant (or product) to (or from) the interface is well defined and this permits the precise modelling of the convective-diffusion-kinetic processes within the cell. The mass transport processes can be described by the following equations:[27,28]

$$\frac{\partial c_i}{\partial t} = D_i \nabla^2 c_i - \left[v_x \frac{\partial c_i}{\partial x} + v_y \frac{\partial c_i}{\partial y} + v_z \frac{\partial c_i}{\partial z} \right] \pm \sum_{i,j} k_{ij} c_i^{n_i} c_j^{n_j} \quad (1)$$

where c_i is the concentration of the i-th species, D_i is its diffusion coefficient and v_x, v_y and v_z are the cartesian components of flow. The third term in this equation represents any homogeneous kinetic terms involving the species i, j of order n_i, n_j having rate constants k_{ij}.

The backwards implicit finite difference (BIFD) method[32,33] can be employed to unravel equation 1 with the incorporation of a heterogeneous kinetic rate law to specify the boundary conditions at the reacting solid substrate surface and of a chemical model of any homogeneous reactions so as to characterise the rate processes denoted by k_{ij}. In this way the detector response and its solution flow rate dependence can be calculated. Comparison of theory and experiment allows selection between candidate rate equations thus giving mechanistic information. The optimal kinetic parameters are derived by minimising the deviation between the theoretical and experimental data in a least squares sense. Standard optimisation techniques, such as the BFGS (Broyden-Fletcher-Goldfard-Shanno) algorithm[34-36] can be adopted for this purpose. The convective-diffusion equations within the channel cell and their solution will be elaborated in the next chapter.

Imaging the reacting surface

AFM is a recently developed, versatile surface imaging technique, capable of resolving surface details down to the atomic level.[37-40] The method is applicable to the study of the surface topography of a wide range of samples (including insulators). A variety of mineral surfaces, [41,42] biological molecules,[43,44] organic crystals[45] as well as single fibres[46-50] have been studied using AFM. True lateral atomic-scale resolution by AFM was reported recently.[51]

In AFM, a sharp tip is mounted on a microcantilever arm and rastered across the sample surface to obtain the surface topography of the sample. Specifically, the AFM measures a contact force, typically of van der Waals origin, between the tip and the sample surface when the tip is scanned over the sample surface. The contact force can vary rapidly as a function of the distance between tip and sample surface. The abrupt change in contact force keeps the interaction confined to the very end of the sharp tip and hence may achieve high resolution imaging.

A schematic diagram of an AFM is given in Fig. 8.[52] The sample is positioned in a x-y-z piezo stage. AFM imaging can be done in either contact (repulsive) mode or noncontact (attractive) mode. In contact imaging, the AFM tip is pressed against the surface and the contact force acting on the tip will cause deflections of the microcantilever, which are monitored by a sensor. The output of the sensor is connected to a feedback loop which regulates the force and therefore the distance between the tip and the sample surface. In a typical configuration, forces as small as 10^{-9} N[39] are applied to the surface to reduce the damage to the sample. All images reported in this work were acquired in the contact mode. On the other hand, noncontact imaging maintains the AFM tip several nanometres above the

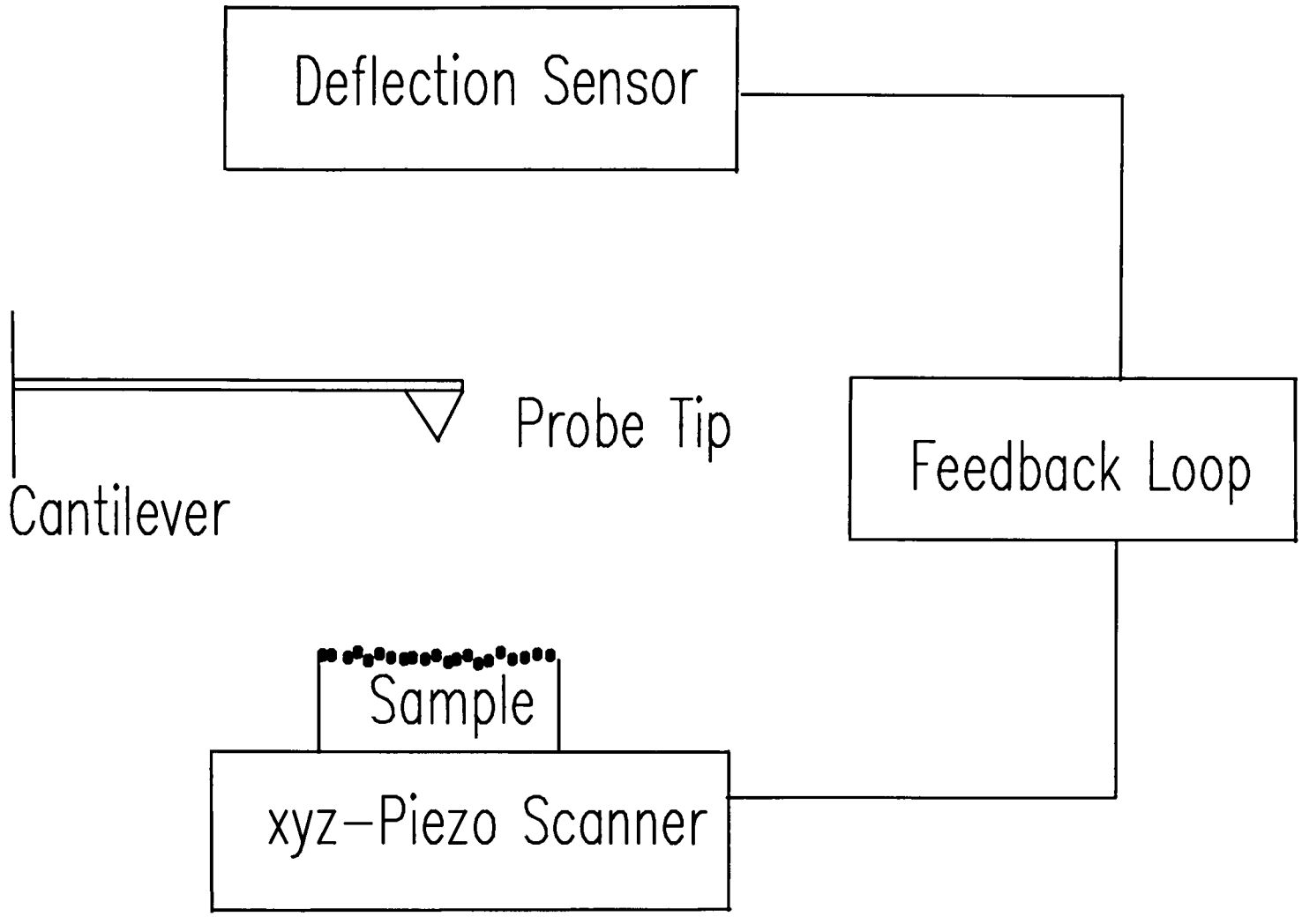


Fig. 8 Schematic diagram of an AFM.

sample surface and detects the weak attractive forces that are presented in this region. An advantage is that this operating mode is less likely to damage the substrate. However, the resolution is usually lower than the repulsive mode because of the greater distance between the tip and the surface.

The application of AFM for the real time monitoring of surfaces has received considerable attention in recent years. The technique was applied for in situ observation of the morphology of gold electrode surfaces,[53] InSe surfaces[54] and highly orientated pyrolytic graphite[55] in electrochemical studies. Excitingly, Hillner et al.[56,57] have demonstrated the method in the investigation of calcite crystal growth and dissolution in a flow-through fluid cell. AFM was employed to image the (1014) cleavage plane of a single calcite crystal as a supersaturated solution was flowed through the crystal to induce crystal growth. For the dissolution studies,

pure water was utilised instead of supersaturation solution. It was found that the calcite grows from aqueous solution by motion of monomolecular steps and dissolves by a combination of step motion and etch pit expansion.

5. Summary of the work presented in this thesis

Chapters 2 and 3 deal with the solution of the convective-diffusion equation within the channel cell and the experimental details. Chapters 4, 5 and 6 are concerned with the hydrolytic dissolution of solid triphenylmethyl chloride (TPMCl) in aqueous solution. Chapter 4 presents the implementation of an ac impedance measurement system to investigate the hydrolytic dissolution of solid triphenylmethyl chloride in aqueous solution of low ionic strength. Theory is given which enables the measured impedance resulting from the spatially inhomogeneous ion distribution within the flow cell, and its flow rate dependence, to be related to the dissolution process at the solid-liquid interface. The kinetic data will be seen to agree well with the heterogeneous model (chapter 6). In conjunction with the results described in chapter 6, the new method permits the measurement of kinetic salt effects which are consistent with a transition state in which there is essentially full dissociation into the ions TPM^+ and Cl^-. Chapter 5 reports a crystal indexing method based on a genetic algorithm. With the aid of this approach, the Miller indices of the centimetre sized triphenylmethyl chloride crystal planes can be unambiguously identified using a set of interplanar angles. The knowledge of Miller indices of a particular crystal plane is crucial to correlate its availability of exposed chlorine atoms as mentioned in chapter 6.

Chapter 6 establishes the mechanism of the hydrolytic dissolution of TPMCl in aqueous

solution. It will be demonstrated that hydrolysis proceeds via a direct heterogeneous process whereby the reaction occurs at the solid-liquid interface rather than in solution after the prior dissolution of TPMCl. Kinetic parameters are reported including the effect of ionic strength. Further, using experiments conducted with a different surface of single crystals of TPMCl, it will be shown that the rate of the interfacial reaction depends on the availability of exposed chloride atoms in the reacting crystal planes. In situ AFM studies indicate that the TPMCl surface, once in contact with water, is covered rapidly by a porous product layer. However, the latter provides a negligible effect on the reaction rate because of its high porosity.

Chapter 7 reports a channel flow cell specifically devised for ultraviolet-visible spectroelectrochemical experiments. The methodology developed in this chapter forms the basis for the study of reactive dyeing kinetics as described in chapter 8. The performance of this cell was characterised using the oxidation of N,N,N′,N′-tetramethyl-p-phenylenediamine in aqueous electrolytes. Theory was then derived to allow the spectroscopic response to be evaluated and was found to be in excellent agreement with experimental results. The technique was then applied to study the dimerisation rate of methyl viologen radical cations. The mechanistic results confirm that the electrogenerated methyl viologen radical cations dimerise rapidly in aqueous solution and the dimerisation rate constant has now been quantified for the first time.

Chapter 8 applies the spectrochemical channel cell developed in chapter 7 to scrutinise reactive dyeing kinetics on a cotton fabric. The reactive dyeing processes are complicated by the simultaneous hydrolysis of the dye molecule and the physical binding of its hydrolysed form onto the fabric. All these (undesirable) processes will be taken into account in the

reaction model evolved. In particular, the kinetic results indicate that the dye adsorption to the fabric is controlled by a solid-liquid interfacial process which is first order with respect to the surface concentration of the dye. However, the rate of this process is governed by the availability of reaction sites for the adsorption of dye molecules. It will be shown that mercerisation pretreatment of the fabric, and the presence of supporting electrolyte in high pH media, are essential to increase the dye uptake rate. AFM studies will be seen to suggest that mercerisation leads to a disordered fibre surface which may responsible for the enhanced dye absorption rate.

6. References

1. Compton R.G.; Coles B.A.; Spackman R.A., *J. Phys. Chem.*, **1991**, *95*, 4741.

2. Compton R.G.; Fisher A.C.; Wellington R.G.; Bethell D.; Lederer P., *J. Phys. Chem.*, **1991**, *95*, 4749.

3. Compton R.G.; Wellington R.G.; Bethell D.; Lederer P.; O'Hare D.M. *J. Electroanal. Chem.*, **1992**, *322*, 183.

4. Compton R.G.; Fisher A.C.; Wellington R.G.; Winkler J.; Bethell D.; Lederer P. *J. Chem. Soc. Perkin Trans. II*, **1992**, 1359.

5. Stinson S.C. *C&EN*, **1994**, *May 16*, 10.

6. Doraiswamy L.K.; Sharma M.M. *Heterogeneous Reactions* Vol 2; John Wiley; New York, **1984**.

7. Trambouze P. *Chem. Eng. Prog.*, **1990**, *86(2)*, 23.

8. Leng D.E. *Chem. Eng. Prog.*, **1991**, *87(6)*, 23.

9. Bennington C.P.J.; Thangavel V.K. *Can. J. Chem. Eng.*, **1993**, *71*, 667.

10. Bannister D.; Elliott J. *Kirk-Othmer Encyclopedia of Chemical Technology*, Mark H.F; Othmer D.T.; Overberger C.G.; Seaborg G.T. (eds.); 3rd ed.; Vol 23; John Wiley; New York, **1983**.

11. Rys P.; Zollinger H., *Fundamentals of the Chemistry and Applications of Dyes*; Wiley; New York, **1972**.

12. Lubs H.A. (ed.) *The Chemistry of Synthetic Dyes and Pigments*; Am. Chem. Soc. Monograph Series; Reinhold Pub. Corp.; New York, **1955**.

13. Abrahart E.N. *Dyes and Their Intermediates*, 2nd ed.; Chemical Publishing; New York, **1977**.

14. Compton R.G.; Wilson M. *J. Appl. Electrochem.*, **1990**, *20*, 793.

15. Compton R.G.; Harding M.S.; Pluck M.R.; Atherton J.H.; Brennan C.M. *J. Phys. Chem.*, **1993**, *97*, 10416.

16. Compton R.G.; Harding M.S.; Atherton J.H.; Brennan C.M. *J. Phys. Chem.*, **1993**, *97*, 4677; Harding M.S. *D.Phil. Thesis*; Oxford, **1993**.

17. Tam K.Y.; Compton R.G.; Atherton J.H.; Brennan C.M.; Docherty R. *J. Am. Chem. Soc.*, **1996**, *118*, 4419.

18. Booth J.; Sanders G.H.W.; Compton R.G.; Atherton J.H.; Brennan C.M. *J. Am. Chem. Soc.*, submitted; Booth J. *D.Phil. Thesis*; Oxford, **1996**.

19. Noyes A.A.; Whitney W.R. *Z. Physikal Chem.* **1897**, *23*, 689; Bircumshwa L.L.; Riddiford A.C. *Quart. Rev. Chem. Soc.* **1952**, *6*, 157.

20. Yadav G.D.; Sharma M.M. *Ind. Eng. Chem. Process Des. Dev.*, **1981**, *20*, 385; Atherton J.H.; Umbers J.A. *Paper presented in the 4th European Symposium on Organic Reactivity*, Newcastle, England, July, **1993**.

21. Agar J.N. *Discuss. Faraday Soc.* **1947**, *1*, 26; Gregory D.P.; Riddiford A.C.; *J. Chem.*

Soc., **1956**, 3756.

22. Tominaga H.; Adzumi H.; Isobe T. *Bull. Chem. Soc. Japan*, **1939**, *14*, 348; Levich V.G. *Physicochemical Hydrodynamic*; Prentice-Hall; Englewood Cliffs, **1962**.

23. Macdonald D.D.; Wright G.A. *Can. J. Chem.*, **1970**, *48*, 2847.

24. Guk Y.V.; Golod E.L.; Gidaspov B.V. *J. Org. Chem. USSR*, **1977**, *13*, 14.

25. Touitou E.; Donbrow M. *Int. J. Pharm.*, **1981**, *9*, 97.

26. Serqjuddin A.T.M.; Jarowski C.I. *J. Pharm. Sci.*, **1985**, *74*, 148.

27. Compton R.G.; Unwin P.R. *Phil. Trans. R. Soc. Lond.*, **1990**, *A330*, 1.

28. Unwin P.R.; Compton R.G. *Comprehensive Chemical Kinetics*, Compton R.G. (ed.); *Vol 29*; Elsevier; Amsterdam, **1989**.

29. Brown C.A.; Compton R.G.; Narramore C.A. *J. Colloid Interface Sci.*, **1993**, *160*, 372.

30. Moldoveanu S.; Anderson J.L. *J. Electroanal. Chem.*, **1984**, *175*, 67.

31. Fisher A.C.; Compton R.G. *J. Phys. Chem.*, **1991**, *95*, 7538.

32. Moldoveanu S.; Anderson J.L. *J. Electroanal. Chem.*, **1984**, *175*, 67.

33. Compton R.G.; Pilkington M.B.G.; Stearn G.M. *J. Chem. Soc. Faraday Trans. 1*, **1988**, *84*, 2155.

34. Walsh G.R. *Methods of Optimization*; Wiley; London, **1975**.

35. Yuan Y. *IMA J. Numerical Analysis*, **1991**, *11*, 325.

36. Tam K.Y.; Chau F.T. *J. Auto. Chem.*, **1992**, *14*, 157.

37. Sarid D. *Scanning Force Microscopy*; Oxford University Press; Oxford, **1990**.

38. Binning G.; Quate C.F.; Gerber C.H. *Phys. Rev. Lett.*, **1986**, *56*, 930.

39. Hansma P.K.; Elings V.B.; Marti O.; Bracker C.E. *Science*, **1988**, *242*, 209.

40. Sarid D.; Elings V. *J. Vac. Sci. Technol. B*, **1991**, *9*, 431.

41. Johnsson P.A.; Eggleston C.M.; Hochella M.F. *Am. Mineral*, **1991**, *76*, 1443.

42. Meyer G.; Amer N.M. *Appl. Phys. Lett.*, **1990**, *56*, 2100.

43. Hansma H.G.; Weisenhom A.L.; Gould S.A.C.; Sinsheimer R.L.; Gaub H.E.; Stucky G.D.; Zaremba C.M.; Hansma P.K. *J. Vac. Sci. Technol. B*, **1991**, *9*, 1282.

44. Parpura V.; Hayden G.P.; Henderson E. *J. Cell Sci.*, **1993**, *104*, 427.

45. Overney R.M.; Howald L.; Frommer J.; Meyer E.; Guntherodt H.J. *J. Chem. Phys.*, **1993**, *94*, 8441.

46. Hanley S.J.; Giasson J.; Revol J.; Gray D.G. *Polymer*, **1992**, *33*, 4639.

47. Hanley S.J.; Gray D.G. *Holzforschung*, **1994**, *48*, 29.

48. Phillips T.L.; Horr T.J.; Huson M.G.; Turner P.S.;Shanks R.A. *Textile Res. J.*, **1995**, *68*, 445.

49. Zhdan P.A.; Grey D.; Castle J.E. *Surface and Interface Analysis*, **1994**, *22*, 290.

50. Forder C.; Armes S.P.; Simpson A.W.; Maggiore C.; Hawley M. *J. Mater. Chem.*, **1993**, *3*, 563.

51. Ohnesorge F.; Binning G. *Science*, **1993**, *260*, 1451.

52. Meyer E.; Heinzelmann H. *Scanning Tunneling Microscopy II*; Wiesendanger R.; Guntherodt H.J. (Eds.); Springer-Verlag; Heidelberg, **1992**.

53. Manne S.; Massie J.; Elings V.B.; Hansma P.K.; Gewirth A.A. *J. Vac. Sci. Technol. B*, **1991**, *9*, 950.

54. Uosaki K.; Koinuma M. *J. Electroanal. Chem.*, **1993**, *357*, 301.

55. Goss C.A.; Brumfield J.C.; Irene E.A.; Murray R.W. *Anal. Chem.*, **1993**, *65*, 1378.

56. Hillner P.E.; Gratz A.J.; Manne S.; Hansma P.K. *Geology*, **1992**, *20*, 359.

57. Hillner P.E.; Manne S.; Gratz A.J.; Hansma P.K. *Ultramicroscopy*, **1992**, *42*, 1387.

Chapter 2

Theory

This chapter introduces the concepts of mass transport controlling the transfer of reactants/products to/from the solid-liquid interface. Based on these notions, the convective-diffusion equations for different interfacial reaction models within a channel flow cell (CFC) are developed. Solution using the backwards implicit finite difference (BIFD) method are then derived to solve these equations numerically.

1. Mass transport

Three mechanisms, namely migration, diffusion and convection, are believed to govern the transport of materials to/from the solid-liquid interface. Migration involves the transport of charged particles under an electric field. The driving force for this process is proportional to the electrical potential gradient and the charge of the particles involved. In diffusion, particles move from a region of high concentration to one of low concentration so as to maximise the entropy of the system. In convection, the solution is mechanically moved so that both the solvent and solute are transferred at the same rate.

Migration

The presence of a potential gradient causes the migration of charged particles or ions. If the local potential φ varies along the x-axis then there is an electrostatic force acting on the species in the x-direction which is opposed by viscous forces. This results[1] in the particles

having a velocity v (cm s^{-1}),

$$v = -\frac{zu}{|z|}\frac{\partial \varphi}{\partial x} \quad (1)$$

where u (cm^2 s^{-1} V^{-1}) is the mobility of the ions with charge number z. It should be pointed out that the migration effect in most of the experiments reported in this thesis are rendered negligible by the addition of an electroinactive supporting electrolyte to the bulk solution in large excess of the species under investigation.[1] A discussion on the experiments carried out in low ionic strength conditions is given in chapter 4.

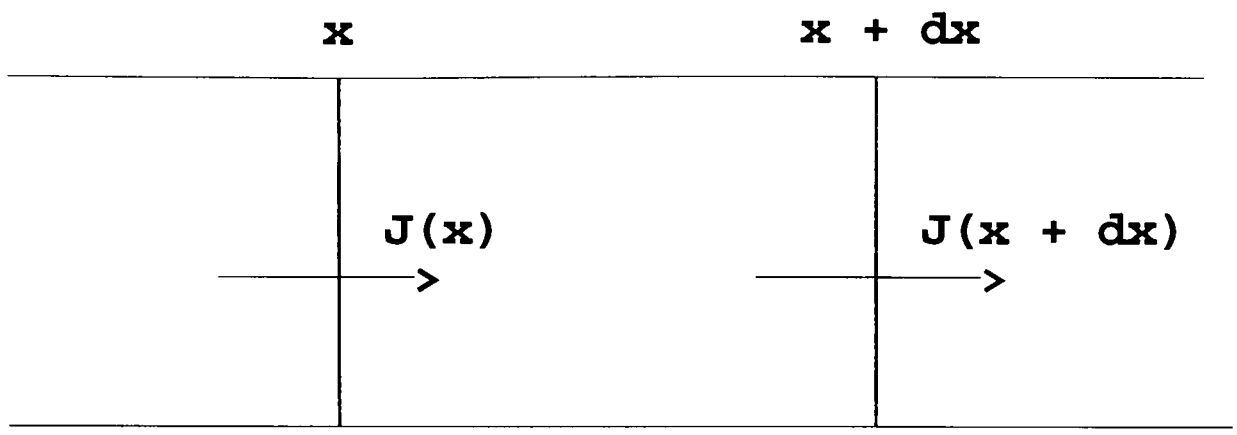
Diffusion

Diffusion involves the movement of particles down a concentration gradient. In solid-liquid reactions, this process arises once the reactant is consumed or the product is generated. The flux of materials, J, transferred by diffusion can be described by Fick's first law,[2,3]

$$J\ (mol\ cm^{-2}\ s^{-1}) = -D\frac{\partial c}{\partial x} \quad (2)$$

where D and c are the diffusion coefficient and the concentration of the species of interest respectively. Diffusion leading to the change of concentration with time can be quantified by Fick's second law as derived below. Consider a fluid element of width dx, the change in concentration is given by

$$\frac{\partial c}{\partial t} = \frac{J(x) - J(x + dx)}{dx} \quad (3)$$



It can be understood that

$$J(x + dx) = J(x) + \frac{\partial J(x)}{\partial x} dx \quad (4)$$

$$= J(x) - \frac{\partial}{\partial x} D \frac{\partial C}{\partial x} dx \quad (5)$$

$$= J(x) - D \frac{\partial^2 C}{\partial x^2} dx \quad (6)$$

By substituting eqn. 6 into eqn. 3, we reach Fick's second law,

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} \quad (7)$$

Consider three dimensions in cartesian coordinates, Fick's laws are written as,

$$J = -D \nabla C \quad (8)$$

$$\frac{\partial C}{\partial t} = -\nabla J = D \nabla^2 C \quad (9)$$

where ∇ is called the Laplace operator[4] and has the form given below

$$\nabla = \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} \quad (10)$$

Convection

Convection concerns movement of the fluid as a result of a pressure gradient. Two types of convections can be identified, namely natural convection and forced convection. The former arises from density differences in the solution. Density differences may result from many factors, such as a process at an interface which removes or adds material into solution and so induces density gradients local to the reacting surface. The latter, forced convection, results from the deliberate application of an external force, such as stirring or pumping. In general, convection can be expressed as follow,

$$J = C v_x \qquad (11)$$

with v_x represents the velocity (cm s^{-1}) of the solution in x-direction. It is important to apply this kind of mass transport in a controllable manner so as to permit the quantification of the material transfer between the solid-liquid interface. We now turn to a consideration of convection, in conjunction with diffusion, and apply it to the specific case of the CFC.

2. Convective-diffusion within a channel flow cell

Consider the mass transport in the x-direction (see eqns. 2 & 11), we can write

$$J = -D\frac{\partial C}{\partial x} + C v_x \qquad (12)$$

In three dimensions this generalises to

$$J = -D\left[\frac{\partial C}{\partial x} + \frac{\partial C}{\partial y} + \frac{\partial C}{\partial z}\right] + C\left[v_x + v_y + v_z\right] \qquad (13)$$

where v denotes the velocities (with the coordinates subscripted). Accordingly, the general convective-diffusion equation (see eqn. 9) in cartesian coordinates is,

$$\frac{\partial c}{\partial t} = D\left[\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2}\right] - \left[v_x\frac{\partial c}{\partial x} + v_y\frac{\partial c}{\partial y} + v_z\frac{\partial c}{\partial z}\right] \quad \textbf{(14)}$$

Fig. 1 depicts a schematic diagram of a CFC. The symbols w, d, 2h, x_e represent respectively the electrode width, channel width, channel depth and electrode length.

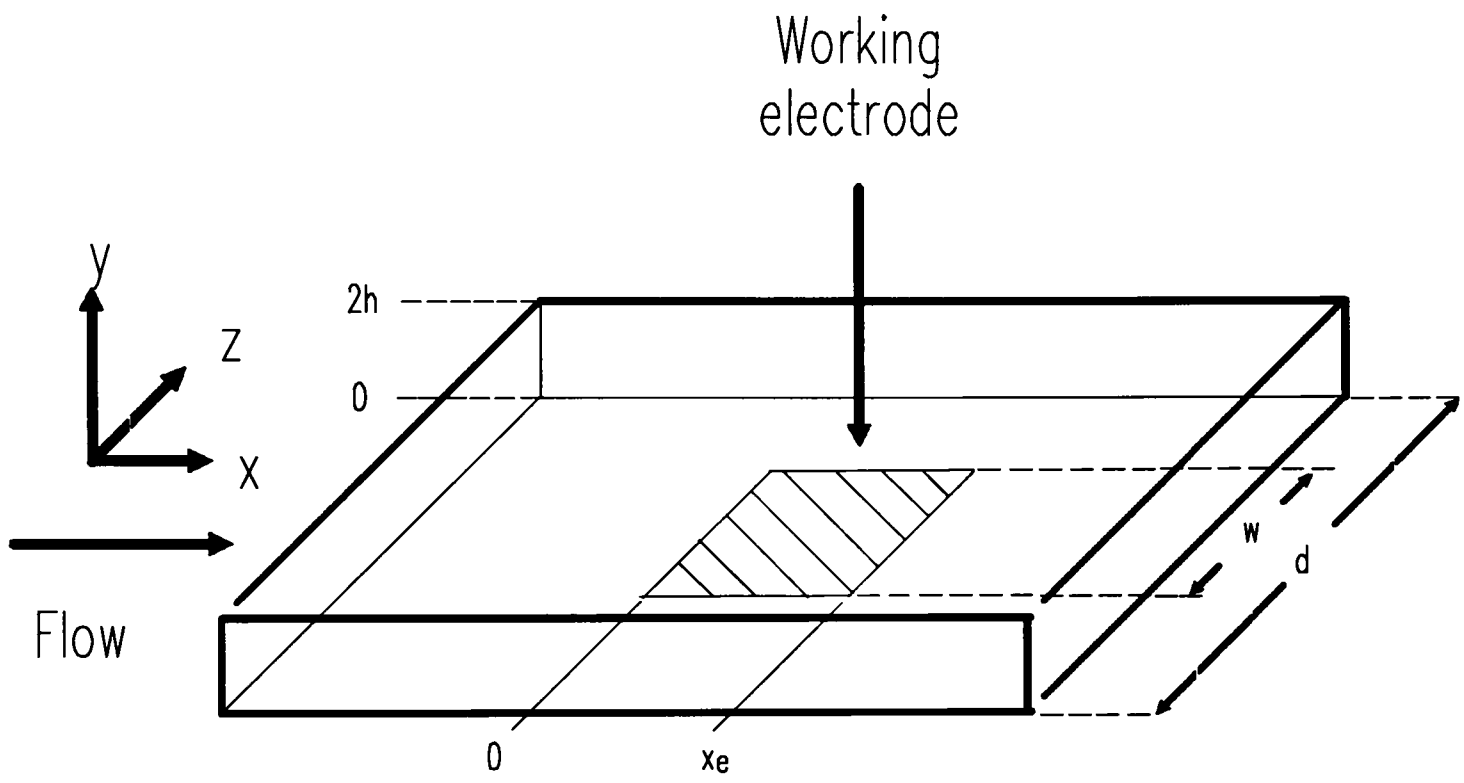


Fig. 1 A schematic diagram of a channel flow cell.

In practical cell geometry, d is usually much greater than h so we may assume that the channel is infinite in the z-direction and consequently the edges of the flow cell do not affect the hydrodynamic flow. In this way, a two dimensional consideration in the x-y plane is sufficient for all modelling purpose. Moreover, the convection in x-direction is large enough to mean that diffusion in this direction is negligible. This consideration is valid for the millimetre sized electrodes that are utilised in this work.[5] Based on these assumptions, the convective-diffusion equation (14) can be simplified to,

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial y^2} - v_x\frac{\partial C}{\partial x} \qquad (15)$$

As shown in eqn. 15, the flow velocity must be quantified before we can derive any solution to model the material transfer within the channel cell. In this thesis, we focus our attention to the laminar flow regime in which the solution advances through the channel in separate, non-mixing layers. This is in marked contrast to alternative turbulent flow in which each element of the solution moves in a chaotic manner. The categorisation of the flow regime is governed by the Reynold's number, N_{Re}:

$$N_{Re} = \frac{2\rho h v_0}{\mu} \qquad (16)$$

where ρ and μ symbolise, respectively, the solution density and viscosity. h is the channel half-height and v_0 represents the solution velocity in the centre of the channel. The ratio (ρ/μ) is known as the kinematic viscosity. If the value of N_{Re} is less than two thousand,[6,7] the flow is designated to laminar otherwise it is regarded as turbulent. All experiments conducted in this work have N_{Re} values less than three hundred.

Upon solution entering the CFC the convective flow is approximately plug shaped. The effect of friction at the walls of the channel will be to slow down the flow and this frictional effect is transferred through the solution towards the centre of the cell. Eventually, a Poiseuille flow regime is developed:

$$v_x = \frac{3 V_f}{4 h d}\left[1 - \frac{(h - y)^2}{h^2}\right] \qquad v_y = v_z = 0 \qquad (17)$$

where V_f is the solution flow rate in $cm^3\ s^{-1}$. The entry length, l_e, required for this to be set up is given approximately by[8]

$$l_e = 0.1\, h\, N_{Re} \qquad (18)$$

l_e values for the CFC used in this thesis are at least two times greater than the required quantities calculated by using eqn. 18.

Combining eqns. 15 and 17, we arrive at the following convective-diffusion equation for the modelling of CFC process described in this thesis:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial y^2} - \frac{3\,V_f}{4\,h\,d}\left[1 - \frac{(h-y)^2}{h^2}\right]\frac{\partial C}{\partial x} \qquad (19)$$

In steady state measurements, eqn. 19 can be simplified to

$$D\frac{\partial^2 C}{\partial y^2} = \frac{3\,V_f}{4\,h\,d}\left[1 - \frac{(h-y)^2}{h^2}\right]\frac{\partial C}{\partial x} \qquad (20)$$

3. Solution of the convective-diffusion equations for a channel flow cell

Two commonly used techniques, namely the Lévêque approximation and backwards implicit finite difference (BIFD) method for the solution of the convective-diffusion equation will be introduced. Although the former one invokes an assumption about the velocity profile, it is applicable within a certain range of solution flow rates and provides a simple approximate relation between limiting current and solution flow rate from which the diffusion coefficients of electroactive species can be evaluated. Here, a brief account of this method will be given. More importantly, it has been demonstrated that the BIFD approach is a versatile strategy to solve the transport equations in CFC applications.[9] In this context, formulations for the

convective-diffusion equations as adopted in the investigation of the hydrolytic dissolution of triphenylmethyl chloride (chapters 4 & 6) are derived in this section. Alternative reaction models for spectroelectrochemical studies (chapter 7) and reactive dyeing process (chapter 8) will be detailed individually in the corresponding chapters.

The Lévêque approximation

Consider a single electron process within a channel electrode,

$$A \rightarrow B \pm e^- \qquad (21)$$

Levich[7] applied a linear velocity profile near the electrode surface to derive an approximate solution for the transport limited currents of a channel electrode as a function of the solution flow rate. This simplification[10] (Lévêque approximation) can be written as

$$v_x \approx \frac{6 V_f y}{4 h^2 d} \qquad (22)$$

Assuming steady state conditions ($\partial[A]/\partial t = 0$), the analytical solution (Levich equation) for eqns. 15 and 22 is given as follows:[7,10]

$$I_{lim} = 0.925 nFw[A]_0 \left(\frac{V_f D^2 x_e^2}{h^2 d}\right)^{1/3} \qquad (23)$$

where F is Faraday's constant, n is the number of electrons transferred and $[A]_0$ is the bulk concentration of species A.

The simplification in eqn. 22 is only satisfactory when the flow is dominated by convection. In particular, the Lévêque approximation breaks down when[10]

$$\frac{x_e D}{v_0 h^2} \geq 1 \quad \Rightarrow \quad \frac{x_e}{v_0} \geq \frac{h^2}{D} \qquad (24)$$

where x_e/v_0 is approximately the time taken to flow over the electrode and h^2/D is the approximate time to diffuse across the height of the cell. In the case of very slow flows, the diffusion layer thickness becomes comparable to the channel height leading to exhaustive electrolysis and hence the linearisation of the flow profile is not valid. The cell then operates under a "thin layer regime" and the current is proportional to the flow rate.[11] The diffusion coefficients of all electroactive species reported in chapter 7 are evaluated by fitting experimental data against the Levich equation (eqn. 23). It can be seen (chapter 7) that all solution flow rates employed in the determination of diffusion coefficients are at least one order of magnitude greater than the breakdown value as predicted using eqn. 24. In contrast to the analytical expression devised using the Lévêque approximation, the BIFD method offers an effective way to unravel eqn. 20 and has found many useful applications.[9,12-15] We next turn to this approach with particular attention to the solid-liquid processes studied in this thesis.

The backwards implicit finite difference method

The BIFD method requires vector calculations[12] within a 2-dimensional grid that represents the x-y plane (see fig. 1). The vectors describe concentrations in the y-direction for different values of x. The computation proceeds at the beginning of the reacting interface to the end of the x-y plane under investigation with the incorporation of the appropriate boundary conditions. In this manner, the concentration of the relevant species over the x-y plane can be solved. This approach retains the full parabolic flow rate profile (eqn. 17) and can be

extended easily to involve coupled homogeneous kinetics, interfacial rates and adsorption processes as described in this thesis. An additional advantage over the alternative numerical techniques such as the explicit finite difference[16] or the Crank-Nicholson[12] methods is that the calculation is independent of the value at (0, 0), i.e. the concentration at the extreme upstream edge of the reacting interface while the other two may give answers dependent on this choice and so may be regarded as less satisfactory.[17] In the subsequent discussion, the BIFD formulations for the triphenylmethyl chloride (TPMCl) system reported in this thesis are given to illustrate how this numerical technique can be applied to model solid-liquid reaction within the CFC.

Based on the three types of reaction patterns for organic solid-liquid reactions mentioned in chapter 1, we can write down the following kinetic models,

Constant flux model (Type A)

$$TPMCl(s) \xrightarrow{k_d} TPMCl(aq) \qquad (25)$$

$$\underset{A}{TPMCl(aq)} \xrightarrow{k_1} \underset{B}{TPM^+(aq) + Cl^-(aq)} \qquad (26)$$

$$TPM^+(aq) + H_2O(l) \xrightarrow{fast} TPMOH(aq, s) + H^+(aq) \qquad (27)$$

where k_d represents the dissolution rate constant of TPMCl and k_1 is the homogeneous hydrolysis rate constant of TPMCl. When k_d is sufficiently large that the surface of the solid substrate is saturated with TPMCl (aq), this is designated as the saturated surface model.

Heterogeneous reaction model (Type B)

$$TPMCl(s) \xrightarrow{k_i} \underset{B}{TPM^+(aq)} + Cl^-(aq) \qquad (28)$$

$$TPM^+(aq) + H_2O(l) \xrightarrow{fast} TPMOH(aq, s) + H^+(aq) \qquad (29)$$

where k_i denotes the interfacial reaction rate constant

Surface passivation model (Type C)

$$TPMCl(s) \xrightarrow{k_d} TPMCl(aq) \qquad (30)$$

$$TPMCl(aq) \xrightarrow{k_1} TPM^+(aq) + Cl^-(aq) \qquad (31)$$

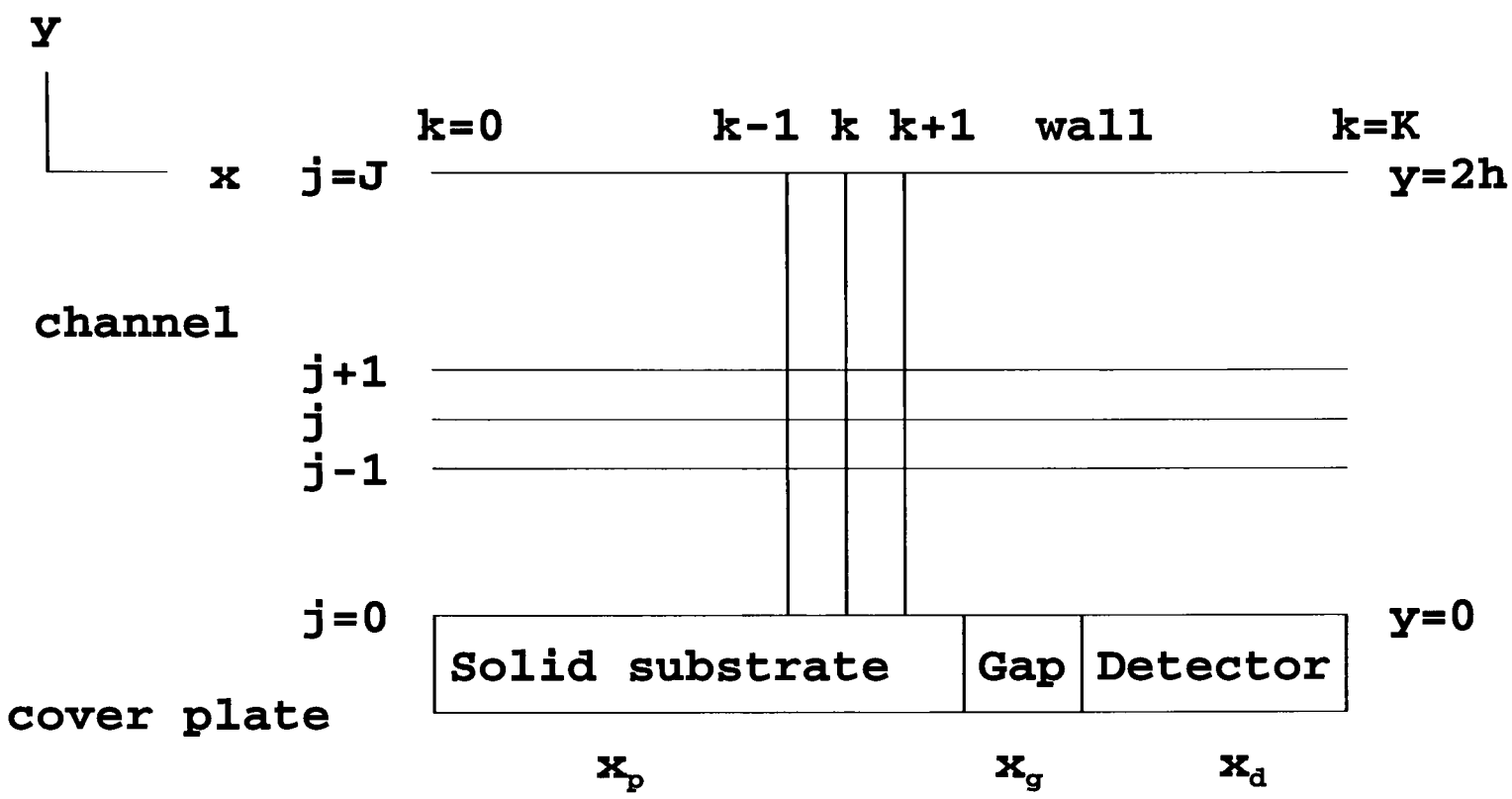
$$TPM^+(aq) + H_2O(l) \xrightarrow{k_p} TPMOH(s) + H^+(aq) \qquad (32)$$

where k_p is the precipitation rate constant of triphenylmethanol (TPMOH).

As shown in chapters 4 and 6, there was no evidence for any formation of a thick or impervious product overgrowth to stop the hydrolysis process so the surface passivation model is omitted in subsequent discussions. We first consider the heterogeneous reaction model; the convective-diffusion equation within the CFC is written as (see eqn. 20):

$$D_B \frac{\partial^2 g^B}{\partial y^2} = \frac{3 V_f}{4 h d} \left[\frac{1 - (h - y)^2}{h^2} \right] \frac{\partial g^B}{\partial x} \qquad (33)$$

where D_B and g^B represent, respectively, the diffusion coefficient and concentration of the chloride ion. Consider the x-y plane to be covered by a 2-dimensional finite difference grid as shown below:



Let the increments in the x-direction be Δx and in the y-direction, Δy,

$$\Delta y = \frac{2h}{J}, \quad y_j = j\Delta y \quad (j = 0,1,...J) \tag{34}$$

$$\Delta x = \frac{x_p + x_g + x_d}{K}, \quad x_k = k\Delta x \quad (k = 0,1,...K) \tag{35}$$

where J and K denote, respectively, the number of grid points in the y and x directions. The derivatives in eqn. 33 are approximated to

$$\frac{\partial g^B}{\partial x} = \frac{g^B{}_{j,k+1} - g^B{}_{j,k}}{\Delta x} \tag{36}$$

$$\frac{\partial^2 g^B}{\partial y^2} = \frac{g^B{}_{j-1,k+1} - 2g^B{}_{j,k+1} + g^B{}_{j+1,k+1}}{(\Delta y)^2} \tag{37}$$

Combining eqns. 33, 36, 37 and after simplification:

$$g^B_{j,k} = -\lambda^B_j g^B_{j-1,k+1} + (2\lambda^B_j + 1) g^B_{j,k+1} - \lambda^B_j g^B_{j+1,k+1} \quad (38)$$

$$\lambda^B_j = \frac{D_B \Delta x (2h)^3 d}{6 V_f \, j (\Delta y)^3 (2h - j\Delta y)} \quad (39)$$

where $g^B_{j,k}$ is the concentration of chloride at the j,k-th grid. For a particular k, the J - 1 simultaneous equations can be expressed in matrix form as

$$\{d\} = \{T\} \{u\}$$

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_j \\ \vdots \\ \vdots \\ d_{J-2} \\ d_{J-1} \end{bmatrix} = \begin{bmatrix} b_1 & c_1 & & & & & & & \\ a_2 & b_2 & c_2 & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & a_j & b_j & c_j & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & \ddots & \ddots & \ddots & \\ & & & & & & a_{J-2} & b_{J-2} & c_{J-2} \\ & & & & & & & a_{J-1} & b_{J-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_j \\ \vdots \\ \vdots \\ u_{J-2} \\ u_{J-1} \end{bmatrix} \quad (40)$$

with

$$u_j = g^B_{j,k+1} \quad (j = 1,2....J-1) \quad (41)$$

$$d_j = g^B_{j,k} \quad (j = 2,3,...J-1) \quad (42)$$

$$a_j = -\lambda^B_j \quad (j = 2,3...J-1) \quad (43)$$

$$b_j = 2\lambda^B_j + 1 \quad (j = 2,3...J-2) \quad (44)$$

$$c_j = -\lambda^B_j \quad (j = 1,2...J-2) \quad (45)$$

The above matrix equation can be elucidated by using the Thomas algorithm[18] as given in Appendix One. The boundary conditions are now developed to solve eqn. 38:

Chloride concentration before solution reaches the solid substrate

$$g^{B}_{j,0} = 0 \qquad (j = 1,2,....J-1) \tag{46}$$

At the far wall, gap and detector (potentiometric, see chapter 6)

$$\frac{\partial g^{B}}{\partial y}\Bigg|_{y = 0,\ 2h} = 0 \tag{47}$$

After substitution into eqn. 38 we have:

$$b_{J-1} = \lambda^{B}_{J-1} + 1, \quad b_1 = \lambda^{B}_{1} + 1, \quad d_1 = g^{B}_{1,k} \tag{48}$$

At the surface of the solid substrate,

$$k_i\ (mol\ cm^{-2} s^{-1}) = -D_B \frac{\partial g^{B}}{\partial y}\Bigg|_{y = 0} \tag{49}$$

In finite difference form:

$$g^{B}_{0,k} = g^{B}_{1,k} + \frac{\Delta y k_i}{D_B} \tag{50}$$

and after substitution into eqn. 38 we have:

$$b_1 = \lambda^{B}_{1} + 1, \quad d_1 = g^{B}_{1,k} + \frac{\lambda^{B}_{1} \Delta y k_i}{D_B} \tag{51}$$

The above calculations are continued along the x-direction to compute all the concentration elements in the finite difference grid. The average value of the chloride concentration over the detector surface is calculated as

$$[Cl^-] \approx \overline{g^B} = \frac{\int_{k_2}^{K} W(k)\ g^B{}_{1,k}\ dk}{\int_{k_2}^{K} W(k)\ dk} \qquad (52)$$

where k_2 represents the grid number corresponding to the beginning of the detector. W(k) denotes a weighting function to account for the circular geometry of the detector which is defined as

$$W(k) = \sqrt{(x_d/2)^2 - (\Delta x\,(k - k_m))^2} \qquad (53)$$

where

$$k_m = \frac{k_2 + K}{2} \qquad (54)$$

The integral of eqn. 52 can be evaluated by using a trapezoidal method.[19]

For the constant flux model, the following steady state convective-diffusion-kinetic equations are obtained

$$D_A \frac{\partial^2 g^A}{\partial y^2} = \frac{3\,V_f}{4\,h\,d}\left[\frac{1 - (h - y)^2}{h^2}\right]\frac{\partial g^A}{\partial x} + k_1 g^A \qquad (55)$$

$$D_B \frac{\partial^2 g^B}{\partial y^2} = \frac{3\,V_f}{4\,h\,d}\left[\frac{1 - (h - y)^2}{h^2}\right]\frac{\partial g^B}{\partial x} - k_1 g^A \qquad (56)$$

When these are cast into finite difference form, we have:

$$g^A{}_{j,k} - \frac{(\Delta y)^2 \lambda^A{}_j k_1 g^A{}_{j,k}}{D_A} = -\lambda^A{}_j g^A{}_{j-1,k+1} + (2\lambda^A{}_j + 1)\, g^A{}_{j,k+1} - \lambda^A{}_j g^A{}_{j+1,k+1} \qquad (57)$$

$$\lambda^A{}_j = \frac{D_A \Delta x (2h)^3 d}{6 V_f \, j (\Delta y)^3 (2h - j\Delta y)} \qquad (58)$$

$$g^B{}_{j,k} + \frac{(\Delta y)^2 \, \lambda^B{}_j \, k_1 \, g^A{}_{j,k}}{D_B} = -\lambda^B{}_j g^B{}_{j-1,k+1} + (2\lambda^B{}_j + 1) g^B{}_{j,k+1} - \lambda^B{}_j g^B{}_{j+1,k+1} \qquad (59)$$

a_j, b_j, c_j, u_j, λ_j for chloride are the same as before while the formula for d_j is given below:

$$d_j = g^B{}_{j,k} + \frac{(\Delta y)^2 \, \lambda^B{}_j \, k_1 \, g^A{}_{j,k}}{D_B} \qquad (j = 1,2, \ldots . J-1) \qquad (60)$$

Similarly, the following equations for the tridiagonal matrix (see eqn. 40) and boundary conditions are established for TPMCl (aq)

$$u_j = g^A{}_{j,k+1} \qquad (j = 1,2 \ldots J-1) \qquad (61)$$

$$d_j = g^A{}_{j,k} - \frac{(\Delta y)^2 \lambda^A{}_j k_1 \, g^A{}_{j,k}}{D_A} \qquad (j = 2,3, \ldots J-1) \qquad (62)$$

$$a_j = -\lambda^A{}_j \qquad (j = 2,3 \ldots J-1) \qquad (63)$$

$$b_j = 2\lambda^A{}_j + 1 \qquad (j = 2,3 \ldots J-2) \qquad (64)$$

$$c_j = -\lambda^A{}_j \qquad (j = 1,2 \ldots J-2) \qquad (65)$$

The TPMCl concentration before the solution reaches the solid substrate:

$$g^A{}_{j,0} = 0 \qquad (j = 1,2, \ldots . J-1) \qquad (66)$$

At the far wall, gap and detector

$$\left.\frac{\partial g^{A}}{\partial y}\right|_{y = 0,\ 2h} = 0 \qquad (67)$$

and after substitution into eqn. 57 we have:

$$b_{J-1} = \lambda^{A}_{J-1} + 1, \quad b_1 = \lambda^{A}_1 + 1, \quad d_1 = g^{A}_{1,k} - \frac{(\Delta y)^2 \lambda^{A}_1 k_1 g^{A}_{1,k}}{D_A} \qquad (68)$$

At the surface of the solid substrate,

$$k_d\ (mol\ cm^{-2} s^{-1}) = -D_A \left.\frac{\partial g^{A}}{\partial y}\right|_{y = 0} \qquad (69)$$

In finite difference form,

$$g^{A}_{0,k} = g^{A}_{1,k} + \frac{\Delta y\ k_d}{D_A} \qquad (70)$$

and after substitution into eqn. 57 we have:

$$b_1 = \lambda^{A}_1 + 1, \quad d_1 = g^{A}_{1,k} - \frac{(\Delta y)^2 \lambda^{A}_1 k_1 g^{A}_{1,k}}{D_A} + \frac{\lambda^{A}_1 \Delta y k_d}{D_A} \qquad (71)$$

Likewise, the computations proceed along the x-direction for the concentration of TPMCl, chloride to give the average concentration over the detector electrode surface.

Finally, we turn to the saturated surface model in which the surface of the solid substrate is assumed to maintain a saturated layer of TPMCl (aq). The formulation of this model is almost in line with the constant flux model except that the solubility of the material (C_{sat}) is now imposed as a boundary condition at the solid substrate surface.

$$g^{A}_{0,k} = C_{sat} \qquad (72)$$

thus, putting eqn. 72 into eqn. 57, we arrive at

$$b_1 = 2\lambda^{A}_{1} + 1, \quad d_1 = g^{A}_{1,k} - \frac{(\Delta y)^2 \lambda^{A}_{1} k_1 g^{A}_{1,k}}{D_A} + \lambda^{A}_{1} C_{sat} \qquad (73)$$

A literature value of 1.959 x 10^{-5} cm^2 s^{-1} for the diffusion coefficient of chloride[20] at similar ionic strength to this work will be adopted in the calculations. The diffusion coefficient for TPMCl was estimated as 5.36 x 10^{-6} cm^2 s^{-1} by means of the Wilke-Chang correlation.[21] It can be seen that for the heterogeneous reaction model, only the kinetic parameter, k_i, has to be known while for the constant flux model, both k_d and k_l need to be sought. Bentley[22] reported a value of 3.9 s^{-1} (in 90% v/v acetone at 25 °C) for k_l. A more recent estimation has shown that k_l is approaching 1.7 x 10^7 s^{-1} in water at 25 °C.[23]

4. Numerical simulation

In this section, attention is directed to the evaluation of suitable J and K values for the convergence of the chloride concentration-solution flow rate (C-F) profile. Then, the behaviour of the constant flux model (A) and heterogeneous reaction model (B) as described previously are compared and the latter one will prove to be the preferable choice in the numerical sense. The corresponding comparison with the saturated surface model will be given in the discussion section of chapter 6. The following cell geometry which is selected from a typical flow cell experiment is utilised to generate all the C-F profiles reported here.

$2h = 0.102$ cm, $d = 0.581$ cm, $x_p = 0.888$ cm, $x_g = 0.304$ cm, $x_d = 0.163$ cm

In addition, an hypothetical rate constant of 1.22×10^{-8} mol cm^{-2} s^{-1}, for either k_l (constant flux) or k_i (heterogeneous reaction), is employed. Yet, this quantity is chosen purposely closed to the experimental kinetic parameter (see chapter 6) in order to confront the present simulations with the experiments.

It is noted that the values of J and K play a crucial role in the accuracy of the computed C-F profiles. The aforementioned numerical method provides a discretised model for the physical process. Thus, large J, K of course, could give a much more accurate profile, but may, however, require prohibitively long computational time and be subject to underflow error. Fig. 2 show a comparison of the percentage deviation of chloride concentration from J = 1000, K = 1000 for different grids size using the type B model.

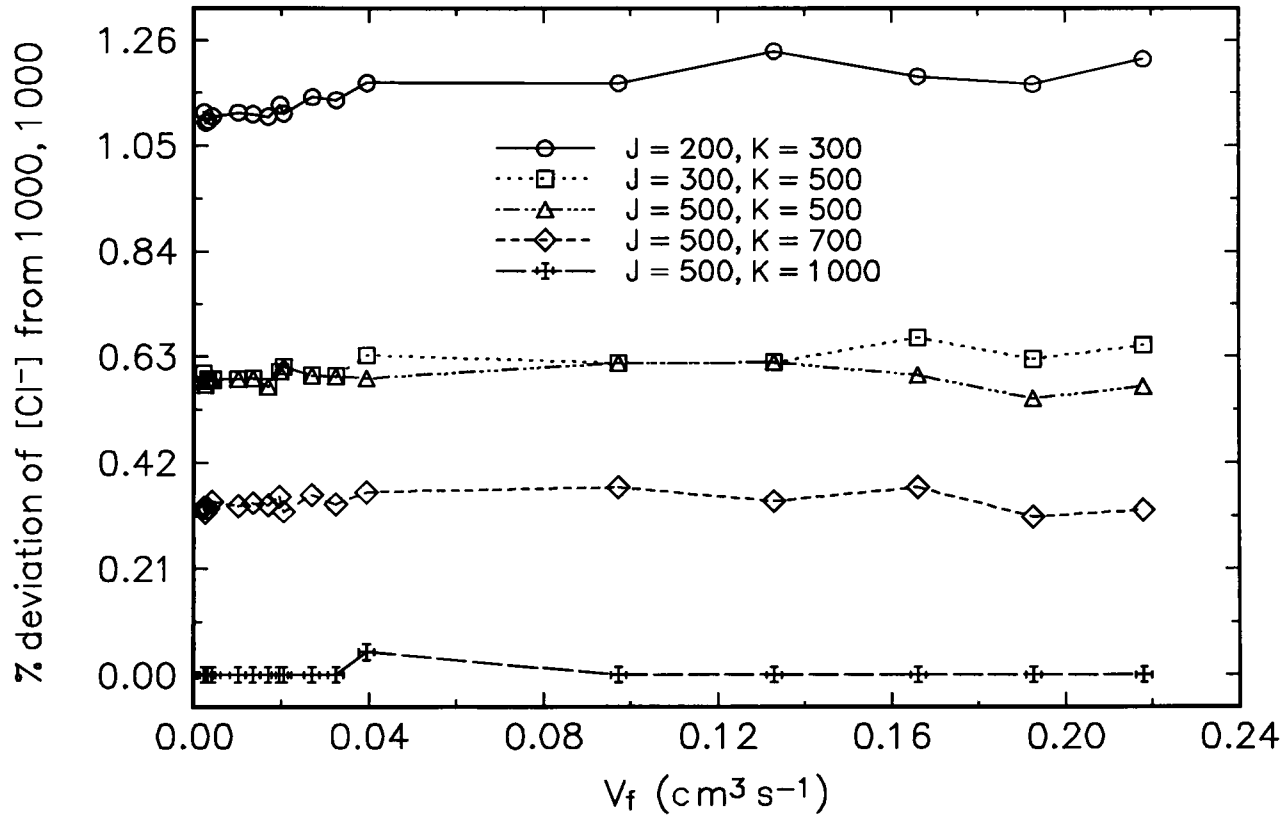


Fig. 2 A test for convergence for the C-F profiles generated using model B. The cell geometry and kinetic parameter are given in the text.

It can be seen that when a grid size equal to or larger than J = 300, K = 500 is used, the percentage deviation of chloride concentration from J = 1000, K = 1000 is less than 0.7%. Hence, one can conclude that a grid size of 300 x 500 is sufficient for convergence of the C-F profile for model B. This quantity (300 x 500) will be employed for subsequent analysis of model B.

Similar comparisons are made between the C-F profiles generated using model A, with a hypothetical homogeneous hydrolysis rate constant k_1 = 0.14 s^{-1}. As shown in fig. 3, the profile computed using J = 1000, K = 1000 is converged to less than 0.5% deviation from those calculated with J = 1000, K = 10000. However, for a homogeneous hydrolysis rate constant k_1 = 10.0 s^{-1}, a grid size of J = 200, K = 4000 is required to attain a 0.1% deviation from the profile computed with J = 1000, K = 10000.

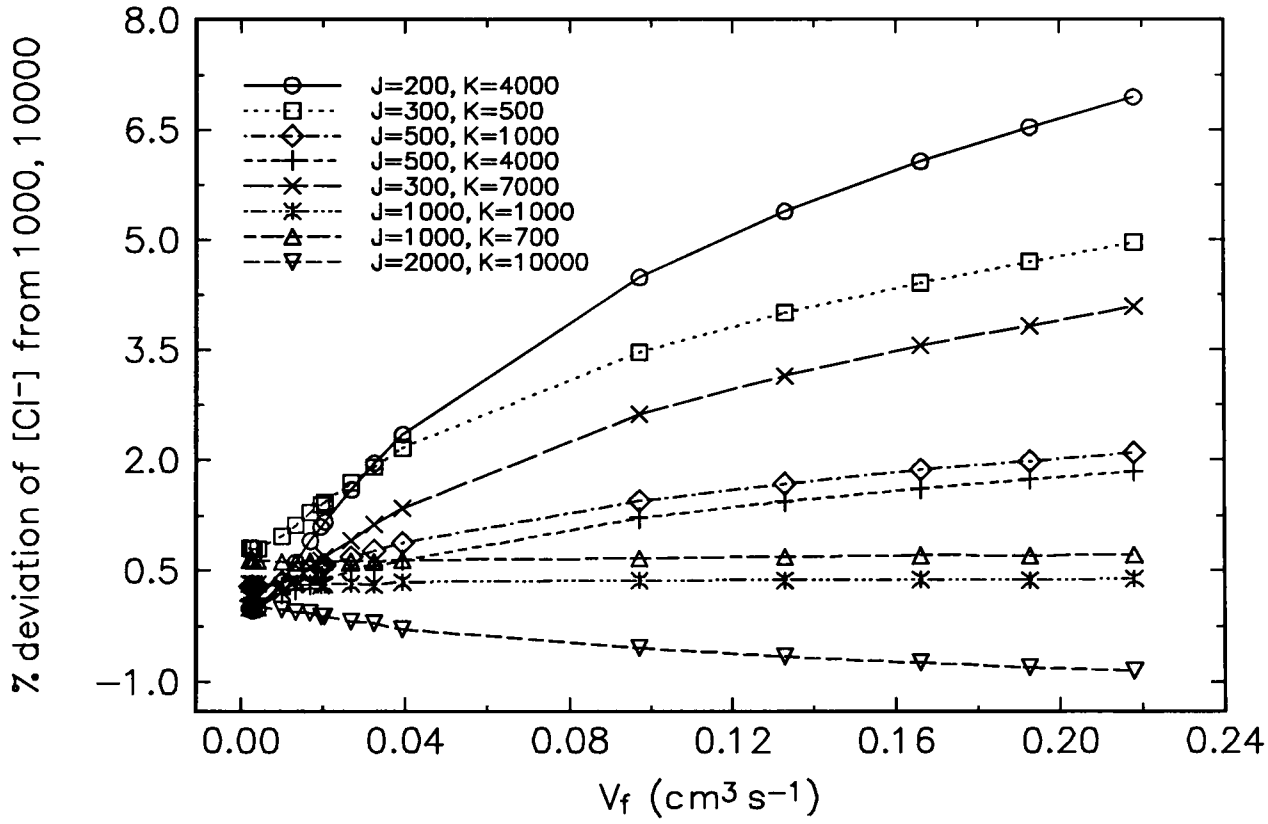


Fig. 3 A test for convergence for the C-F profile generated using model A with k_1 = 0.14 s^{-1}. Other parameters are given in the text.

Fig. 4 illustrates the behaviour of the heterogeneous reaction model and constant flux model. As k_1 increases (from 0.14 to 10.0 s^{-1}), the C-F profile almost overlaps with the one calculated using the heterogeneous reaction model. This is expected because as homogeneous hydrolysis rate constants increase, the life time of TPMCl (aq) in solution decreases. Below a certain k_d/k_1 ratio, the resistance of the TPMCl homogeneous hydrolysis reaction is so low that the process can be regarded as virtually occurring at the solid-liquid interface as predicted by using the heterogeneous reaction model. In other words, the TPMCl molecules are hydrolysed in the solid surface instead of going into the solution. The k_d/k_1 ratio is approximately equal to 3 x 10^{-9} mol cm^{-2} (1.22 x 10^{-8} mol cm^{-2} s^{-1}/3.9 s^{-1}) in this simulation. Note that the homogeneous hydrolysis rate constant for TPMCl in water is believed to be larger than 3.9 s^{-1}.[22-24]

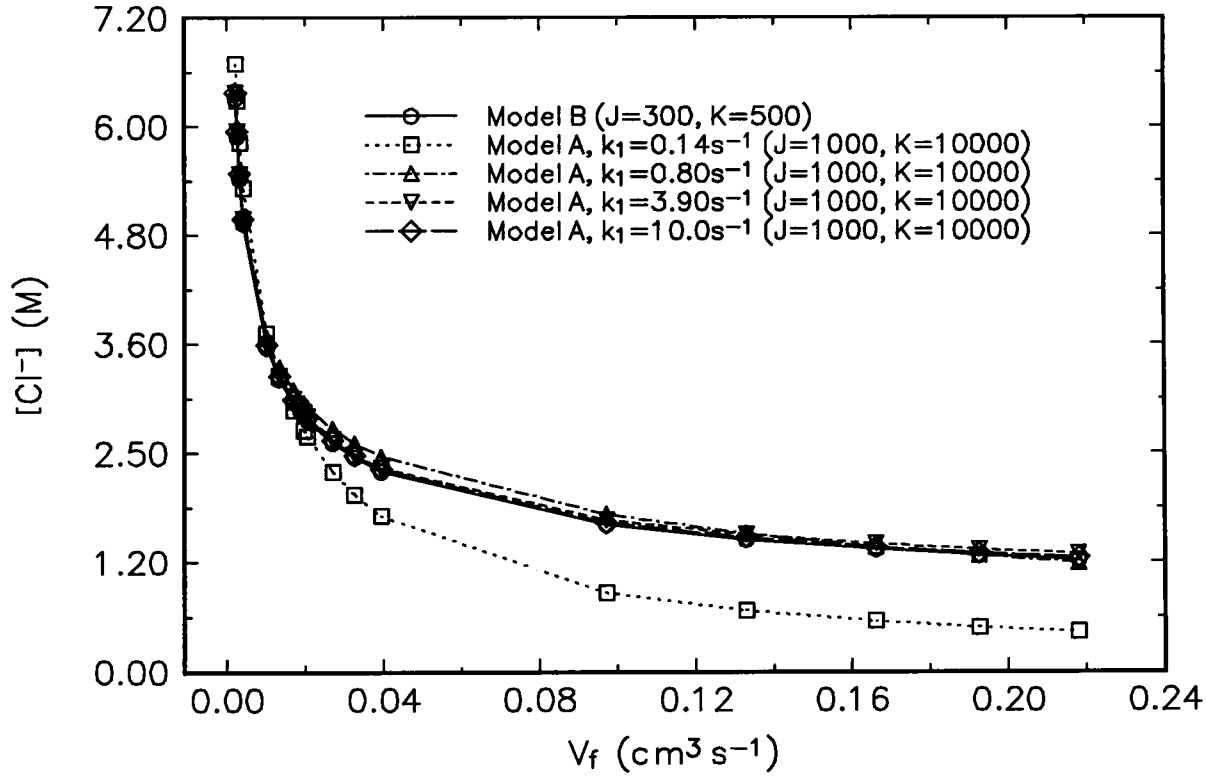


Fig. 4 A comparison of C-F profiles as calculated using model A and model B.

Figs. 5 and 6 show, respectively, the concentration of chloride computed using model B along the x-y plane at solution flow rate of 0.1 and 0.01 cm^3 s^{-1}. Both profiles show a maximum at the end of the solid substrate. The chloride concentration then decreases along the gap and detector direction owing to the diffusion-convective dilution. Comparing the two figures (figs. 5 and 6), the profile is compressed toward the solid substrate surface ($y \rightarrow 0$) by the higher solution flow rate as shown in fig. 6.

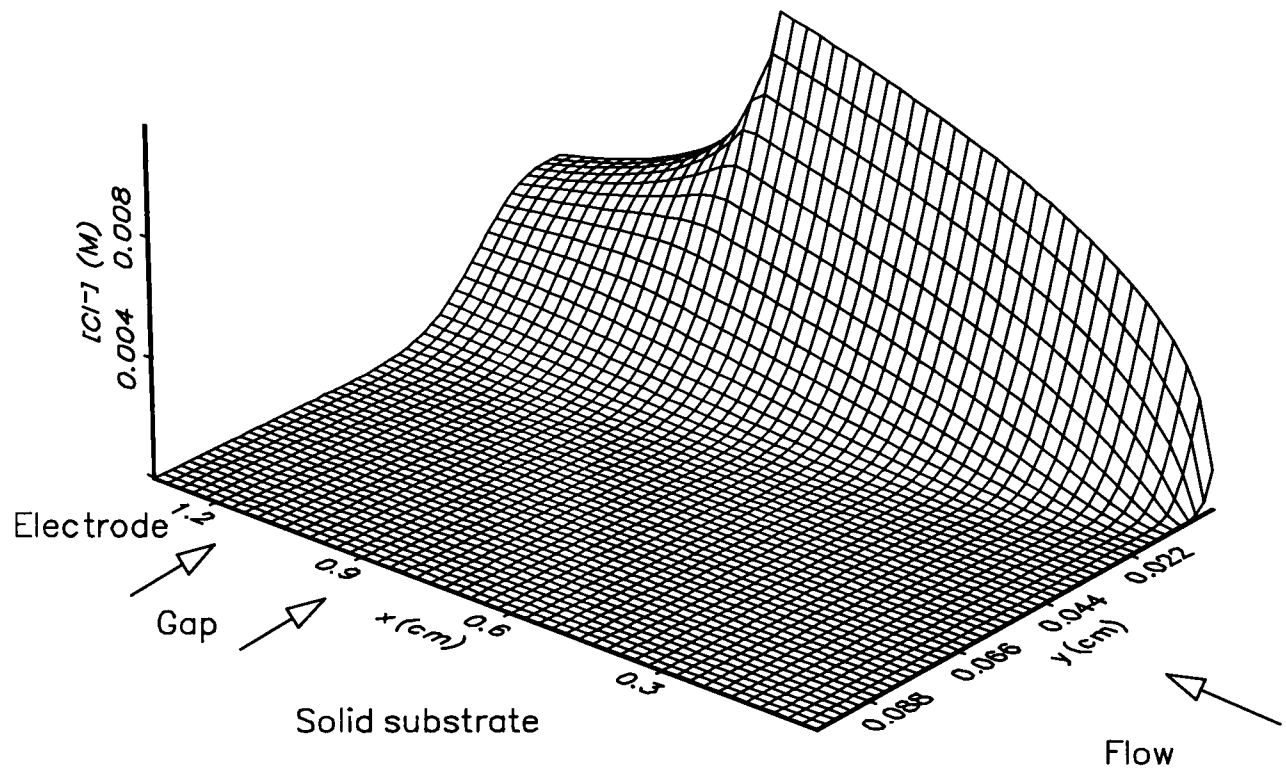


Fig. 5 Concentration of chloride calculated using model B along the x-y plane at a solution flow rate of 0.01 cm^3 s^{-1}. Cell geometry and kinetic parameter are given in text.

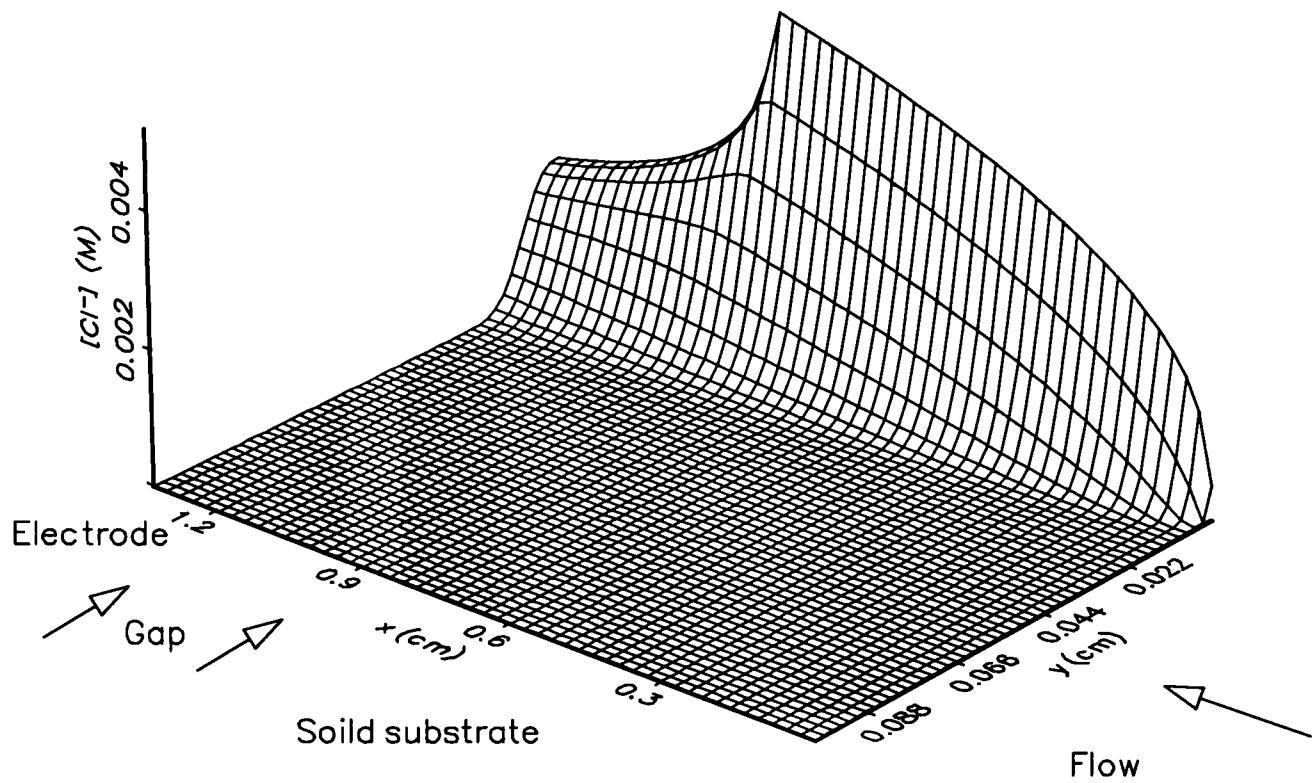


Fig. 6 Concentration of chloride calculated using model B along the x-y plane at a flow rate of 0.1 cm^3 s^{-1}. Cell geometry and kinetic parameter are given in text.

5. References

1. Oldham K.B.; Zoski C.G. *Comprehensive Chemical Kinetics*, Bamford C.H.; Compton R.G. (eds.); *Vol 26*; Elsevier; Amsterdam, **1986**.

2. Fick A. *Philosophica Magazine*, **1855**, *10*, 30.

3. Fick A. *Poggendorf's Annalen der Physic und Chemie*, **1855**, *94*, 59.

4. Brett C.M.A.; Brett A.M.O. *Electrochemistry*; Oxford University Press; Oxford, **1993**.

5. Compton R.G.; Fisher A.C.; Wellington R.G.; Dobson P.J.; Leigh P.A. *J. Phys. Chem.*, **1993**, *97*, 10410.

6. Bird R.B.; Stewart W.E.; Lightfoot E.N. *Transport Phenomena*; Wiley; New York, **1960**.

7. Levich V.G. *Physicochemical Hydrodynamics*; Prentice-Hall, Englewood Cliffs, **1962**.

8. Compton R.G.; Coles B.A. *J. Electroanal. Chem.*, **1983**, *144*, 87.

9. Unwin P.R.; Compton R.G. *Comprehensive Chemical Kinetics*, Compton R.G. (ed.); *Vol*

29; Elsevier; Amsterdam, **1989**.

10. Lévêque M.A. *Ann. Mines. Mem. Ser.* **1928**, *12, 13*, 201.

11. Vassos B.H.; Ewing G.W. *Electroanalytical Chemistry*, Wiley; New York, **1983**.

12. Anderson J.L.; Moldoveanu S. *J. Electroanal. Chem.*, **1984**, *179*, 107.

13. Compton R.G.; Pilkington M.B.G.; Stearn G.M. *J. Chem. Soc. Faraday Trans. 1*, **1988**, *84*, 2155.

14. Compton R.G.; Barghout R.; Eklund J.C.; Fisher A.C.; Bond A.M.; Colton R. *J. Phys. Chem.*, **1993**, *97*, 1661.

15. Compton R.G.; Wellington R.G.; Fisher A.C. *J. Phys. Chem.*, **1991**, *95*, 4749.

16. Britz D. *Digital Simulations in Electrochemistry*, Springer-Verlag; Berlin, **1988**.

17. Moldoveanu S.; Anderson J.L. *J. Electroanal. Chem.*, **1984**, *175*, 67.

18. Lapidus L.; Pinder G.F. *Numerical Solution for Partial Differential Equations in Science and Engineering*, Wiley; New York, **1982**.

19. Stephenson G. *Mathematical Methods for Science Students*; Longman; Harlow, **1973**.

20. Turq P.; Lantela F.; Chemla M. *Electrochim. Acta*, **1969**, *14*, 1081.

21. Wilke C.R; Chang P. *A.I.Ch.E.*, **1955**, *1*, 261.

22. Bentley T.W.; Harris H.C.; Koo I.S. *J. Chem. Soc. Perkin Trans. II*, **1988**, 783.

23. Atherton J.H. *Research in Chemical Kinetics*, Compton R.G.; Hancock G. (eds.); *Vol 2*; Elsevier; Amsterdam, **1994**.

24. Swain C.G.; MacLachlan A. *J. Am. Chem. Soc.*, **1960**, *82*, 6095.

Chapter 3

Experimental

This chapter describes the general procedures concerning the experimental methods developed in this thesis. These include the experimental setup for flow cell kinetic studies and experimental equipments employed as well as the source and purity of chemicals used. Discussion on the use of detection techniques such as AC impedance spectroscopy (chapter 4) and spectroelectrochemical measurements (chapters 7 and 8) for particular purposes will be addressed in the corresponding chapters.

1. The channel flow system

A schematic diagram of the channel flow cell (CFC) is given in fig. 7 of chapter 1. In general, the cell is composed of a rectangular duct cut (about 4.5 cm long, 0.1 cm deep and 0.6 cm wide) in a block of suitable material and closed by a cover plate. The cell geometry was measured to $\pm$ 0.002 cm using a travelling microscope with a vernier scale. The cell depth was determined either by means of a depth gauge (Mitutoyo, Japan) to $\pm$ 0.001 cm or spectroscopically to $\pm$ 0.002 cm via measurements on a series of known concentrations of potassium ferricyanide solution. There were slight differences in the design of various flow cells utilised in this work and the precise descriptions are given along with their applications.

Solution flow was accomplished by gravity feeding from a 500 cm^3 glass reservoir into the CFC. The reservoir solution was degassed using argon (fig. 1). A saturated calomel reference

electrode (Radiometer, Copenhagen) was placed upstream of the CFC for all electrochemical measurements. In the case of voltammetric experiments, a platinum gauze counter electrode (not shown in fig. 1) was located immediately downstream of the channel cell. Teflon tubing of 1.5 mm bore was adopted to direct the solution to/from the channel cell. Silicone rubber tubing was used to connect the teflon tubing to the cell. At the end of the teflon tubing, a capillary was attached to regulate the solution flow rate.

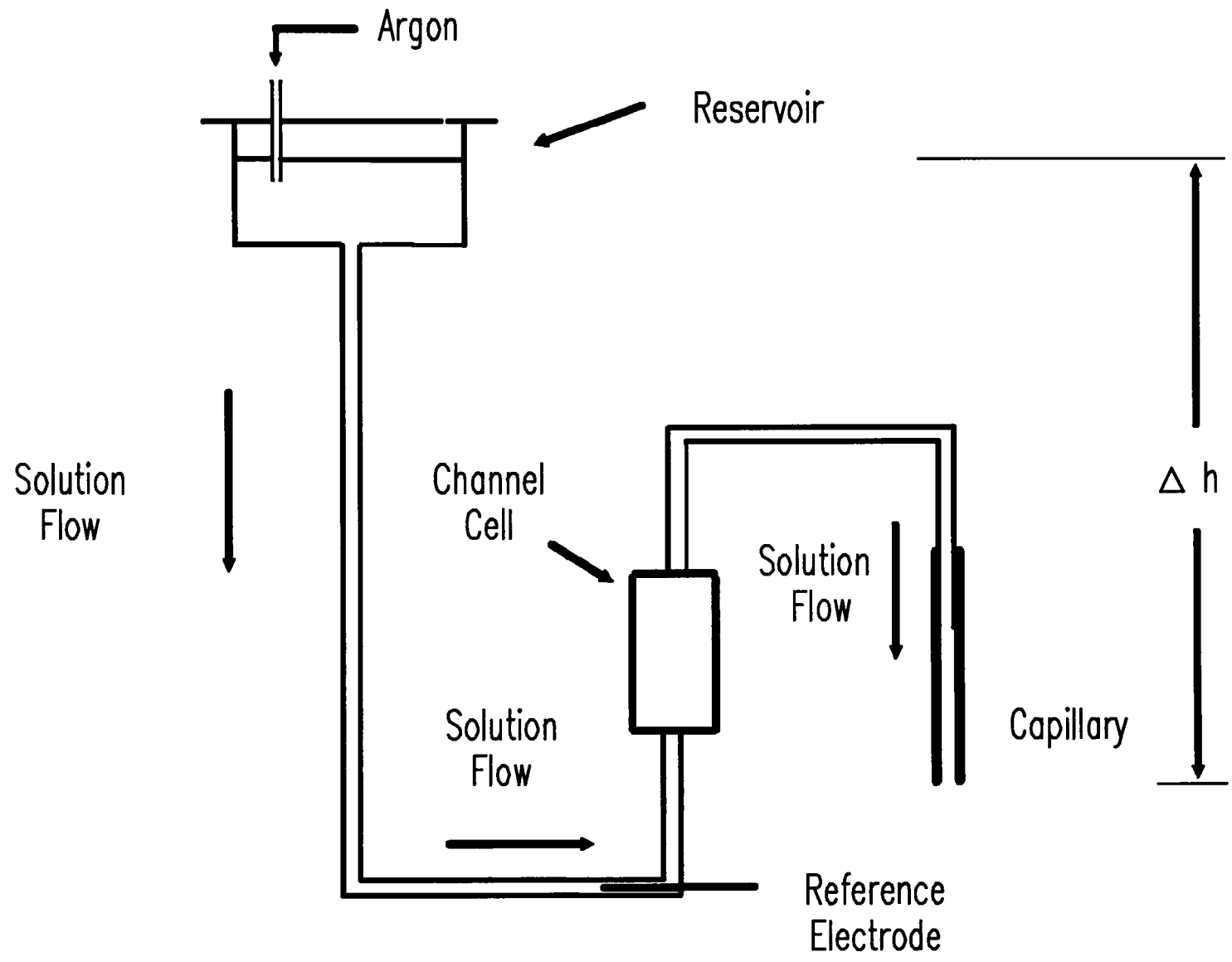


Fig. 1 The gravity fed flow system for the channel flow cell.

The solution flow rate and Δh (see fig. 1) was related by a calibration constant. This was accomplished by measuring the total volume of solution collected over a set time as a function of Δh. The flow rate was controlled in the range 10^{-4} - 10^{-1} cm^3 s^{-1} by selecting one of the three calibrated capillaries of different bores and varying the Δh value. In every flow

cell experiment, the temperature of the flow system was kept at the required temperature using an air-thermostat or thermostatted water.

2. Chloride detection

Chloride produced from the relevant kinetic processes was monitored by using either a home-made silver/silver chloride (Ag/AgCl) electrode or a commercially available chloride ion selective electrode. These sensors were calibrated in situ after each experiment by flowing known concentrations of chloride solution through the flow cell. All potentiometric measurements (with reference to a saturated calomel electrode) were performed using a digital pH meter (Jenway 3030, Essex). The procedures to prepare the sensor electrodes are given below:

Home-made Ag/AgCl electrode

A silver rod of diameter about 1.5 mm (Goodfellow Advanced Materials, Cambridge, purity > 99.95%) was polished until smooth by using diamond lapping compounds (Kemet, Kent). The finest size used was 1.0 μm. The silver wire was cleaned in 5 M nitric acid for about 1 minute and then chloridised by anodically polarising at 1.0 mA cm^{-2} for 45 minutes in 0.2 M potassium chloride solution.[1-3] This treatment produces a layer of silver chloride at the tip of the silver wire. Prior to use, the electrode was aged in 0.2 M potassium chloride solution for 15 minutes. The electrode was inserted through a hole of the same size drilled adjacent to the solid substrate. Molten wax (Wax-a-Way, from Vychem Ltd., Poole) was applied to the junction of the stem of the electrode with the external side of the cover plate. The wax was cooled to harden for sealing purpose.

Commercial chloride ion-selective electrode

A chloride ion-selective electrode (Orion 9417SC, Boston) was utilised in this study. The solid state sensing element (0.80 cm diameter) in the form of a disc was situated concentrically in a plastic sheath (1.15 cm diameter). The sensing element was flush with the surface of the sheath. The electrode was inserted through a hole of the same size drilled adjacent to the solid substrate. Sealing was accomplished by using two O-rings mounted at both ends of the hole.

3. Atomic force microscopy

A SFM-BD2 Park Scientific Instrument (Sunnyvale) or a TopoMetrix TMX 2000 machine (Santa Clara) were employed for all atomic force microscopic measurements. A commercial TopoMetrix liquid cell in conjunction with the latter machine was implemented for in situ imaging of the reacting crystal plane.

4. Ultraviolet-visible spectroscopy

Ultraviolet-visible measurements made use of a Unicam UV2-100 spectrophotometer (ATI Unicam, Cambridge). Spectroscopic access to the channel flow cell was accomplished by using a pair of high quality fused silica windows (U.Q.G. Ltd., Cambridge, 4 mm diameter, 3 mm thickness).

5. AC impedance spectroscopy

A Solartron 1250 frequency response analyzer and a Solartron 1286 electrochemical interface (Schlumberger Electronics Ltd., Franborough) were adopted for impedance studies.

6. Voltammetry experiments

An Oxford Electrodes potentiostat and an Oxford Electrodes linear potential scan generator were used in voltammetric studies. All voltammograms were acquired using a Hewlett Packard 7035B X-Y recorder. Electrodes were made of platinum foil (Goodfellow Advanced Materials, Cambridge, thickness 0.1 mm, purity 99.95%) with dimensions of approximately 4 mm x 4 mm.

7. Computing

All supporting computer programs were developed in SPICE, FORTRAN 77 or UNIX C environments. These were executed either on a VAX/VMS machine or a SUN SPARC workstation. NAG routines[4] were invoked for random number generation and for part of the kinetic parameters optimisation calculations.

8. Chemicals used

All solutions were prepared using deionised water of resistivity > 10^7 Ω-cm and degassed thoroughly with argon or nitrogen prior to use. All chemicals used are tabulated as follows:

Chemical	Abbreviation	Source	Grade
Acetyl chloride	CH_3COCl	Aldrich	99% ACS reagent
Argon	Ar	BOC	> 99.999%
Cotton fabrics/fibres	-	Zeneca	Knitted Indian
Methyl viologen dichloride	$MVCl_2$	Aldrich	98%
Nitrogen	N_2	BOC	> 99.999%
Pentane	C_5H_{12}	Aldrich	> 99%
Potassium chloride	KCl	Aldrich	99% ACS reagent
Potassium ferricyanide	$K_3Fe(CN)_6$	Aldrich	99% ACS reagent
Potassium nitrate	KNO_3	BDH, Poole	AnalaR®
Procion Blue MX-R	-	Zeneca	80.7%
Sodium carbonate	Na_2CO_3	BDH, Poole	AnalaR®
Sodium chloride	NaCl	BDH, Poole	AnalaR®
Sodium hydroxide	NaOH	BDH, Poole	Technical
Sodium sulphate decahydrate	Na_2SO_4	Aldrich	99% ACS reagent
N,N,N′,N′-tetramethyl-p -phenylenediamine	TMPD	Aldrich	98%
Triphenylmethanol	TPMOH	Aldrich	97%
Triphenylmethyl chloride	TPMCl	Aldrich	98%

9. References

1. Ives D.J.G.; Janz G.J. *Reference Electrode*, Academic Press; New York, **1961**.

2. Cammann K. *Working with Ion-Selective Electrode*, Springer-Verlag; Berlin, **1979**.

3. Luo J.L.; Lu Y.C.; Ives M.B. *J. Electroanal. Chem.*, **1992**, *326*, 51.

4. *NAG Fortran Library Manual* - Mark 13, Numerical Algorithm Group Ltd.; Oxford, **1988**.

Chapter 4

A Channel Flow Cell with Downstream AC Impedance Detection

1. Introduction

The measurement of electrochemical impedance is a powerful diagnostic method in the study of electrochemical systems.[1-3] The principle of the impedance technique involves the application of a small perturbation, usually in the form of a sinusoidal potential, to an electrochemical system. The response to this perturbation permits the impedance of the system to be measured and subsequently provides a basis for the analysis of the related processes.[4] Commercial frequency response analysers, such as the one utilised in our laboratory (Solartron Instruments), are able to measure broad impedance values using a wide range of perturbation signal frequencies. This is a competent way of interrogating the relevant chemical and/or physical processes.

To date, the detection systems employed in channel flow cell (CFC) experiments have included the use of fluorescence spectroscopy[5,6] although electrochemical approaches have been more generally implemented. In particular potentiometric[7] and amperometric[8] detection have proved successful. Both these methods, however, operate best under conditions of relatively high ionic strength and, more significantly, are restricted to certain substrates for which either ion selective

electrodes can be fabricated or which display useful voltammetric activity. A more general detection method capable of wide application, good sensitivity and which is appropriate under conditions of low ionic strength would be based on the use of conductivity measurements downstream of the reacting interface.[9,10] This chapter describes a CFC in which impedance spectroscopy measurements are made as a function of frequency to accomplish this purpose. Theory relating the observed impedance to the nature and rate of the interfacial process is developed and requires the modelling of the conductivity response of inhomogeneous ionic concentration profiles established in the flowing system downstream of the solid-liquid reaction.

This new detection system is applied to the study of the hydrolytic dissolution of triphenylmethyl chloride (TPMCl) in water:

$$\mathbf{TPMCl\ (s)\ +\ H_2O\ (l)\ \rightarrow\ TPMOH\ (aq,\ s)\ +\ H^+\ (aq)\ +\ Cl^-\ (aq)}$$

where TPMOH is triphenylmethyl alcohol. The process is demonstrated to proceed *at* the solid-liquid interface rather than after prior dissolution of solid TPMCl.

2. Flow cell design and impedance measurements

The CFC is composed of a rectangular duct (approximately 4.5 cm long, 0.1 cm deep and 0.6 cm wide) cut in a perspex block and closed by a cover plate. The solid substrate, usually in the form of a pellet or single crystal, is embedded in the cover plate and is masked with thin teflon tape so that a known area of solid is exposed to solution. The solution is flowed through the channel by entering through an inlet pipe and exiting via an outlet pipe. These pipes are built into

the ends of the perspex cell unit. For impedance spectroscopy detection two platinum detector electrodes (of approximate dimensions 4 mm x 4 mm) were situated downstream of the solid substrate and directly opposite to one another as shown in fig. 1.

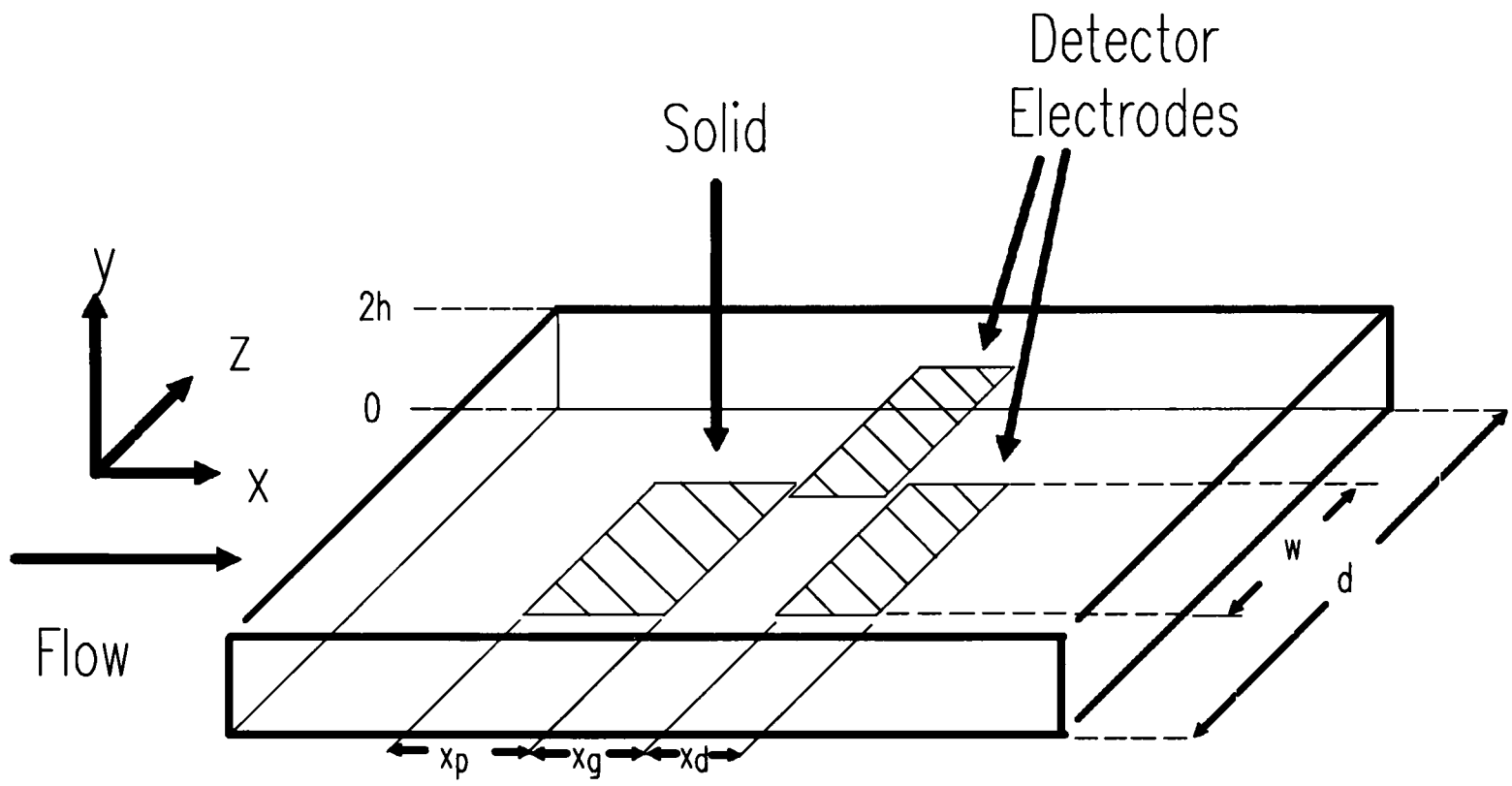


Fig. 1 Schematic diagram of the CFC adapted for impedance studies.

The impedance measurement system consisted of a Solartron Instruments (Schlumberger Electronics Ltd., Farnborough, UK) 1250 Frequency Response Analyser (FRA) and a Solartron Instruments 1286 Electrochemical Interface (ECI). The 1250 FRA and 1286 ECI were fully computer controlled via an IEEE 488 interface. A third party software, ZPLOT (V1.60), was invoked for the electrochemical impedance spectrum acquisition, graphical display of data and simple data analysis (Scribner Associates Inc., 1992). A two-electrode configuration was used. This was accomplished by shorting, respectively, the RE1 (Reference Electrode 1) to the CE (Counter Electrode) and RE2 (Reference Electrode 2) to the WE (Working Electrode) in the ECI.

In addition, a blocking capacitor of 20 μF was connected to the CE-RE1 connector before it was linked to a detector electrode in order to eliminate any DC current that might otherwise have led to undesirable faradaic processes at the lower frequencies employed. Generally impedance spectra in the frequency range of 400-40000 Hz were measured. Preliminary characterisation of the conductivity detection system employed three different concentration of potassium nitrate solutions, namely 1.51 mM, 0.038 M and 0.189 M uniformly distributed in the cell. All impedance measurements reported in this chapter were conducted at 25 $\pm$ 0.5 °C. The resulting impedance spectra are given in fig. 2 and are considered further in the following section.

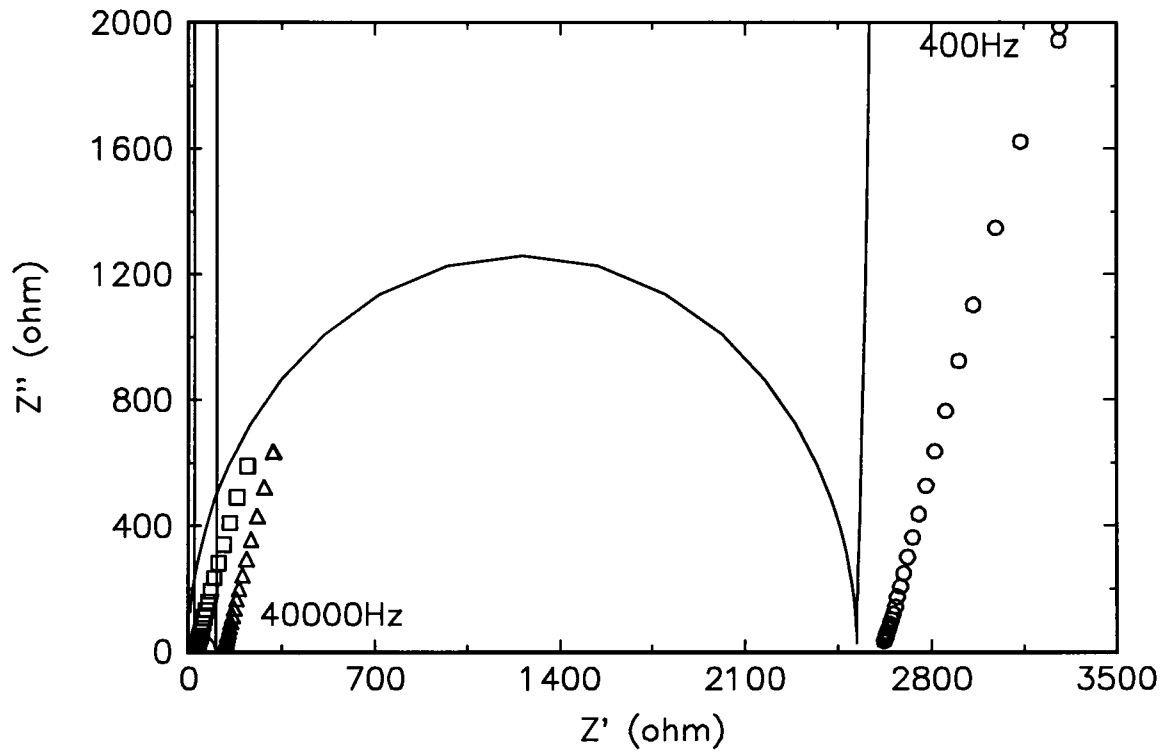


Fig. 2 Experimental impedance spectra obtained from 1.51 mM (o), 0.038 M (Δ) and 0.189 M (□) potassium nitrate solutions. The solid lines represent the predictions of the SPICE theory as outlined in the text.

3. Background theory and the use of SPICE

In order to interpret CFC experimental results acquired using impedance spectroscopy detection it is necessary to model the "conductivity" response of the detector electrodes to the inhomogeneous concentration profiles distributed within the flow cell. To this end the circuit analysis package, Simulation Program with Integrated Circuit Emphasis (SPICE), was employed. SPICE software was originally developed at UC Berkeley in the early 1970s.[11-13] It provides an accurate and cost-effective means of simulating and verifying electronic circuit performance. When using SPICE a description of the related circuit, such as the values of the components and their connections to neighbouring points, which are called nodes, have to be specified. The circuit information is then transformed into a set of equations and solved by SPICE.[14,15]

We consider first the simple circuit shown in fig. 3 as a model to illustrate the application of SPICE. The circuit represents a naive model for the behaviour that might be expected from the CFC if the electrolyte solution was uniformly and homogeneously distributed so that the capacitor connected between nodes 1 and 2 represents the electrode capacitance (C_{elec}) whilst the resistor R_{soln} and the capacitor C_{soln} connected between nodes 2 and 0 denote, respectively, the solution resistance and capacitance.

Let the impedance of the circuit be

$$\mathbf{Z} = \mathbf{Z}' - \mathbf{Z}'' \mathbf{j}$$

with $j = \sqrt{-1}$. Assuming ϖ represents the input ac signal frequency, the analytical expression for

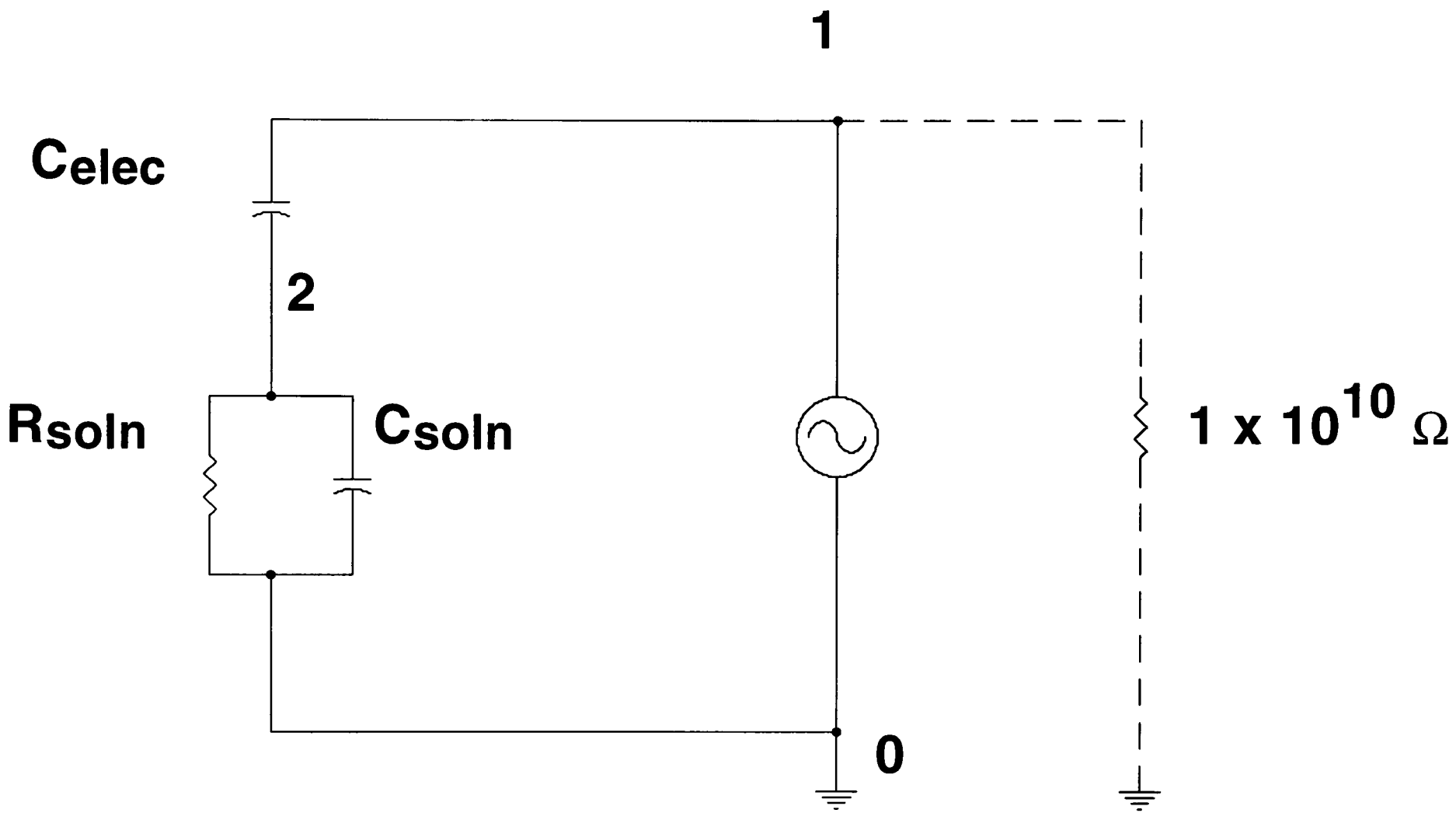


Fig. 3 A simplified circuit to model the impedance spectra.

Z_{10} can be derived as follows:

$$\frac{1}{Z_{20}} = j\varpi C_{soln} + \frac{1}{R_{soln}}$$

$$Z_{20} = \frac{R_{soln}}{1 + j\varpi R_{soln} C_{soln}}$$

$$Z_{10} = Z_{20} + Z_{12}$$

$$= \frac{R_{soln}}{1 + j\varpi R_{soln} C_{soln}} + \frac{1}{j\varpi C_{elec}}$$

$$= \frac{j\varpi R_{soln} C_{elec} + (1 + j\varpi R_{soln} C_{soln})}{j\varpi C_{elec}(1 + j\varpi R_{soln} C_{soln})}$$

$$= \frac{j\varpi R_{soln}(C_{elec} + C_{soln}) + 1}{j\varpi C_{elec}(1 + j\varpi R_{soln} C_{soln})}$$

$$= \frac{j\varpi R_{soln}(C_{elec} + C_{soln}) + 1}{\varpi C_{elec}(j - \varpi R_{soln} C_{soln})} \times \frac{j + \varpi R_{soln} C_{soln}}{j + \varpi R_{soln} C_{soln}}$$

$$= \frac{j\varpi R_{soln}(C_{elec} + C_{soln})(j + \varpi R_{soln} C_{soln}) + (j + \varpi R_{soln} C_{soln})}{-\varpi C_{elec}(1 + (\varpi R_{soln} C_{soln})^2)}$$

$$= \frac{\varpi R_{soln} C_{elec} - j(\varpi^2 R_{soln}^2 C_{elec} C_{soln} + (\varpi R_{soln} C_{soln})^2 + 1)}{\varpi C_{elec}(1 + (\varpi R_{soln} C_{soln})^2)}$$

Collecting the real and imaginary parts,

$$Z' = \frac{\varpi R_{soln} C_{elec}}{\varpi C_{elec}(1 + (\varpi R_{soln} C_{soln})^2)} \quad (1)$$

and

$$Z'' = \frac{-(C_{elec} C_{soln}(\varpi R_{soln})^2 + (\varpi R_{soln} C_{soln})^2 + 1)}{\varpi C_{elec}(1 + (\varpi R_{soln} C_{soln})^2)} \quad (2)$$

The following SPICE program demonstrates how the simulator can be adopted to evaluate the real and imaginary impedance as a function of frequency for the circuit shown in fig. 3. Remarks are added after the symbol "*" for the purpose of clarification. Note that in the program, we assume C_{elec} = 0.47 μF, R_{soln} = 4700 Ω and C_{soln} = 4.7 pF.

```
Test Circuit                    * Title line (required by SPICE)
ISOURCE 1 0 DC 0 AC 1 0         * AC source across node 1-2: 1A, 0V (DC)
CELEC 1 2 0.47E-6               * Define C_elec across node 1-2
RGND  0 1 1E10                  * Virtual DC path to connect node 1 to ground node (0)
CSOLN 0 2 4.7PF                 * Define C_soln across node 0-2
RSOLN 0 2 4700                  * Define R_soln across node 0-2
.AC DEC 10 200 40000            * AC frequency decade-sweep from 200 to 40000Hz
.PRINT AC VR(1) VI(1)           * List the real & imaginary voltage in node 1
.END                            * END statement to terminate (required by SPICE)
```

Since the second line defines an AC source with unit amplitude, the magnitude of the voltages output in the .PRINT statement are equivalent to the sought impedance values. Note that the circuit, as shown in fig. 3, is assembled with node 1 and node 0 hooked to the two-electrode FRA-ECI setup. Fig. 4a depicts the impedance spectra obtained from the SPICE simulation together with the circuit response measured as outlined above from a dummy cell representing the circuit given in fig. 3 and as calculated from the analytical eqns. 1 and 2. There is excellent

agreement between the spectra. Fig. 4b shows the results of the corresponding exercise when C_{soln} is changed to 0.01 μF. Note that in both figs. 4a and 4b the value of R_{soln} can be obtained by extrapolation of the low frequency data to the real impedance (Z′) axis.

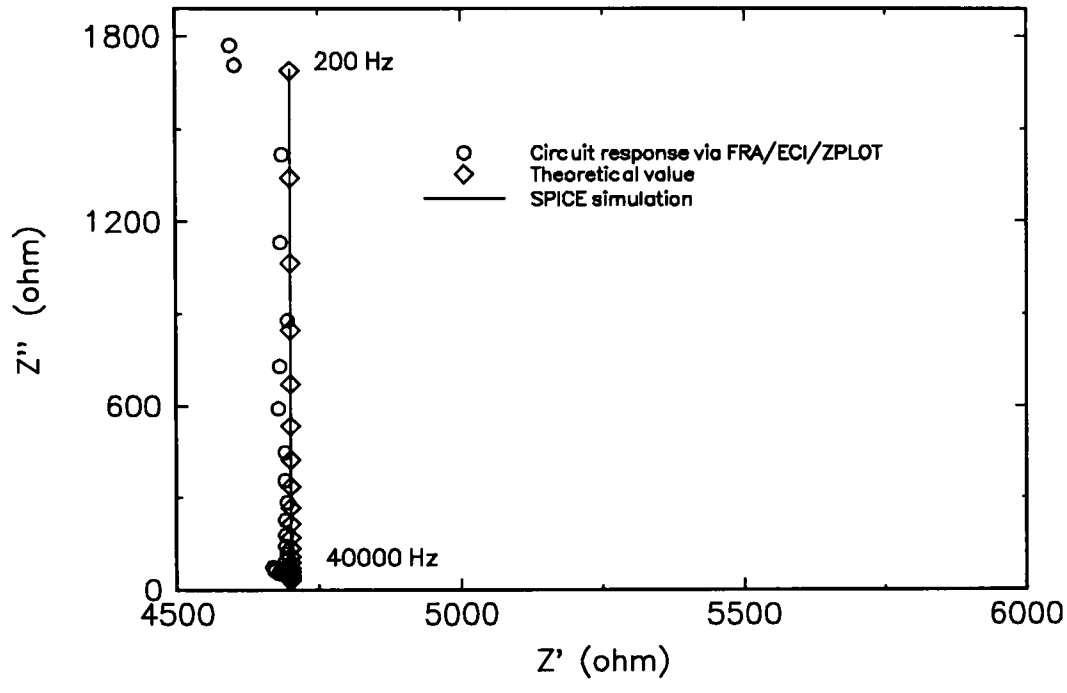


Fig. 4a Impedance spectra for the circuit shown in fig. 3: C_{elec} = 0.47 μF, R_{soln} = 4700 Ω and C_{soln} = 4.7 pF.

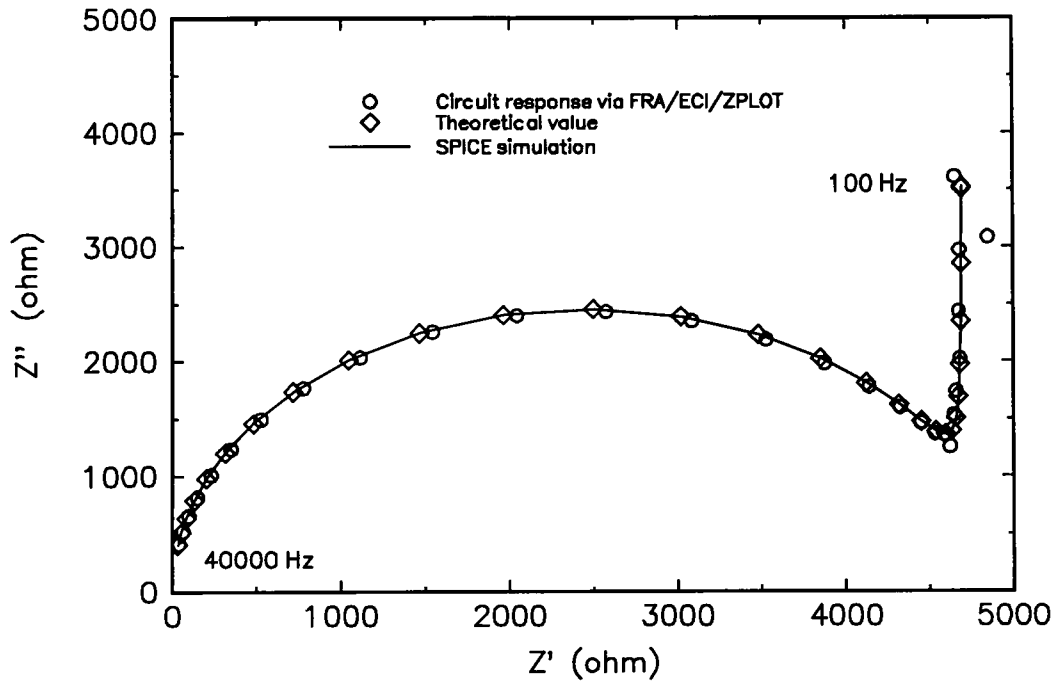


Fig. 4b Impedance spectra for the circuit shown in fig. 3: C_{elec} = 0.47 μF, R_{soln} = 4700 Ω and C_{soln} = 0.01 μF.

We next consider the application of SPICE to the computation of the impedance spectroscopy response of the pair of detector electrodes shown in fig. 1, to the release of ions formed upstream at the solid-liquid interface under interrogation. The problem is now complicated by the fact that the distribution of ions throughout the flow cell is no longer homogeneous - i.e. the concentrations, c_i, of each ion, i, depends on x and y: $c_i = c_i(x,y)$. The values of c_i may be predicted for any model of the interfacial process of interest using established procedures as described subsequently. We then use the network illustrated in fig. 5 to allow for the spatial variation of the solution properties. In this, the impedance is denoted by a 20x80-element resistor/capacitor network with each element consisting of four resistors and four capacitors and such that the middle 40 elements are connected to the electrodes and the remaining 20x20 elements located each side to account for "edge effects" as illustrated in fig. 6 and discussed further below. Fig. 5 shows both the network and the circuit diagrams for each element. The following equations are used to calculate the equivalent resistance (R) and capacitance (C) of each component in an element

$$R = \frac{1}{(\Sigma c_i K_i)} \times \frac{L}{A} \qquad (3)$$

and

$$C = \frac{\epsilon_o \epsilon_r WL}{H} \qquad (4)$$

where L, A, H and W denote, respectively, the length, area, height and width of the element with respect to the particular x or y direction under consideration. K_i (S cm^2 mol^{-1}) denotes the molar ionic conductivity and c_i the concentration. ε_o is the permittivity of a vacuum (8.85 x 10^{-14} J^{-1} C^2

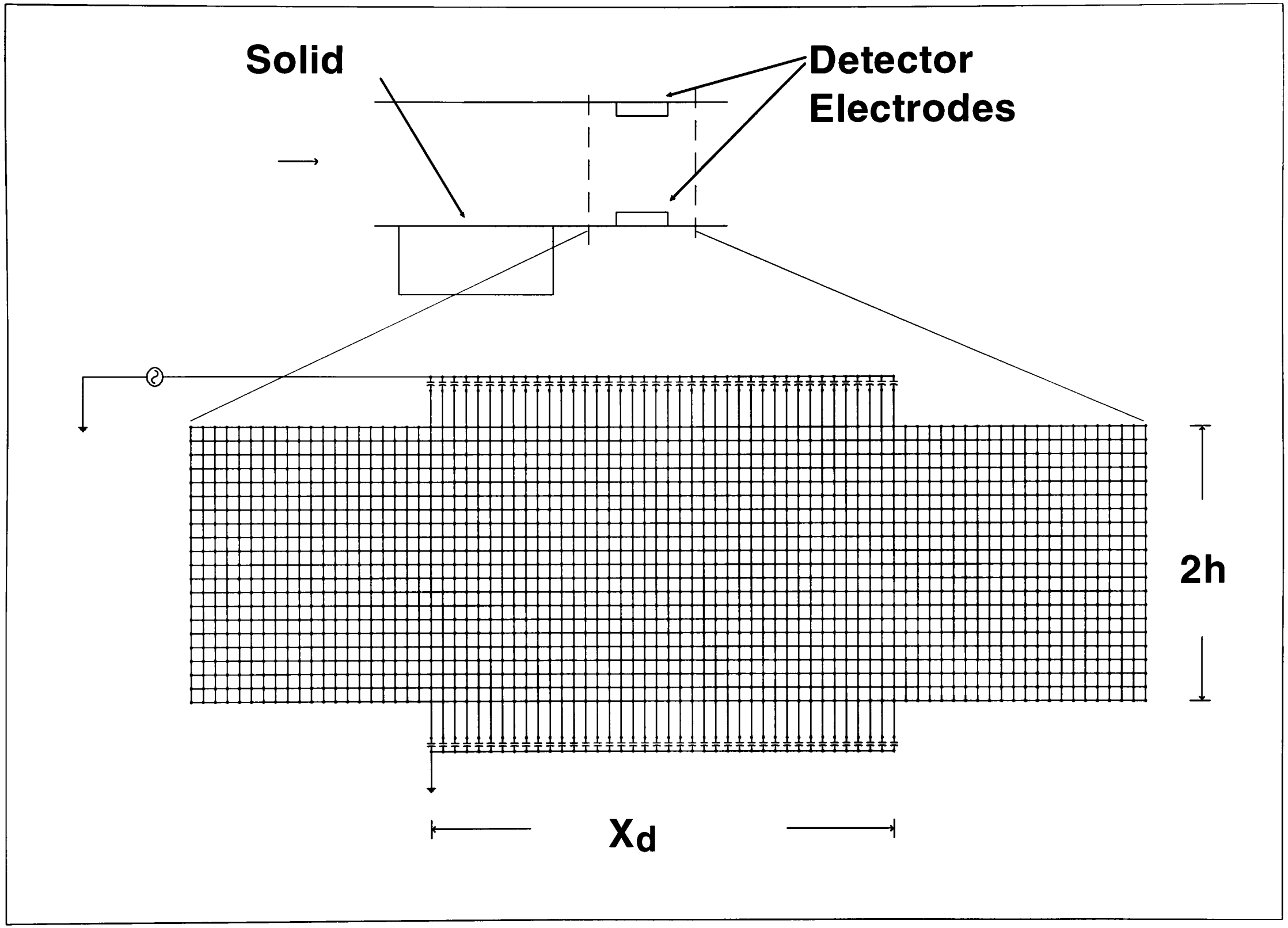


Fig. 5a Schematic diagram showing the network used for the SPICE calculations.

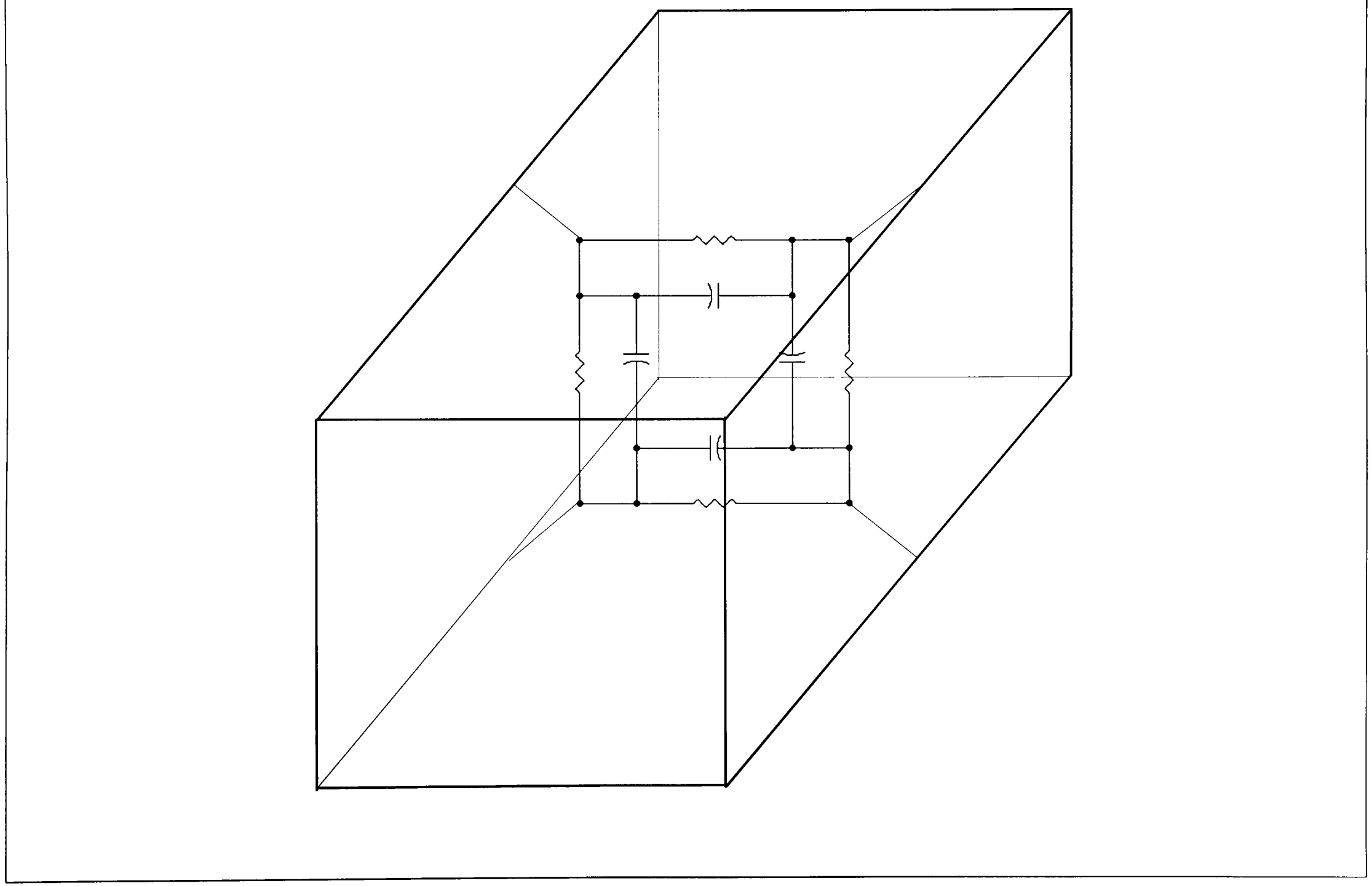


Fig. 5b The electrical circuit used for each element.

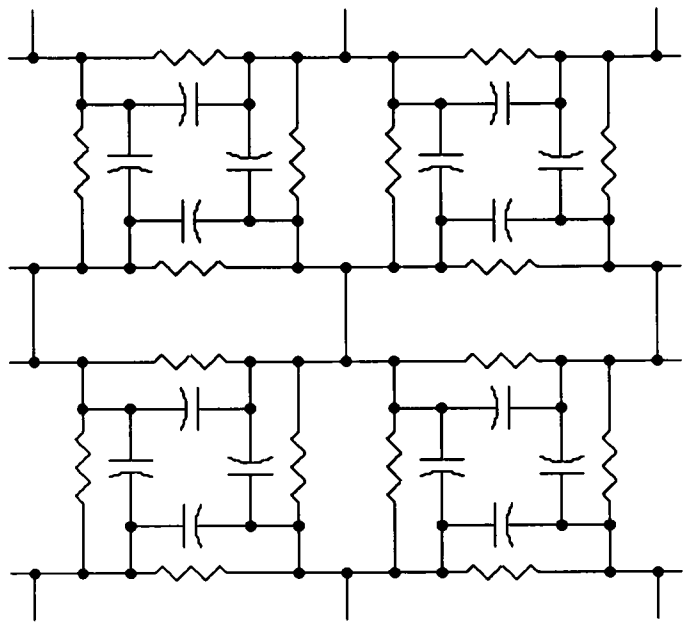


Fig. 5c The electrical circuit representing four neighbouring solution elements.

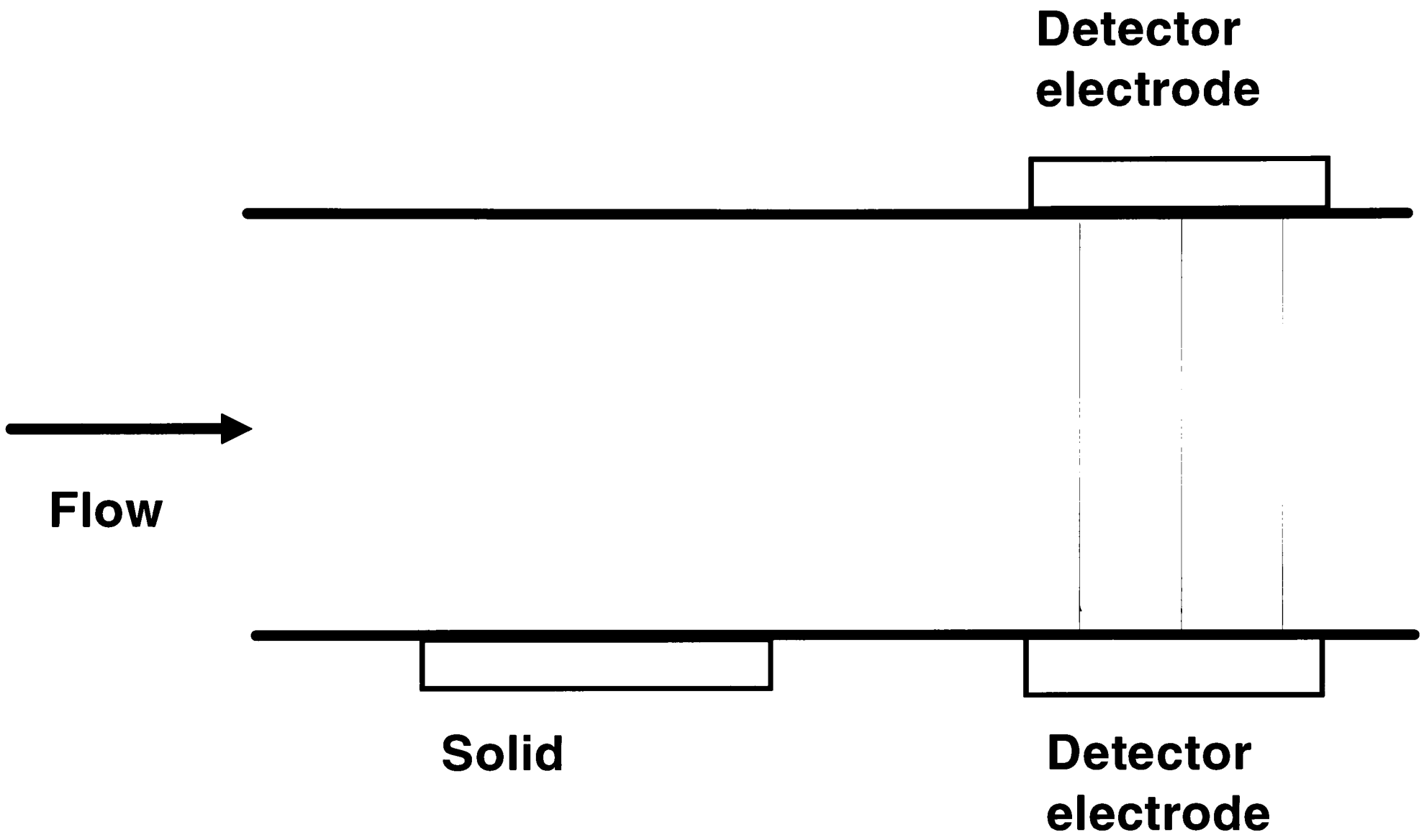


Fig. 6a Schematic behaviour of the current flow between the detector electrodes for a simple model which neglects "edge effects".

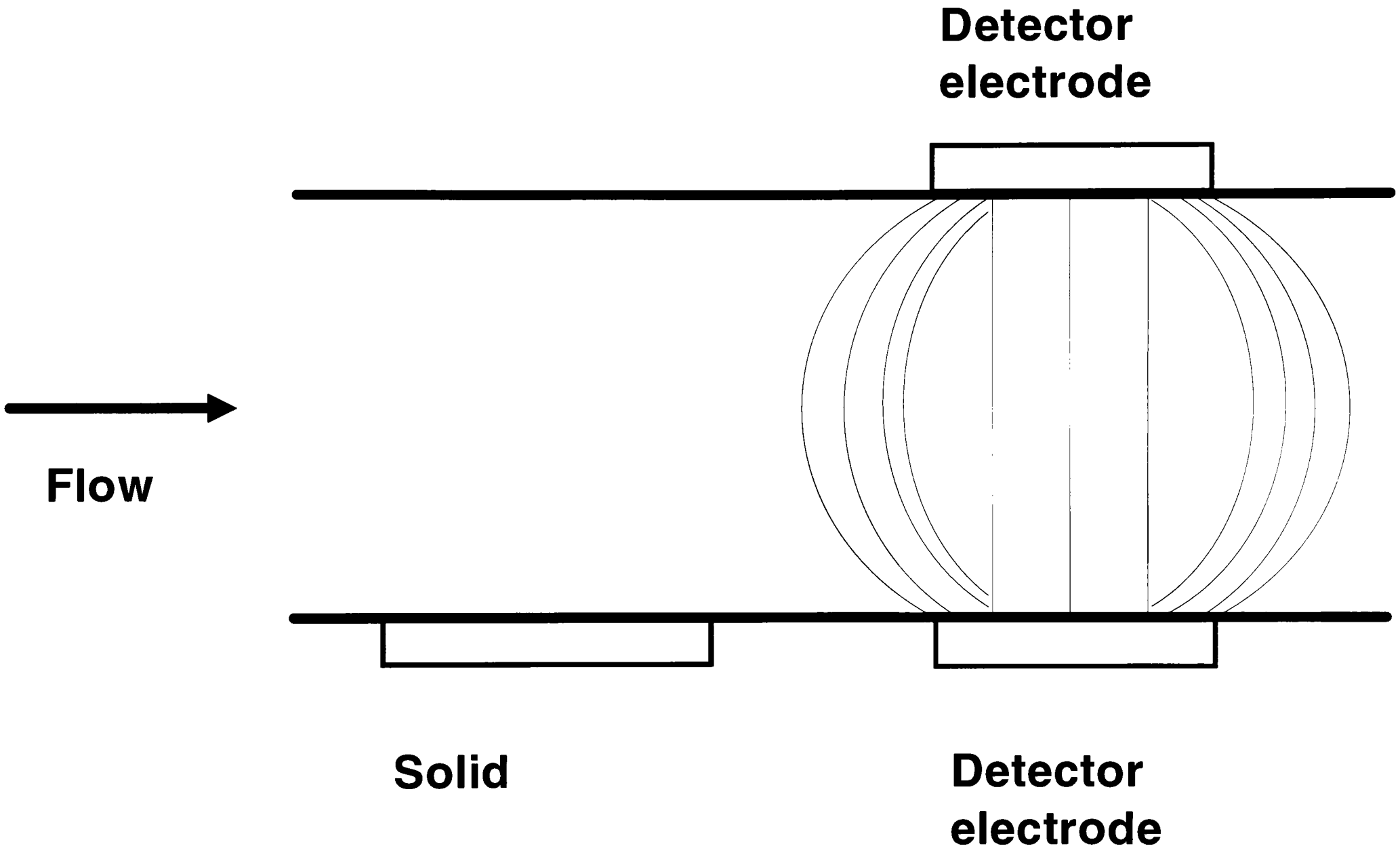


Fig. 6b Schematic behaviour of the current flow between the detector electrodes for a modified model which includes "edge effects".

cm^{-1}) and ε_r is the relative dielectric constant of water at (= 78.54 at 25 °C). Applying eqns. 3 and 4 to fig. 5a gives for the x-direction:

$$R_x = 2\times\frac{1}{c\ \mathrm{K}} \times \left(\frac{x_d\ /40}{w\ 2h\ /20}\right) = \frac{x_d}{2\ c\ \mathrm{K}\ w\ h} \qquad (5)$$

and

$$C_x = \frac{1}{2} \times \frac{\epsilon_o\ \epsilon_r\ (2h\ /20)\ w}{x_d\ /40} = \frac{2\ \epsilon_o\ \epsilon_r\ h\ w}{x_d} \qquad (6)$$

where 2h is the cell height, x_d is the length of the detector electrode and w is the width of the electrodes. For the y-direction:

$$R_y = 2 \times \frac{1}{c\ \mathrm{K}} \times \left(\frac{2h\ /20}{w\ x_d\ /40}\right) = \frac{8h}{c\ \mathrm{K}\ x_d\ w} \qquad (7)$$

and

$$C_y = \frac{1}{2} \times \frac{\epsilon_o\ \epsilon_r\ (x_d\ /40)\ w}{2h\ /20} = \frac{\epsilon_o\ \epsilon_r\ x_d\ w}{8h} \qquad (8)$$

In addition to the 20x80 elements two arrays of capacitors are connected to the network to symbolise the electrode double layer capacitance (see fig. 5a). The values of these capacitors are adjusted according to the experiment being simulated. For example, a value of 0.023 μF was found to be appropriate for the TPMCl experiments described below.

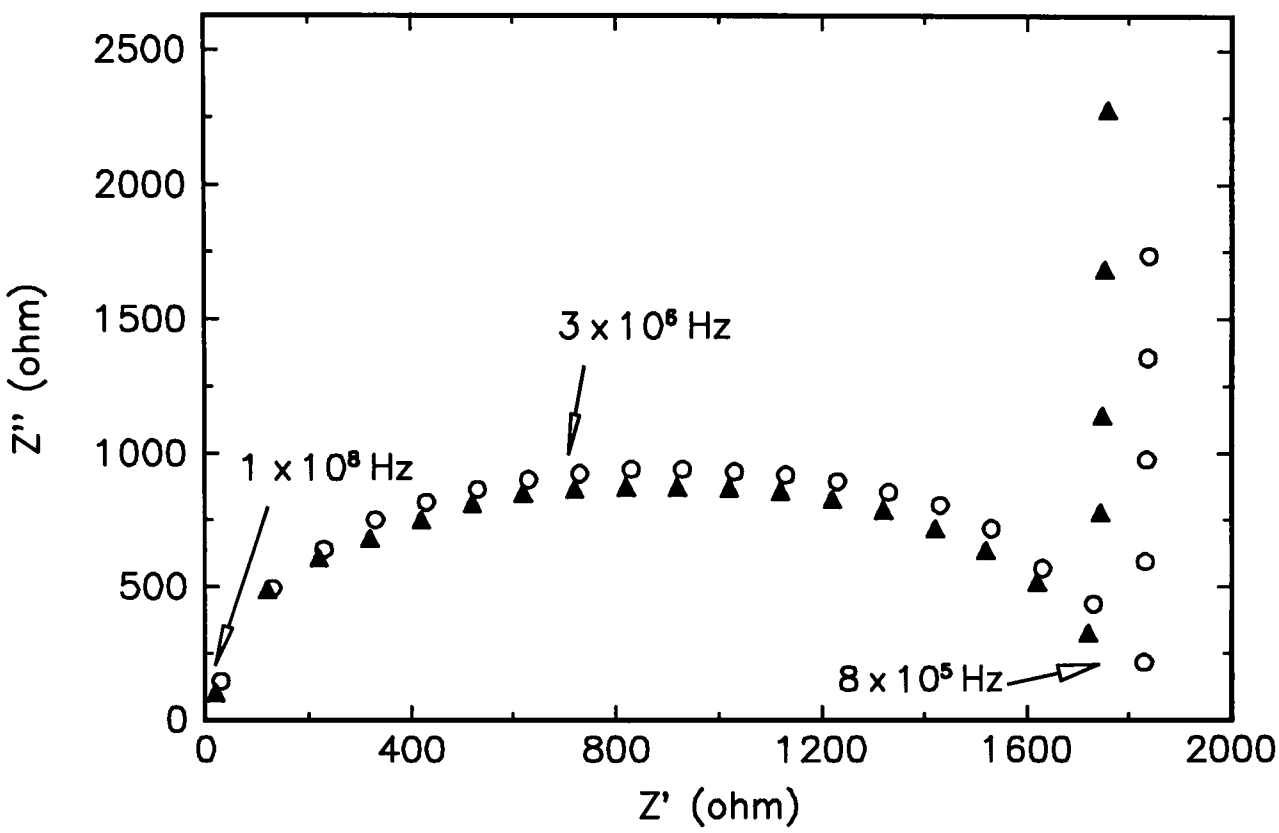
Programs were developed in Fortran 77 and SPICE environments to deduce theoretical impedance spectra. The Fortran program calculated all the required parameters, including values of c_i as described below, needed for the SPICE computation. It then invoked SPICE as a sub-program

via a Unix "system" command and subsequently evaluated by extrapolation, the solution resistance value, R_{soln}, from the SPICE output impedance spectrum file.

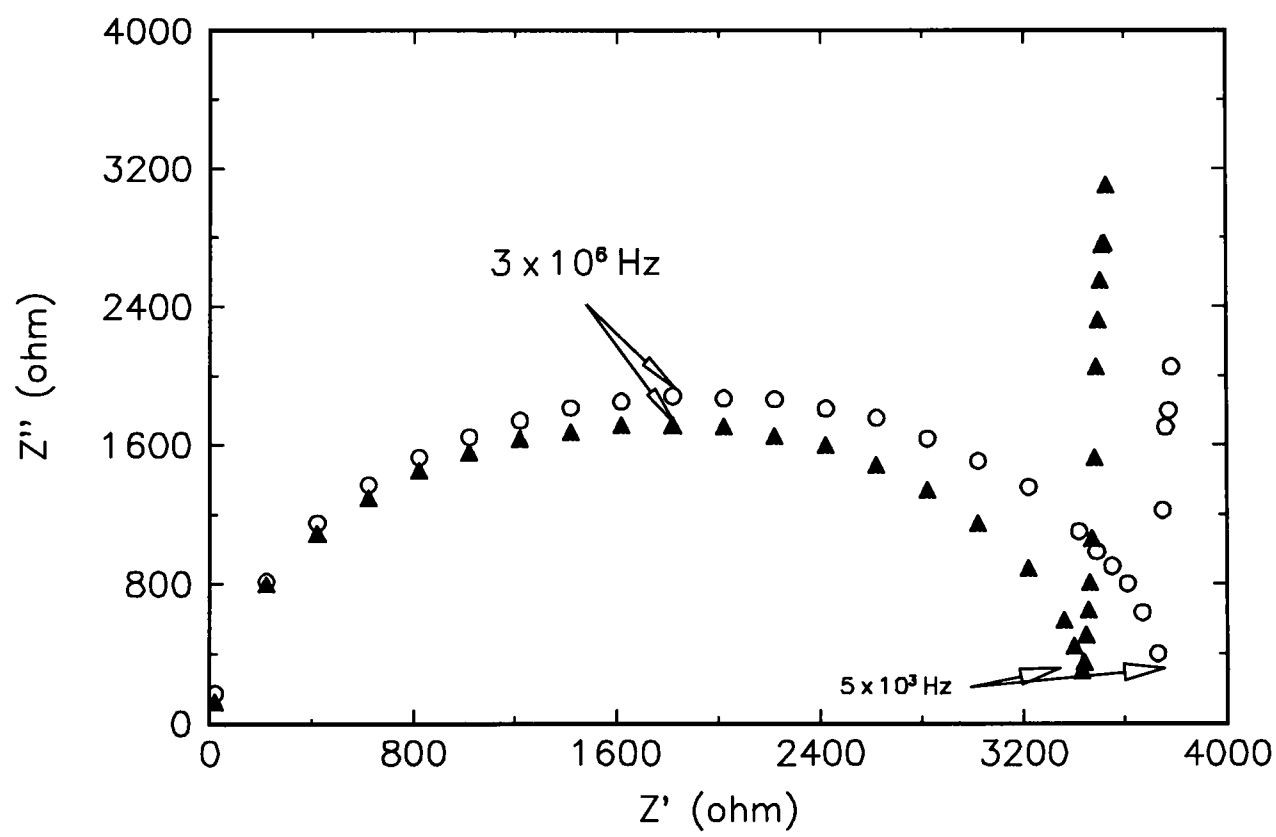
Fig. 2 shows the theoretical impedance spectra computed for homogenous distributions of KNO_3 in the flow cell corresponding to the concentrations cited at the end of section 2. Comparison of the theoretical and experimental points shows good agreement, particularly with respect to the intercepts on the real axis (Z') as are used in subsequent analysis. This vindicates both the general approach and the implementation described above.

Finally, it is interesting to examine the consequences of neglecting "edge effects".[9] Fig. 7 shows impedance plots obtained theoretically using SPICE for detector electrodes of size 4 x 4 mm with different cell heights (2h): 0.0445, 0.0890, 0.1780, 0.3560 and 0.7120 cm for a uniformly distributed electrolyte containing 1 mM KCl (molar conductivity 146.95 S cm^2 mol^{-1} [16]). The corresponding extrapolated resistance values are summarised in table 1. It can be seen that appreciable deviations from the naive no-edge effects model are significant for all, except for, perhaps, the shallowest cell height examined. This emphasises the need for the inclusion of significant distances in the cell both up- and down-stream of the detector electrode pair. It is interesting to note that when edge effects are neglected the high frequency loop is authentically semi-circular but that this is no longer the case when edge effects are included.

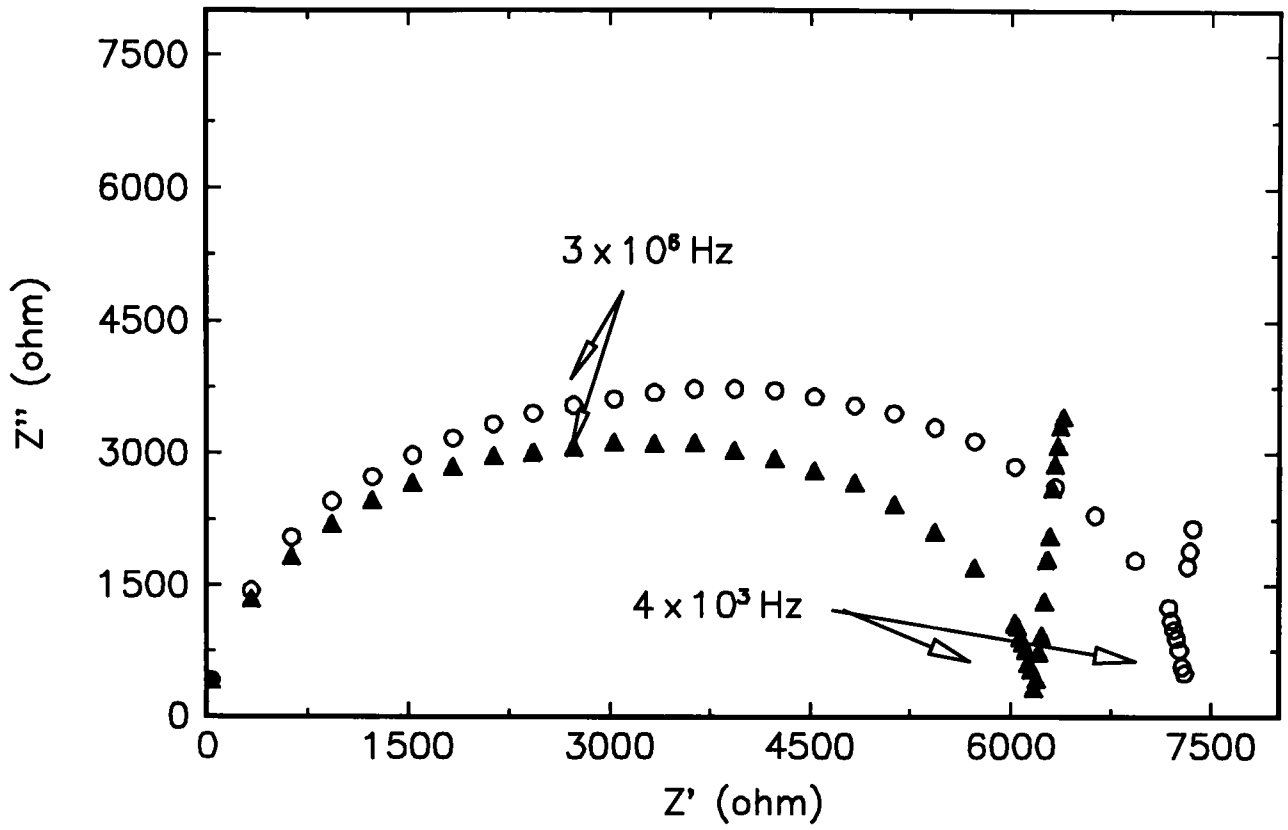
Fig. 7 Impedance spectra generated through SPICE showing the behaviour calculated with (▲) and without (o) "edge effects" for cell heights (2h) of (a) 0.0445, (b) 0.0890, (c) 0.1780, (d) 0.3560 and (e) 0.7120 cm



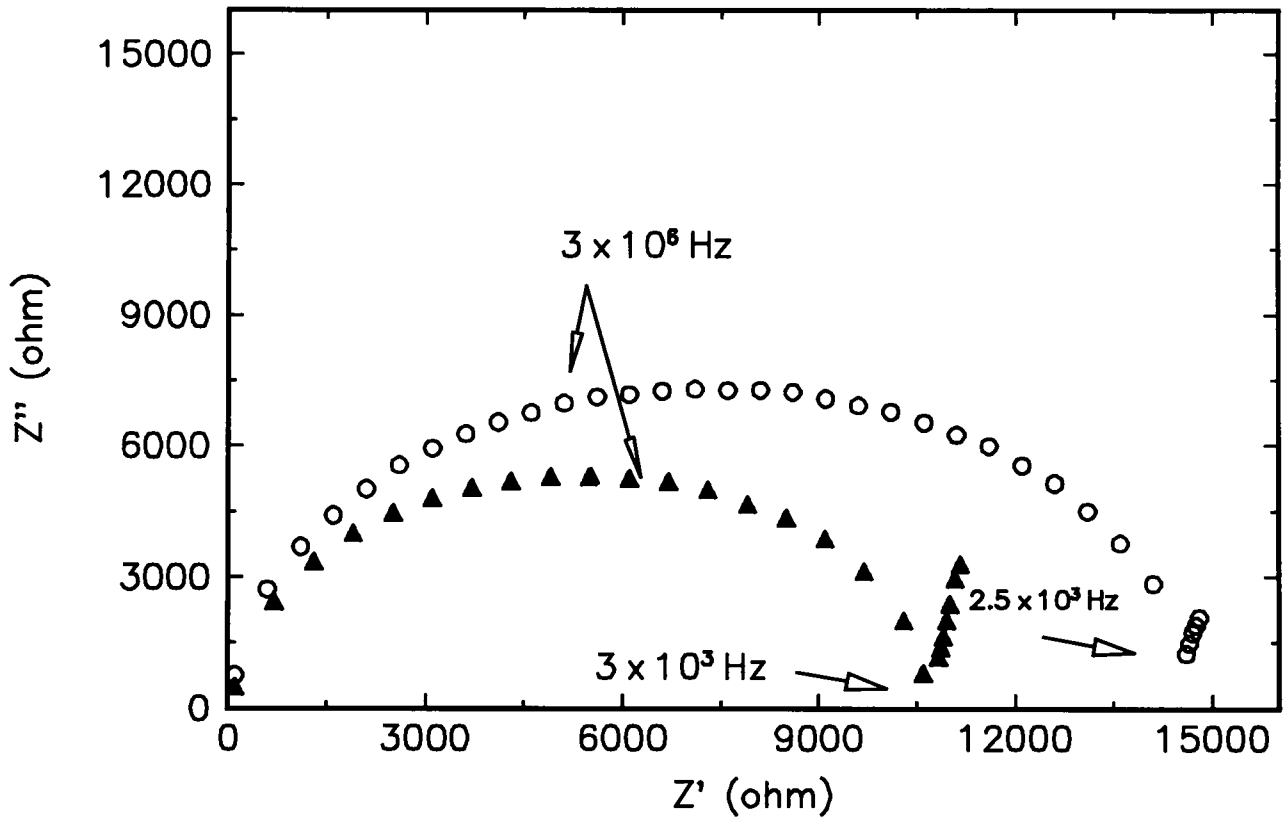
(a)



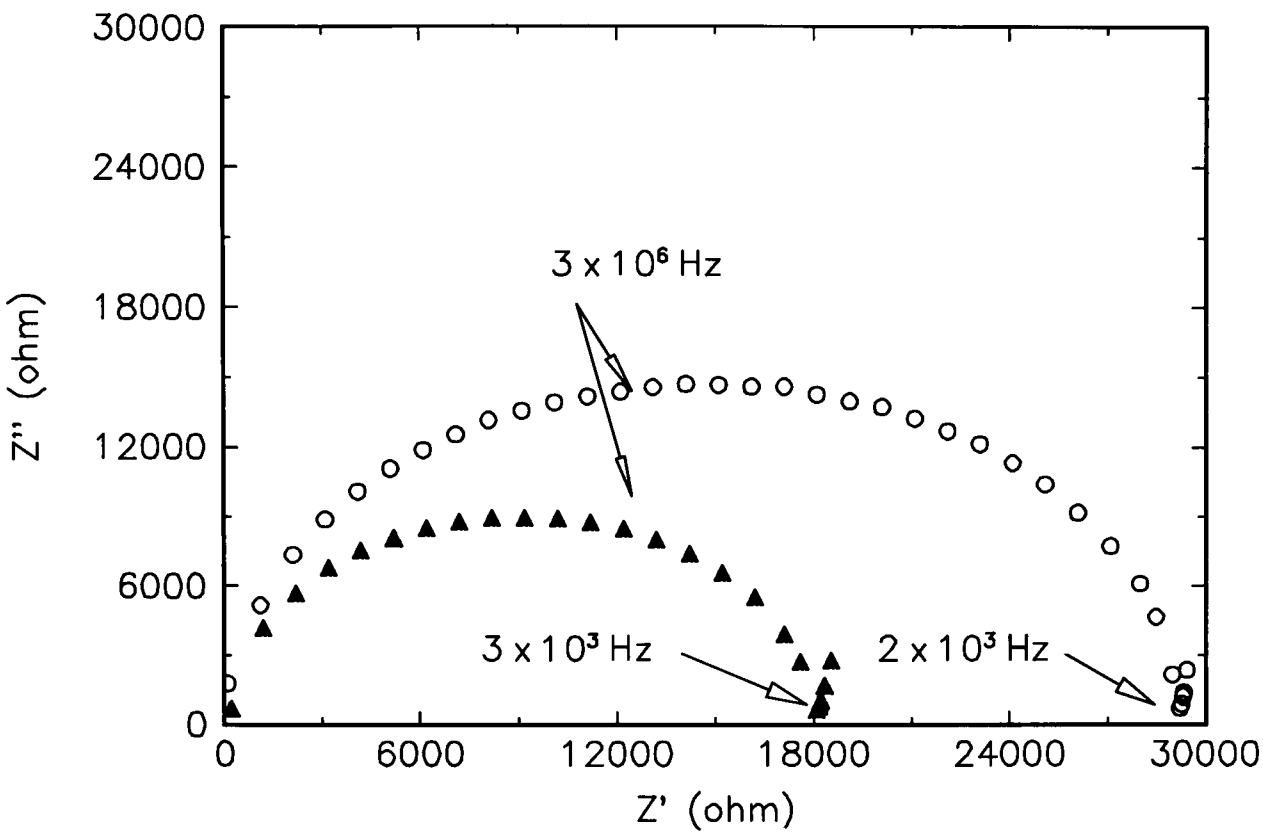
(b)


7500
6000
4500
3000
1500
0
Z'' (ohm)
3 x 10⁶ Hz
4 x 10³ Hz
0
1500
3000
4500
6000
7500
Z' (ohm)

(c)


15000
12000
9000
6000
3000
0
Z'' (ohm)
3 x 10⁶ Hz
2.5 x 10³ Hz
3 x 10³ Hz
0
3000
6000
9000
12000
15000
Z' (ohm)

(d)



(e)

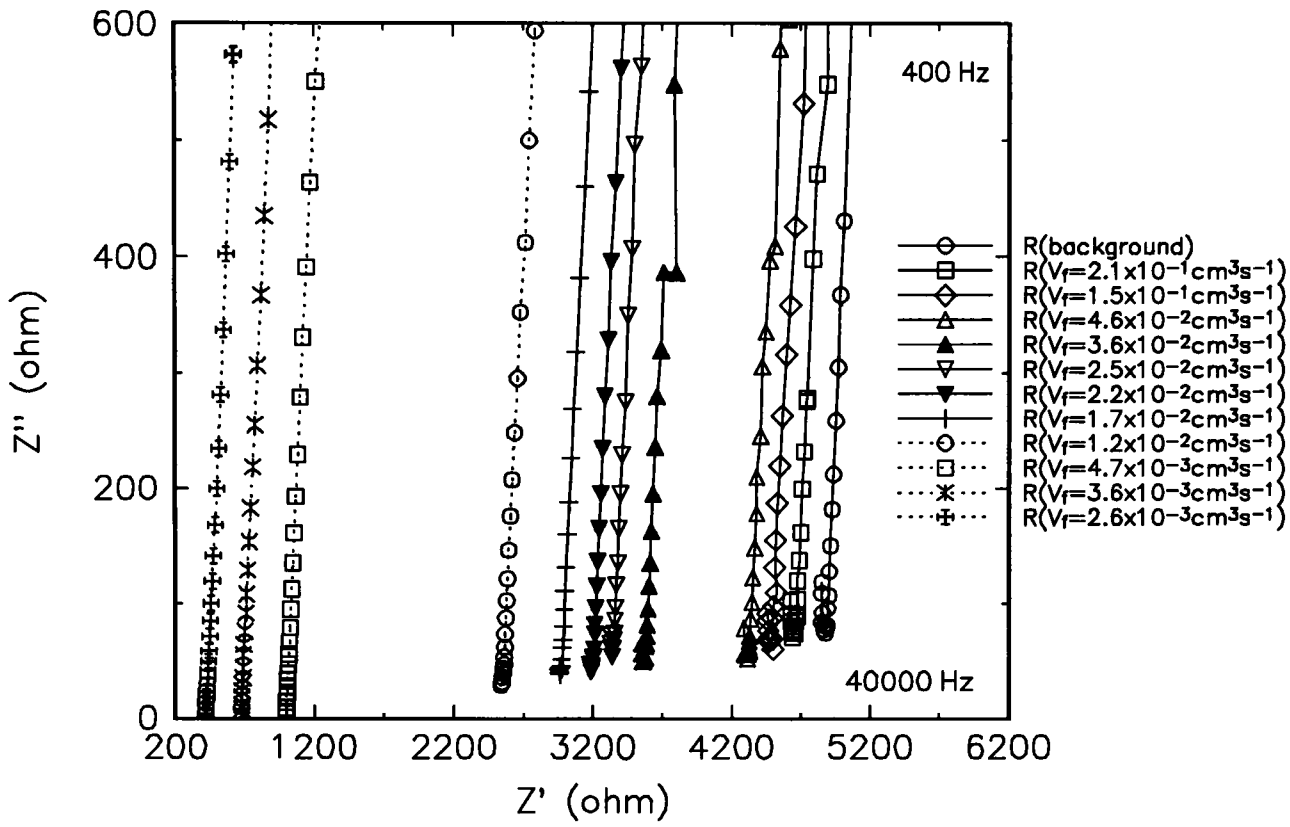
Table 1 A comparison of extrapolated resistance values obtained from "edge effects" and "no-edge effects" SPICE simulations using different cell heights.

Cell height 2h (cm)	Extrapolated Resistance: "no-edge effect" R_{ne} (ohm)	Extrapolated Resistance: "edge effect" R_e (ohm)	R_e / R_{ne}
0.0445	1833	1763	1.04
0.0890	3680	3370	1.09
0.1780	7333	6151	1.19
0.3560	14667	10609	1.38
0.7120	29286	18000	1.63

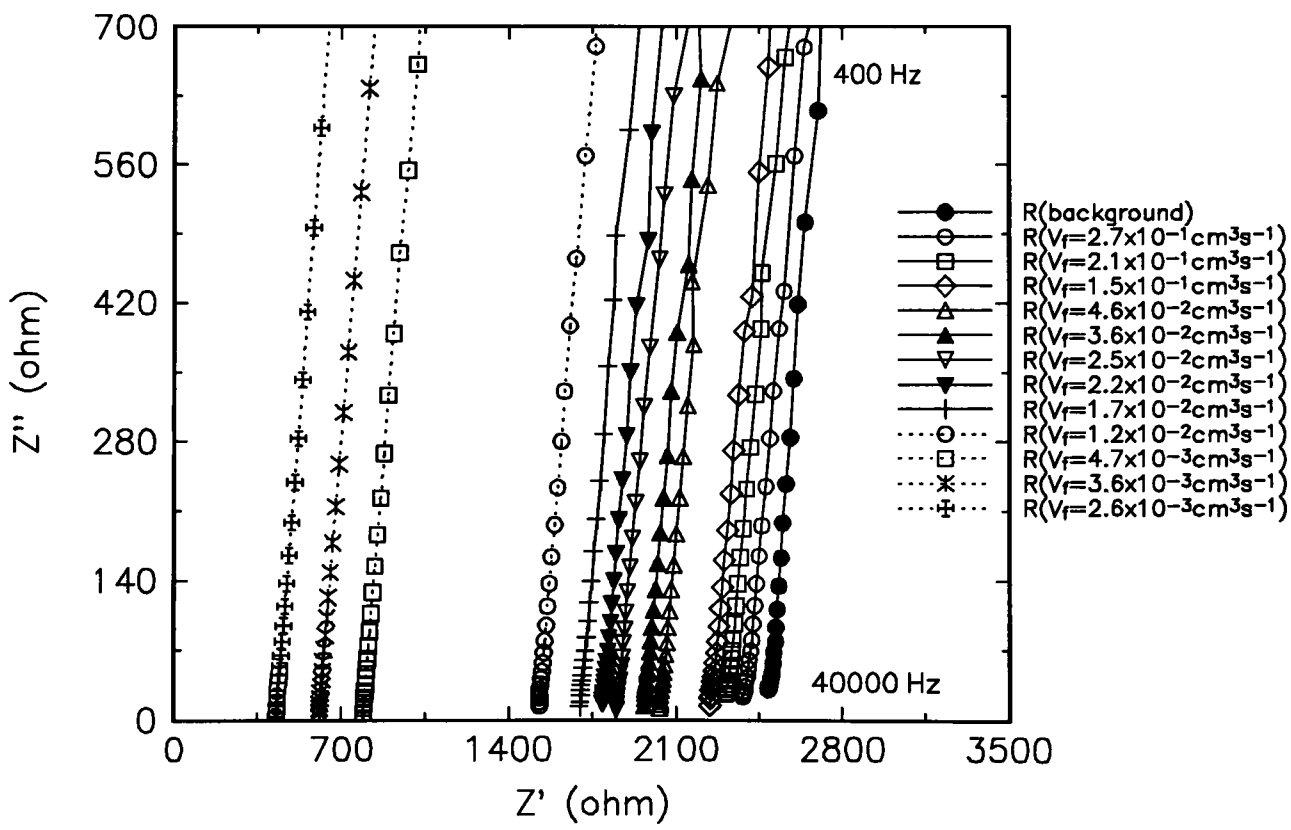
4. Triphenylmethyl chloride (TPMCl) dissolution experiments: experimental

Pellets of TPMCl for use in CFC experiments were made by compressing TPMCl powder (Aldrich, > 98%) directly into a hole cut into the perspex CFC cover plate. The particle size distribution was measured by a light scattering method and characterised by the sizes corresponding to cumulative frequencies of 10% (47 μm) and 90% (445 μm) (see chapter 6). A pressure of 5.6 x 10^9 Nm^{-2} was employed to compact the powdered material. A detailed description of the CFC experimental setup can be found in chapter 3. Solution containing milli-molar KNO_3 electrolyte was flowed through the cell under laminar conditions and the ionic products, protons and chloride ions, released from the hydrolysing interface, were subsequently detected by the conductivity monitoring system as described above. Impedance spectra obtained from the dissolution of pressed TPMCl pellets into 0.75 mM and 1.51 mM KNO_3 electrolyte at different solution flow rates are shown, respectively, in figs. 8a and 8b. Figs. 9a and 9b depict the flow rate dependence of the differential resistance, **dR**, defined as the difference between the solution resistance without dissolution and solution resistance with dissolution at the designated flow rate. The former quantity is readily determined by reversing the flow through the cell so that solution passes over the detector electrodes before it reaches the TPMCl pellet. Note that dR is strongly dependent on the solution flow rate.

Fig. 8 Experimental impedance spectra of TPMCl pressed pellets dissolution in aqueous solutions containing (a) 0.75 mM KNO_3; Cell geometry: x_p = 0.744 cm, x_g = 0.852 cm, x_d = 0.396 cm, w = 0.394 cm, d = 0.608 cm and 2h = 0.094 cm. (b) 1.51mM KNO_3; Cell geometry: x_p = 0.748 cm, x_g = 0.941 cm, x_d = 0.396 cm, w = 0.394 cm, d = 0.608 cm and 2h = 0.094 cm.

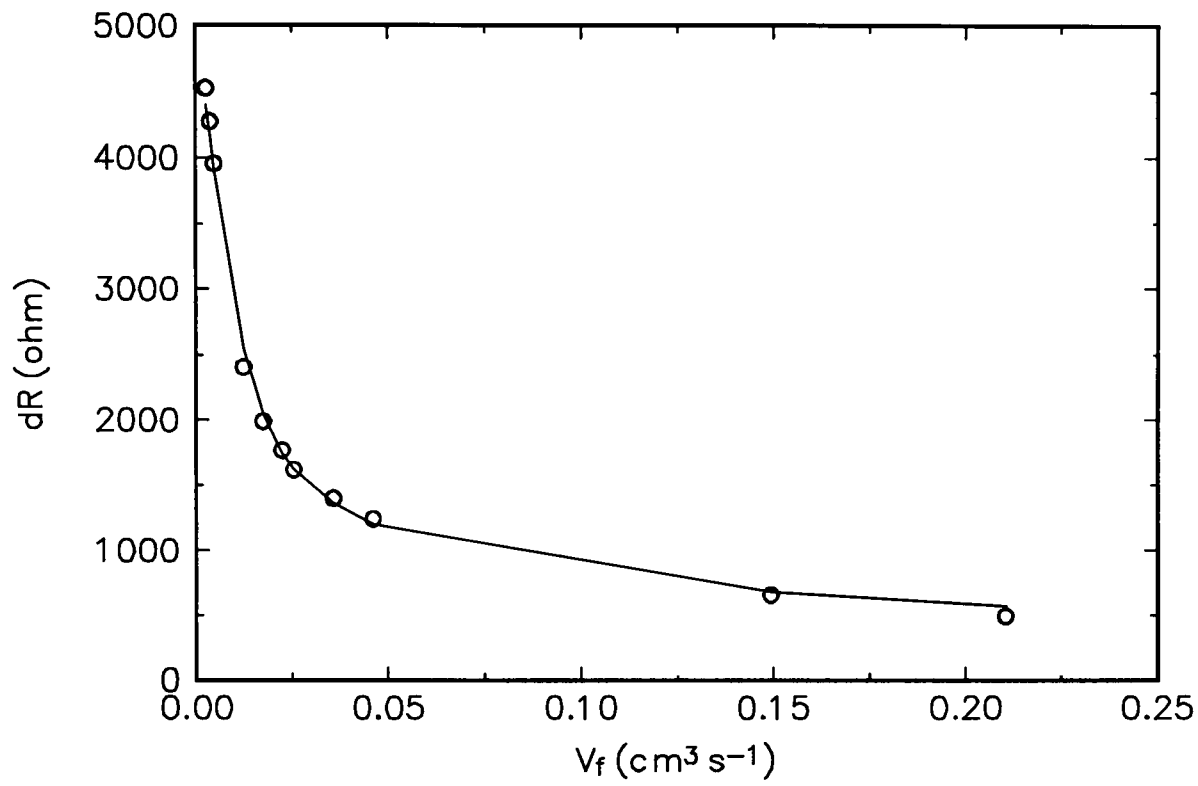


(a)

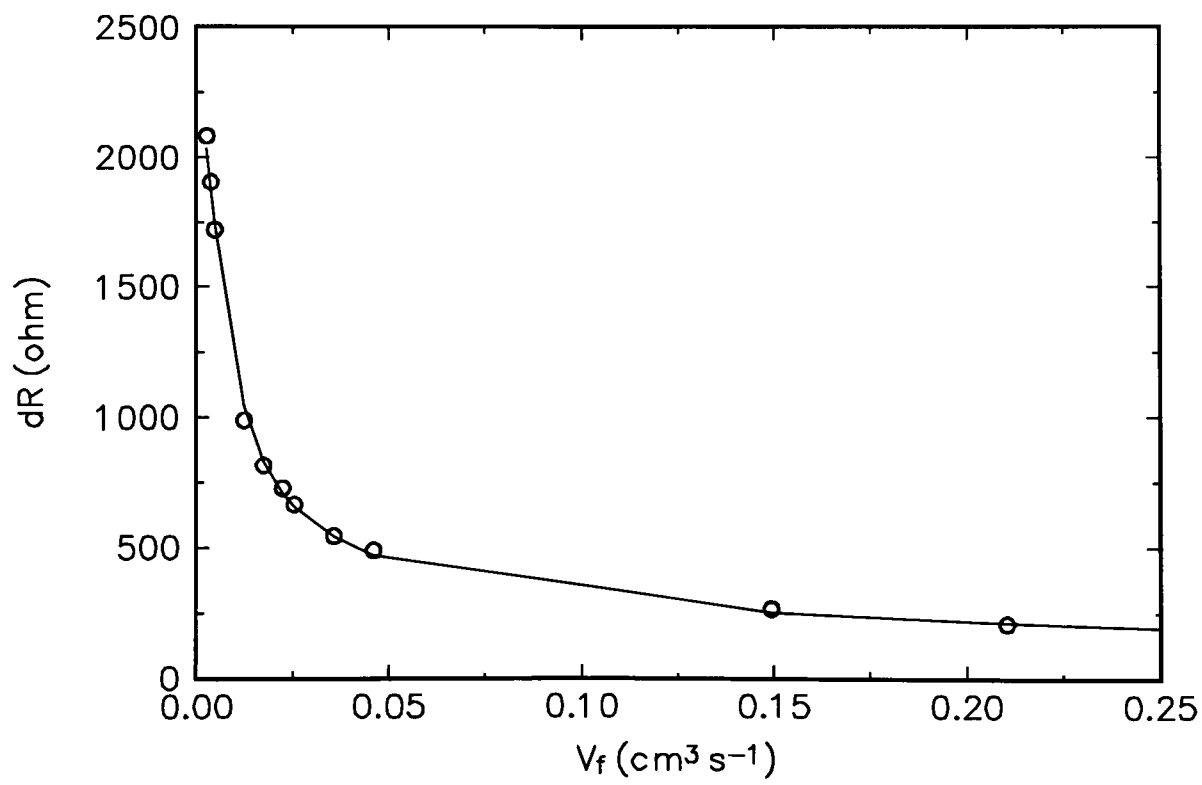


(b)

Fig. 9 Change in differential resistance (dR, as defined in the text) as a function of solution flow rate for the dissolution of TPMCl pressed pellets in aqueous solution containing (a) 0.75 mM KNO_3, (b) 1.51 mM KNO_3. The open circles denote experimental points and the solid lines represent the optimised theoretical fit generated as described in the text.



(a)



(b)

5. Triphenylmethyl chloride (TPMCl) dissolution experiments: modelling of experimental data

In section 3 above it was shown that a network of resistors and capacitors can be used to model the solution resistance provided the concentrations, $c_i(x,y)$, of the various ionic species throughout the cell are known. In the TPMCl dissolution experiment the species are K^+ and NO_3 ions from the electrolyte in the solution, together with H^+ and Cl^- ions released from the reacting solid surface. Since the added KNO_3 will serve as the "supporting electrolyte",[9,17] the region downstream of the reacting pellet can be regarded as effectively electrically neutral. Thus the strategy for analysing experimental CFC impedance spectroscopy data is as follows. First a model for the interfacial process of interest is proposed. This is then used to predict the distribution of H^+ and Cl^- by solving the relevant convection-diffusion equations by means of the Backwards Implicit Finite Difference (BIFD) procedure (see chapter 2).[18,19] The resulting spacial distributions of the H^+ and Cl^- ions is then converted into resistor values using eqns. 5 and 7. Second, the known concentration of KNO_3 is used to generate a further set of resistor values which are connected in parallel (within each element) with those generated via the BIFD computation. Third, the resulting network is then solved by using SPICE to yield the theoretical impedance spectra and hence the solution resistance as a function of both the solution flow rate and any other adjustable parameter in the kinetic model.

We consider the following interfacial reaction scheme for the TPMCl hydrolytic dissolution process:

$$TPMCl\ (s) \xrightarrow{k_i} TPM^+\ (aq) + Cl^-\ (aq) \qquad (9)$$

$$TPM^+\ (aq) + H_2O\ (l) \xrightarrow{fast} TPMOH\ (aq, s) + H^+\ (aq) \qquad (10)$$

where k_i (mol cm^{-2} s^{-1}) denotes an interfacial reaction rate constant describing the release of triphenylmethyl cations and chloride anions directly from the solid surface into solution. This is assumed to be the rate determining step.

The steady state convective-diffusion equations for the products, H^+ and Cl^-, within the CFC are:

$$\frac{\partial[H^+]}{\partial t} = D_{H^+}\frac{\partial^2[H^+]}{\partial y^2} - v_x\frac{\partial[H^+]}{\partial x} = 0 \qquad (11)$$

and

$$\frac{\partial[Cl^-]}{\partial t} = D_{Cl^-}\frac{\partial^2[Cl^-]}{\partial y^2} - v_x\frac{\partial[Cl^-]}{\partial x} = 0 \qquad (12)$$

where t, D_{Cl}, D_H, v_x represent respectively time, the diffusion coefficients of chloride anions and protons, and the solution flow velocity in the x-direction. v_x is given by

$$v_x\ (cm\ s^{-1}) = v_o[1 - \frac{(h-y)^2}{h^2}] \qquad (13)$$

where the solution flow rate, V_f ($cm^3\ s^{-1}$) = 4 v_o h d / 3: d denotes the width of the channel and v_o is the solution velocity at y = h. Values of D_{Cl^-} and D_{H^+} used in this work were 2.00 x 10^{-5} cm^2 s^{-1} and 8.97 x 10^{-5} cm^2 s^{-1} respectively.[20,21]

The chloride and proton concentrations reaching the detector electrodes may be computed by solving the above convective-diffusion equations. This requires the specification of appropriate boundary conditions. At the surface of solid substrate, an interfacial kinetic equation is adopted as the boundary condition:

$$k_i\ (mol\ cm^{-2}\ s^{-1}) = -D_{Cl^-}\frac{\partial[Cl^-]}{\partial y} = -D_{H^+}\frac{\partial[H^+]}{\partial y} \qquad \textbf{(14)}$$

Upstream of the solid the chloride and proton concentrations are specified as zero whilst on all boundaries of the cell other than the reacting surface no-flux conditions pertain:

$$0 = D_{H^+}\frac{\partial[H^+]}{\partial y} = D_{Cl^-}\frac{\partial[Cl^-]}{\partial y} \qquad \textbf{(15)}$$

Solution of the problem is readily accomplished using the BIFD method applied in essentially standard form as described in chapter 2.[18,19] The computations, when made for a known cell geometry and specified D_{H+} and D_{Cl-}, predict the chloride and proton concentrations throughout the flow cell as a function of flow rate, for assumed values of the rate constant k_i. The latter is the only adjustable parameter in the model. The resulting Cl^- and H^+ concentration profiles may be utilised to generate appropriate values of the resistors used in the SPICE network for different values of k_i. However, since the network consists of 20x80 elements, matching one BIFD grid point to one network element would not provide a sufficiently large finite difference grid to

generate sufficiently accurate concentration profiles. Accordingly, the concentration values of several BIFD grid points were averaged for each network element. For the case of the TPMCl problem 15x4 BIFD grid points were mapped into one network element to calculate the resistor values from eqns. 5 and 7. It follows that the number of BIFD grid points needed to represent the channel height in the y-direction (J) is 300 (= 20 x 15). As for the x-direction, 160 (= 40 x 4) and 80 (= 20 x 4) BI grid points are assigned, respectively, to the electrode zone and to each region immediately up- or down-stream of the detectors to allow for "edge effects" - fig. 10 illustrates the conversion from the BIFD grid to the network. It follows that the total number of grid points, K, required to cover the entire x-region ($x_p + x_g + x_d + x_e$), so as to embrace the dissolving solid, the detector electrode region and the "edge effect" zone downstream of the detector electrode, can be calculated by the following relation:

$$K = \frac{x_p + x_g + x_d + x_e}{x_d / 160} \qquad \textbf{(16)}$$

where x_p is the pellet length, x_g the distance between the edges of the pellet and the nearest detector electrode which itself has a length of x_d, and x_e the size of the downstream zone (see fig. 10). For example a value for K of 884 is required for typical experimental parameters: x_p = 0.744 cm, x_g = 0.852 cm, x_d = 0.396 cm and $x_e = x_d/2$ as used to provide the experimental data shown in fig. 8. As shown in chapter 2, a grid size of K = 500 (x-direction) by J = 300 (y-direction) is sufficient for the resulting concentration profiles to converge to better than 1%.

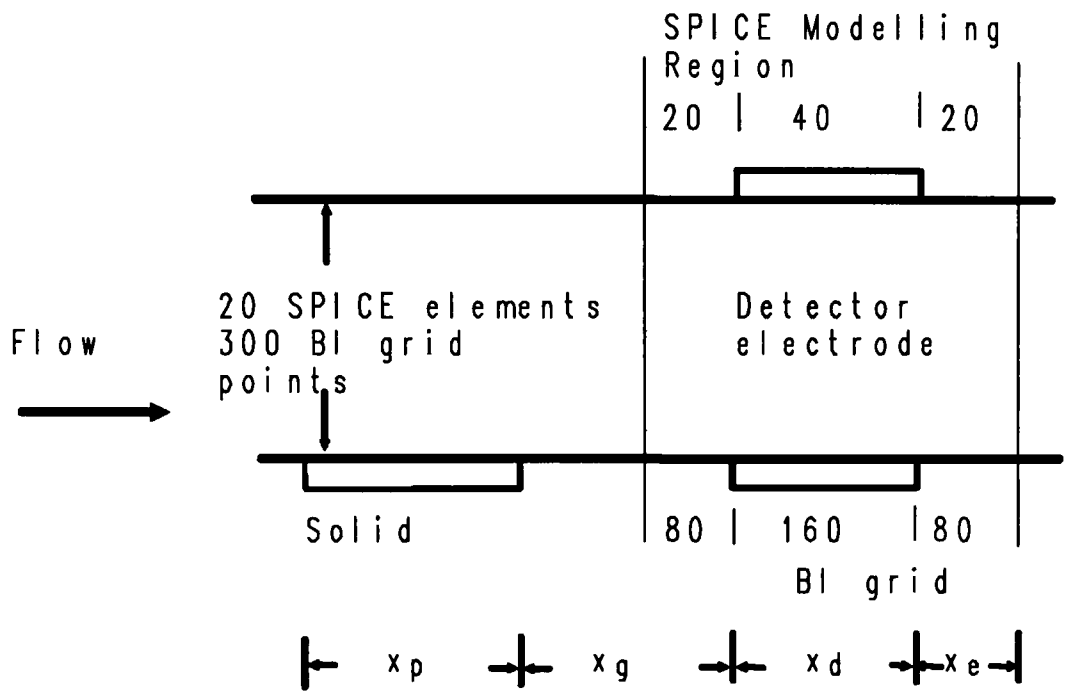


Fig. 10 Schematic diagram showing the relationship between the BIFD and SPICE simulation grids.

Fig. 11 shows theoretical impedance spectra simulated at a wide range of flow rates for a hypothetical TPMCl dissolution process with $k_i = 10^{-8}$ mol cm^{-2} s^{-1}. The cell geometry adopted is the same as defined in fig. 8a.

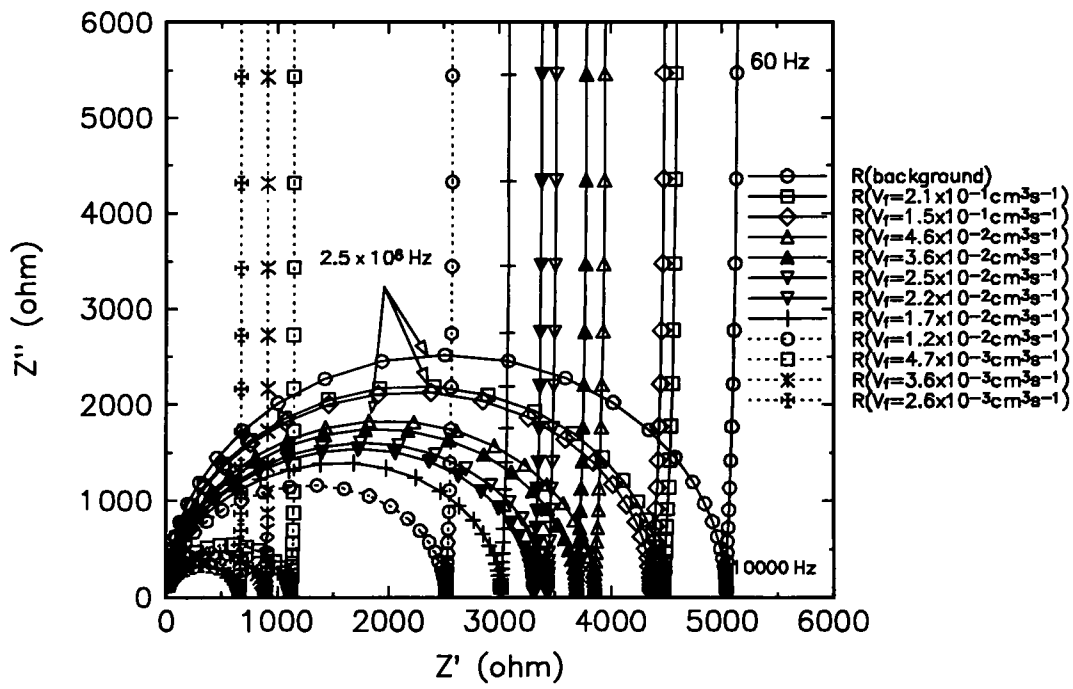


Fig. 11 Theoretical impedance spectra calculated by assuming an interfacial reaction rate constant of 1.0 x 10^{-8} mol cm^{-2} s^{-1} for the TPMCl dissolution in 0.75 mM KNO_3 solution.

The data given in fig. 11 was used to derive fig. 12 which depicts dR as a function of solution flow rate which can be seen to qualitatively parallel the experiments shown in figs. 9a and 9b. Curves such as fig. 12 were generated for different values of k_i and used to model the experimental results such as those given in fig. 9. In particular k_i was optimised to give minimum deviation, Φ, between experimental and theoretical values of dR, where

$$\Phi = \sqrt{\frac{\sum_{i=1}^{N} (dR_{expt,\,i} - dR_{model,\,i})^2}{N}} \qquad (17)$$

and N is the number of data points in the experiment. Standard optimisation method, such as Quasi-Newton algorithm[22] was utilised for this purpose.

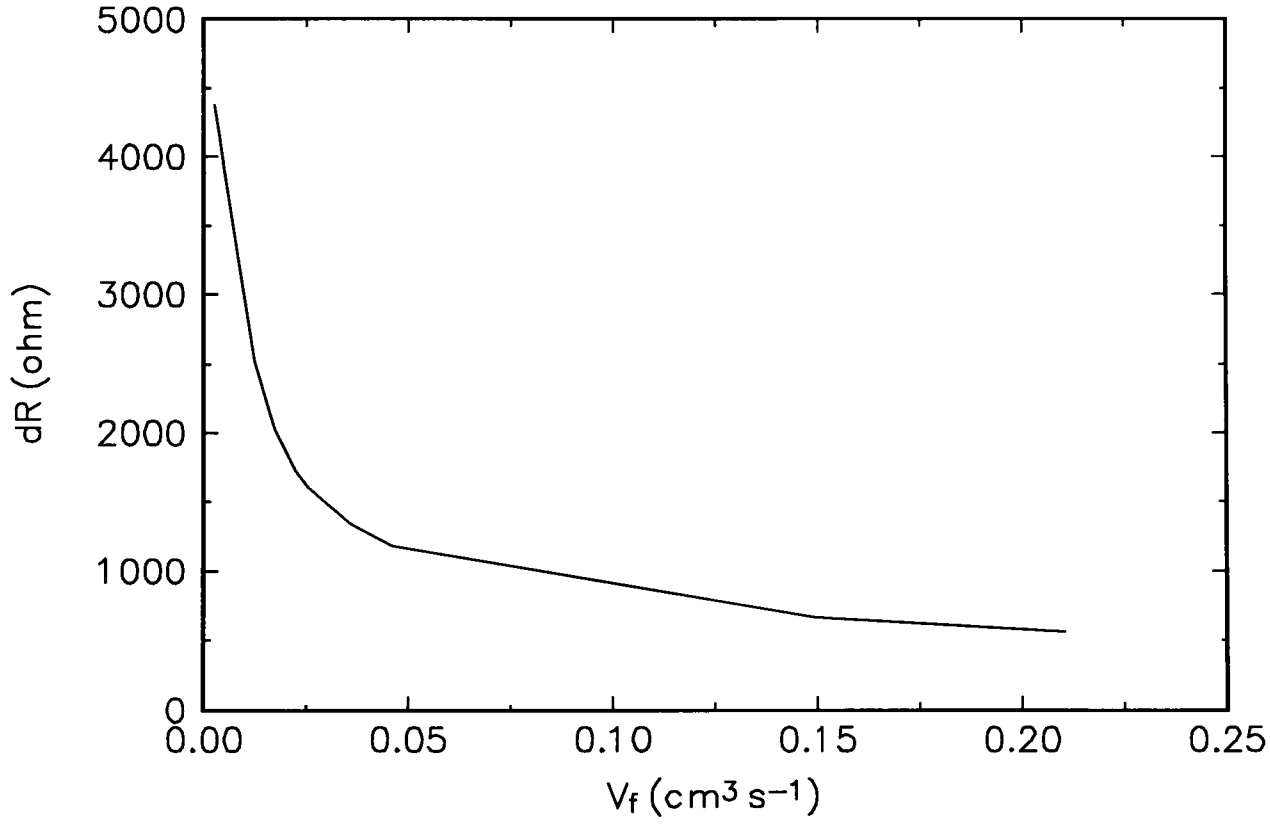


Fig. 12 The flow rate dependence of dR for the hypothetical TPMCl dissolution process (see fig. 11).

Table 2 depicts the optimised rate constants for the TPMCl dissolution process in 0.75 mM and 1.51 mM potassium nitrate solutions. Fig. 9 shows the dR values as a function of solution flow rate; the solid line indicates the theoretical behaviour generated using the optimised rate constants. Excellent agreement is seen between the experimental points and the optimised model over the entire flow rate range studied. This was in marked contrast with other reaction models that were attempted. In particular, poor fits were obtained for models assuming either a saturated surface of TPMCl or a constant flux of TPMCl followed in both cases by homogeneous hydrolysis. In the saturated surface model, the theoretical dR values calculated using the estimated solubility of TPMCl[23,24] were far below the experimental value, regardless of the magnitude of the homogeneous hydrolysis rate constant employed. As for the constant TPMCl flux model, it was found that the theoretical behaviour tended towards the proposed heterogeneous reaction model as the homogeneous hydrolysis rate constant was increased to extremely high values (see chapter 2). This observation can be rationalised since, as the life time of TPMCl (aq) decreases, the process can be regarded as occurring virtually at the solid-liquid interface as predicted by the heterogeneous model. Moreover, based on the measured hydrolysis reaction flux of about 10^{-8} mol cm^{-2} s^{-1} (see table 2) and the solubility of TPMCl ($\approx 10^{-8}$ M), it can be estimated that the reaction layer thickness is around 1 Å. This strongly suggests that the process is authentically interfacial.

Table 2 The optimised interfacial reaction rate constants for the reactive dissolution of pressed TPMCl pellets in low concentration of supporting electrolyte.

$[KNO_3]$ (mM)	Optimised rate constant* (mol cm^{-2} s^{-1})	Uncertainty** (mol cm^{-2} s^{-1})
0.75	1.068×10^{-8}	8.250×10^{-10}
1.51	1.264×10^{-8}	5.647×10^{-10}

* *Mean value from at least three separate experiments.*

** *Standard deviation from at least three separate experiments.*

Finally, we note a slight increase in k_i, reported in table 2, as the supporting electrolyte concentration is increased. This is attributed to a kinetic salt effect related to the increased charge developed in the transition state of the hydrolysis reaction in which the C-Cl bond in TPMCl has been essentially fully ionised with the formation of nearly free triphenylmethyl cations and chloride anions. These observations, and others, will be discussed in detail in chapter 6.[23]

In conclusion we have shown that the use of impedance spectroscopy detection constitutes a valuable extension of the CFC methodology and that the joint use of the BIFD and SPICE modelling techniques permits the quantitative interpretation of experimental results, in terms of mechanistic descriptions of the processes occurring at the solid-liquid interface under study.

6. References

1. Itagaki M.; Tsuru T. *J. Japan. Inst. Metals*, **1993**, *57*, 1412.

2. Kedig M; Mansfield F.; Tsai S. *Corros. Sci.*, **1983**, *23*, 317.

3. You D.; Pebere N.; Dabosi F. *Corros. Sci.*, **1993**, *34*, 5.

4. Brett C.M.A.; Brett A.M.O. *Electrochemistry*, Oxford University Press; Oxford, **1993**.

5. Compton R.G.; Fisher A.C.; Wellington R.G.; Winkler J. *J. Phys. Chem.*, **1992**, *96*, 8153.

6. Compton R.G.; Winkler J.; Riley D.J.; Bearpark S.D. *J. Phys. Chem.*, **1994**, *98*, 6818.

7. Brown C.A.; Compton R.G.; Narramore C.A. *J. Colloid Interface Sci.*, **1993**, *160*, 372; Compton R.G.; Brown C.A. *J. Colloid Interface Sci.*, **1994**, *165*, 445.

8. Orton R.; Unwin P.R. *J. Chem. Soc. Faraday Trans.*, **1993**, *89*, 3947; Compton R.G.; Pritchard K.L.; Unwin P.R. *J. Chem. Soc. Chem. Commun.*, **1989**, 249.

9. Larsen J.P. *D.Phil. Thesis*, Oxford, **1993**.

10. Tam K.Y.; Larsen J.P.; Coles B.A.; Compton R.G. *J. Electroanal. Chem.*, **1996**, *407*, 23.

11. Cohen E. "Program Reference for SPICE2", *Electronics Research Report No. ERL-M592*; Univ. of California; Berkeley, **1976**.

12. Nagel L.W.; Pederson D.O. "Simulation Program with Integrated Circuit Emphasis (SPICE)", *Electronics Research Laboratory Report No.ERL-M382*; Univ. of California; Berkeley, **1973**.

13. Nagel L.W. "SPICE2: A Computer Program to Simulate Semiconductor Circuits", *Electronics Research Report No. ERL-M520*; Univ. of California; Berkeley, **1975**.

14. Antognetti P.; Massobrio G. *Semiconductor Device Modelling with SPICE*, McGraw Hill; New York, **1988**.

15. Thorpe T.W. *Computerised Circuit Analysis with SPICE*, John Wiley and Sons; New York, **1992**.

16. Weast R.C. *Handbook of Chemistry and Physics*, 55th Edition; Chemical Rubber Company; Cleveland, **1975**.

17. Oldham K.B.; Zoski C.G. *Comprehensive Chemical Kinetics*, Bamford C.H.; Compton R.G. (eds.); *Vol. 26*; Elsevier; Amsterdam, **1986**.

18. Compton R.G.; Pilkington M.B.G.; Stearn G.M. *J. Chem. Soc. Faraday Trans 1*, **1988**, *84*, 2155.

19. Fisher A.C.; Compton R.G. *J. Phys. Chem.*, **1991**, *95*, 7538.

20. Turq P.; Lantela F.; Chemla M. *Electrochim. Acta*, **1969**, *14*, 1081.

21. Mills R.; Lobo V. *Self-Diffusion in Electrolyte Solutions*, Elsevier; Amsterdam, **1989**.

22. *NAG Fortran Library Manual - Mark 13*, Numerical Algorithm Group Ltd.; Oxford, **1988**, *Vol. 4*.

23. Tam K.Y.; Compton R.G.; Atherton J.H.; Brennan C.M.; Doherty R. *J. Am. Chem. Soc.*, **1996**, *118*, 4419.

24. Hansch C.; Quinlan J.E.; Lawrence G.L. *J. Org. Chem.*, **1968**, *33*, 347.

Chapter 5

GAMATCH - A Genetic Algorithm-Based Program For Indexing Crystal Planes[1]

This chapter develops an intelligent genetic algorithm-based crystal indexing program, GAMATCH, by means of which the Miller indices of the relevant planes can be unambiguously identified using a set of interplanar angles. The aim of this work is to index the crystal planes of the centimetre-sized triphenylmethyl chloride crystals which were utilised in the dissolution studies, as described in chapter 6.

1. Introduction

X-ray diffraction is a popular method employed to determine crystal structures, characterise lattice parameters and identify the Miller indices (MIs) of different crystal planes.[2,3] The technique can provide excellent results even with quantities as small as milli-gram of crystals. For centimetre sized crystals, however, contact or reflecting goniometry provide attractive alternatives through the evaluation of the interplanar angles.[3,4] A self-explanatory diagram of a contact goniometer is given in fig. 1. This instrument will be implemented to measure the interplanar angles (ϕ) of the triphenylmethyl chloride crystals reported in this chapter. The merits of this approach include the relatively inexpensive equipment required and the short measurement time needed. Based on the measured angles and morphology, the symmetry axes of the crystal and hence the MIs of various planes can be identified.[5-7] However, the assignment may be problematic as in the case of triclinic systems with asymmetric aspect ratios for different planes.[8]

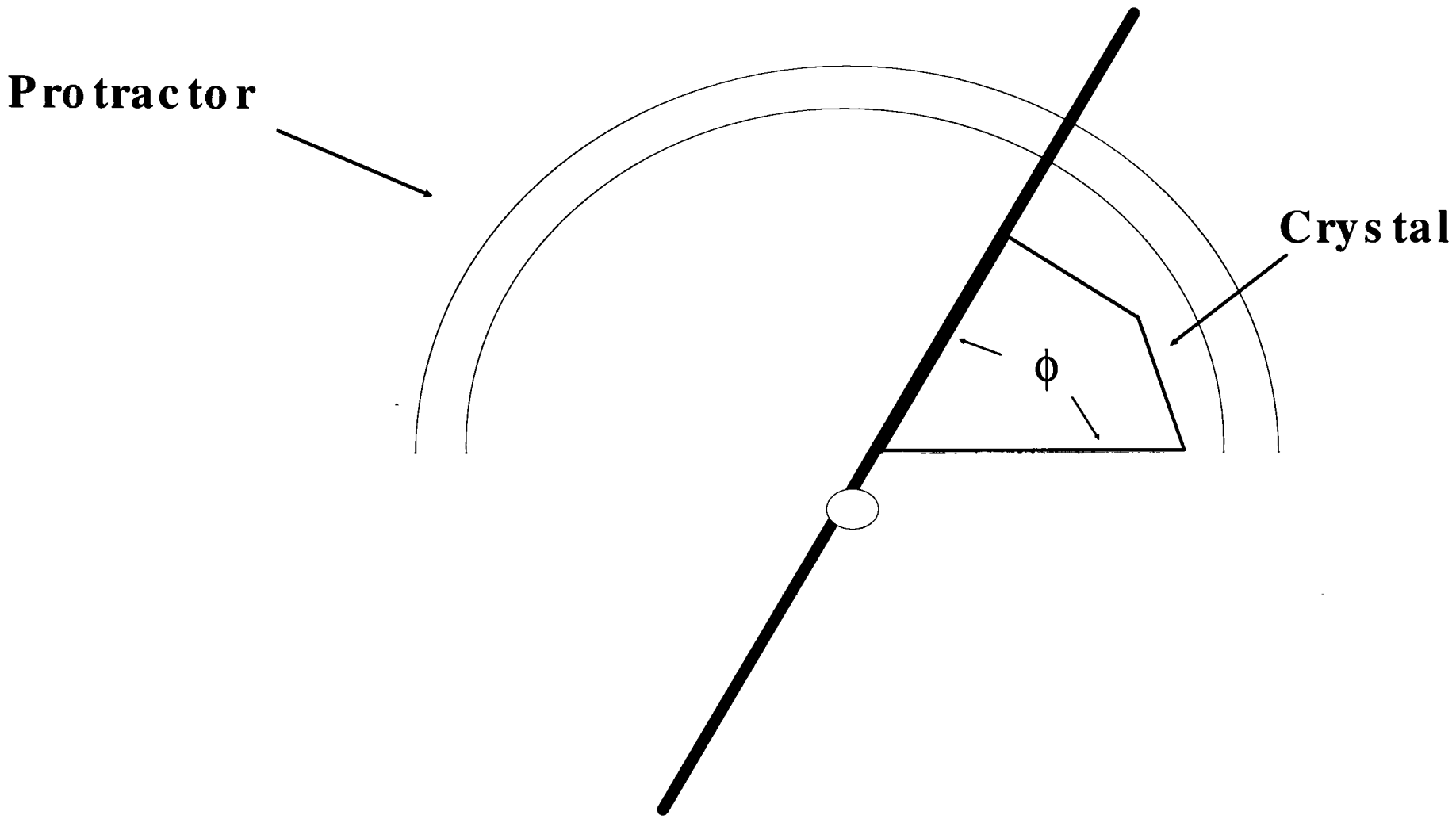


Fig. 1 A schematic diagram of a contact goniometer.

Previously, Docherty et al.[9] proposed the use of a computer program, MORANG, to calculate the possible planes from a given interplanar angle, with an estimated measurement error. Although this method greatly simplifies the matching of observed and calculated interplanar angles to index the corresponding planes, users have to select the designated MIs from a number of possible candidates. Thus, the matching process can become tedious if it involves more than two interplanar angles, even if they are known with insignificant measurement error. It is therefore anticipated that a computer program for the automatic indexing of crystal planes would resolve this issue.

Genetic algorithms (GAs) show many promising results, particularly in optimisation.[10] For instance, Xiao and Williams[11] employed a GA method to optimise the docking of drug molecules onto receptors. The same technique has been adopted to predict polymer folding pathways,[12] estimate kinetic parameters[13] and analyse the distribution of airborne pollution.[14]

In this work, we report a GA-based computer program to automate the MIs assignments. Specifically, a multidimensional GA searching in the permissable MIs domain is carried out to determine the best fit between experimental and calculated interplanar angles. A two-step-divide-and-conquer (TSDC) strategy is proposed for fast convergence to the global minimum. A triclinic crystal, triphenylmethyl chloride ($(C_6H_5)_3CCl$), is used to illustrate how the method works.

2. Method of calculation

Computation of the interplanar angles

The equations for calculating the interplanar angles of triclinic crystals are given in the

International Tables for X-ray Crystallography.[15] For a unit cell of dimensions a, b, c, α, β and γ, the unit cell volume, V is given by the following equation

$$V = 2\,abc\,[\sin(S)\,\sin(S-\alpha)\,\sin(S-\beta)\,\sin(S-\gamma)]^{1/2} \quad (1)$$

where

$$S = (\alpha+\beta+\gamma)/2 \quad (2)$$

The reciprocal lattice elements, denoted by an asterisk, can be obtained from the following relationships:

$$a^* = bc\sin\alpha/V, \quad b^* = ca\sin\beta/V, \quad c^* = ab\sin\gamma/V \quad (3)$$

$$\cos\alpha^* = [(\cos\beta\cos\gamma - \cos\alpha)/\sin\beta\sin\gamma] \quad (4)$$

$$\cos\beta^* = [(\cos\gamma\cos\alpha - \cos\beta)/\sin\gamma\sin\alpha] \quad (5)$$

$$\cos\gamma^* = [(\cos\alpha\cos\beta - \cos\gamma)/\sin\alpha\sin\beta] \quad (6)$$

The interplanar angle ϕ between two planes (hkl) and (h'k'l') can be calculated from

$$\cos\phi = [hh'a^{*2} + kk'b^{*2} + ll'c^{*2} + (kl' + lk')b^*c^*\cos\alpha^* +$$

$$(lh' + hl')c^*a^*\cos\beta^* + (hk' + kh')a^*b^*\cos\gamma^*]/\sqrt{Q^*_{hkl}Q^*_{h'k'l'}} \quad (7)$$

where Q^*_{hkl} is

$$Q^*_{hkl} = h^2a^{*2} + k^2b^{*2} + l^2c^{*2} + 2klb^*c^*\cos\alpha^* + 2lhc^*a^*\cos\beta^* + 2hka^*b^*\cos\gamma^* \quad (8)$$

It can be seen (eqns. 7 & 8) that the interplanar angle is a function of (hkl) and (h′k′l′) as well as the unit cell dimensions, which need to be known beforehand. Suppose *nd* interplanar angles are measured from a given crystal. The searching of MIs for the dedicated planes can be formulated as a combinatorial optimisation problem to minimise the root mean square error (*RMSE*) between the observed (*obs*) and calculated (*calc*) interplanar angles:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{nd} (\phi_{i,obs} - \phi_{i,calc})^2}{nd}} \qquad (9)$$

Optimisation via the genetic algorithm

GA methods are based on the idea of Darwinian evolution[10,16] and can be regarded as an optimisation strategy between random search and hill-climbing.[17] However, the techniques are superior to the conventional ones (e.g. gradient-descent) in solving complex problems.[16]

In the present studies, GAMATCH was coded in a Fortran 77 environment and executed on a SUN SPARC workstation to perform the MIs assignment for a given set of interplanar angles. The structure of GAMATCH is given as follows

```
program GAMATCH
read parameters and angles
iteration = 0
initialise Population(iteration)
  repeat
      iteration = iteration + 1
      evaluate fitness values
      select Population(iteration) from Population(iteration - 1)
      recombine Population(iteration)
            1. Crossover
            2. Elitism
            3. Mutation
  until termination conditions are satisfied
end
```

The program maintains a population of n individuals (chromosomes) for every iteration. Each individual represents a potential solution to the problem under consideration. The fitness of each individual is *evaluated* and then a new population is formed by *selecting* the more fit individuals. Some members of the new population undergo *recombination* by means of genetic operators, namely crossover, elitism and mutation, to form new solutions. The evaluate-select-recombine process is then repeated until the termination conditions are satisfied. The best individual hopefully represents the global solution.

The optimal population size can be estimated by the following relationship[18]

$$\textit{Optimal population size} \approx 1.65 \times 2^{(0.21 \times l)} \qquad (10)$$

where l denotes the bit length of the chromosome. In the present study, optimal population sizes in the range from 130 to 269 are appropriate. However, a value of 200 was chosen arbitrarily in all calculations in order to facilitate the comparison of computation times ($\propto$ generations). The initial population of potential answers is generated in random fashion.[19] The chromosome of the population is represented by using a binary string of length m. Every chromosome is divided into a number of segments (genes) according to the number of crystal planes in the problem.

Each gene is symbolised by a 5-bit binary number which corresponds to the MI of a particular plane. For instance, a 25-bit chromosome is required for a crystal with five planes. Table 1 gives the gene values assigned to different MIs. It should be noted that the first 6 items are just duplications of the others in order to make up the 5-bit (32) combinations. As shown in table 1, the high-index planes and the illegal index (000) are excluded to narrow down the search space.

Table 1 Gene values assigned to the different Miller Indices.

Value of gene (binary)	Value of gene (decimal)	Miller Index (hkl)
00000	0	$(\bar{1}11)$
00001	1	$(1\bar{1}\bar{1})$
00010	2	$(\bar{1}10)$
00011	3	$(1\bar{1}0)$
00100	4	(010)
00101	5	$(0\bar{1}0)$
00110	6	(100)
00111	7	$(\bar{1}00)$
01000	8	(010)
01001	9	$(0\bar{1}0)$
01010	10	(001)
01011	11	$(00\bar{1})$
01100	12	(011)
01101	13	$(0\bar{1}\bar{1})$
01110	14	(101)
01111	15	$(\bar{1}0\bar{1})$
10000	16	(110)
10001	17	$(\bar{1}\bar{1}0)$
10010	18	$(0\bar{1}1)$
10011	19	$(01\bar{1})$
10100	20	$(\bar{1}01)$
10101	21	$(10\bar{1})$
10110	22	$(\bar{1}10)$
10111	23	$(1\bar{1}0)$
11000	24	(111)
11001	25	$(\bar{1}1\bar{1})$
11010	26	$(\bar{1}11)$
11011	27	$(1\bar{1}\bar{1})$
11100	28	$(\bar{1}\bar{1}1)$
11101	29	$(11\bar{1})$
11110	30	$(\bar{1}1\bar{1})$
11111	31	$(1\bar{1}1)$

From each chromosome, we calculate the *Fitness* which is defined as follows

$$Fitness = \frac{Scale \times Penalty^{k}}{RMSE} \tag{11}$$

where the variable *Scale* is an arbitrary constant. $Penalty^{k}$ is a penalty function which reduces the fitness value by a specified amount if any equal gene is found in the same chromosome. k represents the number of equal gene pairs detected and *Penalty* is an arbitrary constant ($0 < Penalty < 1$). Obviously, maximisation of fitness is equivalent to minimisation of the *RMSE*.

In the selection process, the stochastic remainder method[20] is adopted to choose the chromosomes for the next generation. The fitness value of each chromosome is multiplied by a scaling factor so that the average fitness of the population is equal to 1. Then, for every chromosome with above average fitness, a number of copies equal to the integer part of their fitness is made. For instance, a chromosome with scaled fitness of 2.4 would have two copies put into the next generation. The number of copies made is subtracted from the scaled fitness so that the chromosomes have a remainder less than 1. The remaining positions in the population are filled up using a probabilistic procedure. In the method, a chromosome is chosen at random. Another positive random number (less than 1) is generated and compared with the remainder of the fitness. If the random number is less than the remainder of the fitness, then the chromosome is put into the new generation and the remainder is reduced to zero to prevent further copies of the chromosome being made. Otherwise, the probabilistic procedure is continued until all the vacancies in new population are occupied.

The new population is then subjected to the aforementioned recombination in which inventive chromosomes are generated. It should be pointed out that there is no formal rule governing

the probability of crossover (P_c), probability of elitism (P_e) and probability of mutation (P_m).[16] The genetic parameters themselves must also be optimised with regard to convergence speed.[17,21]

In the crossover operation, a random number in the real range (0, 1) is generated for each chromosome. If the random number is smaller than P_c, the chromosome is selected for the crossover operation. An extra chromosome is randomly picked if the number of selected chromosomes is odd. Then, the selected chromosomes are mated in pair. Two locations in the two selected parent chromosomes are randomly chosen. If the first point is to the left of the second one, the block between the two points is exchanged (normal crossover). Otherwise, a wraparound operator is triggered to swap the genes at the outer ends, rather than the inner part between the crossover points (wraparound crossover).

Next, the elitism operator is activated. A random number in the real range (0, 1) is generated and compared with P_e. If the number is smaller than P_e, the "best" chromosome (the one with the highest fitness value from the previous generations) is inserted back into the current population to replace the one with the lowest fitness value. Subsequently, the population undergoes a mutation operation. A random number in the real range (0, 1) is generated for each bit in every chromosome. If the number is smaller than P_m, that bit is superseded by a random integer number (either 0 or 1). Finally, GAMATCH checks the termination conditions. If the number of generations exceeds a predefined number (mcount) or if the fitness of the best individual reaches an expected value then the program will stop and report the individual with the maximum fitness value. Otherwise the whole genetic operation will be **repeated**. The genetic searching process is regarded as successful if the maximum fitness

does attain the required level.

Strategies for rapid convergence to a global solution

The minimisation of the RMSE requires a search for the global minimum in the solution space. However, as the number of planes (variables) increases, the number of generations required to achieve successful termination increases rapidly. Additionally, the search is likely to be trapped in a local minimum before the global optimum can be found. A slight increase in mutation probability may help the program to escape from the local minimum. Thus in some calculations, the mutation probability was raised by about 2×10^{-5} after each generation to avoid premature convergence. In general, a mutation probability within the range 0.15 - 0.2, a crossover probability within 0.1 - 0.6 and an elitism probability within 0.1 - 0.2 are commonly used.

It is found that when more than four planes are involved in the computations, the rate of convergence is greatly speeded up by the application of a two-step-divide-and-conquer (TSDC) strategy. Specifically, the indexing is confined to any three planes in the first step. Then, in the second step, the indices of remaining planes can be evaluated by feeding in the indices obtained from the first step as constants. Normally, less than 100 generations are sufficient (about 2 minutes in our SUN SPARC system) to effect convergence to the global solution. Alternatively, if the indices of the most dominant plane can be determined by other means, the first step can be omitted. The indices of this major plane are then input into the second step as before.

The Bravais-Friedel-Donnay-Harker (BFDH) law is a simple and commonly used method to

estimate the morphological importance of crystal planes.[9,22,23] This method is based on the assumption that the larger the interplanar spacing (d_{hkl}), the higher resistance against the attachment of growth units, which leads to a lower growth rate and hence greater morphological importance. d_{hkl} can be defined as follow:

$$d_{hkl} = \frac{1}{\sqrt{Q^*_{hkl}}} \tag{12}$$

where Q^*_{hkl} is given in eqn. 8. It should be mentioned that the BFDH law can only be used if the unit cell has been characterised, during which process it should be easy to index the planes. Recent work by Liu and Bennema[24] indicates that the BFDH method gives the same results as other more sophisticated recipes (e.g. Ising model, Hartman-Perdok theory) in predicting the most important plane of triclinic organic crystals. This justifies the choice of the BFDH method to identify the most dominant plane.

3. Results

GAMATCH was implemented to index a triclinic organic crystal, triphenylmethyl chloride (TPMCl) with crystal structure has been reported first by Gerdil and Dunand[25] and later by Kahr and Carter.[26] Table 2 gives the crystallographic data for the triclinic crystal (from form III in Kahr & Carter[26]) and table 3 depicts the four most important planes as predicted by using the BFDH law. The BFDH morphology is given in fig. 2. It can be seen that the prediction reveals that the dominant planes are (010), (100), ($\bar{1}$10) and (001) as well as their complements.

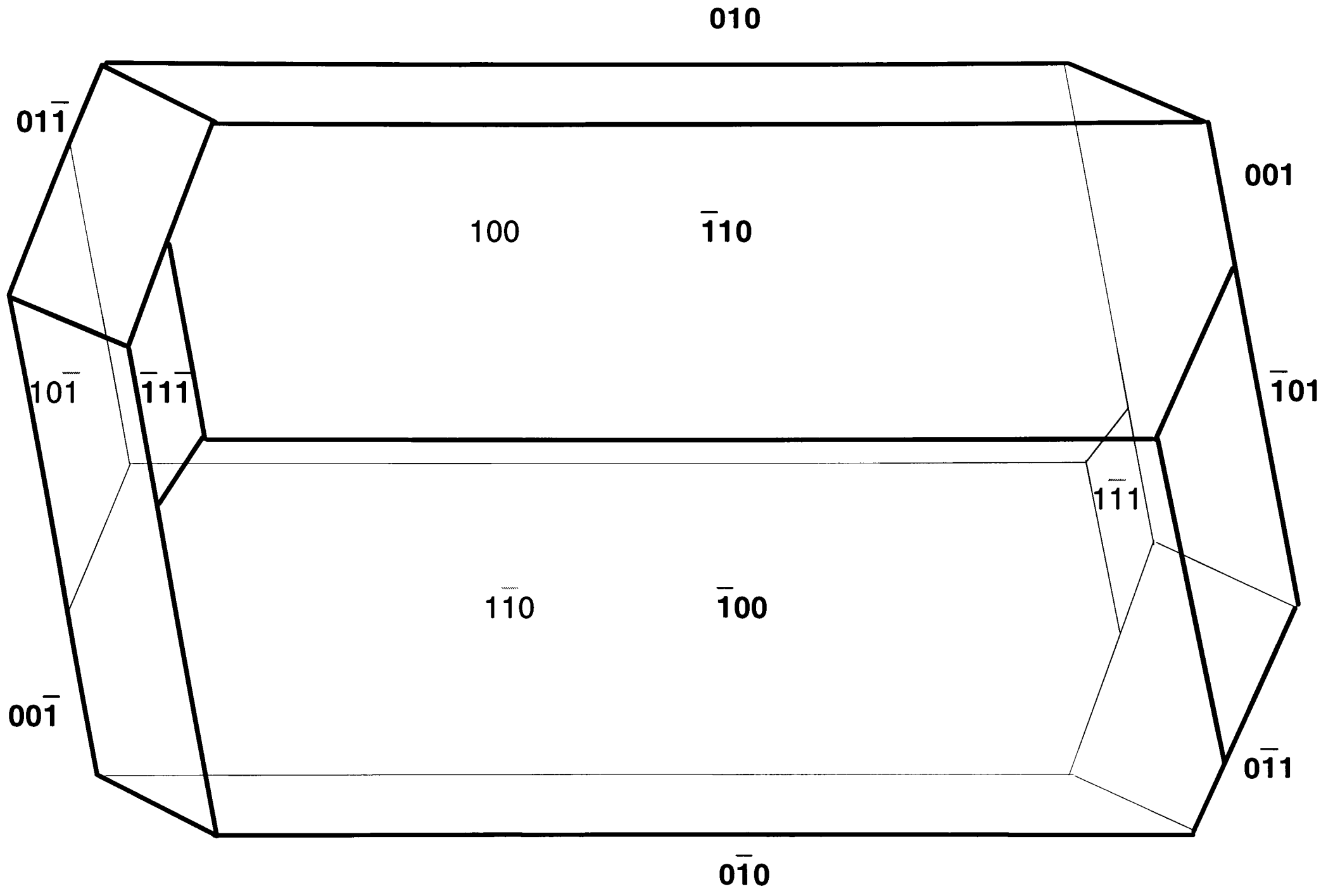


Fig. 2 The BFDH mophology of triphenylmethyl chloride, drawn using the CERIUS nolecular graphics package.

Table 2 Crystallographic data for TPMCl (taken from form III in Kahr & Carter[26]).

Formula	$C_{19}H_{15}Cl$
System	Triclinic
Space group	$P\bar{1}$
a, Å	14.1526(4)
b, Å	21.3190(7)
c, Å	13.0654(5)
α, °	99.92(3)
β, °	92.68(3)
γ, °	106.15(2)
V, Å^3	3712(2)
Z	10

Table 3 The dominant planes of TPMCl and the corresponding interplanar spacing (d_{hkl})

Planes (hkl)	d_{hkl} (Å)	Morphological importance
(010) / (0$\bar{1}$0)	20.0922	1
($\bar{1}$00) / (100)	13.5293	2
($\bar{1}$10) / (1$\bar{1}$0)	13.1303	3
(001) / (00$\bar{1}$)	12.8054	4

Before applying GAMATCH to index the TPMCl crystals grown in our laboratory (see chapter 6), we first used a hypothetical TPMCl crystal (see fig. 2), as predicted by using BFDH law, to evaluate the proposed method. The approximate interplanar angles, supposed to be measured experimentally, were generated using eqn. 7. In addition to the four dominant planes (see table 3), three other planes were arbitrarily selected for evaluation. The values of the approximate angles were rounded off to nearest degree and are given in table 4.

Table 4 The approximate interplanar angles selected for GA searching of Miller indices in the first hypothetical crystal example.

Planes*	Approximate interplanar angle (ϕ_{obs})
1^2	68
1^3	73
2^3	141
1^4	66
2^4	79
3^4	85
1^5	86
2^5	126
3^5	50
4^5	47
1^6	79
2^6	88
3^6	84
4^6	145
5^6	134
1^7	97
2^7	133
3^7	51
4^7	137
5^7	95
6^7	45

*with** 1: (010), 2: $(\bar{1}10)$, 3: (100), 4: $(01\bar{1})$, 5: $(10\bar{1})$, 6: (001), 7: $(1\bar{1}1)$

* *The numbers are assigned to the expected indices arbitrarily for identification purposes.*

^ *denotes the intersection between two crystal planes.*

Table 5 Results from GAMATCH indexing the hypothetical crystal with the interplanar angles given in table 4.

	First step	Second step
n	200	200
mcount	3000	3000
P_c	0.10	0.60
P_e	0.10	0.10
P_m	0.15	0.15
Penalty	0.80	0.80
RMSE	0.22	0.28
Fitness	459.82	361.94
Generation number*	7	34
Assigned indices	1: (010), 2: ($\bar{1}$10), 3: (100)	1: (010), 2: ($\bar{1}$10), 3: (100), 4: (01$\bar{1}$), 5: (10$\bar{1}$), 6: (001), 7: (1$\bar{1}$1)

* *Number of generations required to converge to the corresponding fitness value.*

The TSDC strategy used in conjunction with GAMATCH was invoked to index the planes from the approximate angles in table 4. Typical results for the first and second steps are given in table 5. By comparing the expected indices in table 4 with those resolved in table 5, the correct MIs are found to be successfully assigned to the designated planes without a prior knowledge of their identities. Next, attention was directed to another example involving only the three dominant planes: (010), (100), ($\bar{1}$10) and their complements along the z-axis of the hypothetical crystal. The selected approximate interplanar angles are given in table 6. We supposed that only the largest plane (010) was identified using the BFDH law. With this information in mind, GAMATCH then performs the TSDC second step calculations. Typical

results are shown in table 7. In this case, the deduction from GAMATCH also shows excellent agreement with the expected indices as presented in table 6. It can be seen that table 7 reports 20 times as many generations as table 5.

Table 6 The approximate interplanar angles selected for GA searching of Miller indices in the second hypothetical crystal example.

Planes*	Approximate interplanar angle (ϕ_{obs})
1^2	68
1^3	107
1^4	180
1^5	112
1^6	73
2^3	39
2^4	112
2^5	180
2^6	141
3^4	73
3^5	141
3^6	180
4^5	68
4^6	107
5^6	39

*with** 1: (010), 2: ($\bar{1}$10), 3: ($\bar{1}$00), 4: (0$\bar{1}$0), 5: (1$\bar{1}$0), 6: (100)

* *The numbers are assigned to the expected indices arbitrarily for identification purposes.*

Table 7 Results from GAMATCH indexing the hypothetical crystal with the interplanar angles given in table 6.

	Second step
n	200
mcount	3000
P_c	0.10
P_e	0.10
P_m	0.20
Penalty	0.80
RMSE	0.19
Fitness	514.07
Generation number*	890
Assigned indices	1: (010), 2: ($\bar{1}$10), 3: ($\bar{1}$00), 4: (0$\bar{1}$0), 5: (1$\bar{1}$0), 6: (100)

* *Number of generations required to converge to the corresponding fitness value.*

With the success obtained for the hypothetical TPMCl crystal, GAMATCH was then utilised to index the TPMCl crystals prepared in our laboratory. The triclinic TPMCl crystals were grown from pentane using a slow cooling process (see chapter 6). Fig. 3 depicts a schematic diagram of a typical crystal of approximate size 0.7 x 0.5 x 0.25 cm. The smaller crystals obtained were characterised using x-ray diffraction. The cell dimensions were in excellent agreement (within 0.1 %) with the triclinic form (III) reported by Kahr and Carter (see table 2). As for the crystal shown in fig. 3, the interplanar angles were measured by using contact goniometry and are shown in table 8. According to the BFDH law (see table 3), the dominant plane 1 was assigned as plane (010). Then, GAMATCH was applied to index the other planes and the results are given in table 9. The MIs assignment obtained from GAMATCH is in

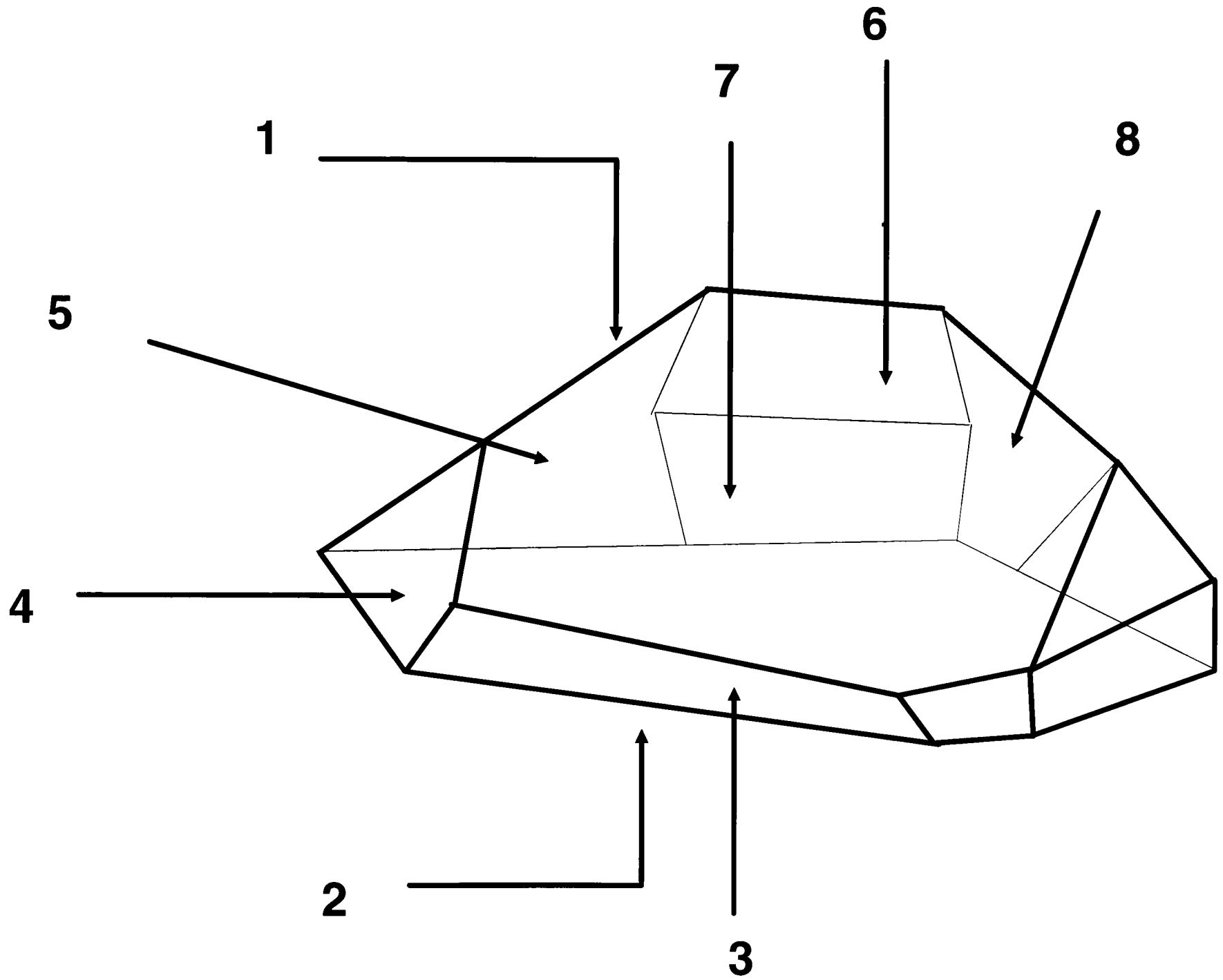


Fig. 3 A schematic diagram of the triclinic TPMCl crystal.

complete agreement with the results obtained using MORANG.[9]

Table 8 *The interplanar angles of the TPMCl crystal as shown in fig. 3.*

Planes*	Interplanar angle (ϕ_{obs})
1^2	41.5
1^3	99.5
1^4	101.5
1^6	86.5
1^7	59.0
1^8	82.0
2^3	115.5
2^4	75.0
2^5	74.5
2^6	50.0
2^7	77.0
2^8	64.0
3^4	115.0

* *The numbers are assigned to the designated planes arbitrarily for identification purposes.*

4. Conclusion

A GA-based program coupled with a divide-and conquer strategy has been applied to solve a crystal planes indexing problem. Using experimentally measured interplanar angles, the program is able to index the crystal planes without involving any manual matching work. The results from the computation show that the proposed program can productively assist the

assignment of MIs for large crystals where the x-ray diffraction method becomes difficult to employ.

Table 9 Results from GAMATCH indexing the TPMCl crystal with the interplanar angles given in table 8.

	Second step
n	200
mcount	3000
P_c	0.60
P_e	0.20
P_m	0.10
Penalty	0.80
RMSE	7.78
Fitness	12.85
Generation number*	291
Assigned indices	1: (010), 2: (110), 3: ($\bar{1}$01), 4: (1$\bar{1}$0), 5: (1$\bar{1}\bar{1}$), 6: (10$\bar{1}$), 7: (01$\bar{1}$), 8: (1$\bar{1}$1)

* *Number of generations required to converge to the corresponding fitness value.*

5. References

1. Tam K.Y.; Compton R.G. *J. Appl. Cryst.*, **1995**, *28*, 640.

2. Buerger M.J. *Contemporary Crystallography*, McGraw Hill; New York, **1970**; Giacovazzo C. *Fundamentals of Crystallography*, Giacovazzo C. (ed.); Oxford University Press; Oxford, **1992**.

3. Mullin J.W. *Crystallization*, 3rd ed.; Butterworth; Oxford, **1993**.

4. Phillips F.C. *An Introduction to Crystallography*, 4th ed.; Longman; London, **1977**.

5. Barker T.V. *Graphical and Tabular Methods in Crystallography*, Murby; London, **1922**.

6. Bishop A.C. *An Outline of Crystal Morphology*, Hutchinson Scientific & Technical; London, **1967**.

7. Porter M.W.; Spiller R.C. *The Barker Index of Crystals*, Vol. 1; Pt. 1.; W. Heffer and Sons; Cambridge, **1951**.

8. Docherty R. Personal communication, 1994.

9. Docherty R.; Roberts K.J.; Dowty E; *Comput. Phys. Comm.*, **1988**, *51*, 423.

10. Goldberg, D.E. *Comm. A.C.M.*, **1994**, *37(3)*, 113.

11. Xiao Y.L.; Williams D.E. *J. Phys. Chem.*, **1994**, *98*, 7191.

12. Judson R.S. *J. Phys. Chem.*, **1992**, *96*, 10102.

13. Hibbert D.B. *Chemom. Intell. Lab. Syst.*, **1993**, *19*, 319.

14. Cartwright H.M.; Harris S.P. *Atmo. Envir. Pt. A*, **1993**, *27*, 1783.

15. *International Tables For X-ray Crystallography*, Kynoch Press; Birmingham, **1973**; Vol. 2.

16. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag; Brelin, **1992**.

17. Li T.H; Lucasius C.B.; Kateman G. *Anal. Chim. Acta*, **1992**, *268*, 123.

18. Goldberg D.E. "Optimal Initial Population Size for Binary Coded Genetic Algorithms", *TGCA Rep. No. 85001*; Univ. of Alabama, **1985**.

19. *NAG Fortran Library Manual - Mark 13*, Numerical Algorithm Group Ltd.; Oxford, **1988**; Vol. 6.

20. Cartwright H.M. *Applications of Artificial Intelligence in Chemistry*, Oxford University Press; Oxford, **1993**.

21. Grefenstette J.J. *IEEE Trans. Syst. Man Cybern.*, **1986**, *16*, 122.

22. Donnay J.D.H.; Harker D. *Am. Mineralogist.*, **1937**, *22*, 446.

23. Hartman P.; Bennema P. *J. Cryst. Growth*, **1980**, *49*, 145.

24. Liu X.Y.; Bennema P. *J. Cryst. Growth*, **1994**, *135*, 209.

25. Gerdil R.; Dunand A. *Acta Cryst.*, **1975**, *B31*, 936.

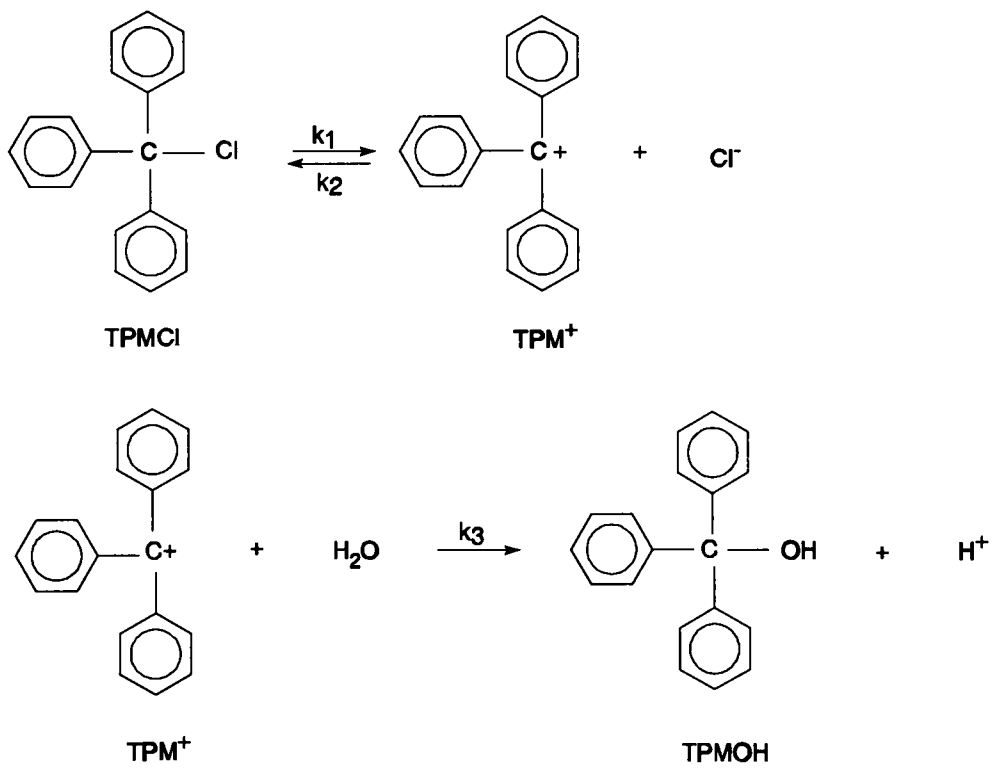
26. Kahr B.; Carter R.L. *Mol. Cryst. Liq. Cryst.*, **1992**, *219*, 79.

Chapter 6

The Hydrolytic Dissolution of Solid Triphenylmethyl Chloride in Aqueous Solution

1. Introduction

In this chapter we apply the channel flow cell (CFC) methodology to study the dissolution and hydrolysis of solid triphenylmethyl chloride, TPMCl. Whilst the homogeneous solvolysis of TPMCl in various solvents has been extensively studied from an early date[1-4] with the conclusion that the reaction follows a unimolecular nucleophilic substitution mechanism as depicted below



Little, if any, work on the hydrolysis of solid TPMCl appears to have been undertaken. Work on the rate of hydrolysis of TPMCl in a two phase water-toluene system[5] shows that particular interfacial process to be exceptionally fast and it has been speculated[5] that the reaction may be likely to occur *at* the liquid-liquid interface. This observation, combined with the recognition that the solubility of TPMCl in water is very low - ca. 10^{-8} M as predicted using

an established correlation[7] - has led us to investigate the reaction of solid TPMCl with water to see if the hydrolysis to form TPMOH is authentically heterogeneous, occurring *at* the solid-liquid boundary, or if it proceeds via initial release of TPMCl from the reacting surface followed by hydrolysis in homogeneous bulk solution. Given the behaviour in liquid-liquid systems[1-4], TPMCl was selected as a likely candidate for the genuine heterogeneous reaction.

In the present study, the CFC method is employed to investigate the hydrolysis of solid TPMCl using as substrates, namely, fused pellets, pellets pressed from particulate material, and single crystals. Both a chloride ion-selective electrode (ISE) and a conductivity detection system are separately employed for the sensitive monitoring of the release of the hydrolysis products in the CFC. The former is optimally employed at relatively high ionic strengths (> 10^{-2} M) and the latter in more dilute solutions ($<10^{-2}$ M) so that the two detection approaches between them cover the full range of salt concentrations. The CFC kinetic results in conjunction with ex situ scanning electron micrographs and in situ atomic force micrographs of the reacted or reacting surface will be shown to allow the interfacial reaction mechanism to be unambiguously identified. In contrary to previous observations in the benzoic acid and cyanuric chloride systems (see chapter 1), the solid TPMCl hydrolysis will be shown to proceed directly *at* the reacting surface and its rate to be governed by the density of exposed chlorine atoms in any particular reacting crystal plane.

2. Experimental

A schematic diagram of the CFC is given in fig. 1. The cell is composed of a rectangular duct (about 4.5 cm long, 0.1 cm deep and 0.6 cm wide) cut in a perspex block and closed by a

cover plate. The solid substrate, either in the form of fused pellet, pressed pellet or single crystal, was embedded into the cover plate together with a downstream detector system (see below) for the monitoring of the amount of product released. The ionic strength of solutions was adjusted to the desired value by adding potassium nitrate. Solution flow rates were obtained in the range 10^{-3} - 10^{-1} cm^3 s^{-1} using gravity feed as described in chapter 3. In every experiment, the temperature of the flow system was maintained at 25 $\pm$ 0.5 °C using an air-thermostat.

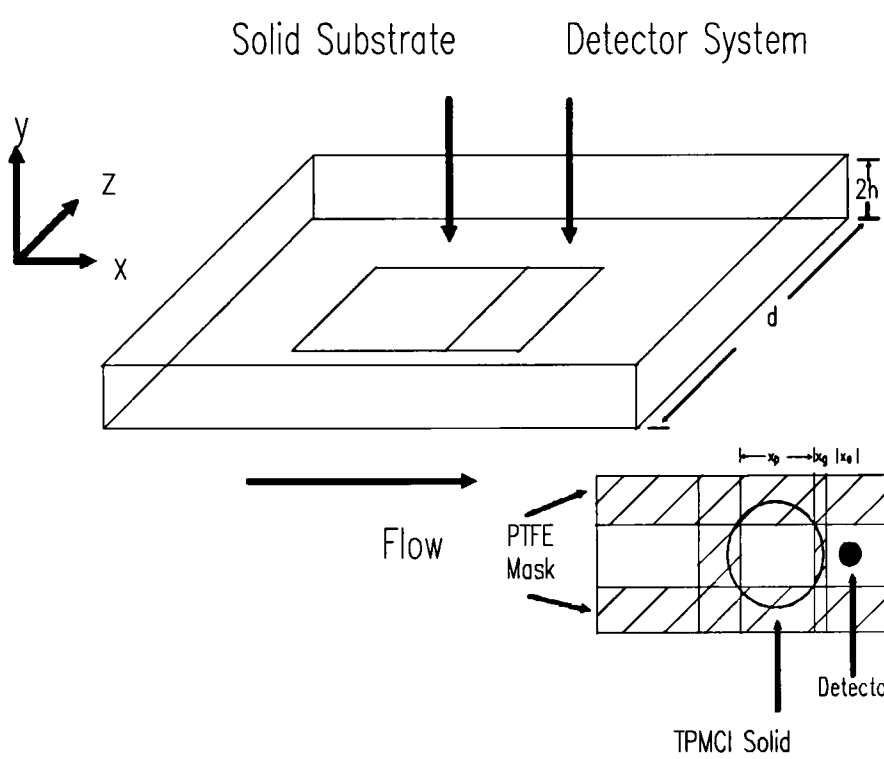


Fig. 1 Schematic diagram of a channel flow cell. The inset shows a top view of the cover plate utilised in this work.

The cover plate was a block of teflon or perspex with a circular hole (see fig. 1) which supported a pellet or single crystal. The solid substrate was masked with thin teflon tape so that a known area of the solid was exposed to solution. Commercially available TPMCl powder (Aldrich, 98%) was employed without further purification. For a fused pellet, the compound was directly melted in the teflon block at 150°C. Upon solidification, the surface of the pellet was cut flush to the cover plate surface. In the case of pressed pellet a pressure

of 5.6 x 10^9 Nm^{-2} was generated by a screw press to compact the powdered material directly into the perspex block.[8] The particle size distribution of the TPMCl powder used to form pressed pellets was measured by a light scattering method (Zeneca, Huddersfield). Three different distributions were used; the finer powders being obtained from the coarser by mechanical grinding. Fig. 2 shows the measured size distributions in each case. The distributions are characterised by the sizes corresponding to cumulative frequencies of 10% and 90%. These values for the three distributions used were (i) 14-90 μm, (ii) 47-445 μm and (iii) 36-738 μm respectively. In the subsequent discussions, the porosity of surface layers formed by compacting these separate powders are scrutinised. It is therefore appropriate to consider a volume average particle diameter, D[4, 3], defined as $\Sigma nd^4/\Sigma nd^3$ where d indicates the diameter of individual particles of number n in the system.[9] D[4,3] for particles with distributions (i), (ii) and (iii) are 53.5 μm, 278.2 μm and 332.4 μm respectively.

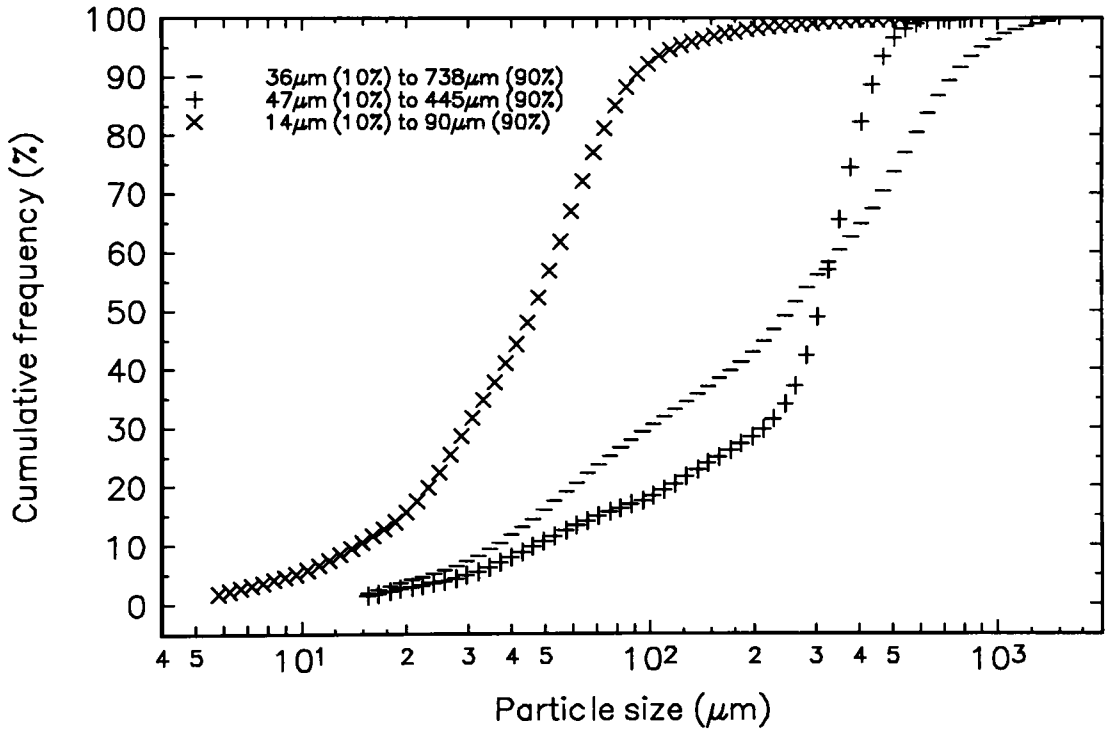


Fig. 2 The particle size distribution of the three different TPMCl samples used to form pressed pellets.

For single crystal experiments it was noted that Kahr and Carter[10] reported that TPMCl could be grown into three different crystal forms. They prepared a trigonal polymorph ("Phase I" in their notation[9]) from toluene solution while two triclinic polymorphs (Phases II and III) were grown from pentane. In the present study, the slow cooling process[11-14] was implemented to grow a single triclinic form (Phase III). First a saturated solution was produced by adding TPMCl powder to 500 cm^3 of anhydrous pentane (Aldrich, > 99%) at 32°C. The resulting solution was filtered twice and 0.5 cm^3 of acetyl chloride added to the filtrate. Then the saturated solution was transferred to a 500 cm^3 crystal growing flask. Seed crystals were added and the solution was kept in motion by a slowly rotating paddle to ensure a homogeneous TPMCl concentration over the crystal surface. The solution cooling rate was controlled by using a home-built programmable temperature water bath. It consisted of an electronic controlling system which triggered the illumination of a powerful light bulb (275 W IR reflector, GEC Electronics), which shone through one side of the water bath to provide heating. The rate of cooling was adjusted to 0.04 °C per hour from 32 °C to 20 °C. The crystals formed were approximately 0.5 x 0.5 x 0.5 cm in size. Fig. 3 shows a typical crystal grown from this procedure. Suitably smaller sized crystals prepared in this way were characterised using x-ray diffraction techniques (Chemical Crystallography Laboratory, Oxford). The unit cell dimensions (a = 14.1506 **Å**, b = 21.3190 **Å**, c = 13.0559 **Å**) and the unit cell volume (V = 3710 Å^3) are in excellent agreement with the results of Kahr and Carter[9] as quoted in chapter 5. Contact goniometry together with a genetic algorithm-based matching procedure[15] (see chapter 5) were used to index large crystals. Different crystal planes - $(\bar{1}11)$, (010) and $(\bar{1}10)$ - were employed for CFC experiments. To this end the crystal was embedded into cover plate using wax so that only a specific plane was exposed to the solution.



Fig. 3 Single crystals grown from pentane using a cooling rate of 0.04 °C hr^{-1}. The approximate dimensions of the crystals are 0.6 cm x 0.6 cm x 0.4 cm. The uppermost plane of the left crystal is $(\bar{1}11)$ *while for the right one is* (010) *. At the top of the figure is a British 5 pence coin with diameter of about 1.7 cm.*

A chloride ISE or a conductivity detection system were utilised to monitor the release of the ionic products H^+ and Cl^- from the reacting interface. For the ISE a silver rod of diameter about 1.5 mm was polished smooth and formed into a silver/silver chloride electrode via an established method as given in chapter 3.[16] All potentiometric measurements were made with reference to a saturated calomel electrode and were accomplished using a Jenway 3030 pH meter. The ISE was calibrated *in situ* after each experiment by flowing known concentrations of chloride solution through the flow cell. A typical calibration plot for the ISE is given in fig. 4. A gradient of -57 (±1) mV per decade was usually obtained in the calibration plots in good agreement with the theoretical Nernstian response of -59.6 mV per decade expected at 25 °C. Full details of the conductivity detection system can be found in chapter 4.[17] In summary it comprises a pair of 0.4 x 0.4 cm platinum electrodes positioned about 0.9 cm

downstream of the reacting surface. The impedance between the two Pt electrodes was measured using a Solarton 1250 Frequency Response Analyser in conjunction with a Solarton 1286 Electrochemical Interface (Suhlumberger Electronics Ltd., Farnborough, UK).

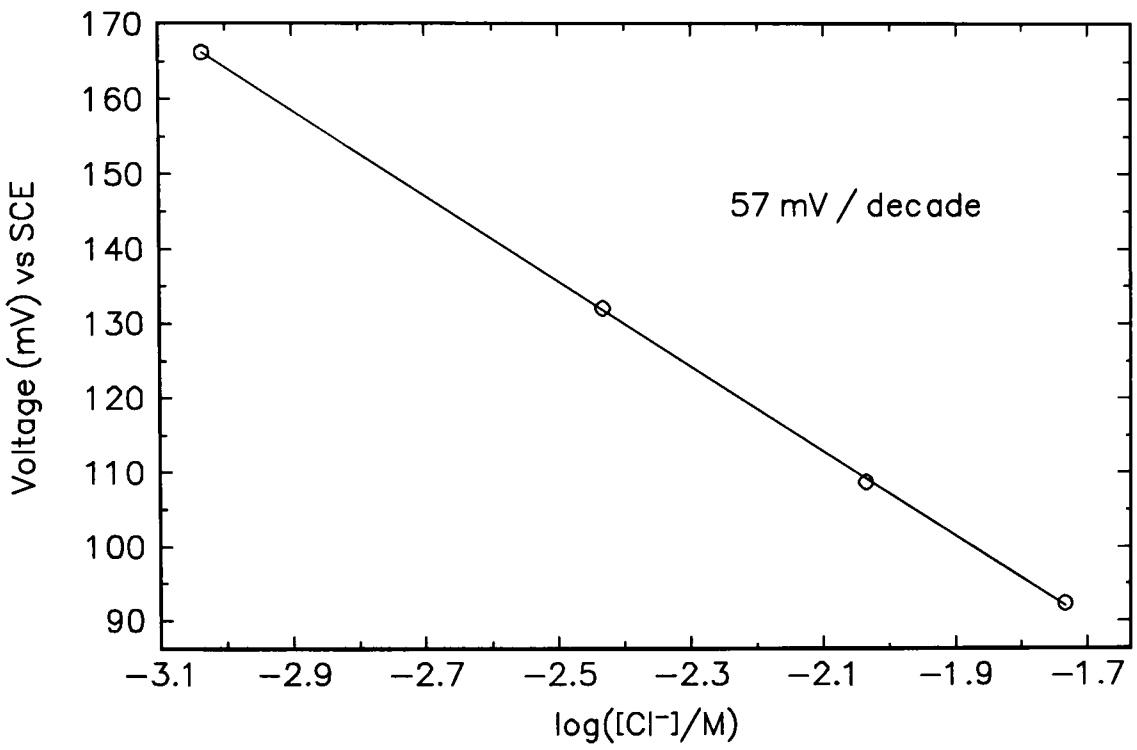


Fig. 4 Calibration plot for the Ag/AgCl ion selective electrode.

A TopoMetrix TMX 2000 atomic force microscope (AFM) and a Hitachi S570 electron microscope were employed to image the surfaces of the solid substrate. A commercial TopoMetrix liquid cell was used for in situ AFM imaging of the reacting crystal surface. Prior to a given AFM experiment the crystal surface was cleaned using pentane. The AFM was operated in repulsive force mode.

3. Results and Discussion

Preliminary experiments were pursued, before undertaking CFC studies, in which a 0.7 g pellet pressed from particles in the size range 47-445 μm (fig. 2) was allowed to hydrolyse in 20 cm^3 of 0.094 M KNO_3 solution. The chloride concentration was monitored using a

silver/silver chloride ISE. As shown in fig. 5, the concentration of the chloride ion released was found to increase over a period of up to 5 hours. This suggests that the reacting surface remains active during this period.

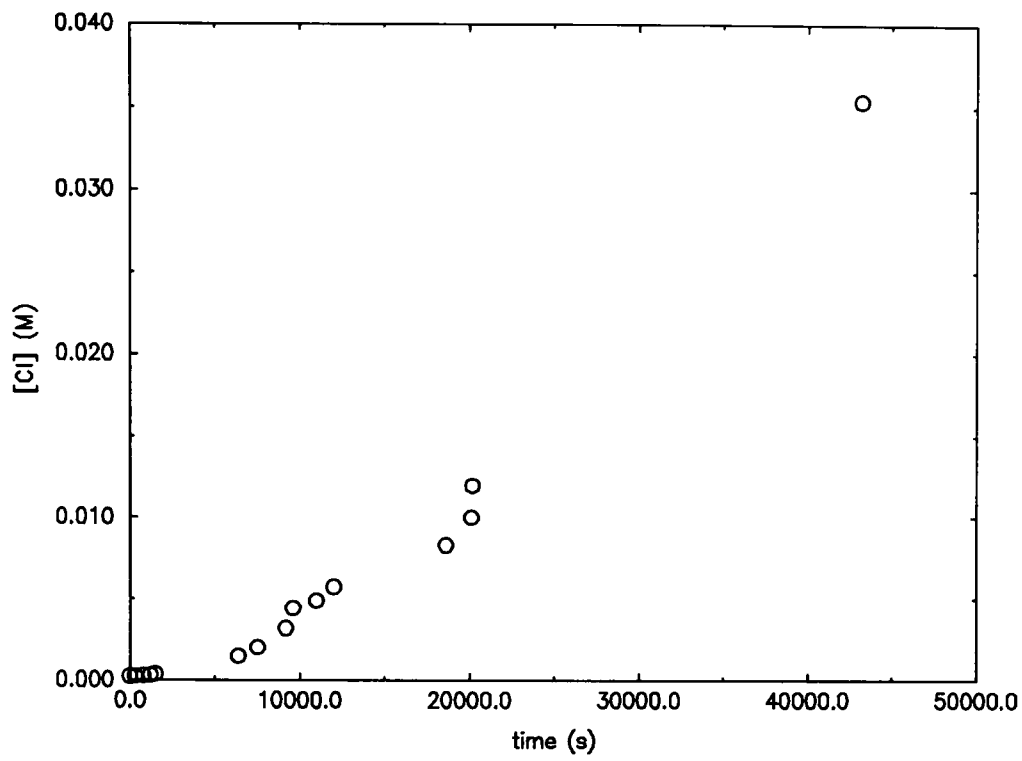


Fig. 5 A plot of chloride concentration as a function of time for the hydrolysis of pressed pellet (47-445 μm particles) in 0.094 M KNO_3 solution.

Attention was then directed to CFC measurements. A series of experiments were conducted in which a solution of 0.094 M KNO_3 was flowed in turn over pressed pellets, fused pellets and single crystals. In each case a downstream chloride ISE was used to monitor the hydrolysis. Figs. 6-10 show representative experimentally measured chloride concentrations in each case as a function of the solution flow rate. It should be noted that the chloride concentration decreases with increasing flow rate as a result of "convective dilution" of the product before it reaches the detector. As the flow rate (convection) increases, the concentration of Cl^- in the zone of the detector decreases since the flux of Cl^- injected from the reacting surface remains essentially fixed.

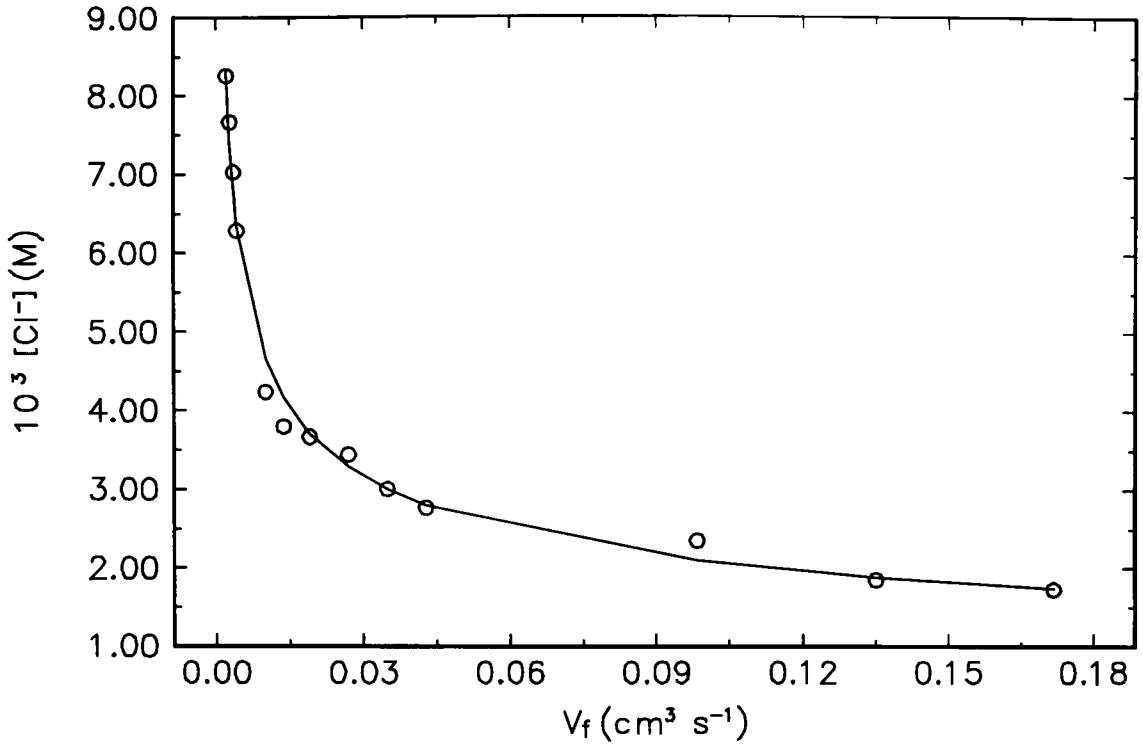


Fig. 6 The flow rate dependence of the downstream chloride concentration measured using the ISE for the dissolution of a pressed pellet (36-738 μm particles) into 0.094 M potassium nitrate solution: (o) experimental data and —— theoretical behaviour predicted using the interfacial reaction model and the optimised rate constant shown in table 1. Cell geometry (see fig. 1): pellet length, x_p = 9.15 mm, distance between pellet and detector, x_g = 3.11 mm, detector length, x_e = 1.59 mm, cell depth, 2h = 0.102 cm and channel width, d = 0.581 cm.

Fig. 7 The flow rate dependence of the downstream chloride concentration measured using the ISE for the dissolution of a fused pellet into 0.094 M potassium nitrate solution: (o) experimental data and —— theoretical behaviour predicted using the interfacial reaction model and the optimised rate constant shown in table 1. Cell geometry: x_p = 7.20 mm, x_g = 2.45 mm, x_e = 1.45 mm, 2h = 0.102 cm and d = 0.581 cm.

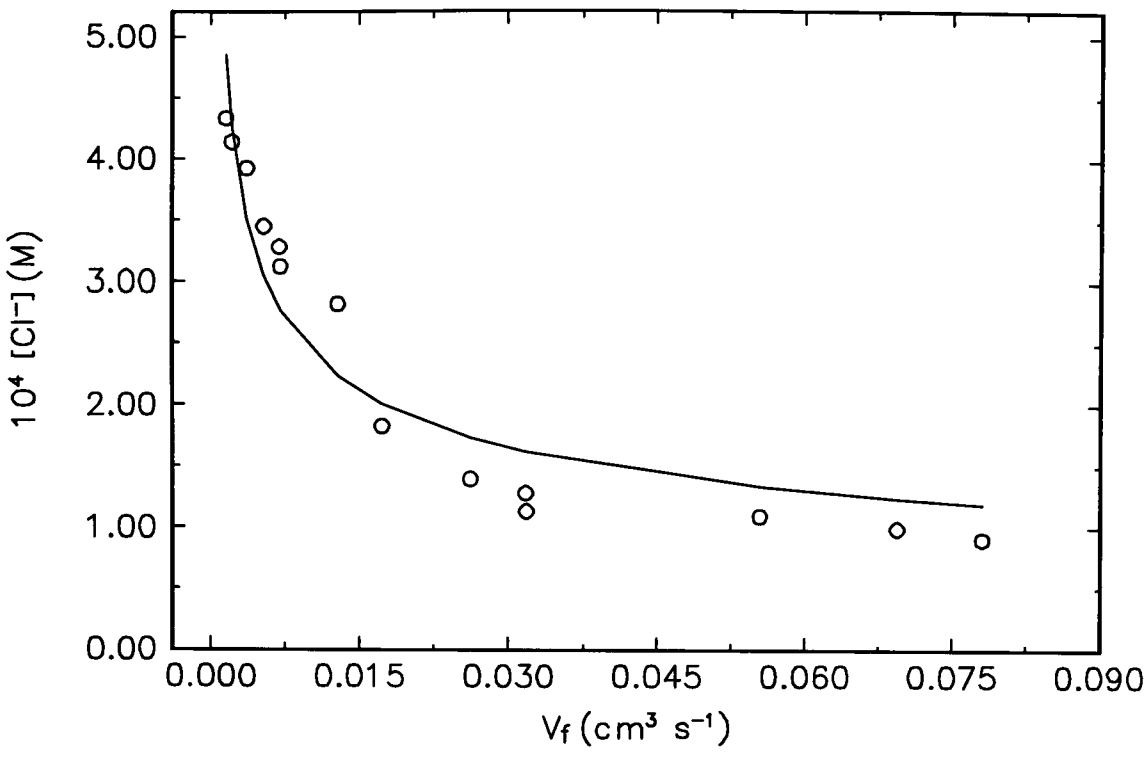


Fig. 8 The flow rate dependence of the downstream chloride concentration measured using the ISE for the dissolution of the single crystal plane $(\bar{1}11)$ *into 0.094 M potassium nitrate solution: (o) experimental data and —— theoretical behaviour predicted using the interfacial reaction model and the optimised rate constant shown in table 1. Cell geometry:* x_p *= 1.91 mm,* x_g *= 3.13 mm,* x_e *= 1.59 mm, 2h = 0.102 cm and d = 0.581 cm.*

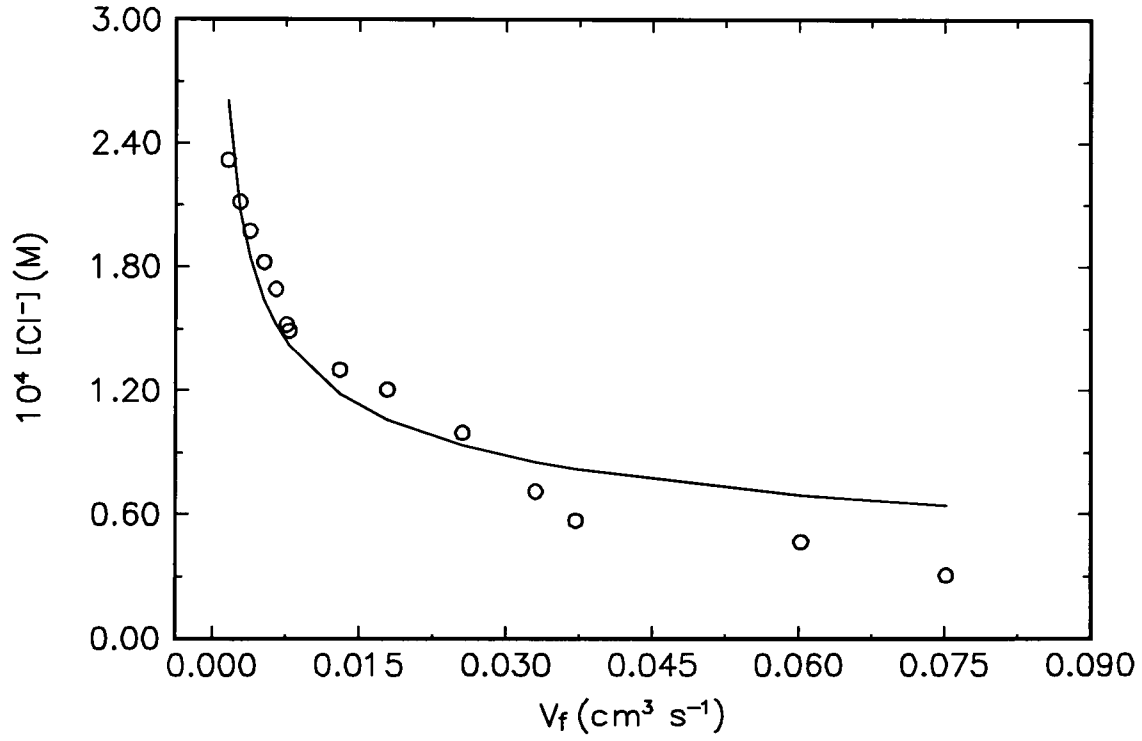


Fig. 9 The flow rate dependence of the downstream chloride concentration measured using the ISE for the dissolution of the single crystal plane (010) *into 0.094 M potassium nitrate solution: (o) experimental data and —— theoretical behaviour predicted using the interfacial reaction model and the optimised rate constant shown in table 1. Cell geometry:* x_p *= 2.30 mm,* x_g *= 5.91 mm,* x_e *= 1.60 mm, 2h = 0.102 cm and d = 0.581 cm.*

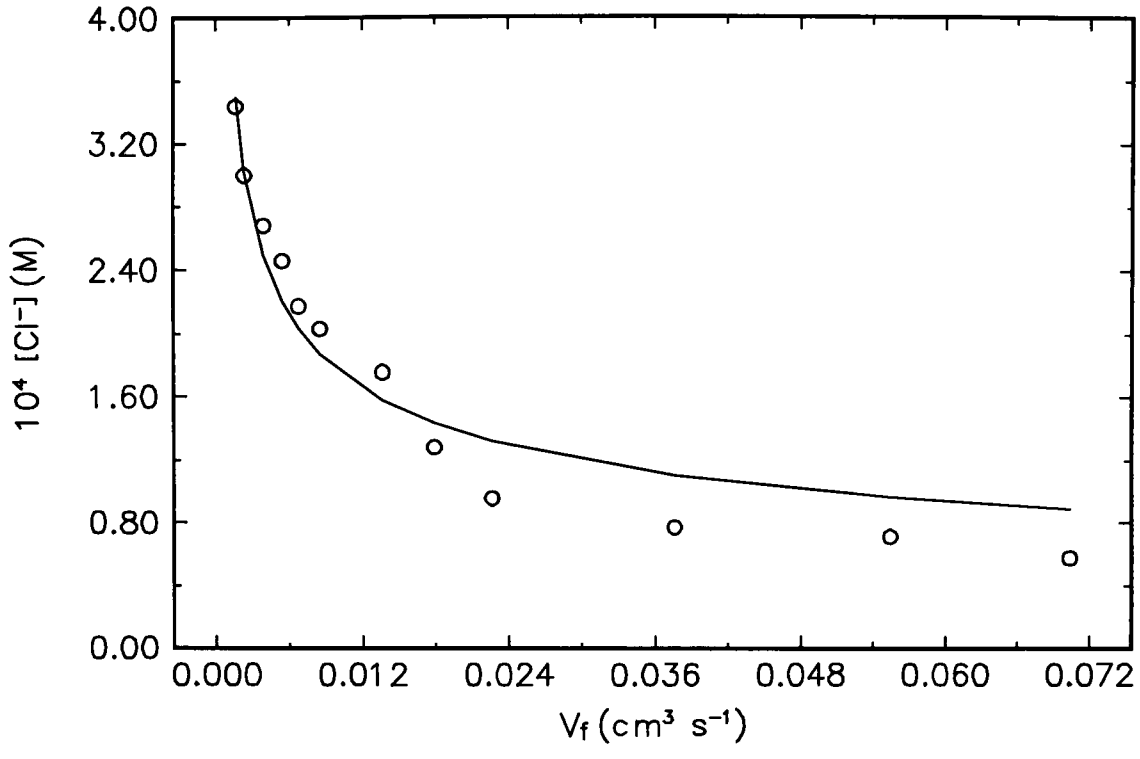


Fig. 10 The flow rate dependence of the downstream chloride concentration measured using the ISE for the dissolution of the single crystal plane $(\bar{1}10)$ *into 0.094 M potassium nitrate solution: (o) experimental data and —— theoretical behaviour predicted using the interfacial reaction model and the optimised rate constant shown in table 1. Cell geometry:* x_p *= 4.03 mm,* x_g *= 5.35 mm,* x_e *= 1.46 mm, 2h = 0.102 cm and d = 0.581 cm.*

The following reaction model was examined to see whether it was consistent with the CFC kinetic data:

$$TPMCl\ (s) \xrightarrow{k_i} TPM^+\ (aq) + Cl^-\ (aq) \qquad (1)$$

$$TPM^+\ (aq) + H_2O\ (l) \xrightarrow{fast} TPMOH\ (aq, s) + H^+\ (aq) \qquad (2)$$

where k_i (mol cm^{-2} s^{-1}) denotes an interfacial reaction rate constant describing the release of triphenylmethyl cations and chloride anions directly from the solid surface into solution. This is assumed to be the rate determining step. The steady state convective-diffusion equation describing the distribution of the chloride anion within the CFC is

$$\frac{\partial[Cl^-]}{\partial t} = D_{Cl^-}\frac{\partial^2[Cl^-]}{\partial y^2} - v_x\frac{\partial[Cl^-]}{\partial x} = 0 \qquad (3)$$

where the coordinates x and y are defined in fig. 1 and D_{Cl^-}, v_x and t represent, respectively, the diffusion coefficient of the chloride anion, the solution flow rate in the x-direction and time. Values of D_{Cl^-} in aqueous media of different ionic strengths at 25 °C have been given by Turq et al.[18] In the present study, a quantity of 1.96 x 10^{-5} cm^2 s^{-1} was employed in the modelling of kinetic data obtained from high ionic strength media (> 10^{-2} M).

The theoretical flow rate dependence of the chloride concentration registered by the detector ISE may be computed by solving the above convective-diffusion equation. This requires the specification of appropriate boundary conditions. At the surface of the solid substrate, an interfacial kinetic equation is adopted as the appropriate boundary condition:

$$k_i\ (mol\ cm^{-2}\ s^{-1}) \quad = \quad -D_{Cl^-}\frac{\partial[Cl^-]}{\partial y} \qquad (4)$$

Upstream of the solid the chloride concentration is specified as zero whilst on all boundaries of the cell other than the reacting surface a no-flux condition is pertinent:

$$0 \quad = \quad D_{Cl^-}\frac{\partial[Cl^-]}{\partial y} \qquad (5)$$

Solution of the problem is readily accomplished using the Backward Implicit Finite Difference Method applied in essentially standard form as described in chapter 2.[19,20] The computations, when made for a known cell geometry and specified D_{Cl^-}, predict the chloride ion concentration throughout the flow cell as a function of flow rate for assumed values of the rate constant k_i. The latter is the only adjustable parameter in the model. The resulting Cl^- concentration profiles may be used to predict the average [Cl^-] over the surface of the potentiometric sensor and hence to predict the dependence of the detector signal on flow rate for different values of k_i. The theoretical data can be related to the experimental results to give a best-fit value of k_i by minimising the deviation between the theoretical and

experimental concentrations in a least squares sense. An error function, root-mean-square-deviation (RMSD) is defined as

$$RMSD = \sqrt{\frac{\sum_{j=1}^{N} ([Cl^-]_{expt,\,j} - [Cl^-]_{model,\,j})^2}{N}} \qquad (6)$$

where N is the number of data points. The problem then reduces to a non-linear minimisation of RMSD for optimal rate constants. The quasi-Newton method[21] was adopted for this purpose.

The results of the CFC experiments described above were fitted with the proposed heterogeneous reaction model. Figs. 6-10 depict the theoretical behaviour generated using the optimised rate constants which are reported in table 1. Excellent agreement between experimental data and theoretical prediction is noted over the entire flow rate range.

Table 1 The optimised interfacial rate constants, k_i, for the hydrolytic dissolution of TPMCl in $[KNO_3]$ = 0.094 M.

Sample	Rate constant* (mol cm^{-2} s^{-1})	Uncertainty** (mol cm^{-2} s^{-1})	RMSD*** (see eqn. 6)
Pressed (14-90 μm)	2.65×10^{-8}	$\pm 2.42 \times 10^{-10}$	6.47×10^{-4}
Pressed (47-445 μm)	2.37×10^{-8}	$\pm 5.84 \times 10^{-10}$	3.63×10^{-4}
Pressed (36-738 μm)	1.48×10^{-8}	$\pm 1.88 \times 10^{-9}$	2.23×10^{-4}
Fused	2.00×10^{-9}	$\pm 1.51 \times 10^{-10}$	5.33×10^{-5}
Crystal ($\bar{1}11$)	3.28×10^{-9}	$\pm 2.65 \times 10^{-10}$	2.83×10^{-5}
Crystal (010)	1.94×10^{-9}	$\pm 1.91 \times 10^{-10}$	2.59×10^{-5}
Crystal ($\bar{1}10$)	1.22×10^{-9}	$\pm 3.13 \times 10^{-10}$	2.07×10^{-5}

* *Mean value from at least three separate experiments*

** *Standard deviation from at least three separate experiments*

*** *Mean value from at least three separate experiments. RMSD value obtained from fitting the saturated surface model to the experimental data is 2.82×10^{-3} (see fig. 11)*

This was in marked contrast with other reaction models that were attempted. In particular, poor fits or unsatisfactory interpretations were obtained for models assuming either a saturated surface of TPMCl or a constant flux of TPMCl, followed in both cases by homogeneous hydrolysis (see chapters 2 & 4). As shown in fig. 11, the theoretical product concentration calculated using the estimated solubility of TPMCl[7] (8.64 x 10^{-8} M) was far below the experimental one regardless of the magnitude of the homogeneous hydrolysis rate constant employed.

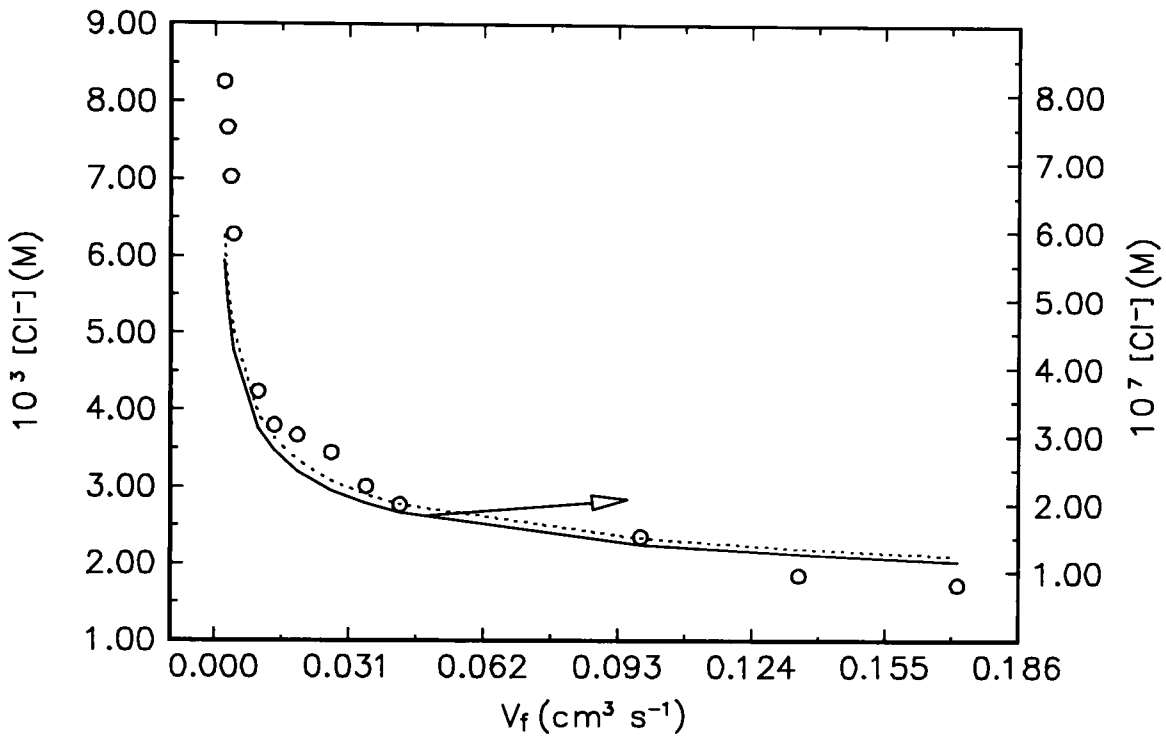


Fig. 11 The flow rate dependence of the downstream chloride concentration measured using the ISE for the dissolution of a pressed pellet (36-738 μm particles) into 0.094 M potassium nitrate solution: (o) experimental data and theoretical behaviour predicted using the saturated surface model with a homogeneous hydrolysis rate constants of 50 s^{-1} (——) and 60 s^{-1} (---). Cell geometry is the same as shown in caption of fig. 6.

Returning to table 1 it can be seen that for the pressed pellets the rate constants decrease as the particle size used to form them increases. Also the fused pellet samples show a slower rate of rate interfacial hydrolysis. These trends probably reflect the surface morphology of the different samples. Scanning electron microscopy (SEM) analysis of the pellet surfaces before dissolution revealed that the pressed pellet surfaces were more porous than that of the fused

one. Figs. 12 and 13 show, respectively, the SEM micrographs of a fused pellet and a pressed pellet formed from particles in the size range 36-738 μm from which this difference is readily apparent. Figs. 14 and 15 show the corresponding SEM images for the TPMCl particles size ranges 14-90 and 47-445 μm which are qualitatively similar to the 36-738 μm picture; however the surfaces formed by compressing the materials with wide particle size distributions are appreciably less porous than that from those with narrow distributions. This is probably due to the fact that particles with dissimilar diameters are moulded more easily in a tightly packed manner. Obviously, the larger the surface porosity the larger the apparent interfacial rate constant because of both the increased surface area and the fact that the pores can act as reservoirs of solution containing a high concentration of hydrolysis products.[22,23] Clearly, more structural work is required to definitely establish the effect of surface porosity; this is beyond the aim of the present work which seeks to establish the interfacial nature of the hydrolysis process.

Table 1 also shows rate constants, k_i, measured for three different crystal planes of single crystals of TPMCl. These were found to obey the sequence:

$$k_i\ (\bar{1}11) \ > \ k_i\ (010) \ > \ k_i\ (\bar{1}10)$$

This trend may be rationalised by assuming that the experimental crystal planes approximate to flat surfaces and by generating images of each of the three crystal faces studied, using the known crystal structure (as deduced from diffraction) and including molecules located within a depth of 5 Å of the nominal surface. The plots are shown in figs. 16-18. It is possible to distinguish repeat boxes for each of the surfaces. These are in effect 2-dimensional "unit cells" for the surface: each figure shows four of these cells. Inspection of the three figures shows that the density of exposed chlorine atoms changes between the different surfaces. Fig.

Fig. 12 SEM micrograph of a fused pellet before dissolution. The dotted bar represents a distance of 30 μm.

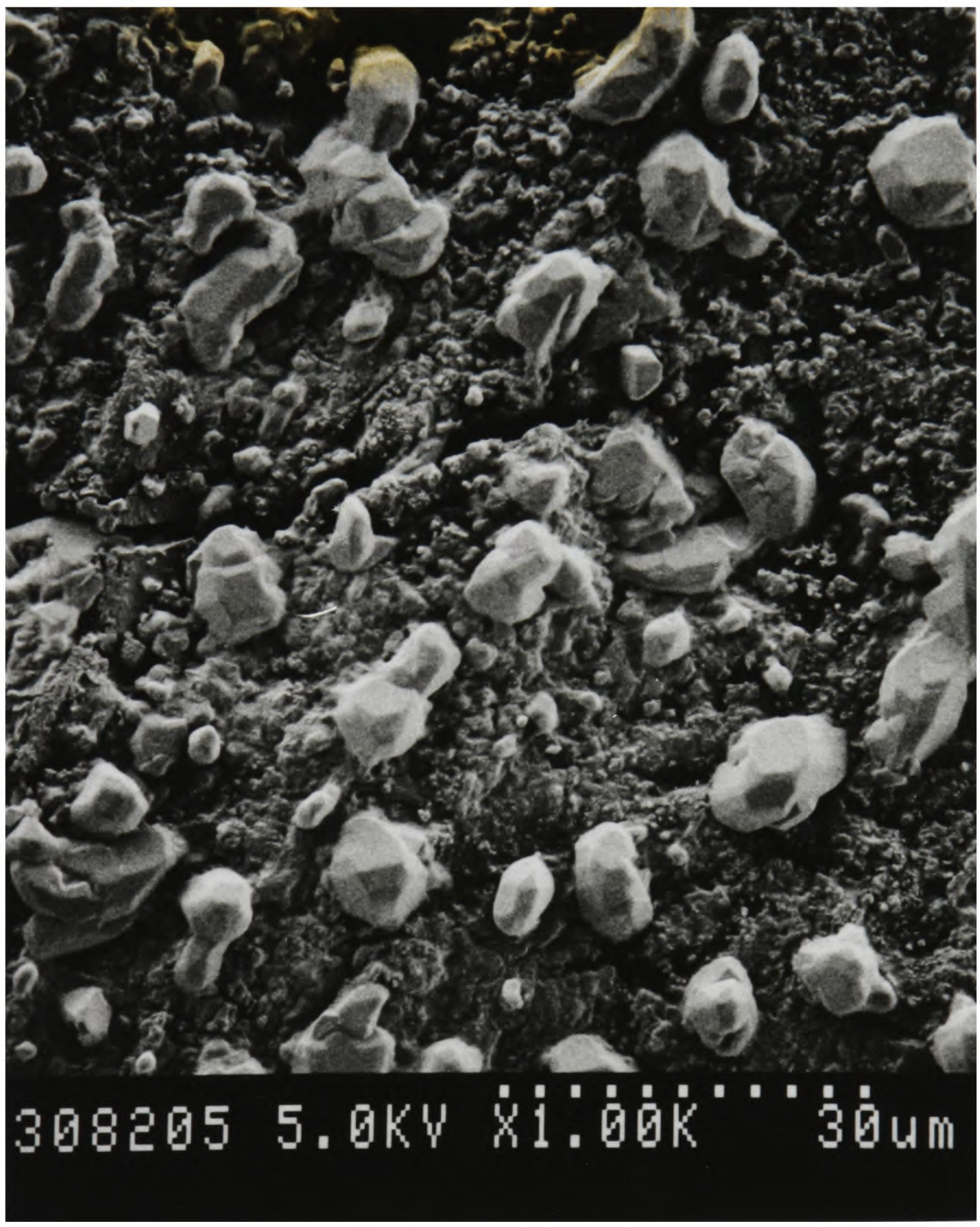


Fig. 13 SEM micrograph of a pressed pellet (36-738 μm particles, D[4,3] = 332.4 μm) before dissolution. The dotted bar represents a distance of 30 μm.

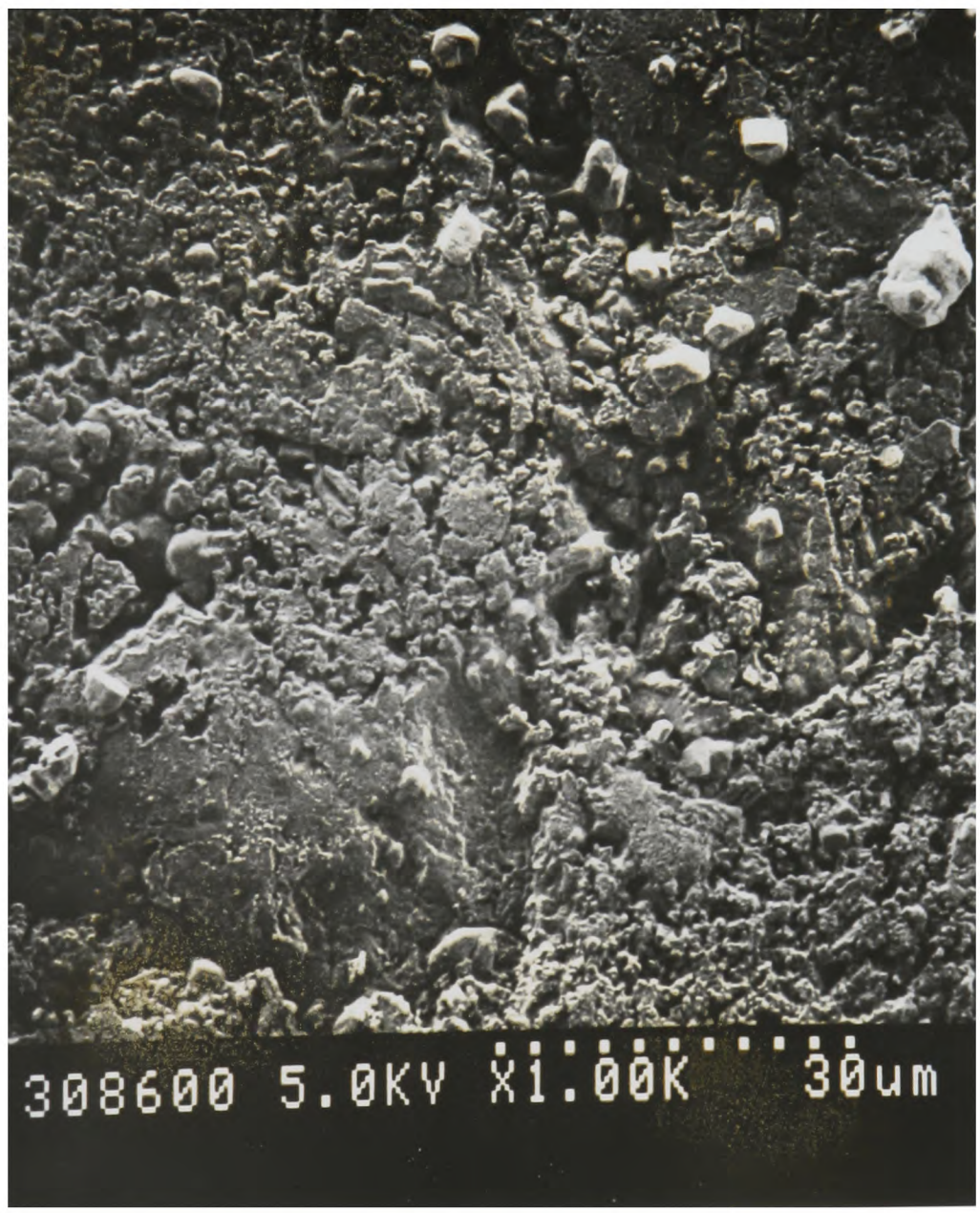


Fig. 14 SEM micrograph of a pressed pellet (14-90 μm particles, D[4,3] = 53.5 μm) before dissolution. The dotted bar represents a distance of 30 μm.



Fig. 15 SEM micrograph of a pressed pellet (47-445 μm particles, D[4,3] = 278.2 μm) before dissolution. The dotted bar represents a distance of 30 μm.



Fig. 16 Phase III TPMCl single crystal surface: (T11) *plane.*

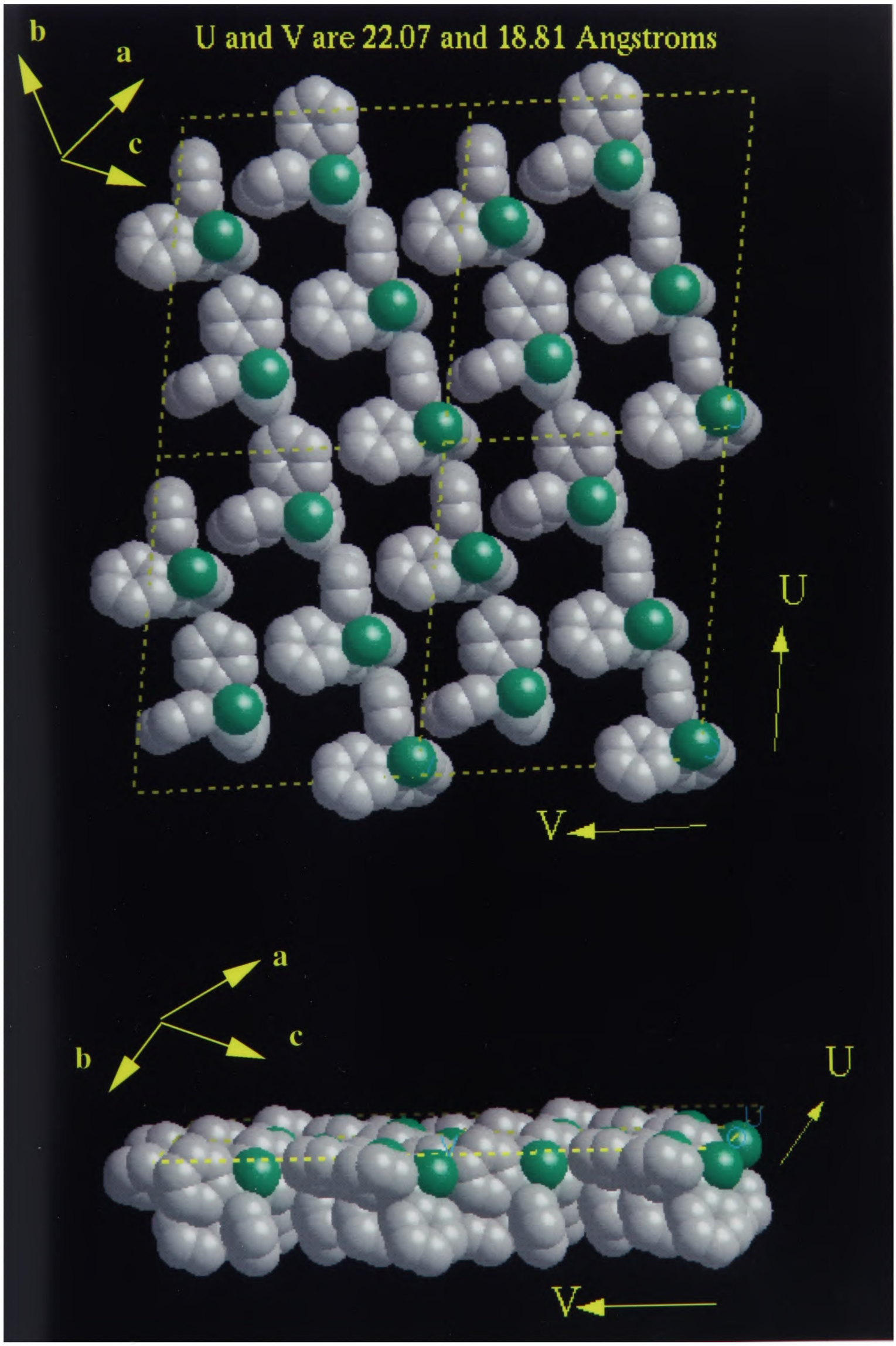


Fig. 17 Phase III TPMCl single crystal surface: (010) *plane.*

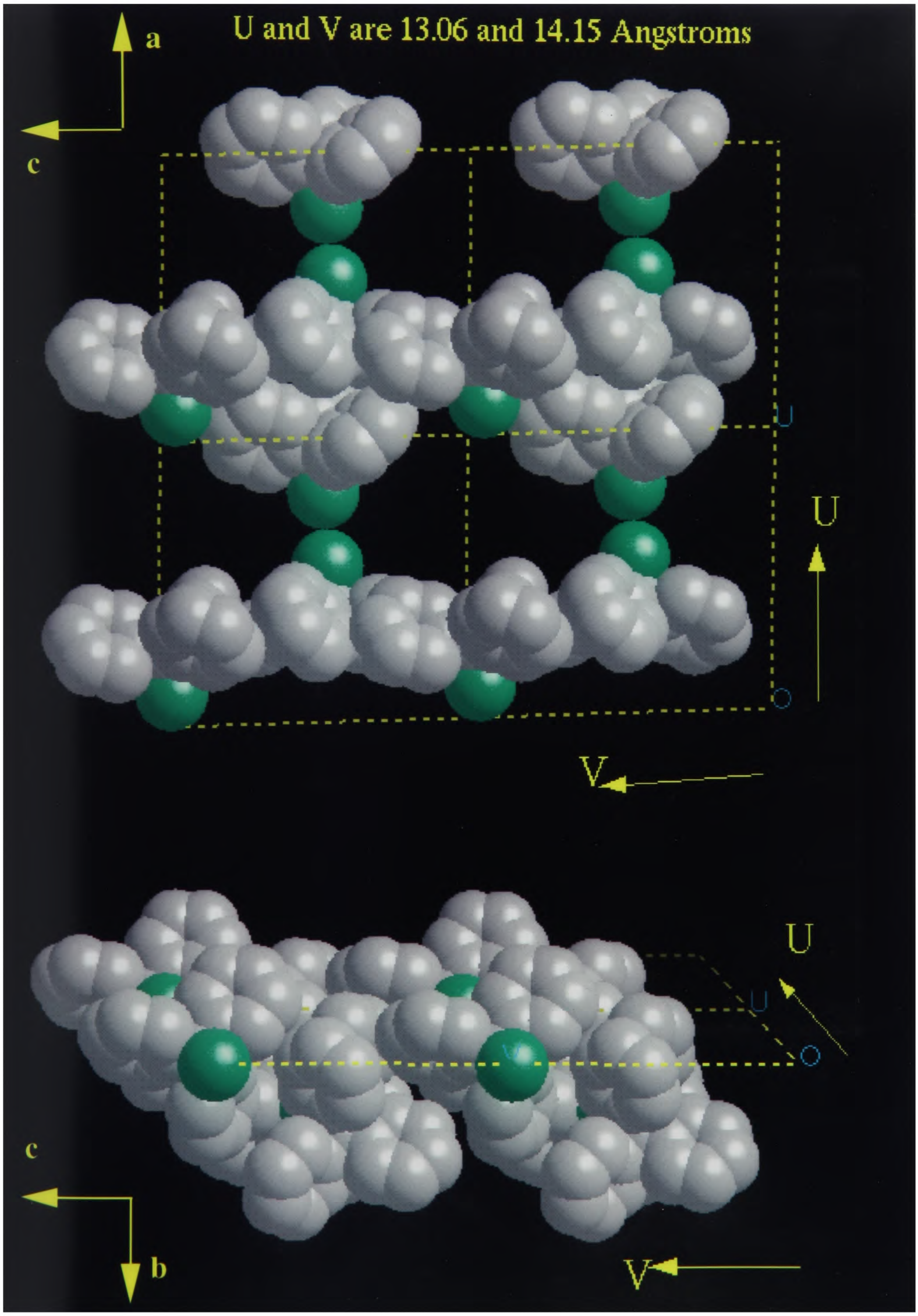
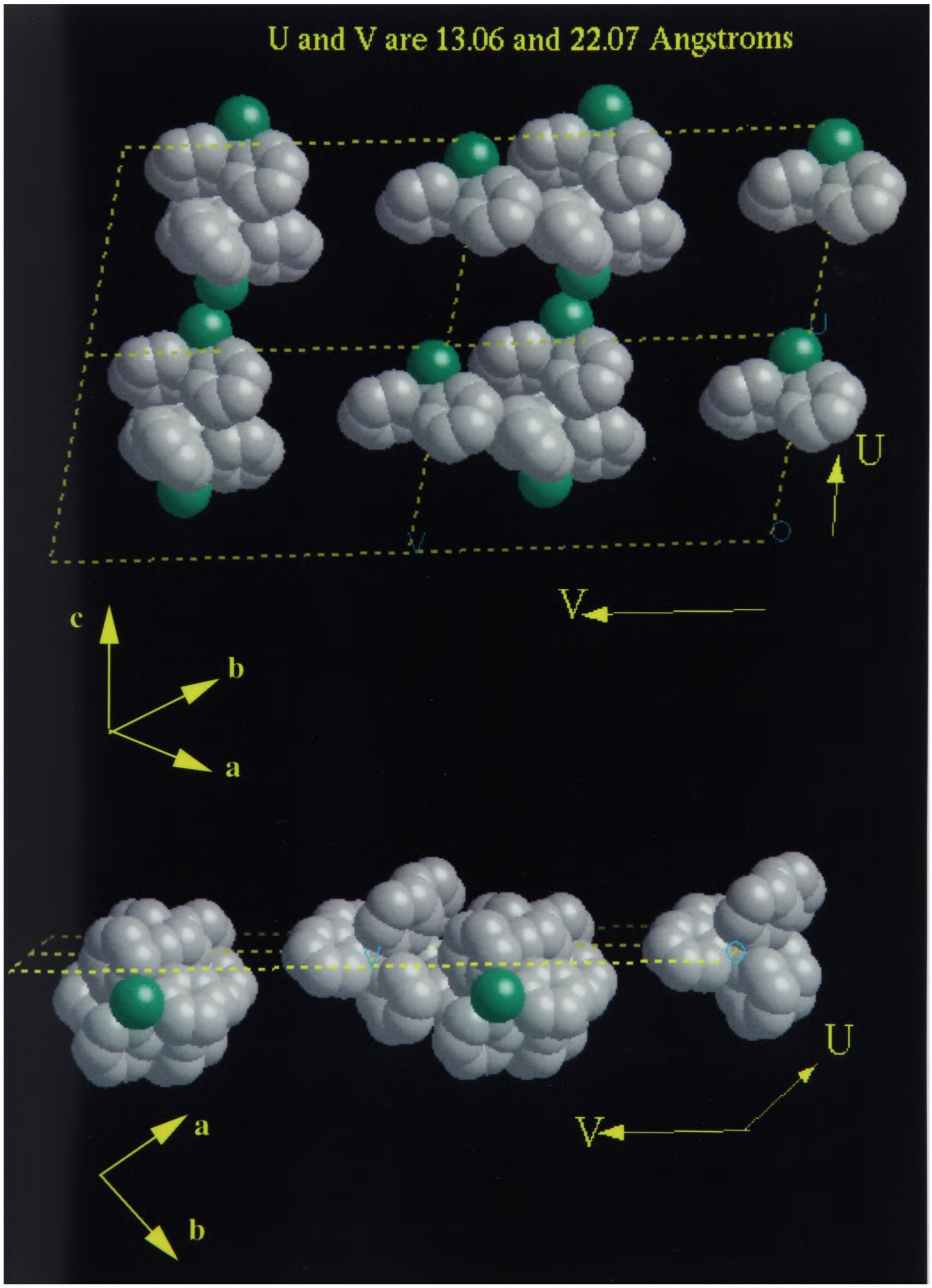


Fig. 18 Phase III TPMC single crystal surface: (T10) plane.



16 relates to the $(\bar{1}11)$ surface and contains 5 molecules per repeat box, which is of size 22.1 x 18.9 Å. All the chlorine units are pointing up out of this surface so that there is one reactive unit per 82.7 $Å^2$. Fig. 17 shows the (010) surface. The basic repeat unit is 13.1 x 14.1 Å and contains three molecules. Only one is actually on the surface; the other two are buried at 3.5 Å below. There is thus one "surface" chlorine atom every 180.0 $Å^2$. The $(\bar{1}10)$ surface is shown in fig. 18. This again has three "surface" chlorine atoms per repeat box (22.1 x 13.1 Å). Two of these are about 3 Å below the other which is again located in the surface plane. There is one "surface" chlorine atom 288.8 $Å^2$. The molecule(s) at the surface have chlorine atoms which point out of the surface in the case of the $(\bar{1}11)$ plane but in the other two planes the surface molecules have their chlorine atoms lying "in" and along the surface plane. It is likely that the $(\bar{1}11)$, (010) and $(\bar{1}10)$ crystal planes will have different solubilities. However, the kinetic experiments presented above indicate that molecules of TPMCl react *without* leaving the solid surface and so we believe solubility contributions are not pertinent to the discussion.

Table 2 summarises the density of the chlorine atoms within 5 Å of the surface of the planes $(\bar{1}11)$, (010) and $(\bar{1}10)$. Comparison of these values with the interfacial rate constants measured for the hydrolytic dissolution of TPMCl shows a strong correlation and suggests that the measured differences between the different crystal planes reflects the chlorine atom "availability" in each plane. Note that the three dimensional unit cells of volume defined by the two-dimensional cells specified above and by a depth of one interplanar spacing contain a total of 10 molecules for *each* of the three planes considered. If removal of one surface molecule leads to the dissolution of a unit cell then the dissolution rate will reflect the relative chlorine densities given in table 2. Therefore, the observed correlation provides further

evidence to that obtained from the CFC kinetic data that the hydrolytic dissolution is indeed an authentically *heterogeneous* process.

Table 2 A comparison of the relative hydrolytic dissolution rates to the density of chlorine atoms at the surface of the crystal planes $(\bar{1}11)$, (010) *and* $(\bar{1}10)$.

Plane	Area (Å^2) per reactive Cl atom	Relative Cl density in surface	Rate constant (mol cm^{-2} s^{-1})	Relative reactivities
$(\bar{1}11)$	83	3.5	3.28×10^{-9}	2.7
(010)	184	1.6	1.94×10^{-9}	1.6
$(\bar{1}10)$	288	1.0	1.22×10^{-9}	1.0

Next CFC dissolution experiments using pressed pellets (47-445 μm particles) were performed in aqueous media of different ionic strength. The chloride ISE sensor was used to study KNO_3 solutions of concentration 0.0940 M and 0.0189 M while the conductivity detection system was adopted for 1.50 mM and 0.754 mM concentrations. In each case the interfacial model of hydrolytic dissolution was found to give an excellent fit to the detector signal/flow rate data. The optimised rate constants for the different concentrations of potassium nitrate are listed in table 3.

Table 3 The optimised rate constants for the hydrolytic dissolution of TPMCl pressed pellets (47-445 μm particles) in aqueous solutions of different ionic strength.

[KNO_3] (M)	Rate constant* (mol cm^{-2} s^{-1})	Uncertainty** (mol cm^{-2} s^{-1})
9.40×10^{-2}	2.37×10^{-8}	5.85×10^{-10}
1.89×10^{-2}	1.48×10^{-8}	1.05×10^{-9}
1.51×10^{-3}	1.26×10^{-8}	5.64×10^{-10}
7.54×10^{-4}	1.07×10^{-8}	8.25×10^{-10}

* *Mean value from at least three separate experiments*

** *Standard deviation from at least three separate experiments*

The values of k_i increase systematically with ionic strength suggesting, as expected, that the transition state has a greater charge than the reactants.[24] This is qualitatively compatible with the proposed mechanism. Considering the rate determining step (1) of the hydrolysis process, we can write

$$TPMCl\ (s) \overset{K}{\rightleftharpoons} \{TPM^+,\ Cl^-\}^* \overset{k^*}{\rightarrow} TPM^+\ (aq) + Cl^-\ (aq)$$

We assume the kinetic salt effect to operate on a transition state comprising the "separated" ions TPM^+ and Cl^- with charges Z_+ and Z_- respectively,

$$K = \frac{a^*_{TPM^+}\ a^*_{Cl^-}}{a\ (TPMCl,\ s)} = \gamma^*_{TPM^+}\ [TPM^+]^*\ \gamma^*_{Cl^-}\ [Cl^-]^* = \gamma_\pm^2\ [TPM^+]^*\ [Cl^-]^*$$

$$\therefore \quad [TPM^+]^*\ [Cl^-]^* = \frac{K}{\gamma_\pm^2}$$

Relating $[TPM^+]^*$ and $[Cl^-]^*$ to the interfacial kinetic rate constant (k_i),

$$k_i\ (mol\ cm^{-2}\ s^{-1}) = k^*\ [TPM^+]^*\ [Cl^-]^* = \frac{k^*\ K}{\gamma_\pm^2}$$

$$\log_{10} k_i = \log_{10} k^* K - \log_{10} \gamma_\pm^2$$

Applying Debye Hückel limiting law[25] for aqueous solution at 25 °C

$$\log_{10} \gamma_\pm = -0.509\ |Z_+ Z_-|\ I^{\frac{1}{2}}$$

where I represents the ionic strength of the solution, we can write

$$\log_{10} k_i = \log_{10} k^* K + 1.018\ \log_{10}\ |Z_+ Z_-|\ I^{\frac{1}{2}} \qquad (7)$$

Fig. 19 shows a plot of $\log(k_i)$ against $\sqrt{I}$. A reasonable straight line is apparent and the measured slope has a value of 1.13 which is within 10% of that predicted by eqn. 7 for $Z_+ = Z_- = 1$. This observation may suggest a fully dissociated transition state.

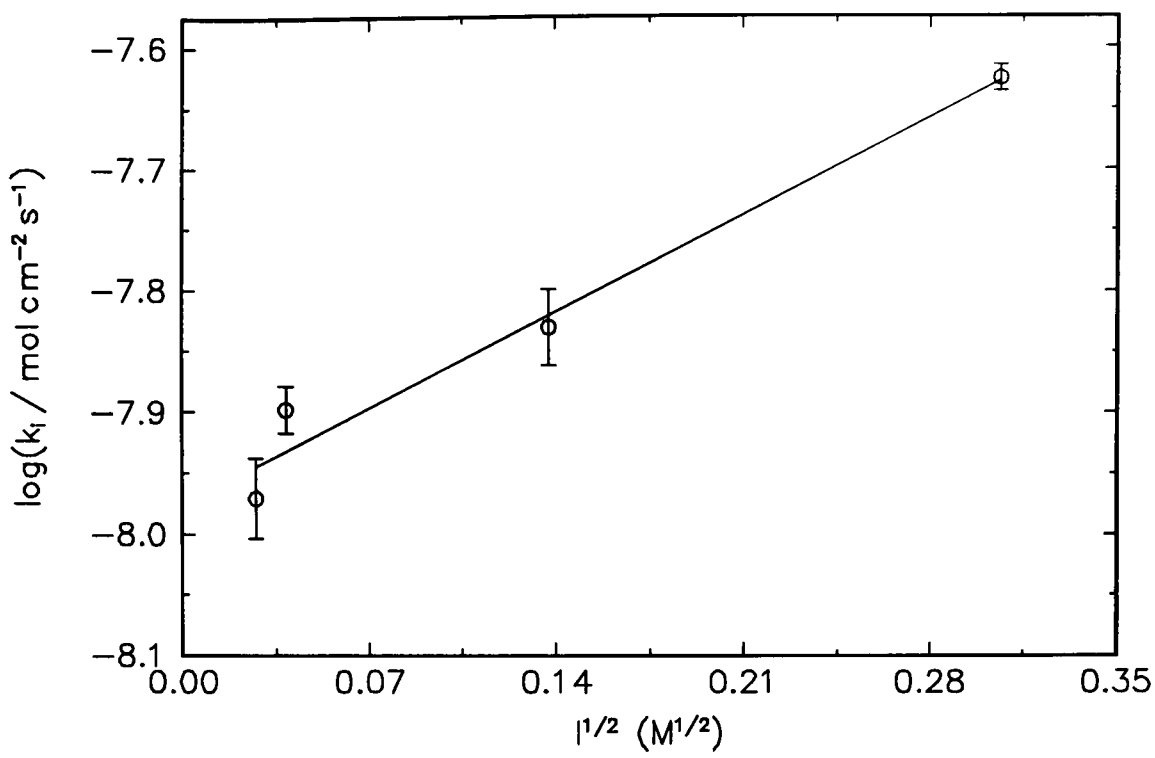


Fig. 19 A plot of log(k_i) against the square root of the ionic strength ($\sqrt{I}$) of the solution for the hydrolytic dissolution of TPMCl pressed pellets (47-445 μm particles) as suggested by the form of eqn. 7.

We next consider the fate of the reaction product TPMOH. CFC experiments with conductivity detection (see chapter 4) were conducted using pressed pellets (47-445 μm) at a fixed flow rate in the presence of varying amounts of added TPMOH (up to 3.6 x 10^{-5} M) to see if, over a period of time, any build-up of TPMOH at the surface might passivate the surface so inhibiting further hydrolytic dissolution. As shown in fig. 20, after a short initial transient, an essentially constant release of H^+ and Cl^- was observed for periods up to ca 3.5 hours indicating that no significant surface passivation occurs. These CFC experiments are in agreement with the preliminary dissolution work as mentioned above.

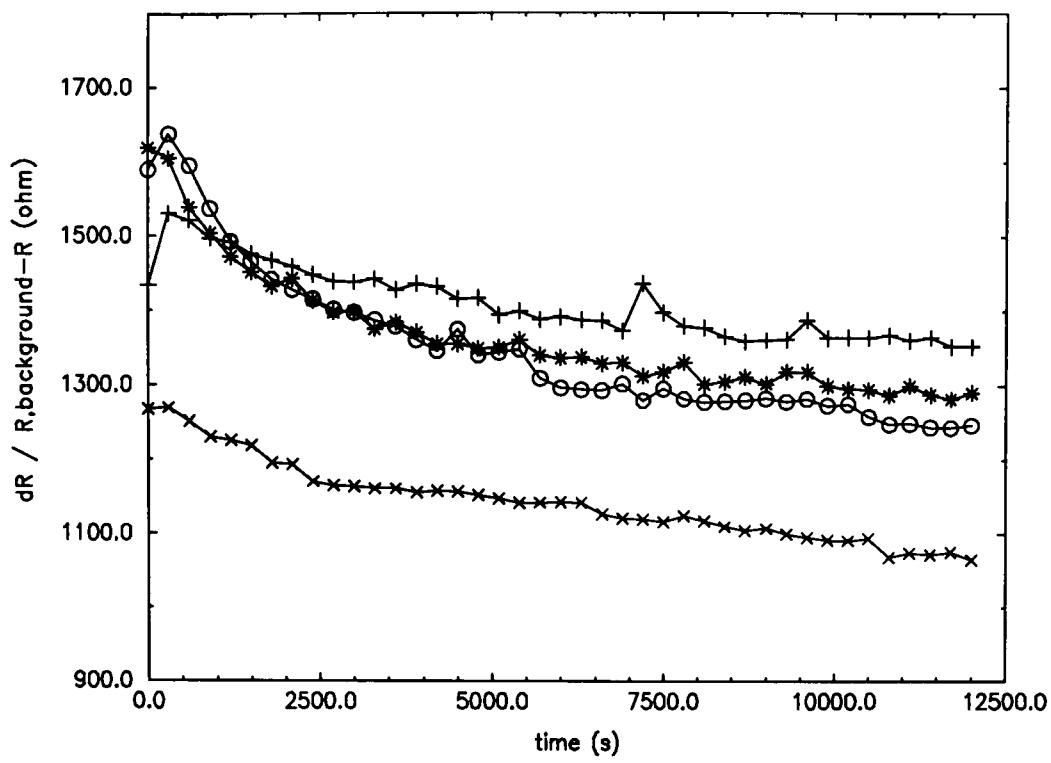


Fig. 20 Plots of dR (defined in chapter 4) as a function of time for the hydrolytic dissolution of a pressed pellet (47-445 μm particles) into 1.51 mM KNO_3 at a flow rate of 1.23 x 10^{-2} cm^3 s^{-1}. (o) [TPMOH] = 1.46 x 10^{-5} M, x_p = 7.46 mm, x_g = 9.34 mm; (+) [TPMOH] = 3.61 x 10^{-5} M, x_p = 7.79 mm, x_g = 9.44 mm; (x) [TPMOH] = 0, x_p = 7.11 mm, x_g = 9.24 mm; (*) [TPMOH] = 3.15 x 10^{-5} M, x_p = 7.79 mm, x_g = 9.28 mm. Cell geometry: 2h = 0.094 cm, d = 0.608 cm, w = 0.394 cm and x_d = 0.396 cm.

Next the surface of fused and pressed pellets after 2 hours dissolution in the CFC were examined using SEM. Typical images are shown in figs. 21 and 22. Comparison of these with figs. 12 and 13 reveals that a porous overlayer composed of small crystalline solids has become deposited on the pellet surface. We speculate that this is solid TPMOH on account of the estimated low solubility of this material[7] (ca. 10^{-6} M) but note that the similar sized crystals nucleated together. TPMOH is known to self-associate by forming H-bond tetramers which pack in a rather loosely crystalline state.[26,27] The overgrowth generated is rather porous since a negligible effect on the dissolution kinetics is observed. Subsequently, the (010) plane of a single TPMCl crystal was imaged in real time under water using atomic force microscopy to scrutinise the development of the porous overlayer. In situ micrographs are depicted in fig. 23. Before reaction, the surface was imaged in air (fig. 23a). The presence of

Fig. 21 SEM micrograph of fused pellet after 2 hours dissolution in a flow cell (flow rate 4 x 10^{-2} cm^3 s^{-1}). The dotted bar represents a distance of 30 μm.

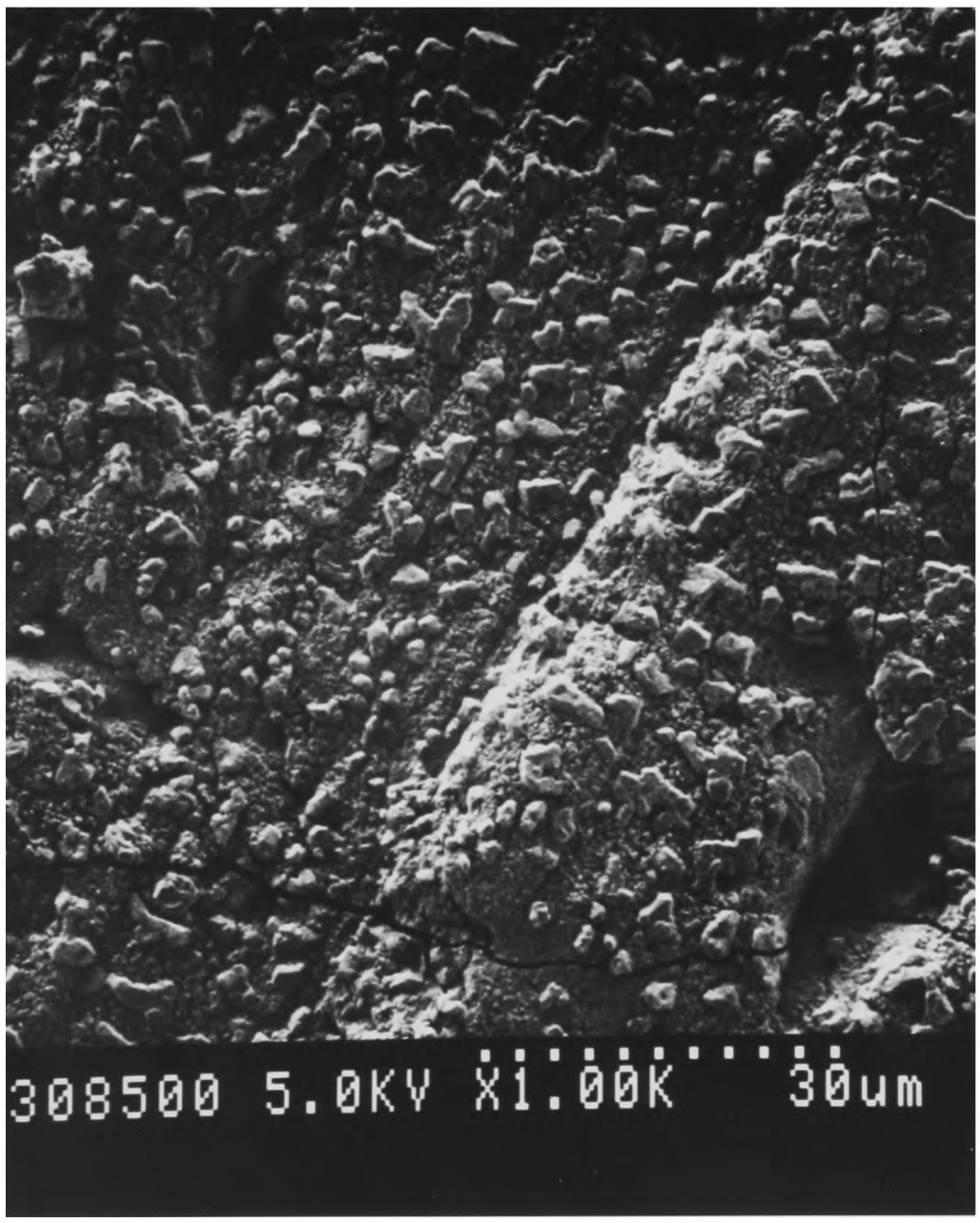


Fig. 22 SEM micrograph of pressed pellet (36-738 μm) after 2 hours dissolution in a flow cell (flow rate 4 x 10^{-2} cm^3 s^{-1}). The dotted bar represents a distance of 30 μm.

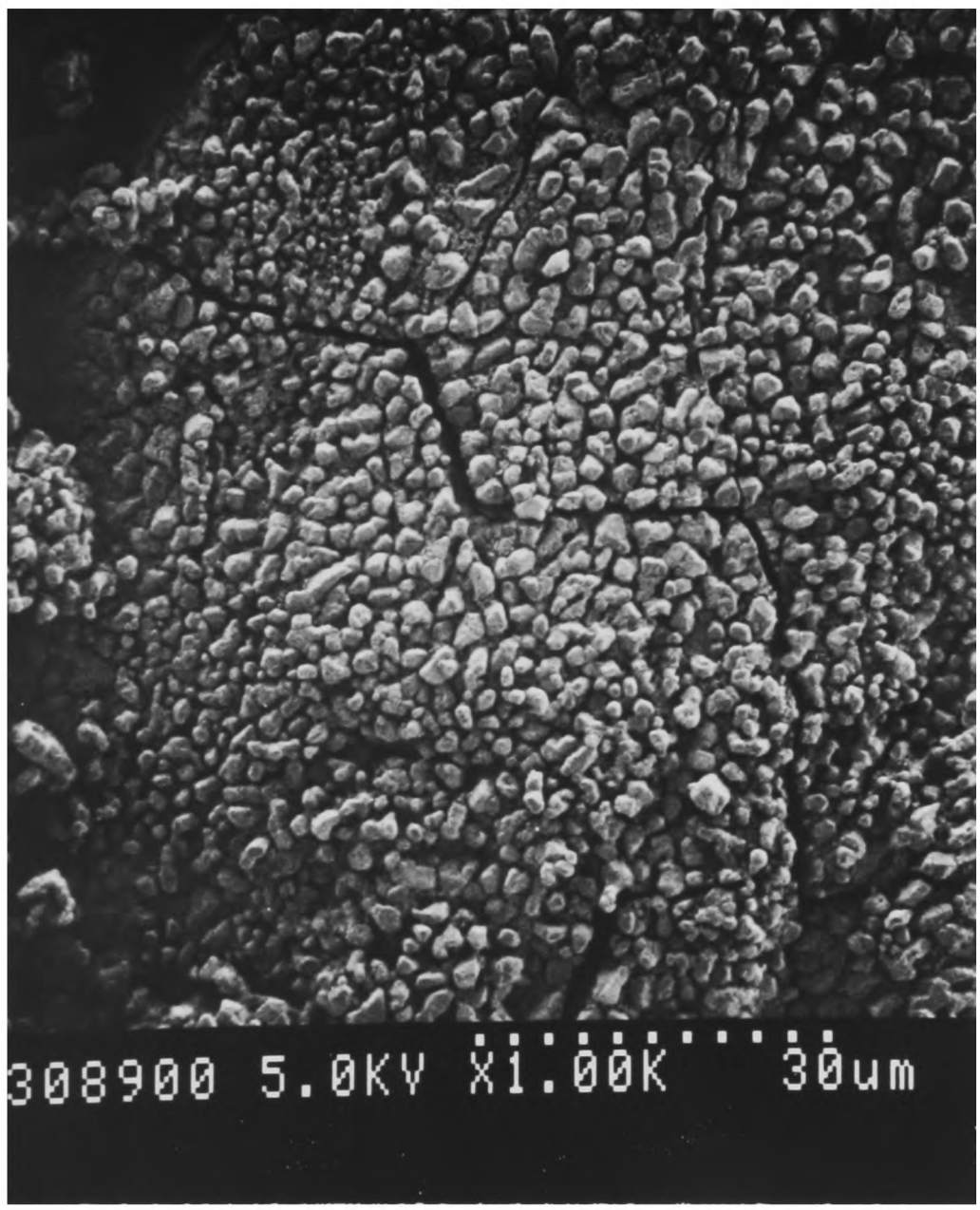
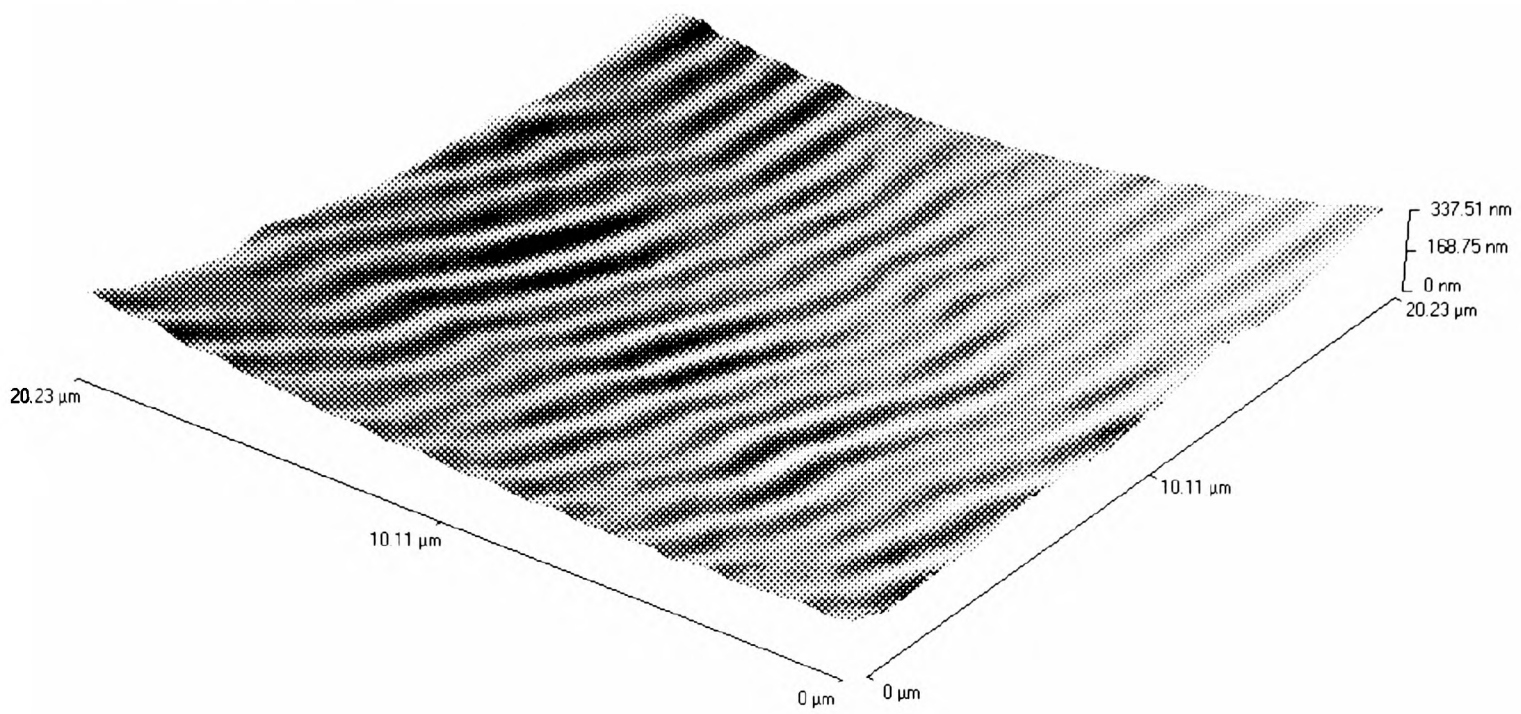
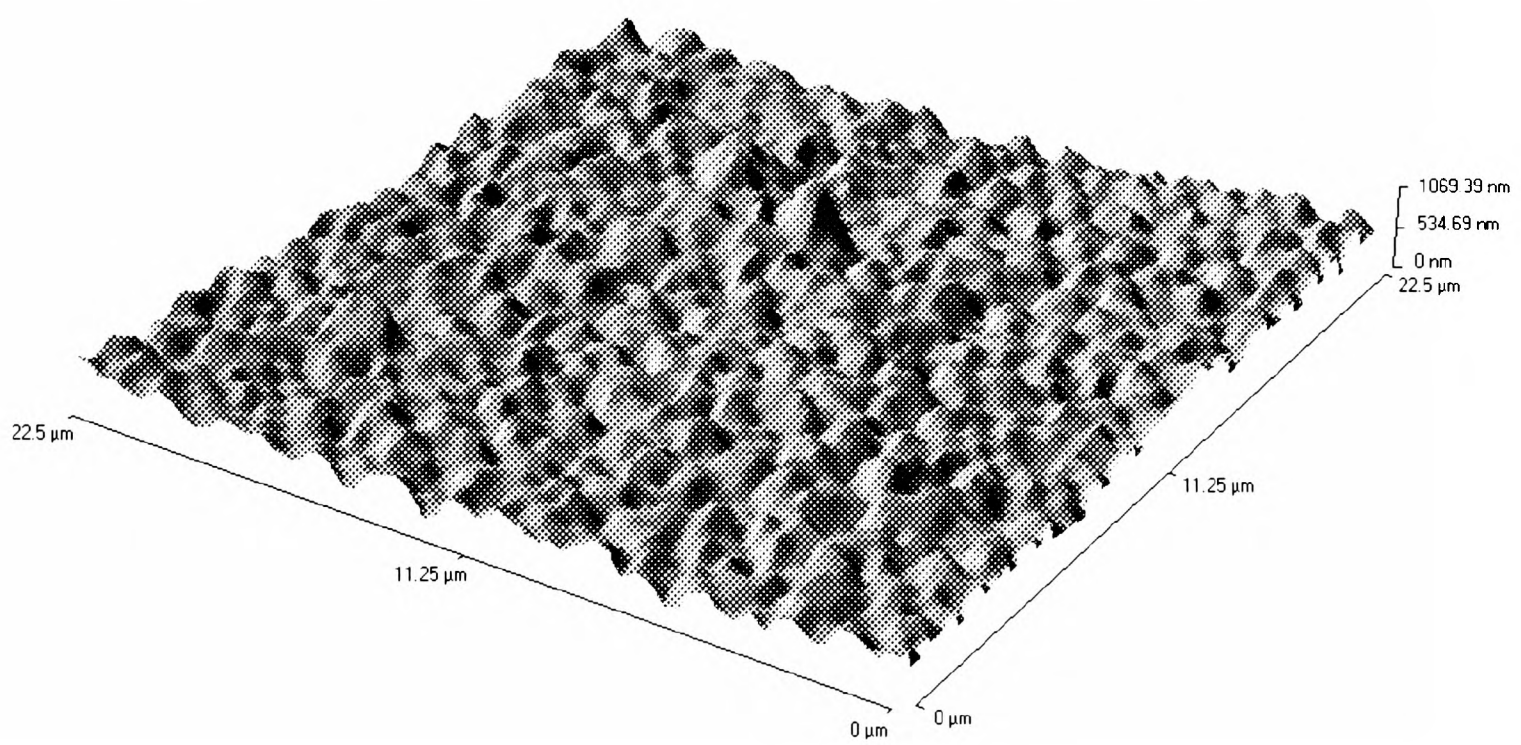


Fig. 23 In situ AFM micrograph of the (010) plane showing the evolution of an overgrowth layer.

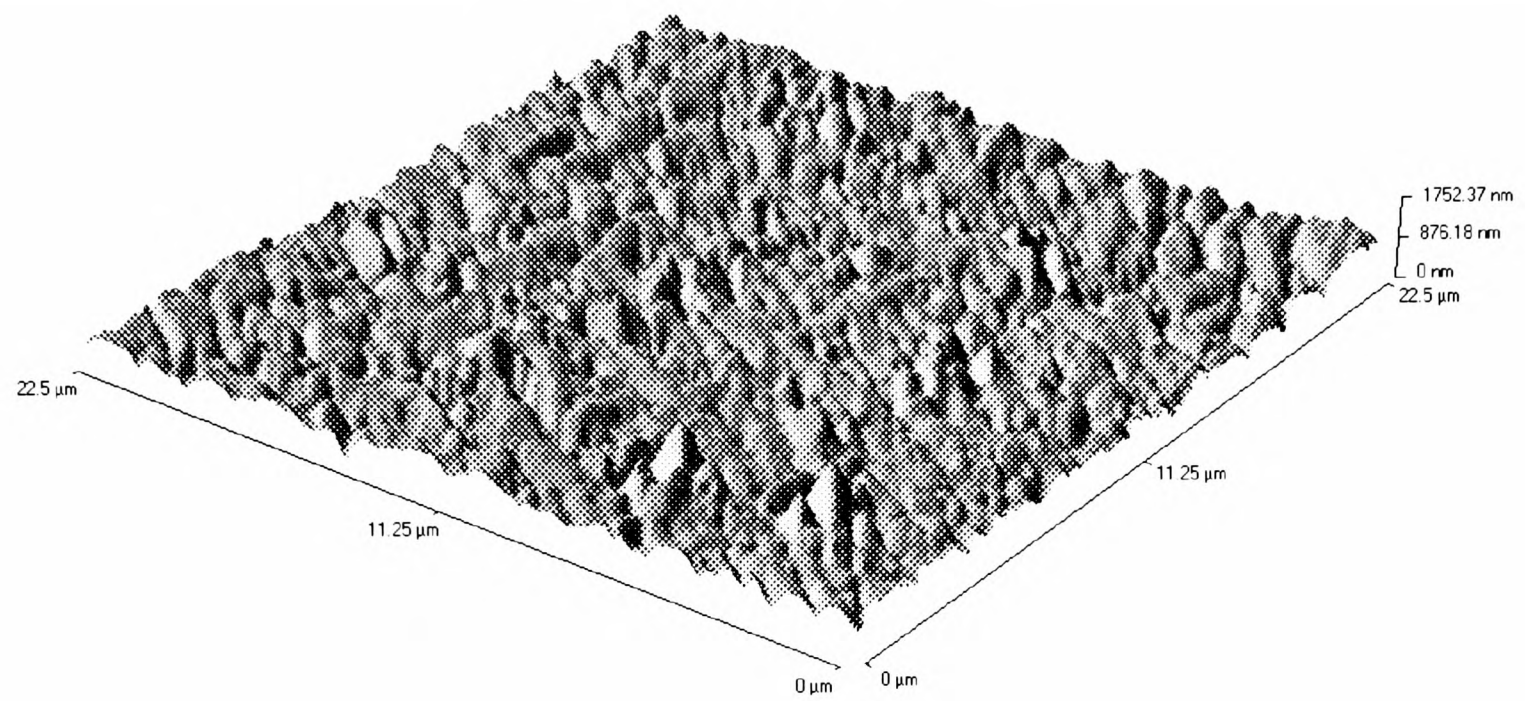
(a) taken in air



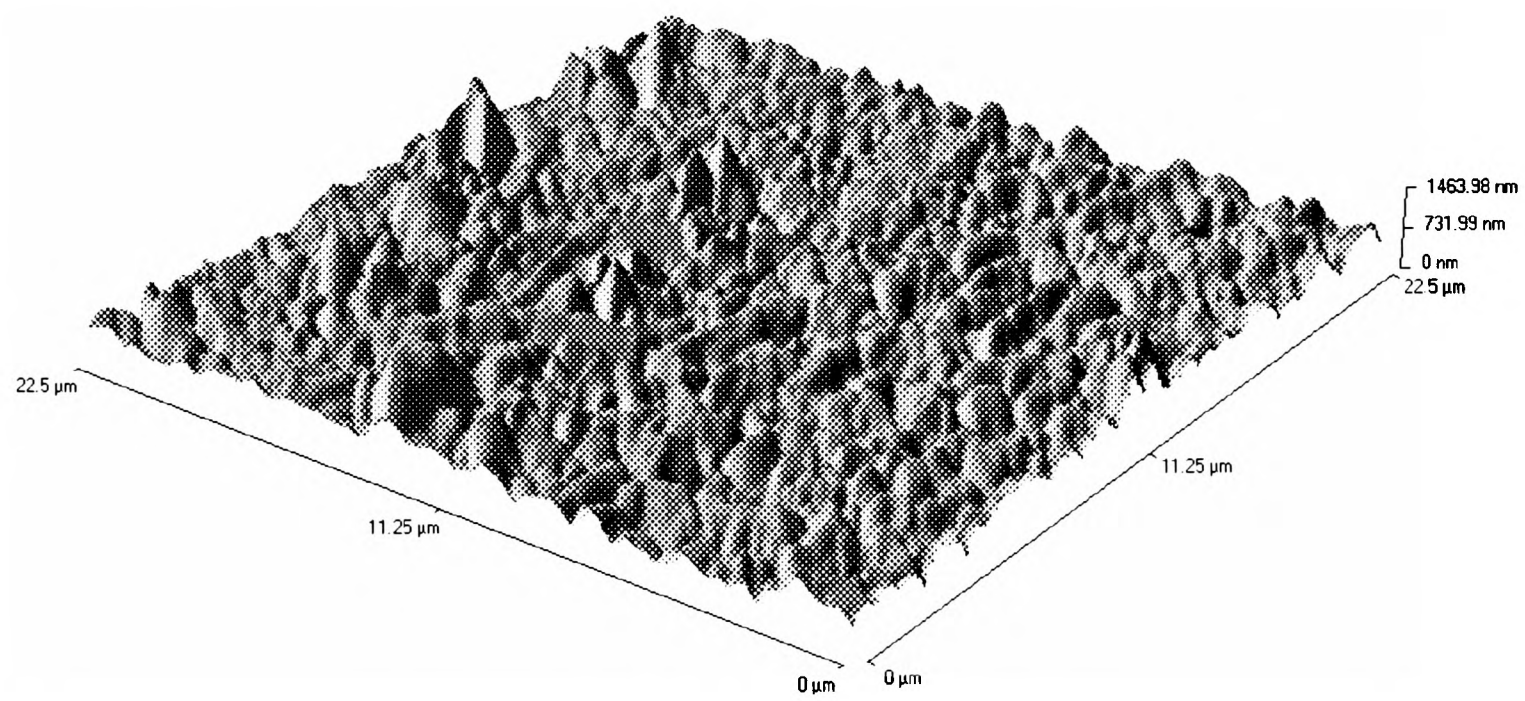
(b) recorded under initially pure water after 2 minutes of reaction



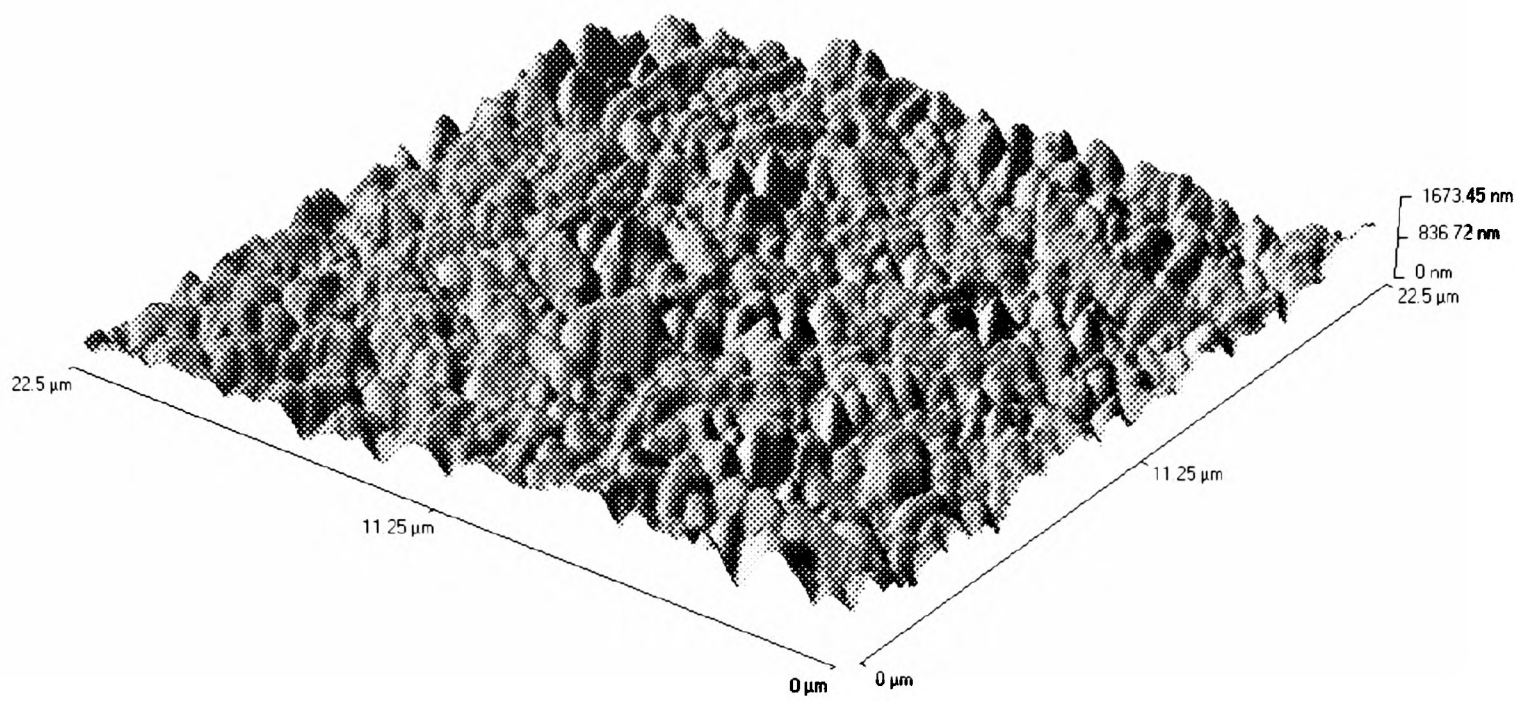
(c) recorded under initially pure water after 10 minutes of reaction



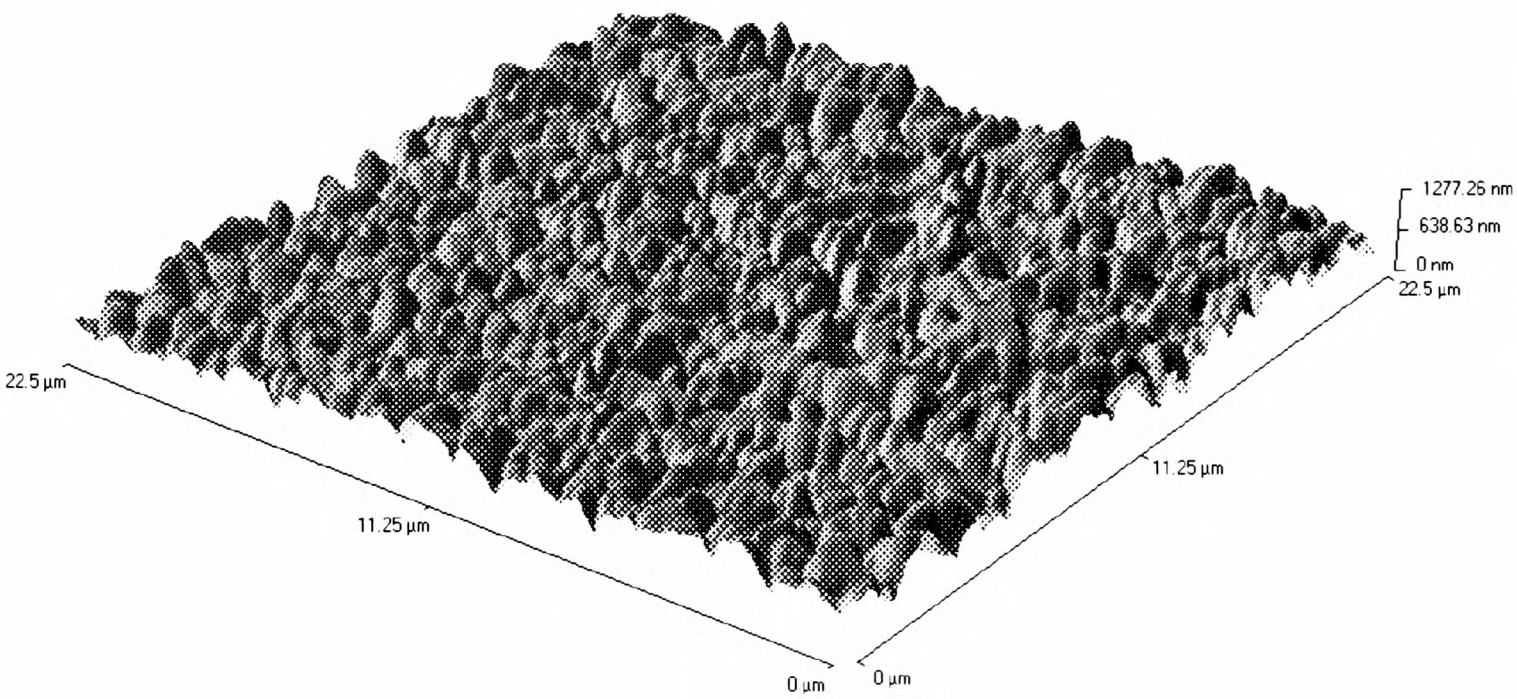
(d) recorded under initially pure water after 25 minutes of reaction



(e) recorded under initially pure water after 40 minutes of reaction



(f) recorded under initially pure water after 58 minutes of reaction



terraces with ledge heights ranging from around 2 nm to 30 nm are observed. Once the surface is exposed to water a porous layer forms rapidly: fig. 23b shows a surface after 2 minutes exposure. Figs. 23c-f, indicate that the layer remains almost unchanged between 10 and 58 minutes exposure. If the developed porous overlayer is contrasted with the fresh surface (fig. 23a), the root-mean-square (RMS) roughness increases from 57 - 80 nm to 180 - 240 nm which corresponds to a three to four times increase. The RMS roughness of the developed overgrowth may be qualitatively correlated to the thickness of the porous layer.

4. Conclusions

The hydrolytic reaction between solid triphenylmethyl chloride and water forming TPMCl takes place at the solid-liquid interface rather than in bulk solution following the prior dissolution of TPMCl. Individual crystal planes appear to react at a rate controlled by the "availability" of exposed chlorine atoms in the reacting surface and kinetic salt effects are consistent with a transition state in which there is essentially full dissociation into the ions TPM^+ and Cl^-. Microscopic studies suggest that a porous overlayer of TPMOH is rapidly formed on the solid surface but contributes a negligible resistance to the hydrolysis process.

5. References

1. Hughes E.D.; Ingold C.K.; Mok S.F.; Patai S.; Pocker Y. *J. Chem. Soc.*, **1957**, 1265.

2. Golomb D. *J. Chem. Soc.*, **1959**, 1327.

3. Swain C.G.; MacLachlan A. *J. Am. Chem. Soc.*, **1960**, *82*, 6095.

4. Ingold C.K. *Structure and Mechanism in Organic Chemistry*, 2nd Ed.; Cornell University Press; New York, **1969**.

5. Silhanek J.; Kondradova L.; Simeckova O.; Horak J. *Coll. Czech. Chem. Commun.*, **1982**, *47*, 2904.

6. Atherton J.H. *Research in Chemical Kinetics*, Vol 2; Compton R.G. (ed.); Elsevier; Amsterdam, **1994**.

7. Hansch C.; Quinlan J.E.; Lawrence G.L. *J. Org. Chem.*, **1968**, *33*, 347.

8. Oberg E.; Jones F.D. *Machinery's Handbook*, 18th ed.; Industrial Press; New York, **1970**.

9. C. Orr, *Kirk-Othmer Encyclopedia of Chemical Technology*, Vol 21; 3rd Ed.; Mark H.F.; Othmer D.T.; Overberger C.G.; Seaborg G.T. (eds.); John Wiley & Sons; New York, **1983**.

10. Kahr B.; Carter R.L. *Mol. Cryst. Liq. Cryst.*, **1992**, *219*, 79.

11. Brice J.C. *The Growth of Crystals from Liquids*, North-Holland Pub. Co.; Amsterdam, **1973**.

12. Brice J.C. *Crystal Growth Processes*, Blackie & Son Ltd.; Glasgow, **1986**.

13. Simon B.; Boistelle R. *J. Cryst. Growth*, **1981**, *52*, 779.

14. Tarjan I.; Matrai M. *Laboratory Manual on Crystal Growth*, Akademiai Kiado; Budapest, **1972**.

15. Tam K.Y.; Compton R.G. *J. Appl. Cryst.*, **1995**, *28*, 640.

16. Ives, D.J.G.; Janz, G.J. *Reference Electrodes*, Academic Press; New York, **1961**.

17. Tam K.Y.; Larsen, J.P.; Coles, B.A.; Compton, R.G., *J. Electroanal. Chem.*, **1996**, *407*, 23.

18. Turq P.; Lantela F.; Chemla M. *Electrochim. Acta*, **1969**, *14*, 1081.

19. Compton R.G.; Pilkington M.B.G.; Stearn G.M. *J. Chem. Soc. Faraday Trans. 1*, **1988**, *84*, 2155

20. Fisher, A.C.; Compton, R.G. *J. Phys. Chem.*, **1991**, *95*, 7538

21. *NAG Fortran Library Manual - Mark 13*, Vol 4; Numerical Algorithm Group Ltd.; Oxford, **1988**.

22. Compton R.G.; Harding M.S.; Atherton J.H.; Brennan C.M. *J. Phys. Chem.*, **1993**, *97*, 4677.

23. Compton R.G.; Harding M.S.; Pluck M.R.; Atherton J.H.; Brennan C.M. *J. Phys. Chem.*, **1993**, *97*, 10416.

24. Gordon J.E. *The Organic Chemistry of Electrolyte Solutions*, John Wiley & Sons; New York, **1975**.

25. Atkins P.W. Physical Chemistry, Oxford University Press; Oxford, **1986**.

26. Weber E.; Skobridis K.; Goldberg I. *J. Chem. Soc. Chem. Commun.* **1989**, 1195.

27. Ferguson G.; Gallagher J.F.; Glidewell C.; Low J.N.; Scrimgeour S.N. *Acta Cryst.*, **1992**, *C48*, 1272.

Chapter 7

The Kinetics of Dimerisation of the Methyl Viologen Radical Cation

1. Introduction

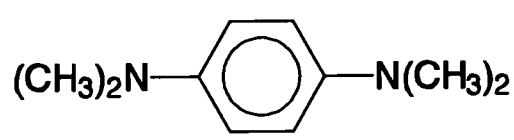
The use of the channel electrode is a sensitive and useful technique for the investigation of the mechanism of electrochemical reactions.[1,2] The cell consists of a rectangular duct in which solution flows over a working electrode under laminar flow conditions. The electrochemical oxidation or reduction process is carried out at the working electrode surface. The electrogenerated species has hitherto been scrutinised by using electron spin resonance[3] or fluorescence[4] spectroscopies with the appropriate detector located immediately downstream of the electrode. Since the working electrode is linked to the detector system via a well defined hydrodynamic regime, the detector response as a function of mass transport can be quantitatively evaluated and compared with experimental values. In this manner, a diversity of electrochemical reaction mechanisms have been resolved.[5,6]

UV-Visible (UV-Vis) spectroscopy provides a simple and effective way to quantify the light absorbing electrogenerated species. Although conventional optically transparent electrodes (OTEs) are available for in situ spectroscopic studies, the transmittance of incident light is generally low - ca. 20 to 80%[7] and may vary over the spectral range of interest. Moreover, the hydrodynamic regime within OTEs is usually ill-defined, rendering the modelling of diffusion and convection processes uncertain. A channel flow cell specifically devised for spectroelectrochemical work has previously been described.[8,9] However, this design was

largely confined to a tailor-made spectrometer and a simplified linear fluid velocity profile was implemented to interpret results instead of the full parabolic one which is known to be preferable in complicated multi-species electrokinetics modelling.[1] This chapter report a spectroelectrochemical channel cell (SCC), which is compatible with commercially available UV-Visible spectrophotometers, for the quantitative monitoring of the light absorbing species generated at a working electrode surface.[10,11] In contrast to conventional OTEs, the SCC enables over 80% transmission of light which should allow high sensitivity with respect to the transient electrogenerated species. Moreover, the full parabolic fluid velocity profile is pursued in our calculations and this permits precise modelling of any homogeneous and/or heterogeneous process within the cell. The proposed method is exemplified by two model systems, namely the oxidation of N,N,N′,N′-tetramethyl-p-phenylenediamine (TMPD) and the reduction of methyl viologen dichloride ($MVCl_2$).

The photochemistry and electrochemistry of TMPD has received considerable attention.[12-17] TMPD is often used as a one-electron reducing agent in the investigation of biologically important compounds,[13,14] and its one electron oxidation in aqueous solution is known to produce the Wurster's Blue radical cation (B) which has absorption maxima at 565 nm and 615 nm.[15]

$$\underset{A}{TMPD} \rightarrow \underset{B}{TMPD^{+\bullet}} + e^{-} \qquad (1)$$

TMPD = 

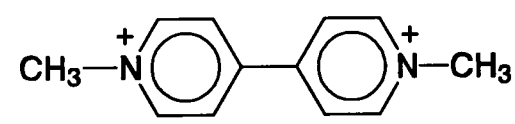
In the following, this simple electro-oxidation will be used to verify the cell hydrodynamics before attention is turned to the kinetically complex $MVCl_2$ system.

The $MVCl_2$ system has been a "hot" research area within the past decade. The $MV^{+\cdot}/MV^{2+}$ couple is frequently invoked in electron transfer investigations and photosensitised processes[18,19] and as an electron mediator for biological systems[20,21] as well as in modified electrodes.[22-24] Moreover, the radical cations have been proposed as electron relays for solar energy conversion.[25-27] To date, the mechanism of the action of MV^{2+} as a herbicide and a human toxin remains unclear although there is evidence to suggest that the toxic pathway probably involves $MV^{+\cdot}$ radical chemistry.[28-30]

In this work the one electron reduction of the methyl viologen dication in aqueous solution is studied. The radical cation is believed to undergo a rapid dimerisation process:[31-35]

$$\underset{C}{MV^{2+}} + e^- \rightarrow \underset{E}{MV^{+\bullet}} \qquad (2)$$

$$2\ \underset{E}{MV^{+\bullet}} \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \underset{F}{(MV^{+\bullet})_2} \qquad (3)$$

MV^{2+} = 

Recently, this association has been confirmed spectroscopically[36,37] and electrochemically.[38] Factor analysis statistically proved that there are two light absorbing components within the equilibrium mixture.[39] In particular, the process (3) has been examined[40] using UV-Vis spectroscopy. An equilibrium expression was proposed for the dimerisation:

$$K_{Dimer} = \frac{k_f}{k_b} = 720\ M^{-1} \quad (4)$$

In the reported work[40], the spectra of $MV^{+\cdot}/(MV^{+\cdot})_2$ mixtures, after background substraction, were deconvoluted by assuming the absorption spectra to be of Gaussian shape. To this end, the absorption maxima at 675.1 nm, 604.4 nm and 395.7 nm were assigned to the monomer (E) while 537.1 nm and 369.1 nm were designated to the dimer (F). Note that the $MV^{+\cdot}$ concentrations also can be reduced by disproportionation. However, it has been estimated that this equilibrium constant is ca. 3.3×10^{-6} which implies that disproportionation is negligible in comparison with the dimerisation process (see eqn. 4).[41]

Interestingly, little, if any, kinetic data for the formation of the dimer (F) has been reported. In the application of $MV^{+\cdot}$ as an electron mediator, if the electron transfer process is relatively sluggish, such as would result from an outer sphere exchange reaction[42], the formation of the dimer may become an undesirable side reaction, especially with an high initial concentration of MV^{2+} (> 3 mM) as described later. An understanding of the dimerisation rate could help to interpret the relevant electron transfer process. In the present study, the suggested spectroelectrochemical channel approach is utilised to scrutinise the above reaction in order to find out whether it is possible to resolve the rapid dimerisation kinetics. The formation rate constants for the dimer and the spectral parameters of the radical species at the absorption maxima are reported.

2. Experimental

A schematic diagram of the channel cell is depicted in fig. 1. The cell is composed of a rectangular duct (about 4.5 cm long, 0.1 cm deep and 0.6 cm wide) cut in a Black Delrin block (Goodfellow Advanced Materials, Cambridge) and closed by a cover plate of the same material. Solution was flowed through the channel via inlet and outlet teflon tubes (not shown in fig. 1) which were attached to the ends of the cell along the y-direction. Spectroscopic access is accomplished by using a pair of high quality fused silica windows (U.Q.G. Ltd., Cambridge, 4 mm diameter, 3 mm thickness) to fit into holes of diameter 4 mm, pre-drilled in the channel unit and cover plate (see fig. 1). The position of the silica window was aligned with the incident beam of the spectrophotometer for maximum sensitivity. The platinum electrode (Goodfellow Advanced Materials, thickness 0.1 mm, purity 99.95%) with dimensions of approximately 4 mm x 4 mm is made flush to the cover plate surface and is located about 1 mm upstream of the silica window. Contact to the electrode was made through a hole in the cover plate. Black silicone rubber (Silastic® 732 RTV, Dow Corning Corp., Michigan) was used as a gasket between the cover plate and the channel unit. The channel unit and cover plate were fastened together by a set of brass clamps compressed along the y-direction. Thermostatted water at 25 °C was circulated through two short brass pipes (along the z-direction, see fig. 1) which were soldered to the brass clamps. The dimensions of the assembled cell is about 12.4 x 12.4 mm which allows it to mount nicely into any conventional spectrophotometer cell holder.

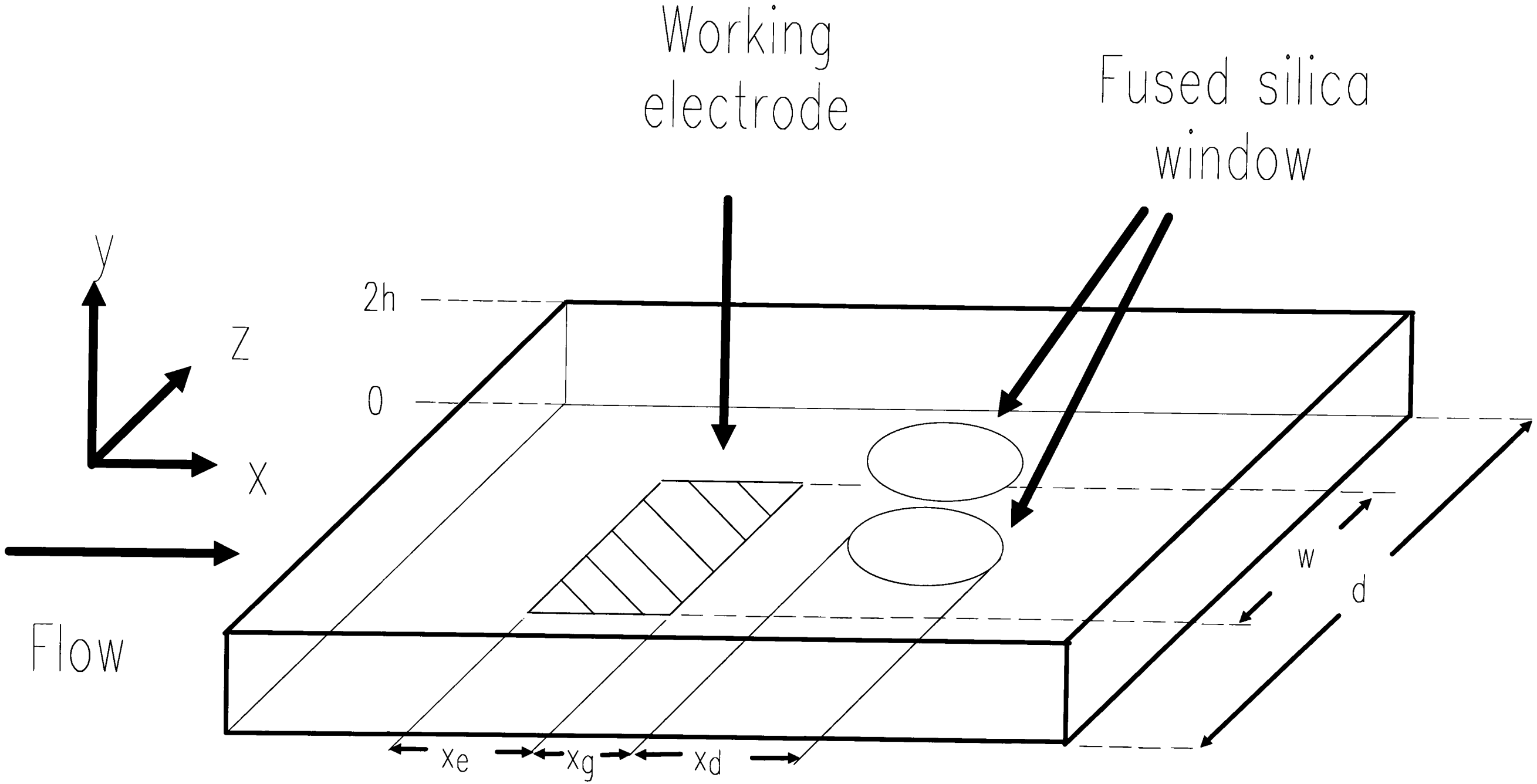


Fig. 1 Schematic diagram of the spectroelectrochemical channel cell.

Typical values for the cell geometry were: x_e = 0.394 cm, x_g = 0.103 cm, x_d = 0.407 cm, w = 0.402 cm and d = 0.601 cm (see fig. 1). The cell depth (2h) was determined spectroscopically via measurements on a series of known concentrations of potassium ferricyanide solution. The extinction coefficient of ferricyanide was found to be 1007 ($\pm$ 7) M^{-1} cm^{-1} (λ_{max} = 420 nm at 30 °C) in our laboratory. A typical value for the cell depth was 0.082 ($\pm$ 0.002) cm.

A saturated calomel electrode (SCE) was placed upstream of the cell as reference electrode while a platinum counter electrode was situated downstream of the cell. Both were positioned outside the spectrophotometer. Solution was flowed into the channel cell from a gravity feed reservoir located outside the spectrophotometer where degassing with nitrogen could take place. The flow velocity was adjusted by linking the solution exit to a pre-calibrated capillary tube. The electrochemical measurements were carried out using a scan generator and a potentiostat (Oxford Electrodes Ltd., UK) in conjunction with an Hewlett Packard 7035B X-Y recorder. A double beam scanning spectrophotometer (ATI Unicam, Cambridge, model UV2-100) was employed for all optical measurements.

3. Modelling of transport within the flow cell

The backwards implicit finite difference (BIFD) method for solving convective-diffusion equations within the channel system has been described in chapter 2. In the following descriptions, the same principle is adopted to formulate mathematical models for the two chemical systems. For the oxidation of TMPD (eqn. 1), the steady state convective-diffusion equation can be written as

$$D \frac{\partial^2[A]}{\partial y^2} - v_x \frac{\partial[A]}{\partial x} = 0 \quad (5)$$

$$D \frac{\partial^2[B]}{\partial y^2} - v_x \frac{\partial[B]}{\partial x} = 0 \quad (6)$$

where

$$v_x = \frac{3\ V_f}{4\ h\ d}\left[1 - \frac{(h - y)^2}{h^2}\right] \quad (7)$$

The symbol D represents the diffusion coefficient and v_x is the solution velocity axially through the channel cell. V_f denotes the solution flow rate in $cm^3\ s^{-1}$. Eqns. 5 and 6 can be readily be solved by dividing the electrode, gap and detector window regions into K (x-direction) x J (y-direction) boxes via the BIFD procedure as given in chapter 2 with the following equations as boundary conditions.

Before the solution reaches the electrode

$$[A] = A_0, \quad [B] = 0 \quad (8)$$

A_o is the bulk concentration of species A. At the electrode surface, assuming $D_A = D_B$

$$A = 0, \quad \frac{\partial[A]}{\partial y} = -\frac{\partial[B]}{\partial y} \quad (9)$$

where the matrix elements b_1 and d_1 are given as

$$\textit{Species A:} \quad b_1 = 2\lambda_1 + 1, \quad d_1 = g^{A}_{1,k} \quad (10)$$

$$\textit{Species B:} \quad b_1 = \lambda_1 + 1, \quad d_1 = g^{B}_{1,k} + \lambda_1\, g^{A}_{1,k+1} \quad (11)$$

At all other surfaces,

$$\frac{\partial [A]}{\partial y} = \frac{\partial [B]}{\partial y} = 0 \qquad (12)$$

with matrix elements b_1 and d_1

$$\textit{Species A:} \quad b_1 = \lambda_1 + 1, \quad d_1 = g^{A}_{1,k} \qquad (13)$$

$$\textit{Species B:} \quad b_1 = \lambda_1 + 1, \quad d_1 = g^{B}_{1,k} \qquad (14)$$

The absorptometric behaviour within the channel cell can be calculated by integrating the following relationship[43]

$$\frac{\partial I}{\partial y} = -\kappa\,[B]\,I \quad \Rightarrow \quad \int_{I}^{I_0} \frac{dI}{I} = \kappa \int_{0}^{2h} [B]\,dy \qquad (15)$$

where I_0 and I represent, respectively, the incident and emergent light intensity at each point in the flow cell and κ is a proportionality coefficient. This is readily accomplished once the concentration profile, [B] (x,y), is generated using the BIFD method as mentioned above. The absorbance at a particular solution flow rate and wavelength (λ) over the silica window region can be calculated by

$$Abs_{\lambda}\,(V_f) = \frac{\varepsilon_{\lambda,B} \int_{x_1}^{x_2} W(x) \int_{0}^{2h} [B]\,(x,y)\,dy\,dx}{\int_{x_1}^{x_2} W(x)\,dx} \qquad (16)$$

$\varepsilon_{\lambda,B}$ represents the extinction coefficient of species B while x_1 and x_2 indicate, respectively, the x-coordinate at the beginning and end of the silica window. The integral of eqn. 16 can be evaluated by using the trapezoidal method.[44] W(x) denotes a weighting function to account

for the circular geometry of the silica window which is defined as

$$W(x) = \sqrt{(x_d/2)^2 - (x - x_m)^2} \tag{17}$$

where

$$x_m = \frac{x_1 + x_2}{2}, \quad x_d = x_2 - x_1 \tag{18}$$

As for the first reduction of methyl viologen dication (eqns. 2 & 3), the steady state mass transport equations can be expressed as

$$D \frac{\partial^2[C]}{\partial y^2} - v_x \frac{\partial[C]}{\partial x} = 0 \tag{19}$$

$$D \frac{\partial^2[E]}{\partial y^2} - v_x \frac{\partial[E]}{\partial x} - 2 k_f [E]^2 + 2 k_b [F] = 0 \tag{20}$$

$$D \frac{\partial^2[F]}{\partial y^2} - v_x \frac{\partial[F]}{\partial x} + k_f [E]^2 - k_b [F] = 0 \tag{21}$$

The boundary conditions which pertain are as follows:

Before the solution reaches the electrode

$$[C] = C_0, \quad [E] = 0, \quad [F] = 0 \tag{22}$$

C_o is the bulk concentration of species C. At the electrode surface, assuming $D_C = D_E = D_F$

$$C = 0, \quad \frac{\partial[C]}{\partial y} = -\frac{\partial[E]}{\partial y}, \quad \frac{\partial[F]}{\partial y} = 0 \tag{23}$$

the matrix elements b_1 and d_1 are

$$\textit{Species C:} \quad b_1 = 2\lambda_1 + 1, \quad d_1 = g^{C}_{1,k} \tag{24}$$

$$\textit{Species E:}\quad b_1 = \lambda_1 + 1,\quad d_1 = g^{E}_{1,k} + \frac{2\,(\Delta y)^2\,\lambda_1}{D}\,(k_b\,g^{F}_{1,k} - k_f\,(g^{E}_{1,k})^2) + \lambda_1\,g^{C}_{1,k+1} \tag{25}$$

$$\textit{Species F:}\quad b_1 = \lambda_1 + 1,\quad d_1 = g^{F}_{1,k} + \frac{(\Delta y)^2\,\lambda_1}{D}\,(k_f\,(g^{E}_{1,k})^2 - k_b\,g^{F}_{1,k}) \tag{26}$$

At all other surfaces,

$$\frac{\partial [C]}{\partial y} = \frac{\partial [E]}{\partial y} = \frac{\partial [F]}{\partial y} = 0 \tag{27}$$

with the matrix elements b_1 and d_1

$$\textit{Species C:}\quad b_1 = \lambda_1 + 1,\quad d_1 = g^{C}_{1,k} \tag{28}$$

$$\textit{Species E:}\quad b_1 = \lambda_1 + 1,\quad d_1 = g^{E}_{1,k} + \frac{2\,(\Delta y)^2\,\lambda_1}{D}\,(k_b\,g^{F}_{1,k} - k_f\,(g^{E}_{1,k})^2) \tag{29}$$

$$\textit{Species F:}\quad b_1 = \lambda_1 + 1,\quad d_1 = g^{F}_{1,k} + \frac{(\Delta y)^2\,\lambda_1}{D}\,(k_f\,(g^{E}_{1,k})^2 - k_b\,g^{F}_{1,k}) \tag{30}$$

It can be seen that the equations (25, 26, 29, 30) for solving species E and F are interdependent, i.e. one cannot calculate E without the knowledge of F. An iterative method was adopted, whereby, $g^{F}_{j,k}$ was approximated by $g^{F}_{j,k-1}$ to evaluate $g^{E}_{j,k}$. This was used to obtain a better value for $g^{F}_{j,k}$, and the procedure was continued until the $g^{E}_{j,k}$ and $g^{F}_{j,k}$ values were converged to better than 1%. Then, the absorbance of the equilibrium mixture at a particular solution flow rate and wavelength (λ) is calculated as

$$Abs_{\lambda}\,(V_f) = \frac{\varepsilon_{\lambda,E}\int_{x_1}^{x_2} W(x)\int_{0}^{2h}[E]\,(x,y)\,dy\,dx + \varepsilon_{\lambda,F}\int_{x_1}^{x_2} W(x)\int_{0}^{2h}[F]\,(x,y)\,dy\,dx}{\int_{x_1}^{x_2} W(x)\,dx} \tag{31}$$

$\varepsilon_{\lambda,E}$ and $\varepsilon_{\lambda,F}$ denote, respectively, the extinction coefficients of species E and F.

In all BIFD computations, a grid size of 1000 (x-direction) x 1000 (y-direction) was required for the TMPD system while values of 30000 to 150000 x 500 for $MVCl_2$ system was needed to obtain convergence (within 2%). All supporting programs were coded in an UNIX C environment and executed on a SUN Sparc workstation. As shown in the previous derivation, the theoretical absorbance is a function of solution flow rate, extinction coefficients and rate constants. We define an error function, $RMSD_\lambda$ (root-mean-square-deviation), to indicate the quality of fit between theory and experiment

$$RMSD_\lambda = \sqrt{\frac{\sum_{i=1}^{N} [Abs_\lambda(i)^{theory} - Abs_\lambda(i)^{expt}]^2}{N}} \qquad (32)$$

where N is the number of data points. The sought parameters, such as the rate constant and the extinction coefficients, can be derived by minimising the deviation between the theoretical and experimental data in a least square sense. The BFGS (Broyden-Fletcher-Goldfard-Shanno) method coupled with Powell's quadratic interpolation linear search technique[45,46] will be utilised for this purpose.

It should be pointed out that when using the aforementioned BIFD technique to model kinetic processes involving large rate constant(s), the cost, in terms of computational time need to derive all the parameters within a single calculation, is quite expensive. Here, a two-step optimisation procedure is proposed to deduce the kinetic and spectral parameters for the methyl viologen system. Specifically, the concentration-flow rate profiles of species E and F are first calculated using a systematic range of rate constants. For each set of profiles, the

BFGS method is applied to minimise the corresponding RMSD values (eqn. 32) for the optimal spectral parameters. Then, the kinetic and spectral parameters with the minimal RMSD value are chosen as the answer.

4. Results and discussion

TMPD is known to be oxidised to the radical cation at potential ca. -0.02 V vs SCE.[47,48] Preliminary experiments were performed using 2.18 mM TMPD in 0.20 M KCl solution to generate the absorption spectrum of $TMPD^{+\cdot}$ to see whether it agreed with the literature reports.[49] The solution was flowed into the SCC. Then, flow was stopped and the potential of the working electrode was scanned from -0.2 V to 0.2 V at 10 mV s^{-1} to electrochemically generate the $TMPD^{+\cdot}$ radical. Several minutes were allowed for the radical to diffuse to the silica window region and the absorption spectrum was then recorded.

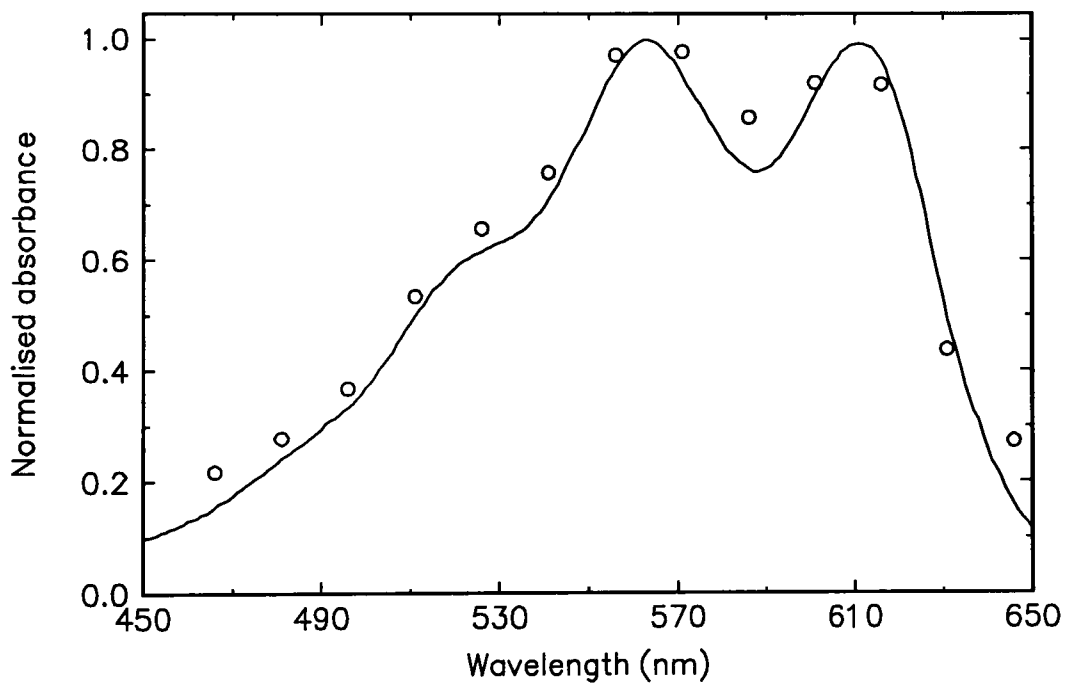


Fig. 2 Normalised absorption spectrum of $TMPD^{+\cdot}$: (——) obtained from the SCC and (o) from literature.[49]

As shown in fig. 2, good agreement can be seen between the normalised spectrum obtained in the channel cell and the reported one.[49]

After confirming the SCC method is capable of producing and monitoring the radical species, we varied the solution flow rate whilst monitoring the electrode potential as before to correspond to the transport limited oxidation of TMPD. For each scan, the potential was held at 0.2 V and then the steady state absorbance at 611.5 nm was recorded and depicted in fig. 3. A half-wave potential of -0.01 V (vs SCE) was observed and the limiting currents obeyed the Levich equation[50] for a one electron oxidation. A diffusion coefficient of 6.37 ($\pm$ 0.31) x 10^{-6} cm^2 s^{-1} was derived from the slope of the Levich plot as illustrated in fig. 4 and found to be consistent with values previously reported in the literature.[47,48] As shown in fig. 3, the good agreement between theory and experiment justifies the validity of our modelling and experimental approaches. The extinction coefficient of $TMPD^{+\cdot}$ radical at 611.5 nm is found to be 1.13 ($\pm$ 0.11) x 10^4 M^{-1} cm^{-1} which is also comparable to a literature value of 1.2 x 10^4 M^{-1} cm^{-1}.[51] Since $TMPD^{+\cdot}$ is relatively stable, the absorption spectra and extinction coefficients can be obtained by chemical generation of the radical.[49,51]

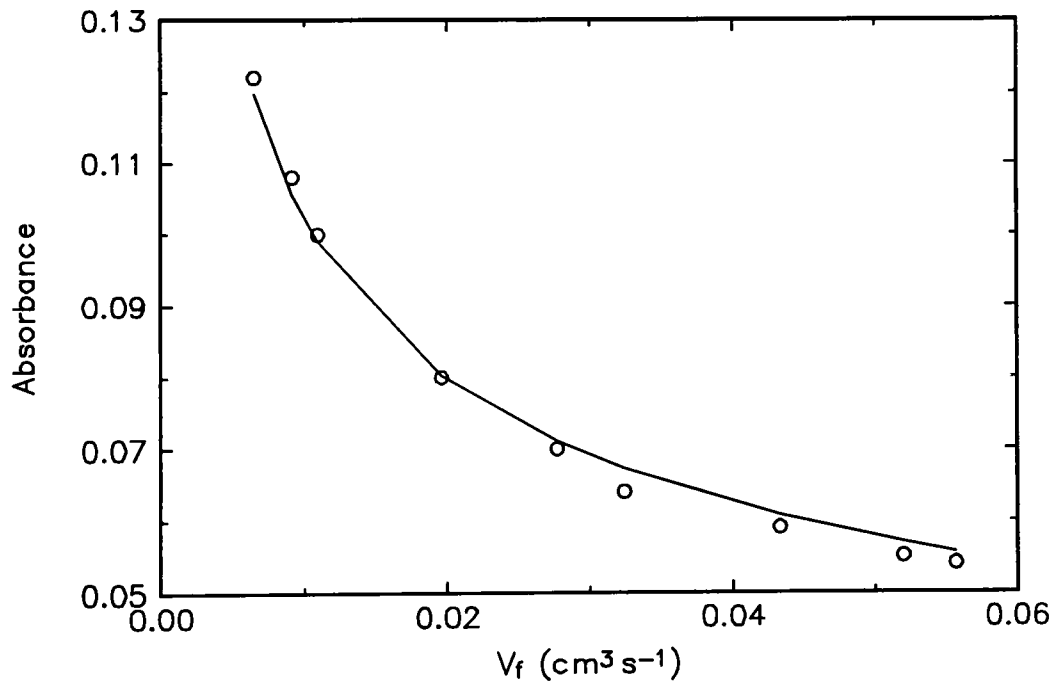


Fig. 3 The flow rate dependence of the absorbance at 611.5 nm for the $TPMD^{+\cdot}$ radical generated using the SCC in 0.20 M sodium chloride solution: (o) experimental data and (——) theoretical behaviour predicted using the model and extinction coefficient as mentioned in the text. Bulk concentration of TMPD is 2.18 mM and cell depth (2h) is 0.0819 cm.

Obviously, for short life time species that are difficult to generate chemically or spectroscopically, the present method is superior in terms of obtaining the spectral data of these transient species.

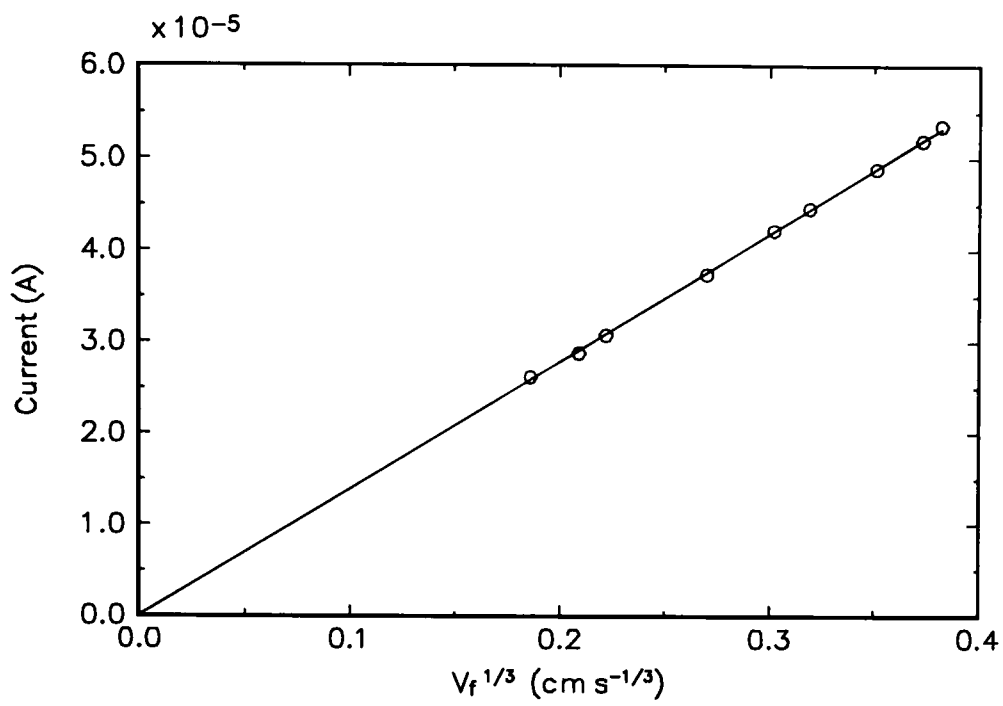


Fig. 4 Levich plot for the first oxidation wave of TMPD. The experimental conditions are the same as fig. 3.

Next, we turn to the one electron reduction of the methyl viologen dication. $MVCl_2$ solutions of 3.51, 4.98 and 7.72 mM in 0.50 M sodium chloride were used for SCC experiments. The potential was scanned from 0 to -0.78 V for the first reduction wave which showed a half-wave potential of -0.64 V (vs SCE). The limiting currents followed the Levich equation for a single electron reduction. Fig. 5 shows a typical Levich plot for 4.98 mM $MVCl_2$ from which a diffusion coefficient of 7.02 ($\pm$ 0.11) x 10^{-6} cm^2 s^{-1} was deduced. The steady state absorption spectra of the electrogenerated $MV^{+\cdot}$ and $(MV^{+\cdot})_2$ mixture from 4.98 mM $MVCl_2$ at different solution flow rates are given in fig. 6. The spectral features of the equilibrium mixture are qualitatively in agreement with Stargard & Hawkridge.[40] It can be seen that the absorbance value decreases as the solution flow rate increases. This trend is clearly demonstrated in fig. 7 for the absorbance at 395 nm and 366 nm. Note that the *peak*

absorbance at these wavelengths were purposely chosen to follow the kinetic process for maximum sensitivity. Similar flow rate dependencies of absorbance at 395 nm and 366 nm for bulk concentrations of 3.51 mM and 7.72 mM are shown, respectively, in figs. 8 and 9.

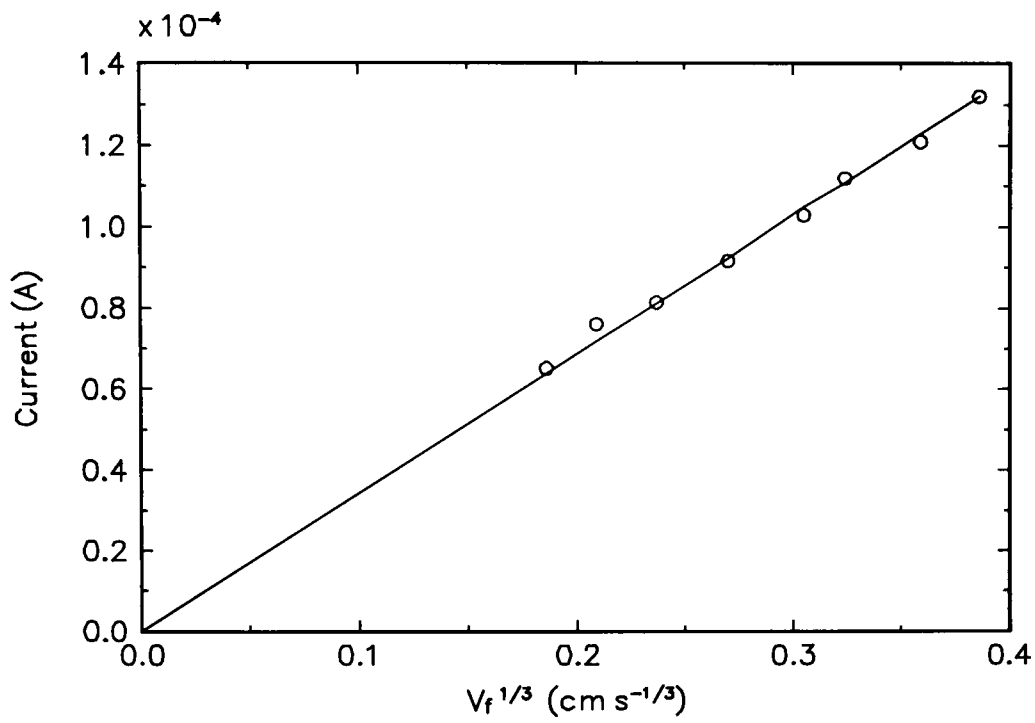


Fig. 5 Levich plot for the first reduction wave of 4.98 mM $MVCl_2$ in 0.50 M NaCl solution. The cell geometry is given in the text and the cell depth (2h) is 0.0831 cm.

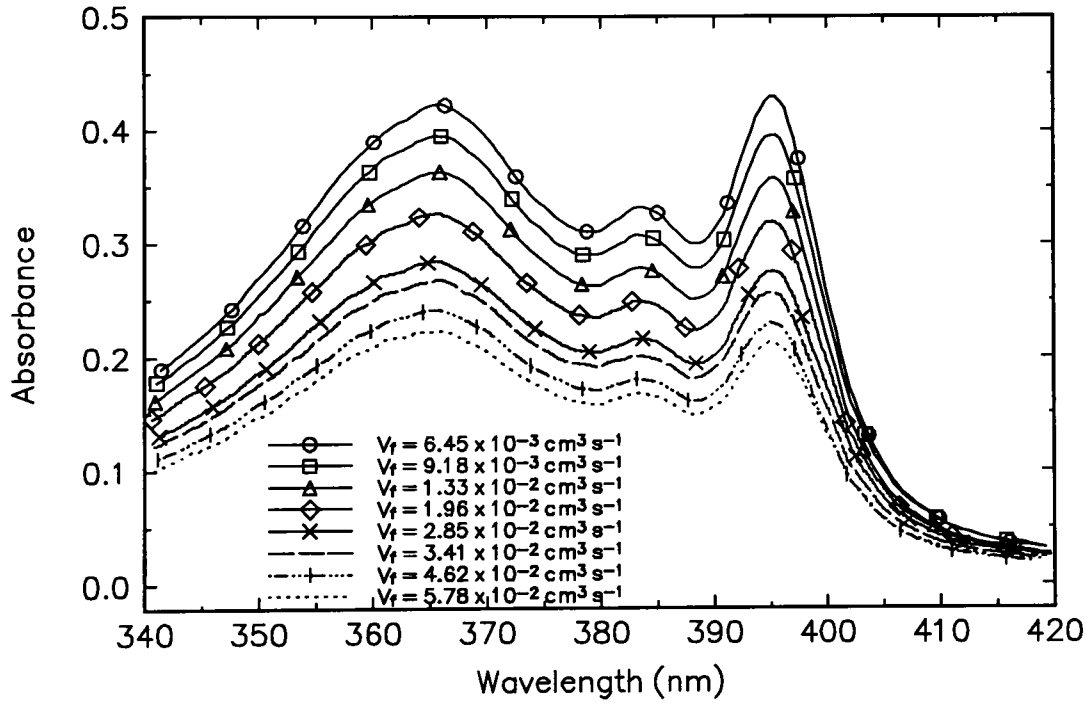


Fig. 6 Steady state absorption spectra of the electrogenerated $MV^{+\cdot}/(MV^{+\cdot})_2$ mixture (from 4.98 mM $MVCl_2$) obtained at different solution flow rates. Cell geometry and cell depth are the same as in fig. 5.

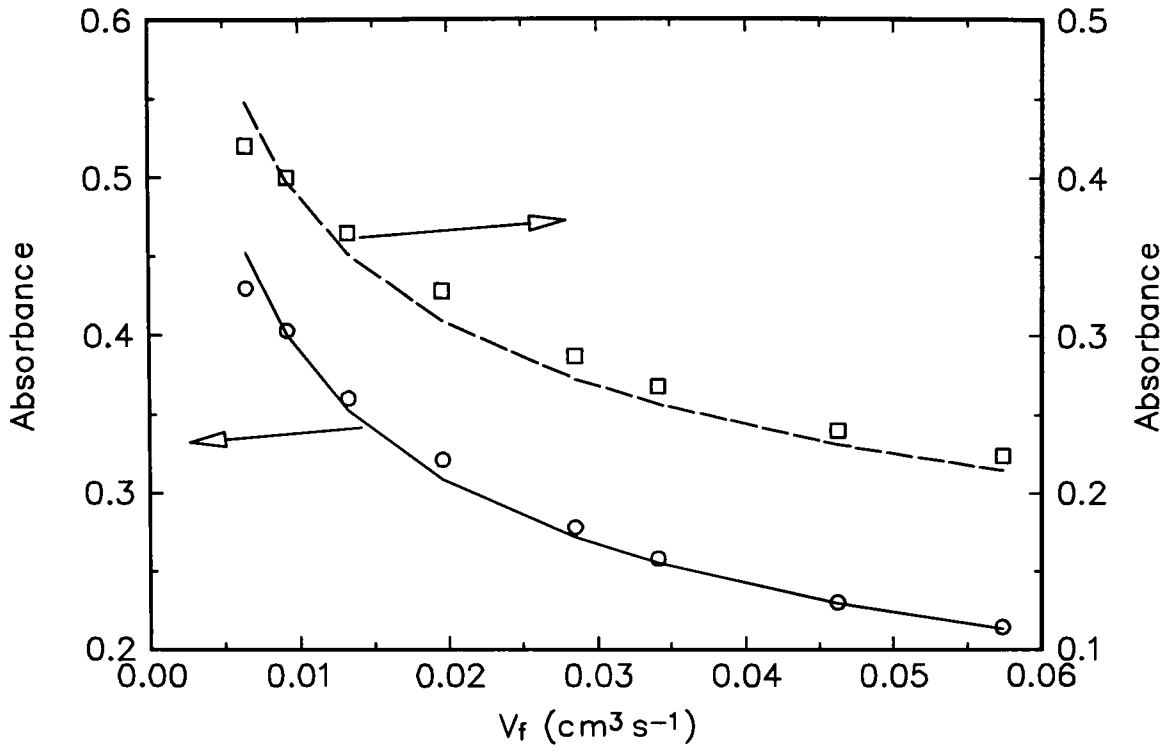


Fig. 7 The flow rate dependence of the absorbance at 395 nm (o) and 366 nm (□) for the $MV^{+\cdot}/(MV^{+\cdot})_2$ mixture generated from 4.98 mM $MVCl_2$ in 0.50 M sodium chloride solution using the SCC method. Solid and dotted lines represent the theoretical behaviour predicted using the model and extinction coefficients as mentioned in the text. Cell geometry and cell depth are the same as in fig. 5.

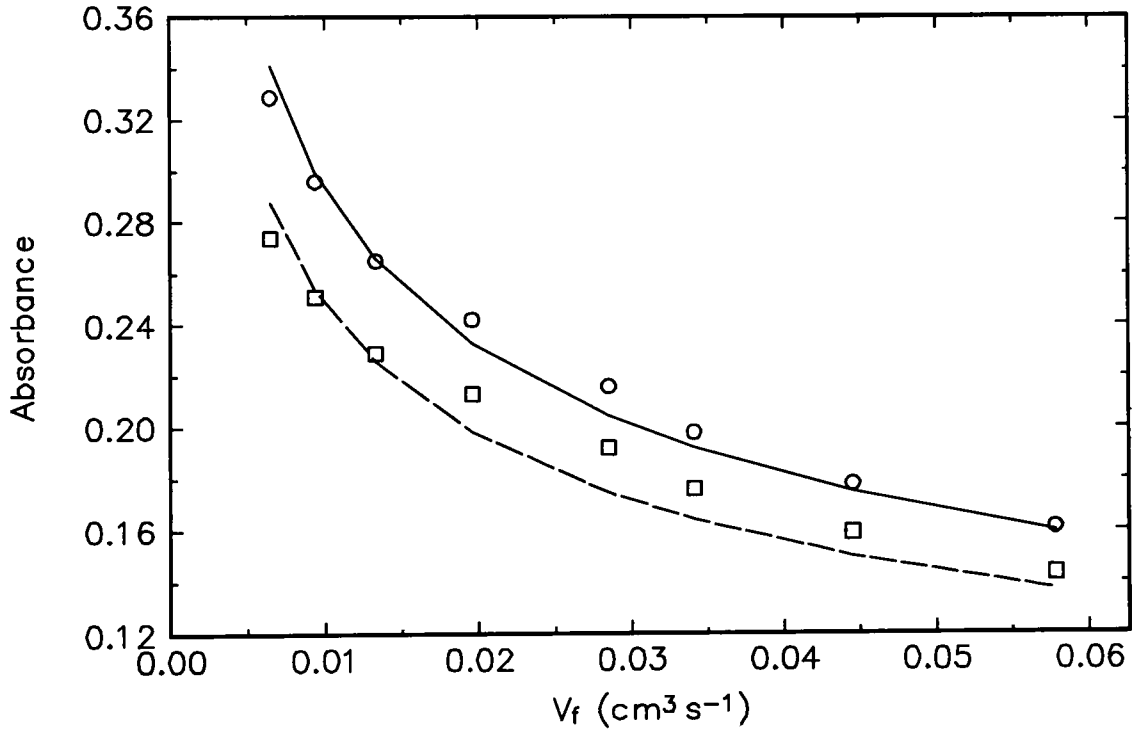


Fig. 8 The flow rate dependence of the absorbance at 395 nm (o) and 366 nm (□) for the $MV^{+\cdot}/(MV^{+\cdot})_2$ mixture generated from 3.51 mM $MVCl_2$ in 0.50 M sodium chloride solution using the SCC method. Solid and dotted lines represent the theoretical behaviour predicted using the model and extinction coefficients as mentioned in the text. Cell geometry and cell depth are the same as in fig. 5.

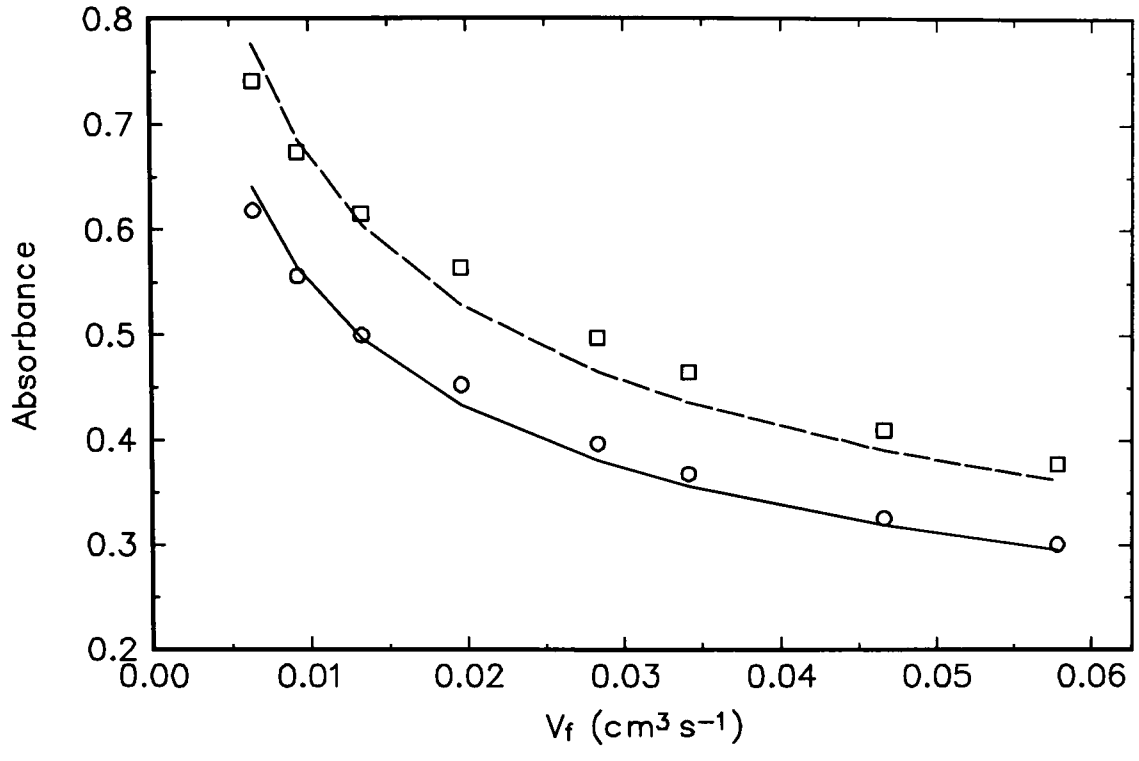
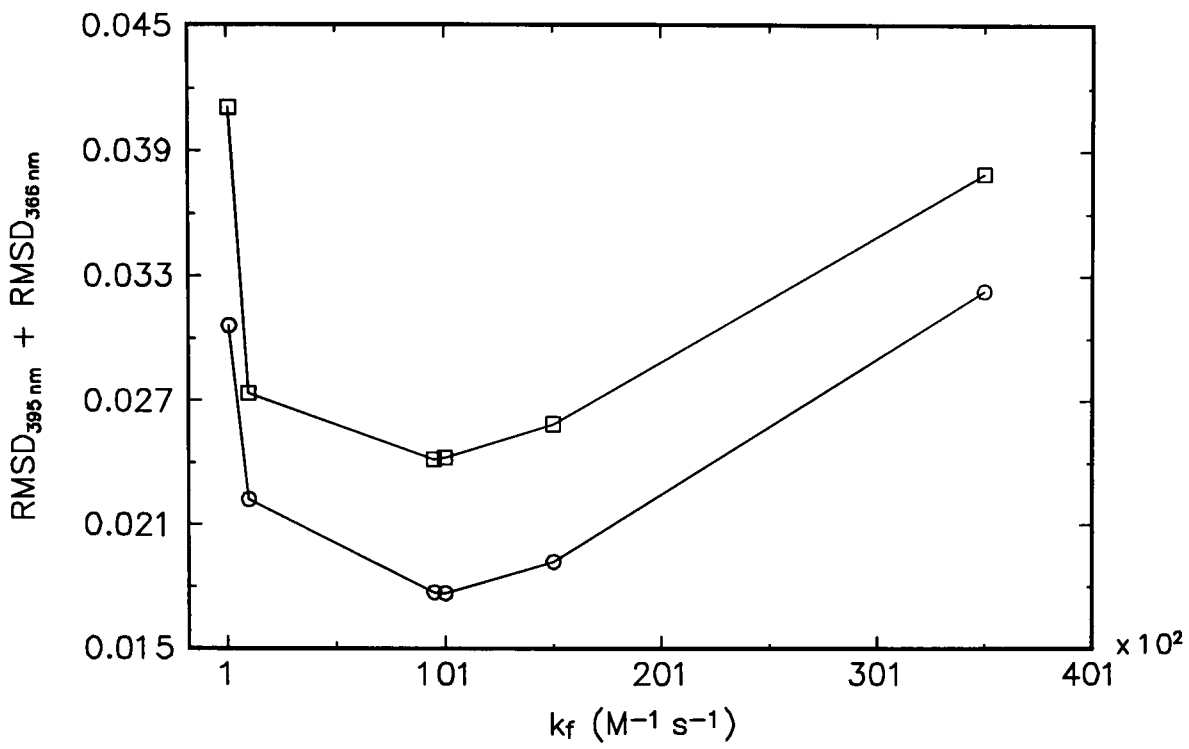


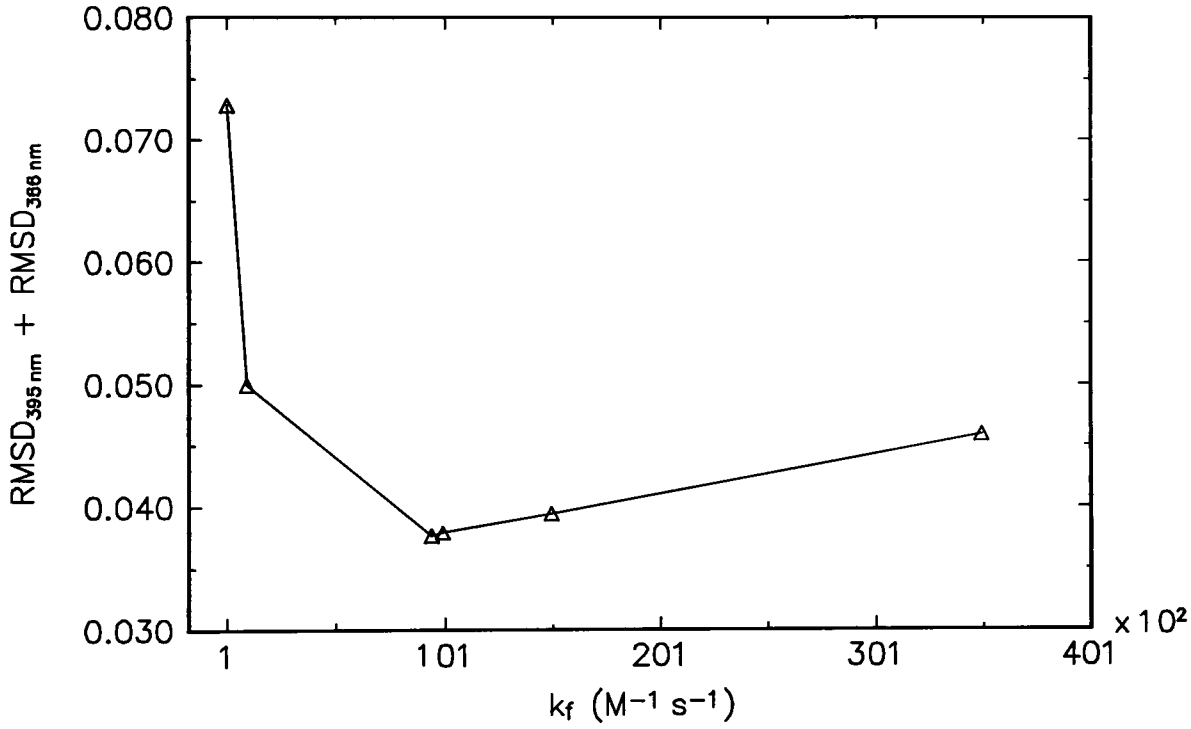
Fig. 9 The flow rate dependence of the absorbance at 395 nm (o) and 366 nm (□) for the $MV^{+\cdot}/(MV^{+\cdot})_2$ mixture generated from 7.72 mM $MVCl_2$ in 0.50 M sodium chloride solution using the SCC method. Solid and dotted lines represent the theoretical behaviour predicted using the model and extinction coefficients as mentioned in the text. Cell geometry and cell depth are the same as in fig. 5.

In modelling the kinetic data, we suppose the diffusion coefficients of all species involved are the same. Recent work using a chronoabsorptometric method demonstrated that the diffusion coefficients of a wide variety of electrogenerated charged radicals are in line with those of the parent compounds.[11] We note that the charge density of $(MV^{+\cdot})_2$ may be comparable to that of $MV^{+\cdot}$ suggesting that the diffusion coefficients of $(MV^{+\cdot})_2$ and $MV^{+\cdot}$ may be similar. In addition, we assume the equilibrium expression (eqn. 4) is valid. For a particular initial $MVCl_2$ concentration, five unknown parameters, namely the extinction coefficients for the monomer and dimer at, respectively, 395 nm and 366 nm as well as the forward rate constant (k_f), are deduced from the spectral data as described before. As depicted in figs. 7-9, the agreement between experiment and theory is apparent. Fig. 10 shows plots of the sum of RMSD values (see eqn. 37) of 395 nm and 366 nm obtained using the BFGS method as a function of k_f for the experimental data of the three initial $MVCl_2$ concentrations.

Fig. 10 Plots of the of the sum of RMSD values (see eqn. 37) for 395 nm and 366 nm obtained using the BFGS method as a function of k_f (a): (o) represent 3.51 mM $MVCl_2$, (□) denote 4.98 mM $MVCl_2$; (b): (Δ) indicate 7.72 mM $MVCl_2$.



(a)



(b)

This vindicates the k_f values as tabulated below. Table 1 lists the rate constants (k_f and k_b) as well as the extinction coefficients for the monomer and dimer derived from the best fit of theoretical and experimental data within a $MVCl_2$ concentration range of 3.51 - 7.72 mM. The extinction coefficients obtained in this work are comparable to those reported by Hawkridge

and others.[40,52,53]

Table 1 Rate constants for the dimerisation[a] of $MV^{+\cdot}$ and the extinction coefficients of $MV^{+\cdot}$ and $(MV^{+\cdot})_2$ in 3.51 mM, 4.98 mM and 7.72 mM $MVCl_2$ solution as determined by the SCC method.

	This work	Stargard & Hawkridge[40]
K_{Dimer} (M^{-1})	720[b] ($\pm$ 80)	720 ($\pm$ 80)
k_f (M^{-1} s^{-1})	9.7 ($\pm$ 0.2)[c] x 10^3	NA[d]
k_b (s^{-1})	13.5[e]	NA
$\varepsilon_{395nm,E}$ (M^{-1} cm^{-1})	3.21 ($\pm$ 0.09) x 10^4	3.49 ($\pm$ 0.01)[f] x 10^4
$\varepsilon_{395nm,F}$ (M^{-1} cm^{-1})	1.27 ($\pm$ 0.22) x 10^3	NA
$\varepsilon_{366nm,E}$ (M^{-1} cm^{-1})	5.95 ($\pm$ 0.46) x 10^3	NA
$\varepsilon_{366nm,F}$ (M^{-1} cm^{-1})	5.31 ($\pm$ 0.06) x 10^4	8.97 ($\pm$ 0.03)[g] x 10^4

[a] *Experiments were conducted at 25 °C with 0.5 M sodium chloride as supporting electrolyte.*

[b] *Value follows Stargard & Hawkridge[40].*

[c] *RMSD values for 3.51 mM, 4.98 mM and 7.72 mM $MVCl_2$ experiments are 1.84 x 10^{-2}, 2.48 x 10^{-2}, 3.87 x 10^{-2} respectively.*

[d] *Not available.*

[e] *Calculated using eqn. 4 (no estimated error quoted).*

[f] *Watanable & Honda[52] and Park et al.[53] reported values of 4.21 ($\pm$ 0.08) x 10^4 M^{-1} cm^{-1} and 4.2 x 10^4 M^{-1} cm^{-1} respectively.*

[g] *Park et al.[53] reported a value of 5.0 x 10^4 M^{-1} cm^{-1}.*

The determination of k_f has been attempted in our laboratory using in situ electron spin resonance (ESR) technique[54] and fast scan voltammetry[55] but the quantification was found to be very difficult. In particular, the in-situ ESR method was proved to be insensitive when applied to the methyl viologen radical cation, if it participates in a rapid reversible reaction, although estimates suggested that k_f was in excess of ca. 1 x 10^3 M^{-1} s^{-1}.[54]

5. Concluding remarks

A novel spectroelectrochemical channel cell has been designed and fabricated for in situ monitoring of the absorption spectra and reaction kinetics of electrogenerated radical species in aqueous solution. The method was applied to study the one electron oxidation of TMPD and, particularly, the one electron reduction of $MVCl_2$. It was found that the former reaction generated a stable $TMPD^{+\cdot}$ radical while the latter process produced a transient $MV^{+\cdot}$ radical which then rapidly underwent a dimerisation reaction. Spectral and kinetic parameters were derived and good agreement found with literature values when available.

6. References

1. Waller A.M.; Compton R.G. *Comprehensive Chemical Kinetics*, Vol 29; Elsevier; Amsterdam, **1989**.

2. Compton R.G; Eklund J.C; Nei L.; Bond A.M.; Colton R.; Mah Y.A. *J. Electroanal. Chem.*, **1995**, *385*, 2498.

3. Coles B.A.; Compton R.G. *J. Electroanal. Chem.*, **1983**, *144*, 87; Compton R.G.; Barghout R.; Eklund J.C.; Fisher A.C.; Davies S.G.; Metzler M.R. *J. Chem. Soc. Perkin Trans. 2*, **1993**, 39.

4. Compton R.G.; Fisher A.C.; Wellington R.G.; Winkler J. *J. Phys. Chem.*, **1992**, *96*, 8153; Compton R.G.; Winkler J.; Riley D.J.; Bearpark S.D.; *J. Phys. Chem.*, **1994**, *98*, 6818.

5. Bond A.M.; Way D.M.; Wedd A.G.; Compton R.G.; Booth J.; Elkund J.C. *Inorg. Chem.*, **1995**, *34*, 3378.

6. Compton R.G.; Dryfe R.A.W. *Progress Reaction Kinetics*, **1995**, *20*, 245.

7. Christensen P.A.; Hamnett A. *Techniques and Mechanism in Electrochemistry*; Blackie Academic & Professional; London, **1994**.

8. Wang Z.; Zhao M.; Scherson D.A. *Anal. Chem.*, **1994**, *66*, 4560.

9. Wang Z.; Scherson D. *J. Electrochem. Soc.*, **1996**, *142*, 4225.

10. Tam K.Y.; Wang R.L.; Lee C.W.; Compton R.G. *Electroanalysis*, in press.

11. Wang R.L.; Tam K.Y.; Marken F.; Compton R.G. *Electroanalysis*, in press; Wang R.L.; Tam K.Y.; Compton R.G. submitted.

12. Bard A.J.; Ledwith A.; Shine H.S. *Adv. Phys. Org. Chem.*, **1976**, *13*, 211.

13. Fujita S.; Steenken S. *J. Am. Chem. Soc.*, **1981**, *103*, 2540.

14. Hazra D.K.; Steenken S. *J. Am. Chem. Soc.*, **1983**, *105*, 4380.

15. Michaelis L.; Schubert M.P.; Granick S. *J. Am. Chem. Soc.*, **1939**, *61*, 1981.

16. Kawamori A.; Honda A.; Joo N.; Suzuki K.; Ooshika Y. *J. Chem. Phys.*, **1966**, *44*, 4363.

17. Kimura K.; Yamada H.; Tsubomura H. *J. Chem. Phys.*, **1968**, *48*, 440.

18. Bakac A.; Espenson J.H. *J. Phys. Chem.*, **1993**, *97*, 12249.

19. Jiang R.; Anson F.C. *J. Phys. Chem.*, **1992**, *96*, 10565.

20. Ohsaka T.; Shintani Y.; Matsumoto F.; Okajima T.; Tokuda K. *Biochem. Bioenergetics*, **1995**, *37*, 73.

21. Suzuki A.; Vergnet C.; Morotgaudry J.F.; Zehnacker C.; Grosclaude J. *Plant Physiol. Biochem.*, **1994**, *32*, 619.

22. Delacey A.L.; Bes M.T.; Gomezmoreno C.; Fernandez V.M. *J. Electroanal. Chem.*, **1995**, *390*, 69.

23. Fry A.J., Sobolov S.B., Leonida M.D. & Voivodov K.I. *Tetrahedron Lett.*, **1994**, *35*, 5607.

24. Jin L.T.; Ye J.S.; Tong W.; Fung Y.Z. *Mikrochim. Acta*, **1993**, *112*, 71.

25. Dai S.; Sigman M.E.; Byrch E.L. *Chem. of Mat.*, **1995**, *7*, 2054.

26. Morales I.; Delarosa F.F. *Solar Energy*, **1992**, *49*, 41.

27. Whitter D.G. *Acc. Chem. Res.*, **1980**, *13*, 82.

28. Ebbesen T.W.; Levey G.; Patterson L.K. *Nature*, **1982**, *298*, 545.

29. Kuo M.L.; Lee K.C.; Lin J.K.; Huang T.S. *Biochim. Biophys. Acta-Mol. Cell Res.*, **1995**, *1268*, 229.

30. Nanni E.J.; Angelis C.T.; Dickson J.; Sawyer D.T. *J. Am. Chem. Soc.*, **1981**, *103*, 4268.

31. Bird C.L.; Kuhn A.T. *Chem. Soc. Rev.*, **1981**, *10*, 49.

32. Blandamer M.J.; Fox M.F.; Symons M.C.R.; Verma G.S.P. *J. Chem. Soc. Chem Comm.*, **1966**, 844.

33. Blandamer M.J.; Brivati J.A.; Fox M.F.; Symons M.C.R.; Verma G.S.P. *Trans. Faraday Soc.*, **1967**, *63*, 1850.

34. Kosower E.M.; Cotter J.L. *J. Am. Chem. Soc.*, **1964**, *86*, 5524.

35. Michaelis L.; Hill E.S. *General Physiol.*, **1933**, *16*, 859.

36. Komers K. *J. Chem. Res. S*, **1994**, 293.

37. Mizuguchi J.; Karfunkel H. *Ber. Bunsenges. Phys. Chem.*, **1993**, *97*, 1466.

38. Heyrovsky M. *J. Chem. Soc. Chem. Comm.*, **1987**, 1856.

39. Meloun M.; Havel J.; Hoegfeldt F. *Computation of Solution Equilibria*; Elllis Horwood; Chichester, **1988**.

40. Stargard J.F.; Hawkridge F.M. *Anal. Chim. Acta*, **1983**, *146*, 1.

41. Winograd N.; Blount H.N.; Kuwana T. *J. Phys. Chem.* **1969**, *73*, 3456.

42. Cotton F.A.; Wilkinson G. *Advanced Inorganic Chemistry*; John Wiley & Sons; New York, 5th ed., **1988**.

43. Atkins P.W. *Physical Chemistry*, Oxford University Press; Oxford, **1986**.

44. Stephenson G. *Mathematical Methods for Science Students*; Longman; Harlow, **1973**.

45. Tam K.Y.; Chau F.T. *J. Auto. Chem.*, **1992**, *14*, 157.

46. Walsh G.R. *Methods of Optimization*; Wiley; London, **1975**; Yuan Y. *IMA J. Numerical Analysis*, **1991**, *11*, 325.

47. Albery W.J.; Compton R.G. *J. Chem. Soc. Faraday Trans. I*, **1982**, *78*, 1561.

48. Larsen J.P. *D.Phil. Thesis*, Oxford University, **1993**.

49. Maruthamuthu P.; Venkatasubramanian L.; Dharmalingam P. *J. Chem. Soc. Faraday Trans. I*, **1986**, *82*, 359.

50. Levich V.G. *Physicochemical Hydrodynamics*; Prentice-Hall; Englewood Cliffs, 1962.

51. Rao P.S.; Hayon E. *J. Phys. Chem.*, **1975**, *79*, 1063.

52. Watanable T.; Honda K. *J. Phys. Chem.*, **1982**, *86*, 2617.

53. Park J.W; Choi N.H.; Kim J.H. *J. Phys. Chem.*, **1996**, *100*, 769.

54. Webster R.; Dryfe R.A.W.; Eklund J.C.; Lee C.W.; Compton R.G. *J. Electroanal. Chem.*, **1996**, *402*, 167.

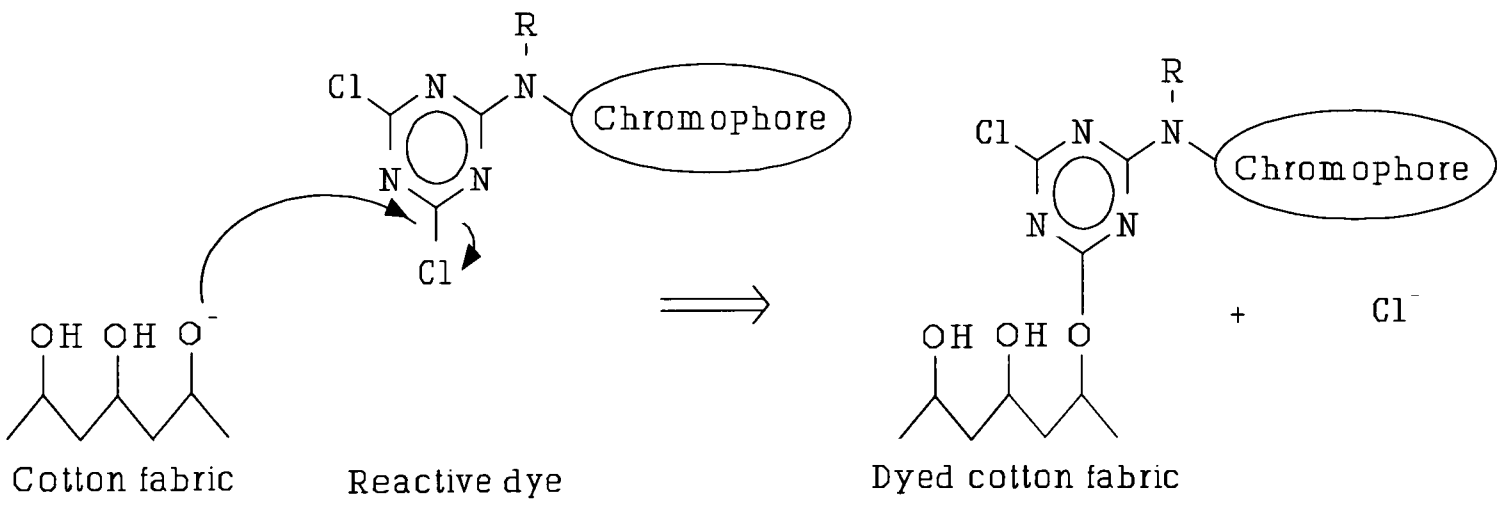
55. Hutchinson F. *Part II Thesis*; Physical and Theoretical Chemistry Laboratory; Oxford, **1996**.

Chapter 8

Applications of the Channel Flow Cell in Kinetic Studies of Reactive Dyeing Processes

1. Introduction

Reactive dyes are one of the most popular class of dyestuffs. They can fix to the fabric by the formation of a covalent bond and therefore have exceptional wetfastness which is an advantage over other classes of dye, such as direct dye, that are immobilised via physical adsorption or mechanical entrapment.[1] Reactive dyes with the dichlorotriazine reactive group are commonly used in the dyeing of cotton fabrics.[2] Dye fixation involves the formation of a covalent bond between a hydroxy group, under alkaline conditions in the fabric, and the dye molecule, by the nucleophilic displacement of chloride ion as shown below:



Studies of the kinetics of reactive dyeing have received considerable attention. One major motivation is to improve the reproducibility of the relevant dyeing processes because of commercial and particularly environmental concerns.[3] However, much previous work has been performed under ill defined hydrodynamic regime and provides little information on the surface reaction mechanism.[4-7] It is therefore anticipated that the reaction mechanisms

associated with reactive dyes will continue to present a challenging problem to the research community.[8]

The channel flow cell method has been widely utilised to study kinetic processes taking place at solid-liquid interfaces.[9,10] The technique was shown to be useful in the investigation of dyeing kinetics in cotton fabrics.[11,12] In this work,[11,12] dye solution was flowed, under laminar conditions, over woven cotton fabric which was embedded in one wall of the flow cell. An electrochemical detector was mounted immediately downstream of the cotton fabric to sensitively monitor the consumption of dye or the release of product (such as Cl^-). Since the hydrodynamics are well defined, the convective-diffusion-kinetic equations within the flow cell could be solved precisely to give a quantitative assessment of the detector response as a function of solution flow rate. In this manner, the interfacial kinetics between dichlorotriazinyl reactive dyes and cotton fabrics were found to be controlled by a surface reaction which was first order with respect to the surface concentration of the dye and controlled by a process at the interface or in the fibre pores.[11,12] Explicit information to separate out the adsorption-diffusion-reaction process is not however obtained.

In the real-world dyebath environment, the fixation of reactive dye is usually carried out in a high pH medium with a substantial concentration of supporting electrolyte, e.g. sodium sulphate. Moreover, in many situations, the dye molecule may not be electrochemically active or the detection of it may become difficult within the dyebath matrix. All these considerations limit the immediate application of the channel flow cell method. However, we recently developed a spectroelectrochemical channel cell based on a channel electrode to interrogate the dimerisation kinetics of the methyl viologen radical cation in water[13] (see chapter 7) and

the diffusion properties of the tris(4-bromophenyl)amine radical cation in acetonitrile.[14] In this chapter, this spectroelectrochemical channel cell is extended to spectrochemically investigate the kinetics of reactive dyeing on knitted cotton fabrics. Specifically, we monitor the in situ consumption of dye by means of UV-visible spectroscopy. In contrast to the previous work[11,12] using electrochemical detection, it is envisaged that the proposed method will be universal to the measurement of dye uptake by fabric under realistic dyebath conditions since all dyestuffs are necessarily spectroscopically active in the visible region. Furthermore, comparison of the results obtained versus those derived from chloride detection should go some way to differentiating the chemical reaction and physical adsorption/diffusion processes.

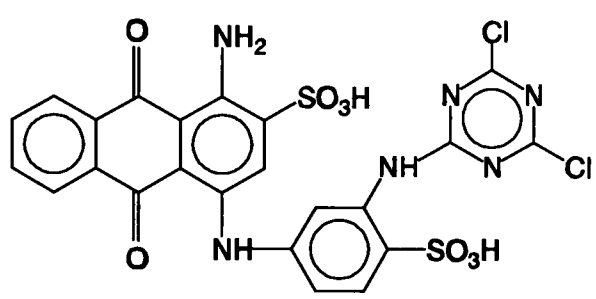
In the present studies, the reactive dyeing kinetics between PROCION Blue MX-R (A) and knitted cotton fabrics are examined using the proposed spectrochemical method. It should be emphasised that the dyeing process is complicated by a parallel hydrolysis reaction of the dye molecules and the uptake of hydrolysed dye which are given, respectively, in eqn. 1 and 2. In particular, we measure the homogeneous hydrolysis rate of the dye (eqn. 1) and the uptake rate of the hydrolysed dye (eqn. 2) independently. In the consideration of the reactive dyeing process, a reaction model based on eqns. 1-3 is proposed to account for the kinetic data obtained over a wide range of initial dye concentrations and supporting electrolyte concentrations. The dyeing rate constants and homogeneous hydrolysis rate constants (k_{DCl}, k_{DOH} and k_{hyd}) are reported.

$$DCl_{A} + OH^{-} \xrightarrow{k_{hyd}} DOH_{B} + Cl^{-} \quad (1)$$

$$DOH_{B} + Cell\text{-}O^{-}_{C} \xrightarrow{k_{DOH}} [DOH \cdots O\text{-}Cell]^{-} \quad (2)$$

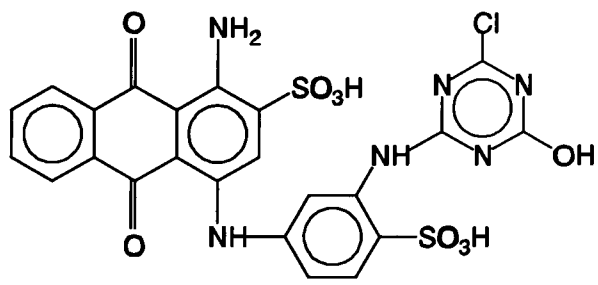
$$DCl_{A} + Cell\text{-}O^{-}_{C} \xrightarrow{k_{DCl}} [DCl \cdots Cell\text{-}O]^{-} \quad (3)$$

A



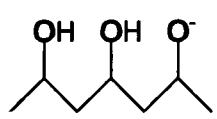
= DCl

B



= DOH

C



Cotton fabric

Note that the dotted lines in eqns. 2 and 3 both represent the adsorption of the dye molecule to the fabric surface whereas the latter step may further involves subsequent chemical bond

formation.

Mercerisation is frequently invoked as a pretreatment technique for cotton fabrics.[2] This process is mainly to "pad" the fabrics through exposure to 25% caustic soda followed by a complete wash out of the caustic. It is believed that this treatment changes the crystalline structure and swells the fibres so as to create more sites for chemical and physical binding of dye molecules.[15] Ex situ atomic force micrographs of cotton fibres provide insight to the effect of mercerisation as a pretreatment process for cotton fabrics.

2. Experimental

A schematic diagram of the channel cell for dyeing studies is given in fig. 1. The cell is made of a rectangular duct (about 4.5 cm long, 0.1 cm deep and 0.6 cm wide) constructed in a transparent perspex block (Goodfellow Advanced Materials, Cambridge) and closed by a cover plate of the same material. The cotton fabric (about 2.0 cm long, 0.1 mm thick and 0.6 cm wide) was adhered into the cover plate with a small recess about 0.15 mm deep, by means of a rubber based glue (Evo Stick, Evode Ltd., Stafford). The resulting fabric surface was made flush to the cover plate surface after swelling. Spectroscopic access was accomplished by using a piece of Black Delrin (Goodfellow Advanced Materials, Cambridge) mask drilled with a rectangular window about 0.4 cm wide and 0.3 cm long. Details of the assembly of the spectrochemical flow cell are given in chapter 7.

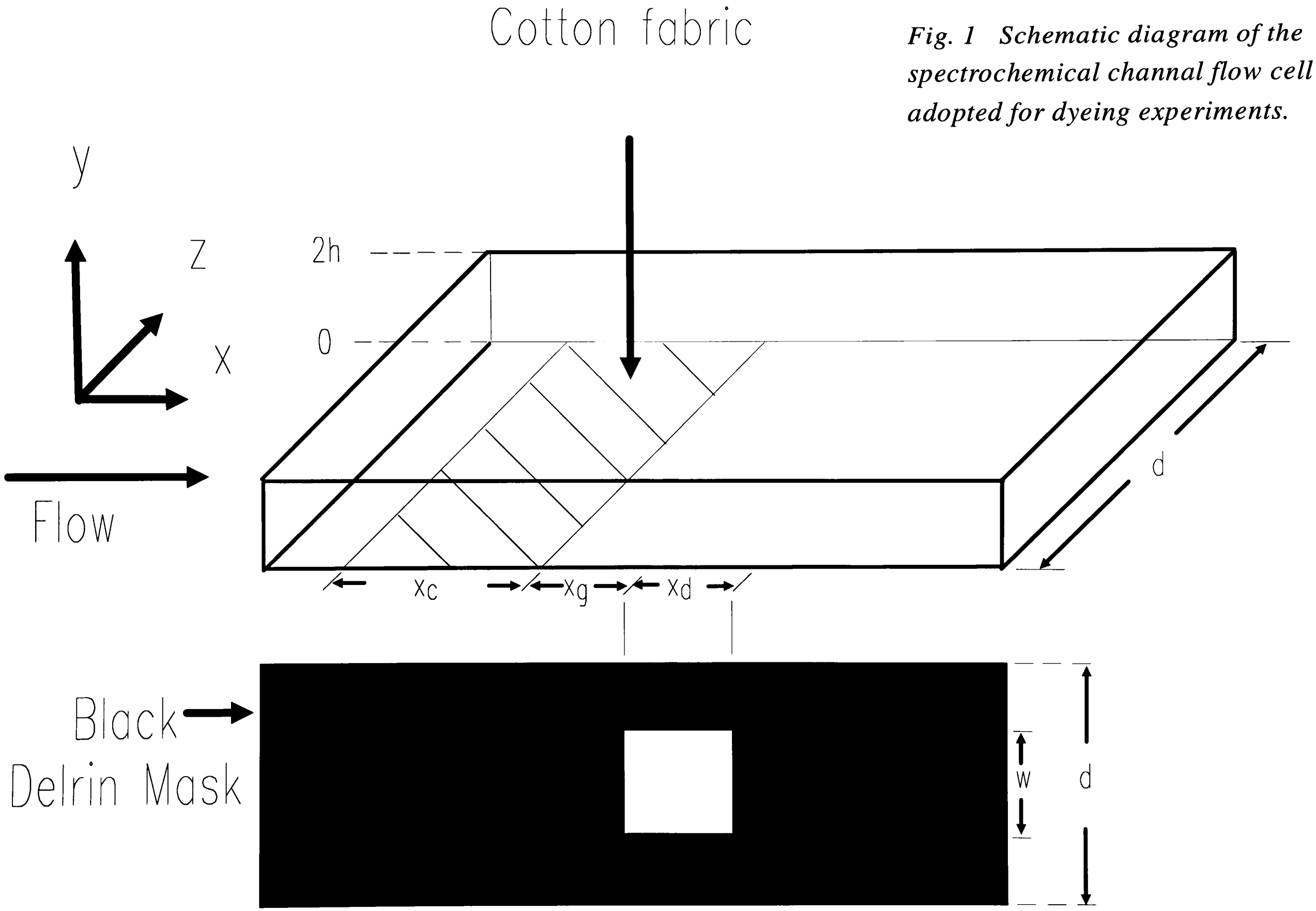


Fig. 1 Schematic diagram of the spectrochemical channal flow cell adopted for dyeing experiments.

Typical values for the cell geometry were: x_c = 1.986 cm, x_g = 0.168 cm, x_d = 0.302 cm, w = 0.396 cm and d = 0.617 cm (see fig. 1). The cell depth (2h) was determined spectroscopically via measurements on a series of known concentrations of potassium ferricyanide solution. A typical value for the cell depth was 0.070 ($\pm$ 0.002) cm. For all flow cell experiments, dye solution was flowed through the channel via inlet and outlet teflon tubes (not shown in fig. 1) which were attached to the ends of the cell along the y-direction. Solution flow was accomplished by using a gravity feeding system capable of delivering flow rates in the range of 10^{-4} - 10^{-2} cm^3 s^{-1}. Kinetic experiments were carried out at 30 °C. A double beam scanning spectrophotometer (ATI Unicam, Cambridge, model UV2-100) was implemented for all optical measurements.

Mercerisation of cotton fabrics or fibres was carried out by boiling the sample in 25 % (w/w) sodium hydroxide solution for about 45 minutes. The resulting fabrics or fibres were washed with hot water until the solution became neutral. Dyeing experiments were performed using both mercerised and unmercerised fabric. Hydrolysed dye solutions were made up by the corresponding dye solution warmed at 70 °C for at least 5 hours. A Parks AFM (SPM-BD2) was adopted to image the surface of the cotton fibres. Single fibres (average diameter 2 - 5 μm) were removed from the dry fibre sample by using a pair of forceps and mounted on the sample stub using double-sided adhesive tape.

A chloride ion selective electrode (Orion 9417SC, Boston) was applied to measure the homogeneous hydrolysis kinetics of the dye. Potentiometric detections were made with reference to a saturated calomel electrode and were accomplished using a Jenway 3030 pH meter. All kinetic measurements were performed at 30 °C. Complementary flow cell dyeing

experiments were conducted using the Orion ion selective electrode. The chloride sensor (outer diameter of 1.148 cm, sensing element of diameter 0.799 cm) was mounted downstream of an unmercerised fabric (see chapter 3). A channel cell width of 1.148 cm and depth of 0.093 cm was employed. This permits the whole sensing element of the electrode to be utilised for experimentation. Typical values of x_c and x_g are 2.003 cm and 0.211 cm respectively.

3. Modelling of the dyeing process

Assuming that the rate of hydrolysis of A is slow compared with the channel flow cell experimental time scale, the convective-diffusion equations describing the transport of A and B within the channel cell can be written as

$$\frac{\partial [A]}{\partial t} = D_A \frac{\partial^2 [A]}{\partial y^2} - v_x \frac{\partial [A]}{\partial x} = 0 \qquad (4)$$

and

$$\frac{\partial [B]}{\partial t} = D_B \frac{\partial^2 [B]}{\partial y^2} - v_x \frac{\partial [B]}{\partial x} = 0 \qquad (5)$$

with

$$v_x = \frac{3\ V_f}{4\ h\ d} \left[1 - \frac{(h - y)^2}{h^2}\right] \qquad (6)$$

where the symbols h, d, x, and y are defined in fig. 1. v_x is the solution velocity axially through the channel and V_f denotes the solution flow rate in $cm^3\ s^{-1}$. D_A and D_B are, respectively, the diffusion coefficients of A and B. Solutions to eqns. 4 & 5 are readily accomplished by dividing the fabric, gap and detector window regions into grids of K (x-

direction) x J (y-direction) boxes via the standard backwards implicit finite difference (BIFD) procedure as detailed in chapter 2.[16] Assuming the initial dye concentration is D_0, the bulk concentration of A and B at particular time t are:

$$[A]_b = D_0 \exp(-k_{hyd}\, t) \qquad (7)$$

$$[B]_b = D_0 - [A]_b \qquad (8)$$

At the fabric surface, the dye uptake kinetics can be formulated as

$$J_A \ (mol\ cm^{-2}\ s^{-1}) = D_A \frac{\partial [A]}{\partial y}\bigg|_{y=0} = k_{DCl}\,[A]_0 \qquad (9)$$

$$J_B \ (mol\ cm^{-2}\ s^{-1}) = D_B \frac{\partial [B]}{\partial y}\bigg|_{y=0} = k_{DOH}\,[B]_0 \qquad (10)$$

where $[A]_0$ and $[B]_0$ denote the surface concentrations of A and B respectively. k_{DCl} and k_{DOH} are, respectively, the dyeing rate constants of A and B. Considering eqns. 9 & 10 in finite difference form, the matrix elements b_1 and d_1 (see chapter 2) can be written as follows:

$$A: \quad b_1 = \frac{\lambda^A{}_1\,(2\,k_{DCl}\Delta y + D_A)}{k_{DCl}\Delta y + D_A} + 1, \quad d_1 = g^A{}_{1,k} \qquad (11)$$

$$B: \quad b_1 = \frac{\lambda^B{}_1\,(2\,k_{DOH}\Delta y + D_B)}{k_{DOH}\Delta y + D_B} + 1, \quad d_1 = g^B{}_{1,k} \qquad (12)$$

At the far wall and detector window,

$$\frac{\partial [A]}{\partial y} = \frac{\partial [B]}{\partial y} = 0 \qquad (13)$$

with matrix elements b_1 and d_1

$$A: \quad b_1 = \lambda^A{}_1 + 1, \quad d_1 = g^A{}_{1,k} \tag{14}$$

$$B: \quad b_1 = \lambda^B{}_1 + 1, \quad d_1 = g^B{}_{1,k} \tag{15}$$

Once the concentration profiles of DCl ($[A]_{x,y,t}$) and DOH ($[B]_{x,y,t}$) at a particular time (t) and solution flow rate (V_f) have been computed, the downstream absorbance at a particular wavelength (λ) can be evaluated by the following equation,

$$Abs_\lambda\ (V_f, t) = \frac{\varepsilon_{\lambda,A} \int_{x_1}^{x_2} \int_0^{2h} [A]_{x,y,t}\, dy\, dx + \varepsilon_{\lambda,B} \int_{x_1}^{x_2} \int_0^{2h} [B]_{x,y,t}\, dy\, dx}{x_2 - x_1} \tag{16}$$

$\varepsilon_{\lambda,A}$ and $\varepsilon_{\lambda,B}$ represent, respectively, the extinction coefficients of A and B while x_1 and x_2 indicate the x-coordinates at the beginning and end of the detector window. The integral of eqn. 16 was evaluated by using a trapezoidal method.[17] In all calculations, a grid size of 5000 x 3000 was required for the absorbance profile to converge within 2%. All supporting programs were coded in an UNIX C environment and executed on a SUN Sparc workstation.

It can be seen that the absorbance (eqn. 16) is a function of solution flow rate (V_f), time (t), initial dye concentration (D_0), diffusion coefficients (D_A, D_B), extinction coefficients ($\varepsilon_{\lambda,A}$, $\varepsilon_{\lambda,B}$), cell geometry and, more importantly, the dyeing rate constants k_{DCl} and k_{DOH}. For a given experiment, V_f, t, D_0, D_A, D_B, $\varepsilon_{\lambda,A}$, $\varepsilon_{\lambda,B}$ and cell geometry are easily determined. Here, the values of k_{DOH} are first measured precisely by using completely hydrolysed dye solutions. Therefore, k_{DCl} can be derived unambiguously from the best fit between theoretical and experimental data. An error function, root-mean-square-deviation (RMSD), is defined to indicate the quality of the fit between theory and experiment

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N} [dA(i)^{theory} - dA(i)^{expt}]^2}{N}} \quad (17)$$

where N is the number of data points and dA is the difference between the absorbance measured through the detector window without and with the fabric, at the specified flow rate and time. The former quantity can be determined by reversing the flow through the cell so that solution passes over the detector window before it reaches the fabric. To this end, the BFGS (Broyden-Fletcher-Goldfard-Shanno) method coupled with Powell's quadratic interpolation linear search technique[13,18-20] was implemented to minimise the RMSD value for the deduction of optimised dyeing rate constants.

4. Results and discussion

The homogeneous hydrolysis kinetics at 30 °C were investigated by using a chloride ion selective electrode (ISE). Fig. 2 shows typical calibrations for the ISE obtained in different solution matrices. The slope of the plot was decreased from 54.2 mV per decade in 0.10 M KNO_3 solution to 6.8 mV per decade in 0.10 M Na_2CO_3 and 1.0 mM dye solution. Addition of 0.35 M Na_2SO_4 in the latter sample gave a similar behaviour of 8.2 mV per decade. Obviously, the sensitivity of the chloride sensor was degraded significantly in the combined presence of dye, sodium sulphate and sodium carbonate.

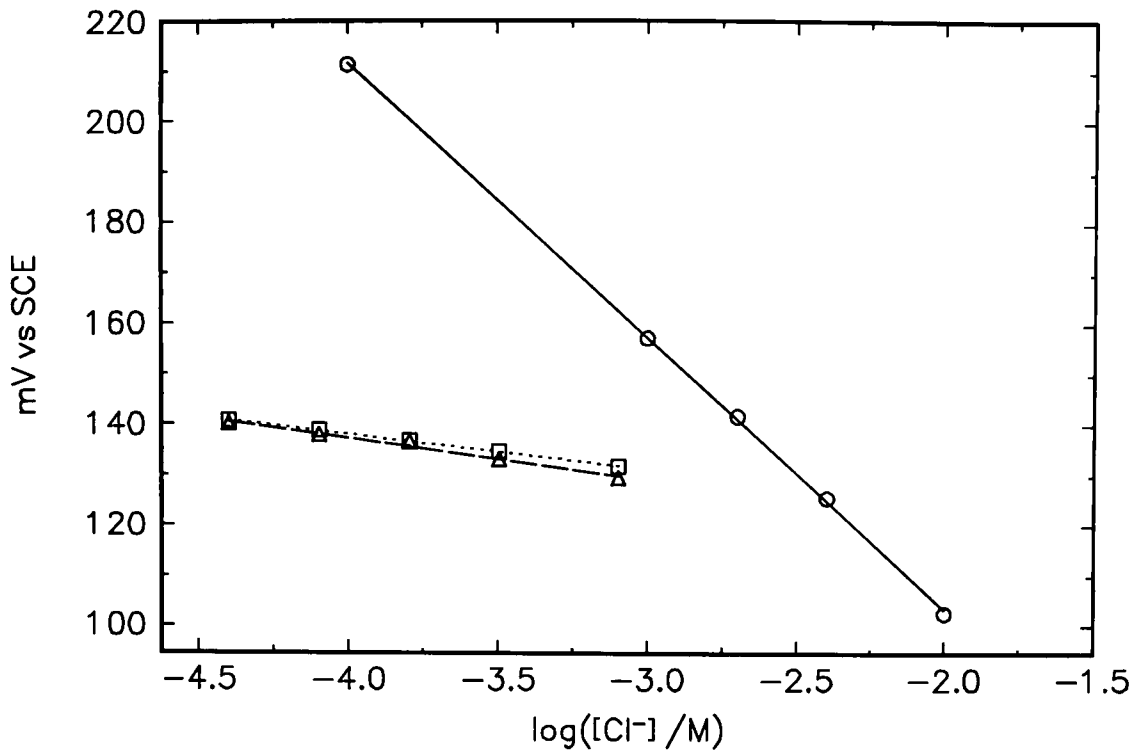


Fig. 2 Calibration plot for chloride ion selective electrode at 30 °C obtained in different solution matrices where (o) represents 0.1 M KNO_3, (□) denote 0.1 M Na_2CO_3 and 1.0 mM DCl and (△) indicates 0.35 M Na_2SO_4, 0.1 M Na_2CO_3 and 1.0 mM DCl. The solid line represents the best fit to the Nernst equation.

In the determination of the homogeneous hydrolysis kinetics, the chloride concentration of the DCl solutions (see table 1) as a function of time was monitored using the ISE. The DCl concentration was calculated (see eqn. 1) by subtracting this quantity from the initial dye concentration. Fig. 3 depicts a typical plot of the natural logarithm of DCl concentration in 0.1 M Na_2CO_3 and 0.35 M Na_2SO_4 solution as a function of time which is in good agreement with first order kinetics. Table 1 summaries the first order hydrolysis rate constants (k_{hyd}). It can be seen that the rate constant is slightly larger in the presence of 0.35 M Na_2SO_4. However, the hydrolysis process is too slow to be detectable in neutral solution.

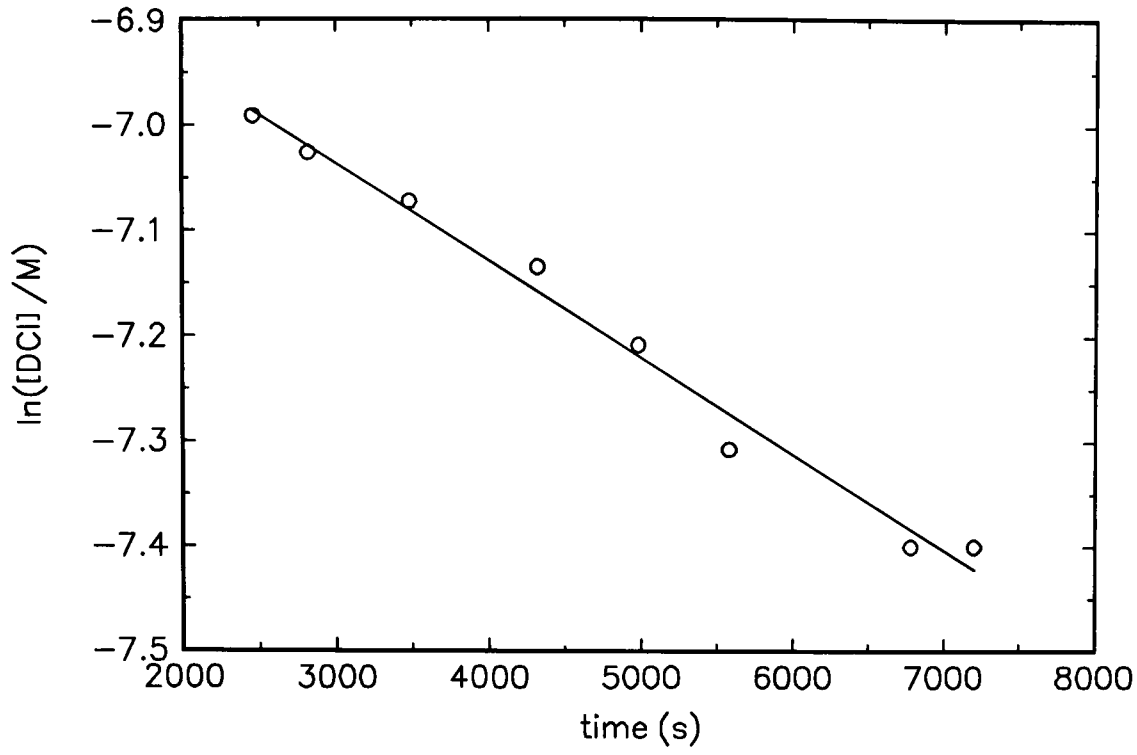


Fig. 3 A plot of the natural logarithm of the DCl concentration against time for the homogeneous hydrolysis of 1.0 mM DCl in the presence of 0.1 M Na_2CO_3 and 0.35 M Na_2SO_4 at 30 °C.

Table 1 Homogeneous hydrolysis rate constants (k_{hyd}) at 30 °C determined using a commercially available chloride sensor.

Na_2SO_4 (M)	Na_2CO_3 (M)	k_{hyd}[a] (10^{-5} s^{-1})
0.35	0.10	9.1 ± 0.7
0.00	0.10	8.3 ± 0.8
0.35	0.00	NA[b]
0.00	0.00	NA

[a] *initial dye concentration was 1.0 mM*
[b] *not available*

Before turning to the dyeing kinetics experiments, we examined the extinction coefficient and diffusion coefficient of DCl and DOH. Fig. 4 gives the absorption spectra of DCl obtained from different dye concentrations in 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4 solution. As shown in fig. 5, the absorption maximum at 604 nm over a wide range of DCl concentration agreed well with Beer's law. Moreover, no isosbestic point was observed in the absorption spectra

(see fig. 4). These observations imply that any self-association of the dye molecules is negligible or unchanging over the concentration range. It was found that the spectral properties of DCl and DOH were essentially indistinguishable - see table 2.

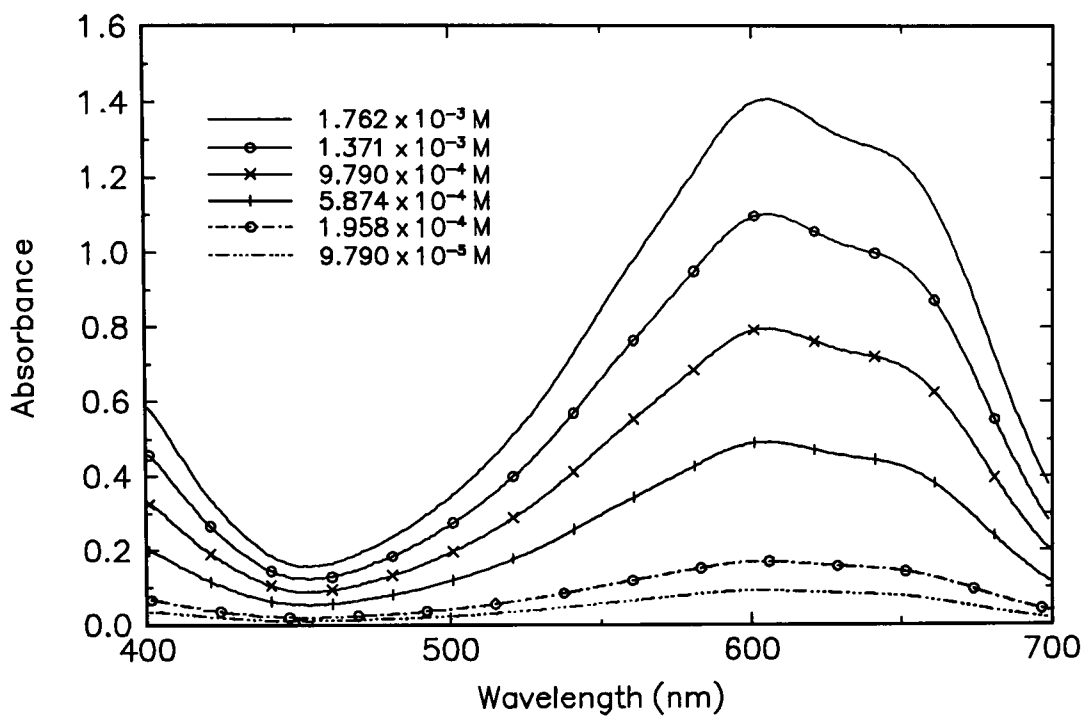


Fig. 4 Absorption spectra of DCl obtained from different dye concentrations in 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4 solution. Cell depth is 0.117 cm.

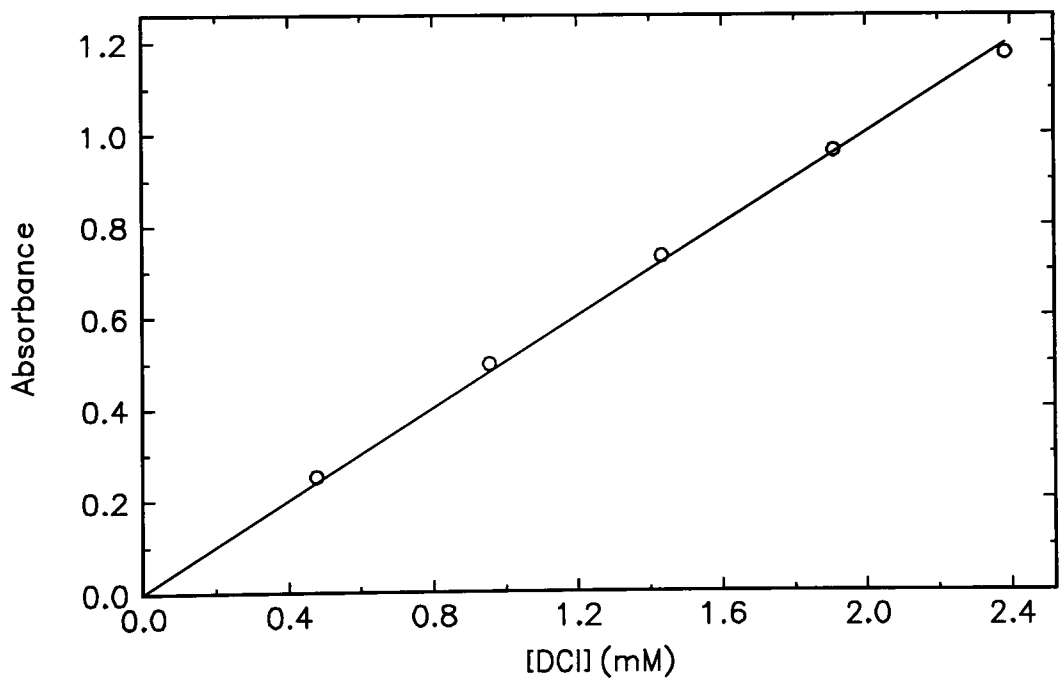


Fig. 5 Absorbance at 604 nm against the concentration of DCl in 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4 solution with cell depth equal to 0.0726 cm. The solid line indicates the theoretical behaviour generated by Beer's law.

Table 2 *Spectral parameters of DCl and DOH at 30 °C.*

Na_2SO_4 (M)	Na_2CO_3 (M)	$\varepsilon_{604nm,A}$ or $\varepsilon_{604nm,B}$ (M^{-1} cm^{-1})
0.35	0.10	6912 ± 48
0.35	0.00	6666 ± 12
0.00	0.10	7360 ± 45
0.00	0.00	7277 ± 9

The diffusion coefficient of DCl was 4.1 (± 0.6) x 10^{-6} cm^2 s^{-1} as determined using rotating disc voltammetry via a mercury plated copper electrode.[11,21] This quantity is in good agreement with a value of 4.14 x 10^{-6} cm^2 s^{-1} predicted by using the Wilke-Chang equation.[22] The diffusion coefficient of DOH was 4.20 x 10^{-6} cm^2 s^{-1} as calculated by the Wilke-Chang equation.

Next, attention was directed to channel flow cell experiments. In the studies of dyeing kinetics, it was first important to ensure the fabric was not dye-saturated within the flow cell experimental time scale. We therefore measured the transient absorptometric behaviour at a fixed flow rate (1.4 x 10^{-4} cm^3 s^{-1}) for DCl and DOH solution in the presence of unmercerised fabric. The corresponding rate constants are calculated by the aforementioned optimisation method and are given in fig. 6. It can be seen that after a short initial transient, the fabric surface remains essentially active for at least 100 minutes. All steady state dyeing experiments reported below were conducted within this time interval.

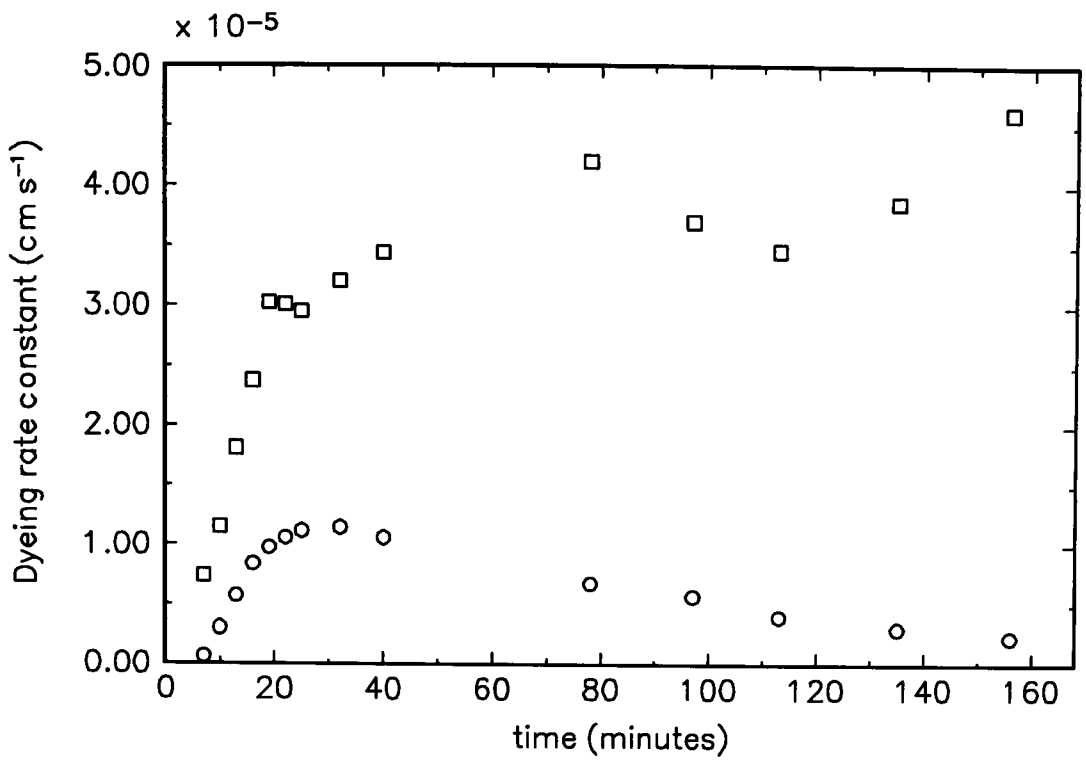


Fig. 6 Variations of k_{DCl} (o) and k_{DOH} (□) for unmercerised fabric as a function of time at a solution flow rate of 1.4 x 10^{-4} cm^3 s^{-1} with initial dye concentration of 1.0 mM in the presence of 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4. Cell geometry: x_c = 2.009 cm, x_g = 0.144 cm and 2h = 0.0759 cm for k_{DCl} measurement; x_c = 2.006 cm , x_g = 0.155 cm and 2h = 0.0691 cm for k_{DOH} measurement.

Figs. 7, 8, 9, and 10 depict, respectively, the flow rate dependence of dA for unmercerised fabric with 3.0 mM DCl, 1.0 mM DCl, 0.5 mM DCl and 1.0 mM DOH in the presence of 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4. All other experiments (see table 3) exhibit a similar trend. Table 3 lists the optimised dyeing rate constants (k_{DCl} and k_{DOH}) obtained from the best fit between experimental and theoretical data for the initial dye concentrations of 3.0 - 0.5 mM. It can be seen that a 2 to 5 times increases in k_{DCl} can be attained in the presence of 0.10 M Na_2CO_3 (pH ≈ 11). Moreover, an addition of 0.35 M Na_2SO_4 enhances the dyeing rate constants appreciably. These are clearly demonstrated in the experiments with 1.0 mM initial dye concentration. Furthermore, the dyeing rate constants are increased slightly for the fabric with mercerisation pretreatment. In summary, k_{DCl} and k_{DOH} are intensified first by using alkaline media - such as 0.10 M Na_2CO_3, second by using 0.35 M Na_2SO_4 and third by mercerisation pretreatment of fabrics.

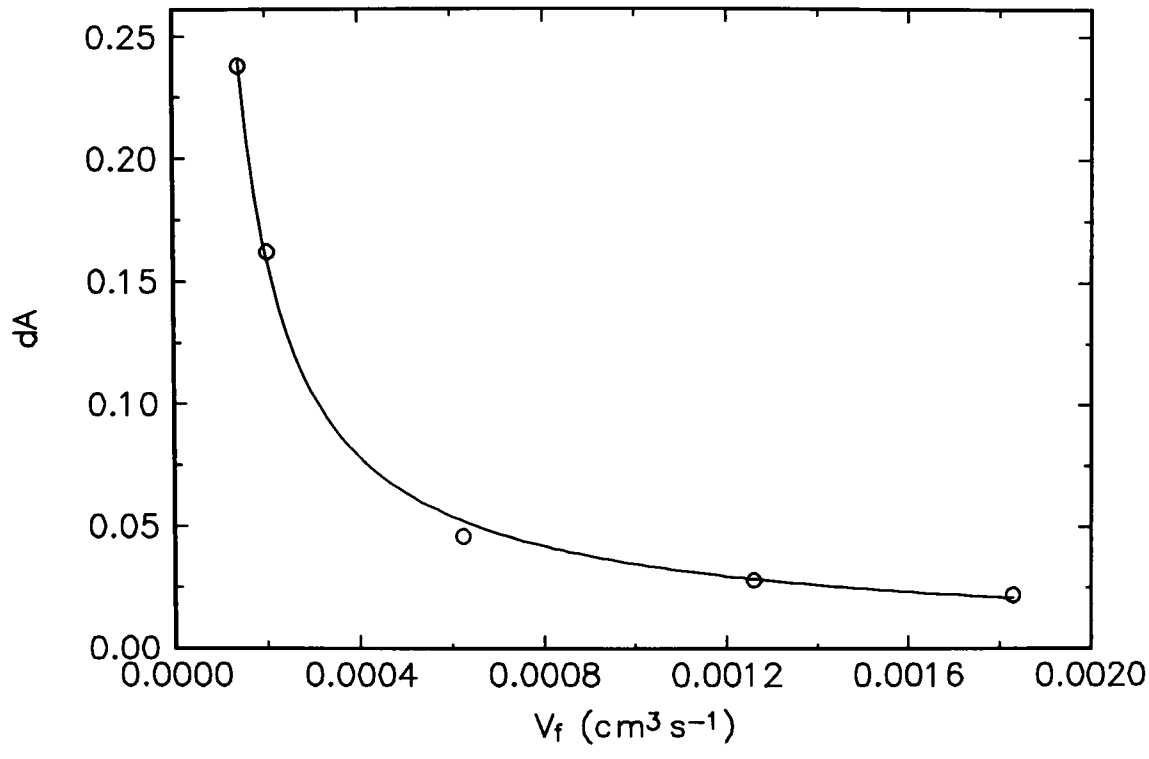


Fig. 7 Flow rate dependence of dA measured using the spectrochemical channel cell for the dyeing of unmercerised fabric with initial dye concentration (DCl) of 3.0 mM in the presence of 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4: (o) experimental data and (——) theoretical behaviour predicted using the interfacial reaction model and the optimised rate constants shown in tables 1 and 3. Cell geometry: x_c = 1.986 cm, x_g = 0.171 cm, d = 0.617 cm and 2h = 0.0669 cm.

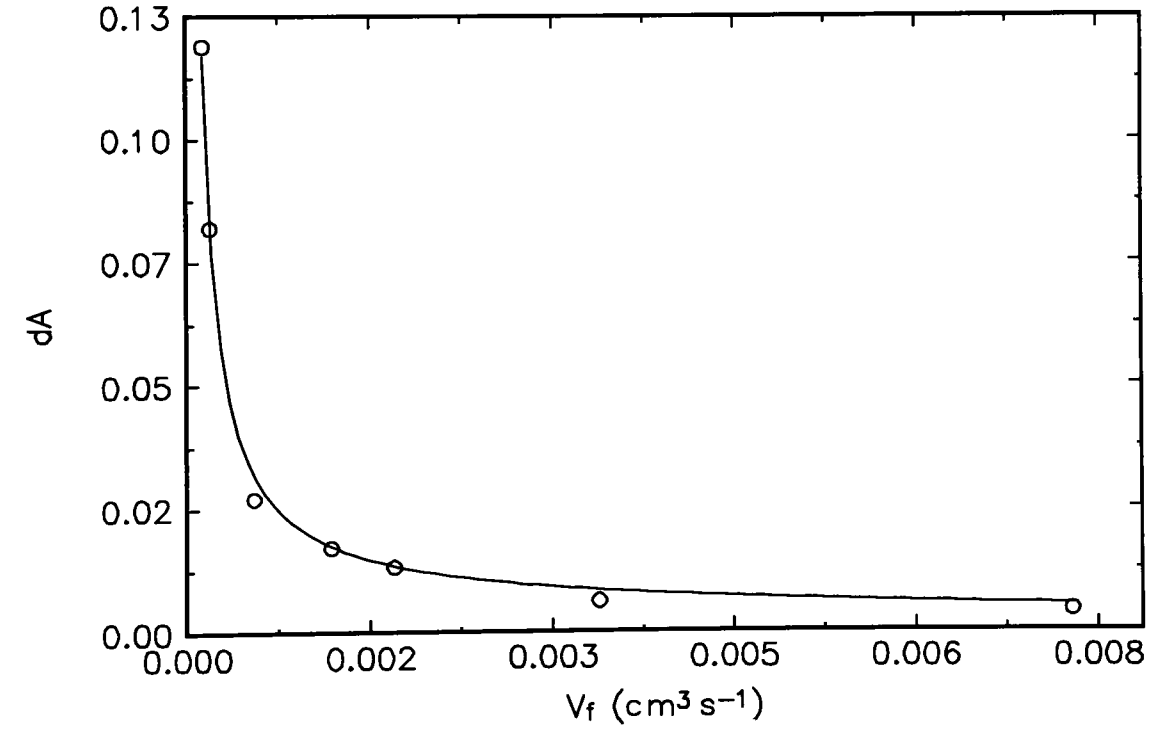


Fig. 8 Flow rate dependence of dA measured using the spectrochemical channel cell for the dyeing of unmercerised fabric with DCl concentration of 1.0 mM in the presence of 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4: (o) experimental data and (——) theoretical behaviour predicted using the interfacial reaction model and the optimised rate constants shown in tables 1 and 3. Cell geometry: x_c = 1.986 cm, x_g = 0.168 cm, d = 0.617 cm and 2h = 0.0698 cm.

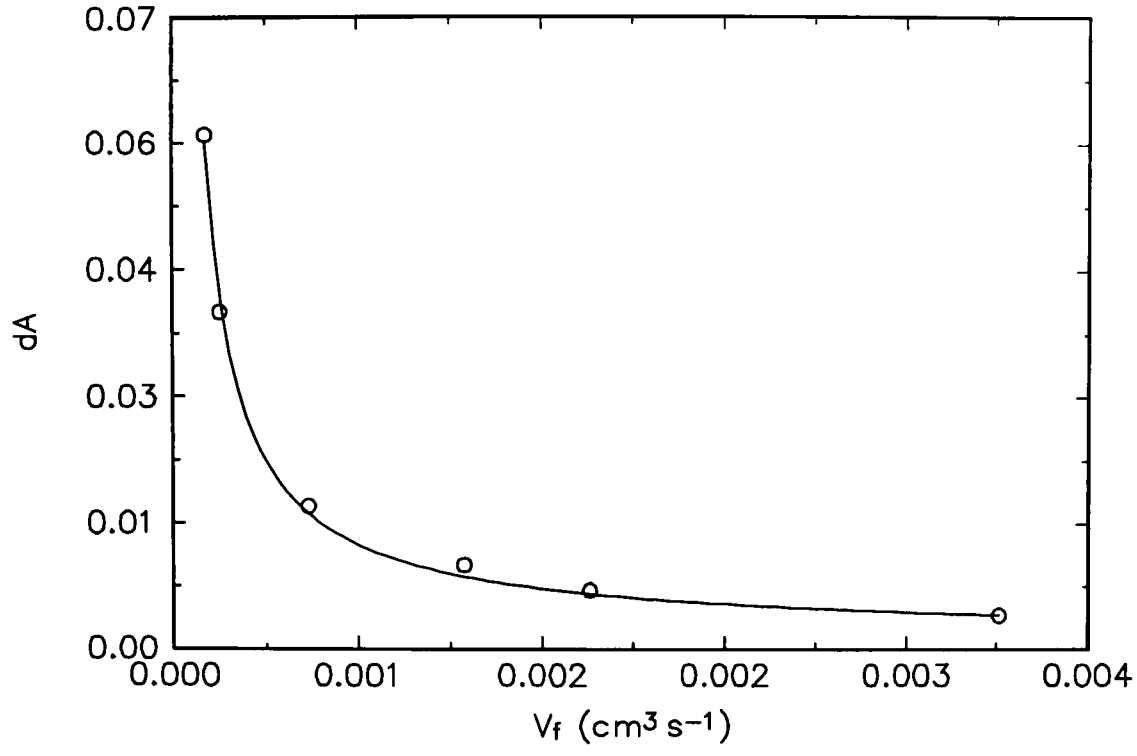


Fig. 9 Flow rate dependence of dA measured using the spectrochemical channel cell for the dyeing of unmercerised fabric with DCl concentration of 0.4 mM in the presence of 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4: (o) experimental data and (——) theoretical behaviour predicted using the interfacial reaction model and the optimised rate constants shown in tables 1 and 3. Cell geometry: x_c = 1.998 cm, x_g = 0.162 cm, d = 0.617 cm and 2h = 0.0688 cm.

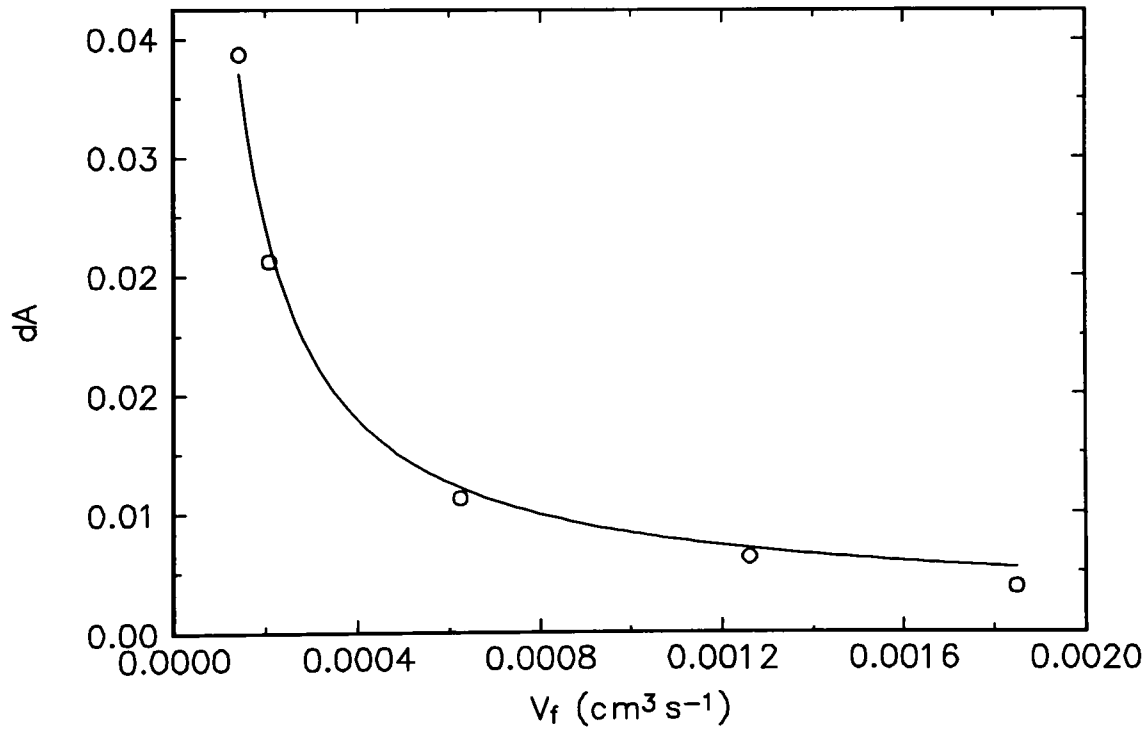


Fig. 10 Flow rate dependence of dA measured using the spectrochemical channel cell for the dyeing of unmercerised fabric with DOH concentration of 1.0 mM in the presence of 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4: (o) experimental data and (——) theoretical behaviour predicted using the interfacial reaction model and the optimised rate constants shown in table 3. Cell geometry: x_c = 1.995 cm, x_g = 0.149 cm, d = 0.617 cm and 2h = 0.0663 cm.

Table 3 *Dyeing rate constants (k_{DCl}, k_{DOH}) determined using a spectrochemical channel cell at 30 °C.*

Na_2SO_4 (M)	Na_2CO_3 (M)	D_0 (mM)	Pretreat[a]	k_{DCl} (10^{-5} cm s^{-1})	k_{DOH} (10^{-5} cm s^{-1})
0.35	0.10	1.0	merc.[b]	5.1 ± 0.4	1.5 ± 0.1
0.35	0.00	1.0	merc.	1.9 ± 0.1	NA[d]
0.35	0.10	1.0	not merc.[c]	4.7 ± 0.4	0.9 ± 0.1
0.35	0.00	1.0	not merc.	0.9 ± 0.1	NA
0.35	0.10	0.5	not merc.	5.1 ± 0.5	1.3 ± 0.1
0.35	0.00	0.5	not merc.	1.3 ± 0.2	NA
0.35	0.10	3.0	not merc.	3.1 ± 0.2	0.5 ± 0.1
0.35	0.00	3.0	not merc.	0.6 ± 0.1	NA
0.00	0.10	1.0	not merc.	2.3 ± 0.3	0.2 ± 0.1
0.00	0.00	1.0	not merc.	NA[e]	NA

[a] *original cotton fabric: knitted, bleached, unmercerised*
[b] *mercerised fabric*
[c] *fabric without mercerisation*
[d] *not available*
[e] *reaction was too slow to be detectable*

Note that under neutral conditions, DOH and DCl are absorbed onto the fabric surface via physical adsorption. However, in the present of alkaline, chemical fixation of DCl is possible. Conjecturally, k_{DCl} (for alkaline conditions) and k_{DOH} (also k_{DCl} for neutral conditions) are assigned, respectively, to the rate of chemical fixation and purely physical adsorption of dye

molecules. We will verify the former when we present the results obtained from chloride detection later. As shown in Table 3, the k_{DCl}/k_{DOH} ratio increases from 3.9 to 6.2 as the dye concentration increases from 0.5 mM to 3 mM. This may imply the adsorption sites on the fabric surface are blocked more at higher concentration than the chemical fixation. Moreover, the k_{DCl} values obtained from mercerised and unmercerised fabrics in basic conditions are consistent to within 10%. Similar comparisons can be made for k_{DCl} (neutral) and k_{DOH} (basic) which are corresponding to a 2.1 and 1.7 times increase respectively. It can be proposed that mercerisation does not significantly enhance the chemical fixation but may offer additional sites at the fabric surface for dye adsorption. We will come back to this point in the subsequent discussion on the AFM results. From the above deductions, we find the dyeing process obeys a solid-liquid interfacial mechanism which is first order with respect to the surface concentration of dye. Most importantly, our results suggest the rate of this reaction is controlled by the availability of the adsorption sites for dye molecules on the fabric surface. We next turn to this argument.

As shown in table 3, the values of k_{DCl} decrease as the initial dye concentrations increase. This can be rationalised by using a surface blocking model.[23] A schematic diagram of this model is visualised in fig. 11. Here, the transport of dye molecules from the bulk solution to the fabric is considered to go through a porous surface layer. The dye molecules are first adsorbed onto the surface layer. Fixation occurs after the dye molecules penetrate to the surface of the bulk fabric. As the concentration of dye increases, more dye molecules are adsorbed onto the surface layer and therefore the dye adsorption rate decreases. Under steady state conditions, the flux of DCl passing through the surface layer is equal to the uptake flux at the fabric boundary[23,24] which was given in eqn. 9. This can be written as

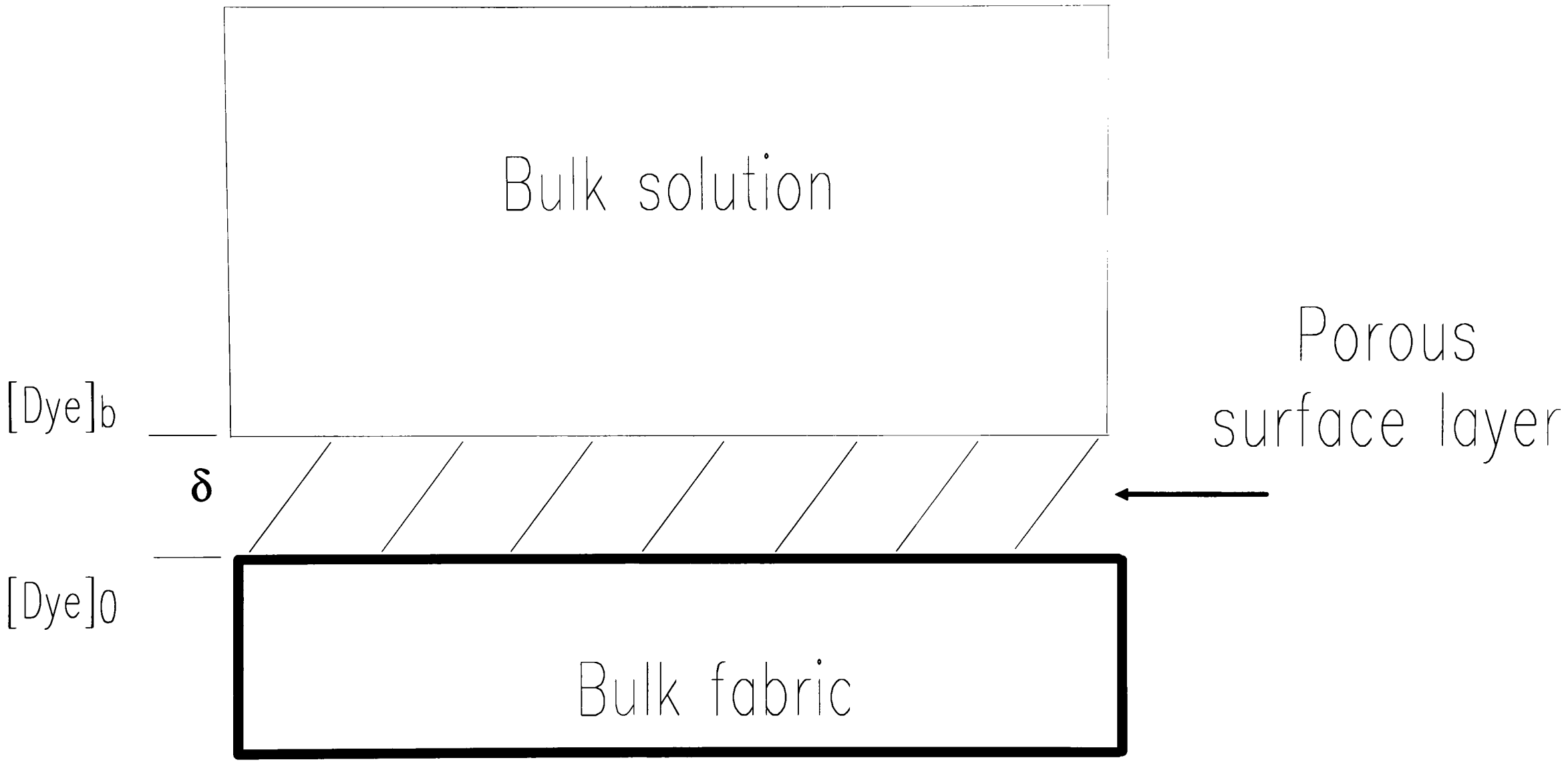


Fig. 11 A schematic diagram of surface blocking model describing the transport of reactive dye to the fabric surface.

$$J_A = \frac{D_{film}}{\delta}([A]_b - [A]_0) = k_{DCl}[A]_0 \quad (18)$$

where D_{film} denotes the diffusion coefficient through the surface layer and δ represents its thickness. We assume D_{film} is directly proportional to the amount of "free" sites,

$$D_{film} = D_{free}(1 - \theta) \quad (19)$$

where D_{free} designates the diffusion coefficient through the surface layer, if no dye molecule is adsorbed, and θ is the fraction of filled sites. Let the adsorption of dye molecule proceed via monolayer adsorption as described by the Langmuir isotherm,[25]

$$\theta = \frac{K[A]_b}{1 + K[A]_b} \quad (20)$$

where K is the constant for the adsorption/desorption process

$$1 - \theta = \frac{1}{1 + K[A]_b}, \quad put \quad k' = \frac{D_{free}}{\delta} \quad (21)$$

$$\therefore \quad J_A = \frac{k'}{1 + K[A]_b}([A]_b - [A]_0) = k_{DCl}[A]_0 \quad (22)$$

Rearranging,

$$\frac{1}{k_{DCl}} = \frac{1}{k'}\frac{[A]_0}{[A]_b - [A]_0} + \frac{K}{k'}\frac{[A]_b[A]_0}{[A]_b - [A]_0} \quad (23)$$

Similarly, we can derive the following equation for DOH,

$$\frac{1}{k_{DOH}} = \frac{1}{k'}\frac{[B]_0}{[B]_b - [B]_0} + \frac{K}{k'}\frac{[B]_b[B]_0}{[B]_b - [B]_0} \quad (24)$$

$[A]_0$ and $[B]_0$, are evaluated by using the average concentrations across the fabric surface from the BIFD calculations. The two kinetic parameters, K and k', in eqns. 23 & 24 can readily be optimised by using multiple variable regression[26] or the BFGS method as mentioned before. The results obtained from both techniques agree well with each other. Fig. 12 gives the variation of k_{DCl} with $[A]_0$ at pH 11 (0.10 M Na_2CO_3) and 7 (0.00 M Na_2CO_3). Fig. 13 depicts the variation of k_{DOH} with $[B]_0$ at pH 11. The solid lines represent the theoretical behaviour generated using eqns. 23 or 24 in conjunction with the optimised K and k' as given in table 4.

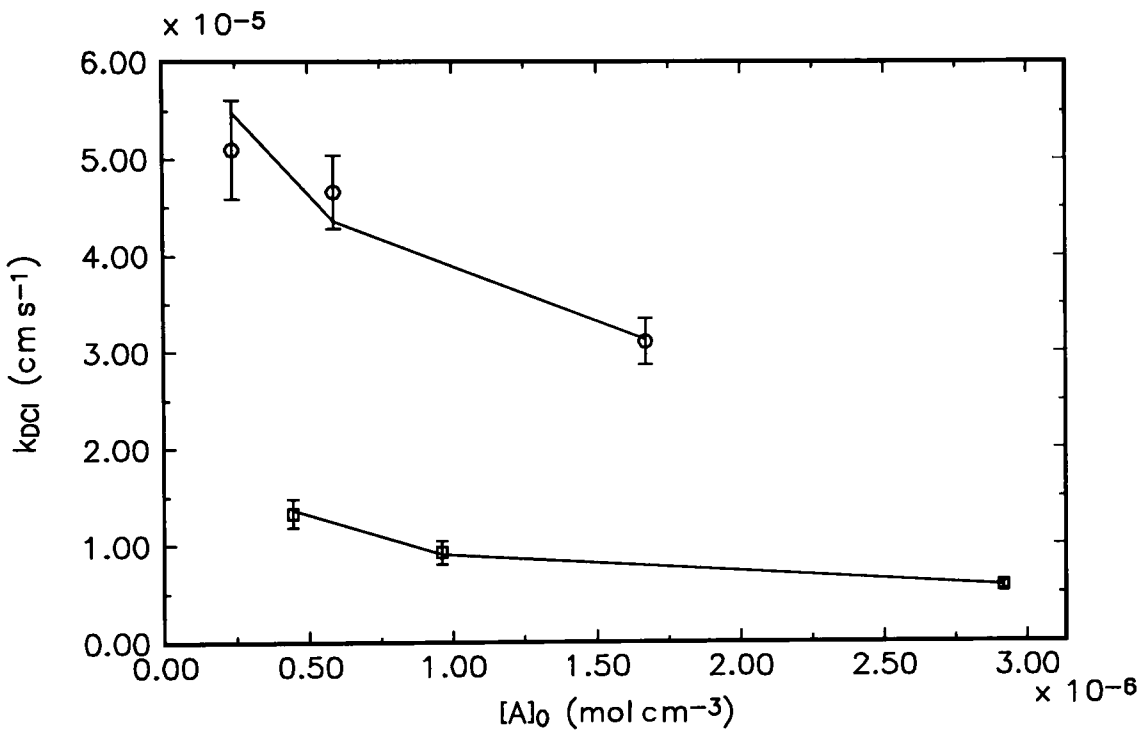


Fig. 12 Variations of k_{DCl} with $[A]_0$ at pH 11 (o) and 7 (□) in 0.35 M Na_2SO_4 solution with the solid lines representing the theoretical behaviour generated by using the surface blocking model and the optimised kinetic parameters as given in table 4.

As shown in table 4, k' are quantitatively the same for DCl and DOH. This indicates that the unreacted dye molecule and its hydrolysed form have similar transport characteristics across the porous surface layer. Note that the K value of DCl is about one order of magnitude greater than that of DOH, which suggests that the affinity for adsorption of the unreacted dye molecule to the fabric surface is greater than that of its hydrolysed form.

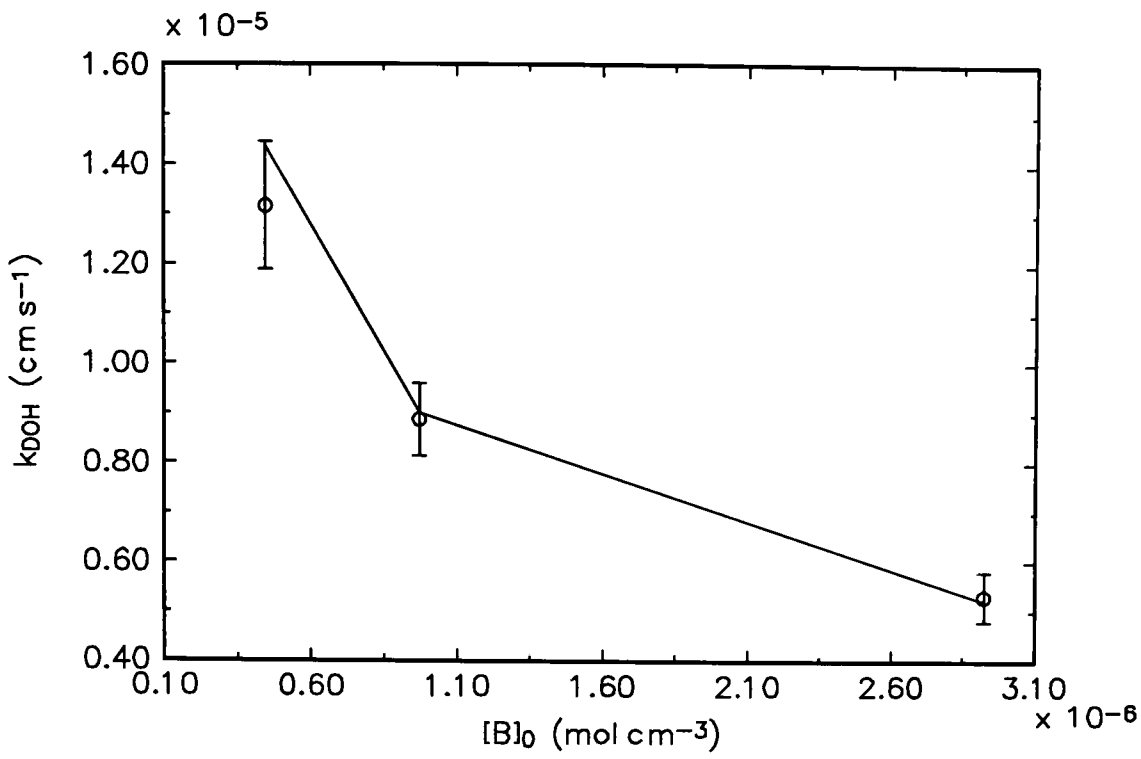


Fig. 13 Variation of k_{DOH} with $[B]_0$ at pH 11 (o) in 0.35 M Na_2SO_4 solution with the solid line representing the theoretical behaviour generated by using the surface blocking model and the optimised kinetic parameters given in table 4.

Table 4 Optimised kinetic parameters[a] (see eqns. 23 and 24) for the adsorption of DCl and DOH on unmercerised fabric based on a surface blocking model.

	DCl (pH = 11)	DCl (pH = 7)	DOH (pH = 11)
K (mol^{-1} cm^3)	2.4 x 10^4	1.2 x 10^4	1.9 x 10^3
k' (cm s^{-1})	1.7 x 10^{-4}	1.6 x 10^{-4}	1.6 x 10^{-4}

[a] $[Na_2SO_4]$ = 0.35 M

This may possibly be explained by considering the pK_a value of DOH which is around 5 to 6.[27] Under the present dyeing environment (pH ≈ 11), the hydrolysed dye molecule exists virtually entirely in the anionic form (DO^-). The electrostatic repulsion between the fabric (Cell-O^-, see below) and DO^- may lead to the lower value of K. It can be seen that at pH 11, the value of k_{DCl} decreases approximately linearly as the dye concentration increases (see fig.

12). This implies the efficiency of the dye fixation with the fabric decreases as the dye concentration increases which may be qualitatively explained by the restricted migration of dye molecules due to the aforementioned blocking effect or alternatively the fabric might become *locally* dye-saturated during experiments.

Motomura and Morita[28] previously pointed out the effect of dye hydrolysis within cellulose, in parallel with the reactive dyeing reaction using CI Reactive Orange 1 and CI Reactive Yellow 4. They examined the dye penetration by using a cylinder of rolled cellulosic film in which the neutral dye solution was first adsorbed. Then, the pH of the solution was increased to induce dye fixation. These experiments were performed under diffusion only conditions. The amount of fixed dye and hydrolysed dye were determined in an ex situ manner. From this they concluded that, within the cellulose specimen, the rate of reactive dye hydrolysis could be as high as 1.5 times the fixation rate. However, the authors may have overlooked the fact that after the rolled film was padded in the alkaline medium, the retained dye molecules could be hydrolysed, under diffusion only conditions, before they were *adsorbed* onto the cellulose surface. Plausibly, the hydrolysed dye could eventually adsorb onto the cellulose surface and may suggest their conclusions to be questionable.

Our approach is in marked contrast to this early empirical work. We take into consideration the hydrolysis in bulk solution and determine, *independently*, the uptake kinetics of the hydrolysed dye by the fabric. This permits us to measure precisely the rate of DCl adsorbed on the fabric surface. Recent work reported that the selectivity for the competing alcoholysis and hydrolysis of reactive dye (PROCION Orange MX-2R) is about 40 to 3000.[29] We anticipate that as the dye molecule *adsorbs* onto cotton fabric, the dye itself may be partially

protected from hydrolysis in terms of the steric hindrance offered by the fabric, because only part of the dye molecule will be exposed to solution. Moreover, it is well established that the [Cell-O^-]/[OH^-] ratio at pH 11 is about 28.[1] Based on the above deductions, we believe the adsorbed dye molecule *preferentially* reacts with Cell-O^- by the formation of a covalent bond.

Complementary flow cell dyeing experiments were conducted using the Orion ISE to see whether the above speculation is justified. By monitoring the chloride release from the fabric, the extent of chemical binding of dye molecules on the fabric surface may be quantified. This can be elucidated by the following equation

$$\underset{A}{DCl} + \underset{C}{Cell\text{-}O^-} \xrightarrow{k_{DCl}} \underset{E}{Cell\text{-}O\text{-}D} + Cl^- \qquad (25)$$

Here, we assume DCl is hydrolysed (eqn. 1) in parallel to eqn. 25 with the rate constant k_{hyd} given in table 1. In this treatment, the boundary conditions for DCl are the same as before (eqns. 9 & 10). As for Cl^-, the following equation is adopted at the fabric surface

$$D_A \frac{\partial [A]}{\partial y} = -D_E \frac{\partial [E]}{\partial y} \qquad (26)$$

with matrix elements b_1 and d_1

$$E: \quad b_1 = \lambda^E{}_1 + 1, \quad d_1 = g^E{}_{1,k} + \lambda^E{}_1 \left[\frac{D_A (g^A{}_{1,k+1} - g^A{}_{0,k+1})}{D_E} \right] \qquad (27)$$

The formula for the evaluation of the average chloride concentration above the detector electrode surface is given in chapter 2. A diffusion coefficient of 2.4 x 10^{-5} cm^2 s^{-1} for the chloride ion[30] was used for modelling purposes. Dyeing experiments were performed using an initial dye concentration of 1.0 mM, sodium carbonate concentration of 0.1 M and sodium

sulphate concentration of 0.35 M. Fig. 14 shows a plot of the experimental and theoretical chloride concentration as a function of solution flow rate. The optimised dyeing rate constant was found to be 6.1 ($\pm$ 3.3) x 10^{-5} cm s^{-1}. Within experimental uncertainties, this quantity is comparable to the dyeing rate constant (k_{DCl}), 4.7 ($\pm$ 0.4) x 10^{-5} cm s^{-1} measured using the proposed spectroscopic method (see table 3). This indicates that the dyeing rate constants (k_{DCl}), as determined spectroscopically, may directly reflect the fixation rate of the dye molecules onto the fabric.

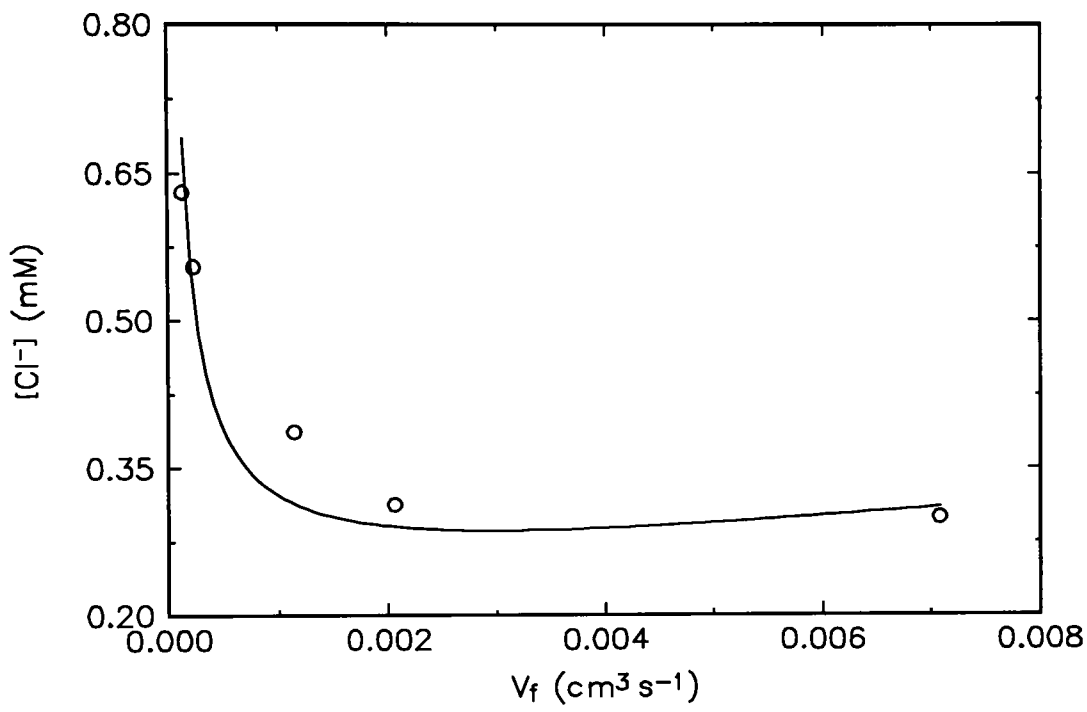
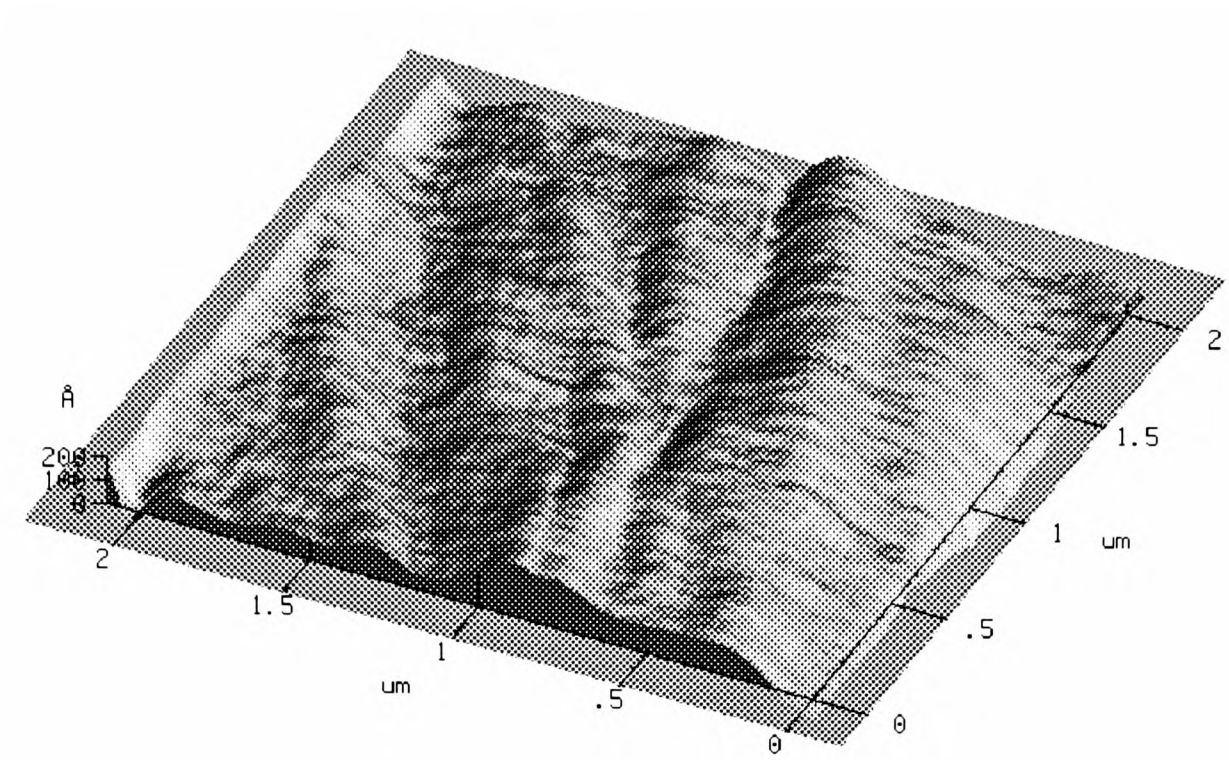


Fig. 14 The flow rate dependence of the downstream chloride concentration measured using a commercial ISE for the dyeing of an unmercerised fabric with initial dye concentration (DCl) of 1.0 mM in the presence of 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4: (o) experimental data and (——) theoretical behaviour predicted using the interfacial reaction model and the optimised rate constant mentioned in the text. Cell geometry: x_c = 2.003 cm, x_g = 0.211 cm, d = 1.148 cm and 2h = 0.093 cm.

We next consider the morphological change induced by mercerisation. Cotton fibres before and after the mercerisation process were imaged in air using an atomic force microscope and are depicted, respectively, in figs. 15 and 16. It can be seen that the mercerised fibres show a more disordered surface in comparison with the unmercerised ones. This observation is in

a



b

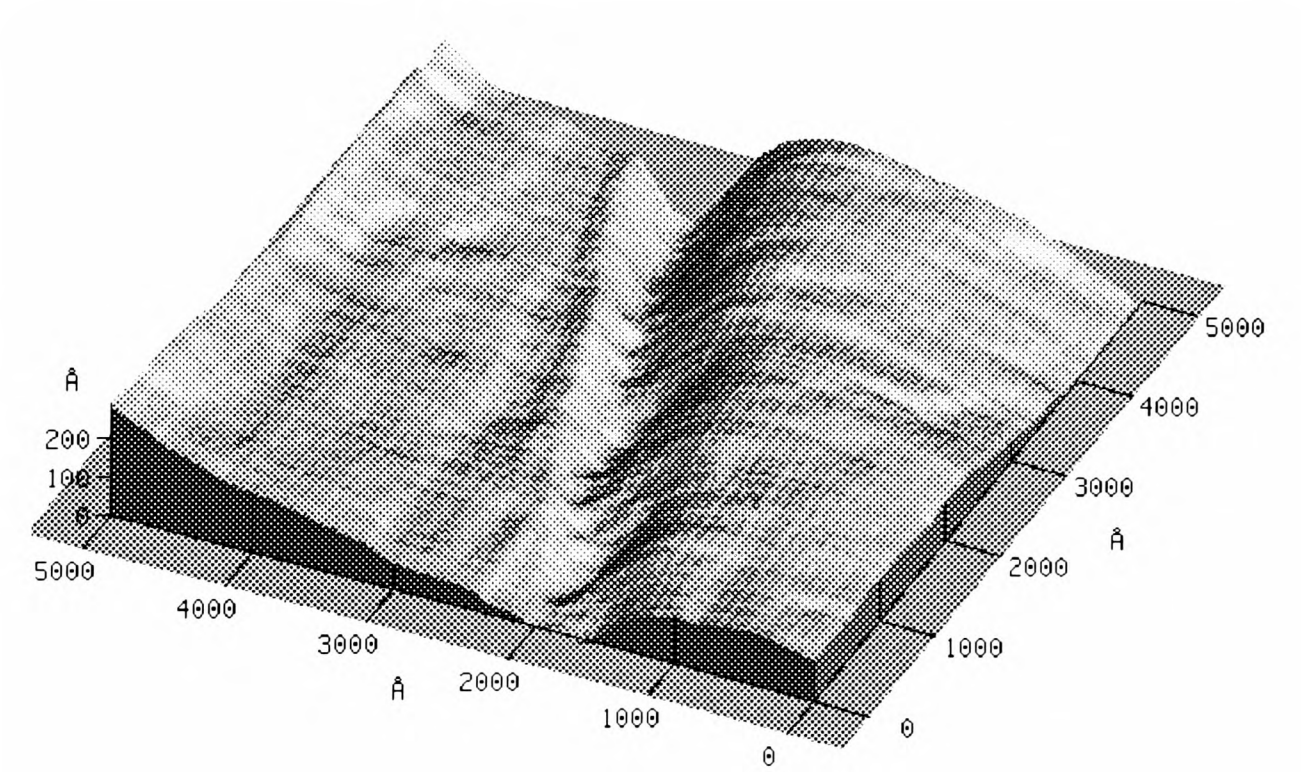
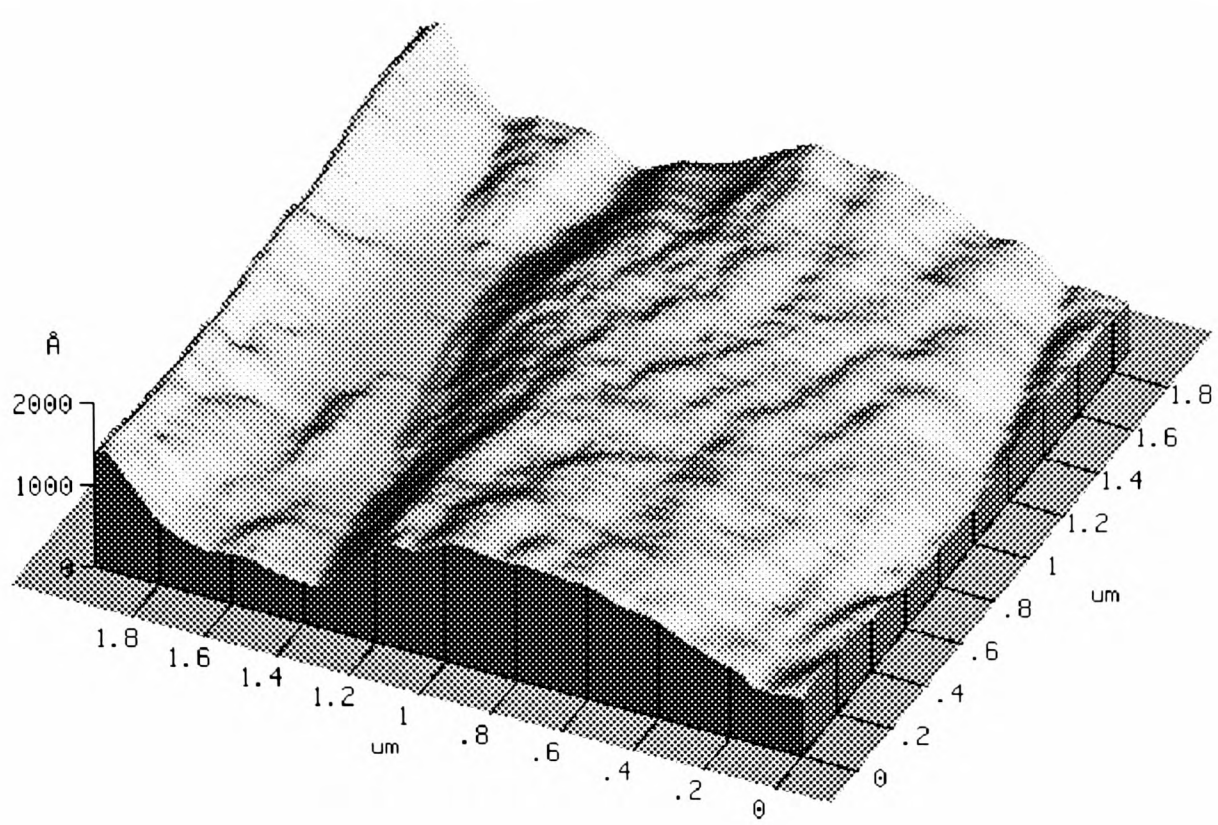


Fig. 15 Ex situ AFM micrographs of unmercerised cotton fibre. (a) 2.0 x 2.0 μm, (b) 0.5 x 0.5 μm.

a



b

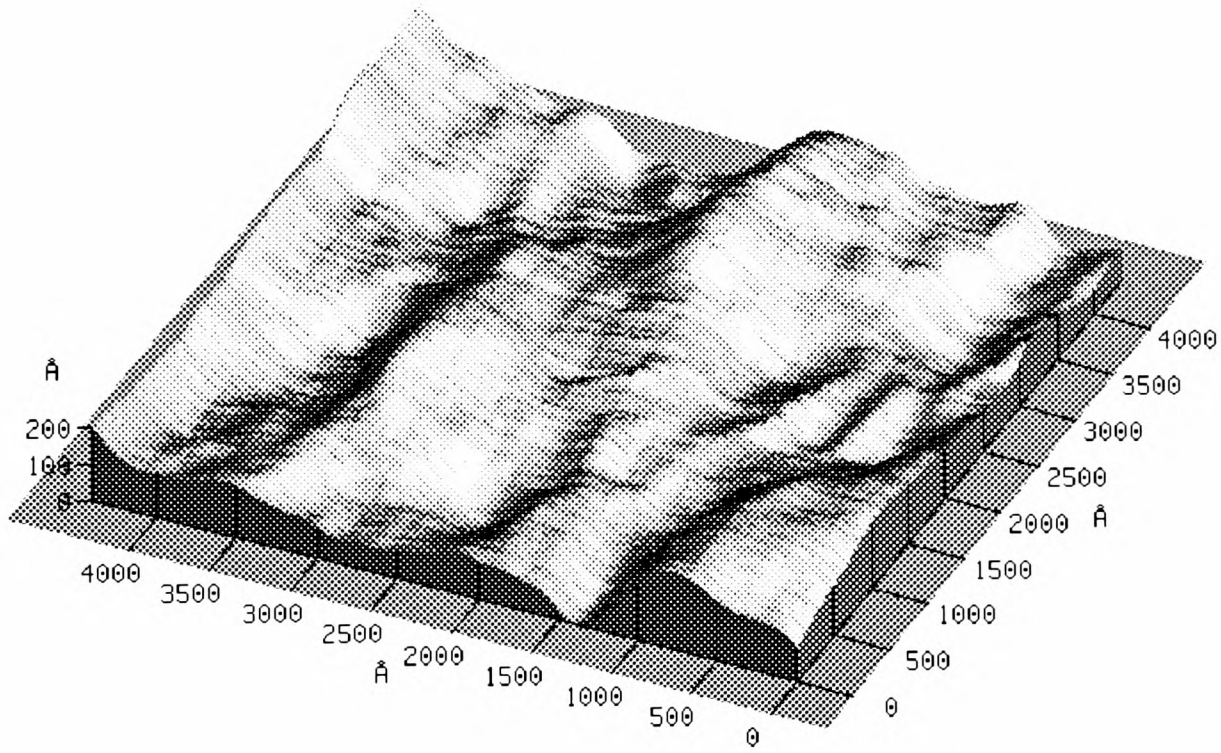
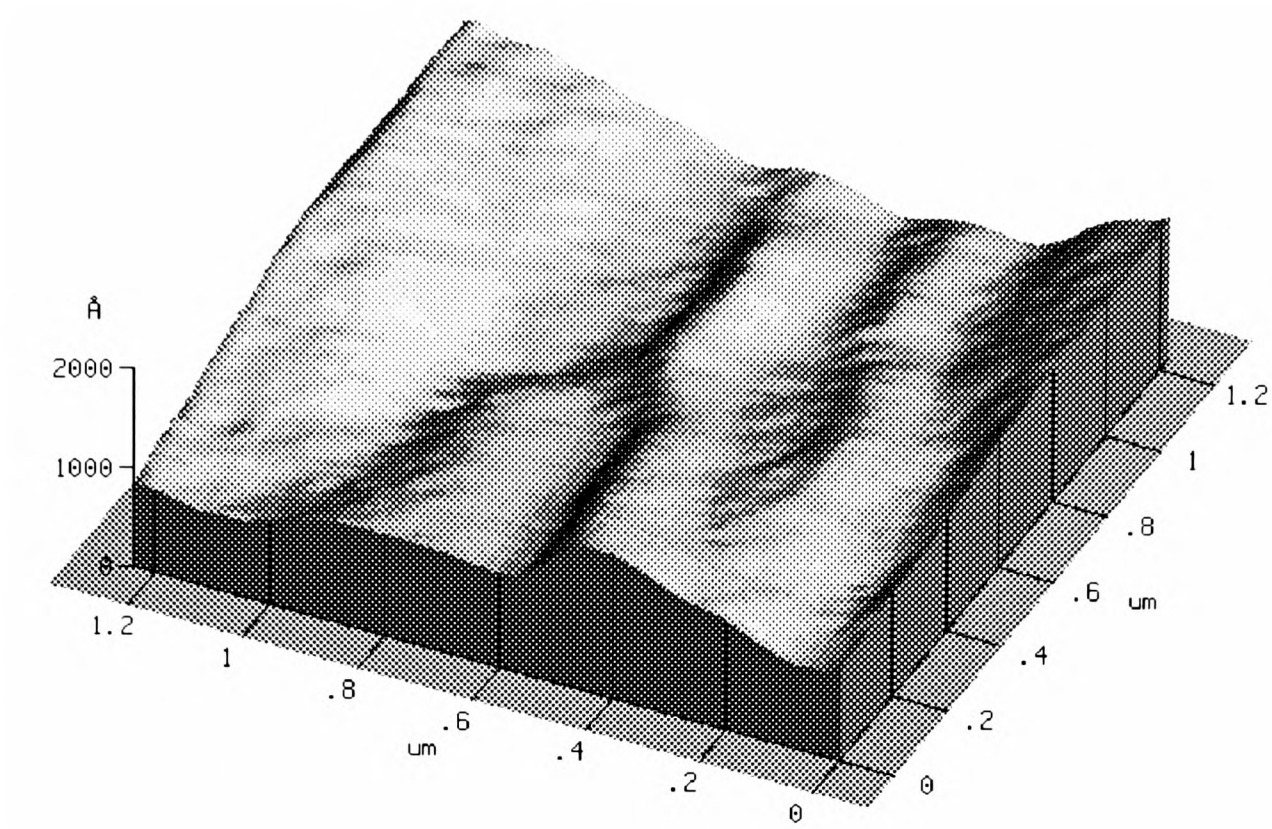


Fig. 16 Ex situ AFM micrographs of mercerised cotton fibre. (a) 1.8 x 1.8 μm, (b) 0.4 x 0.4 μm.

a



b

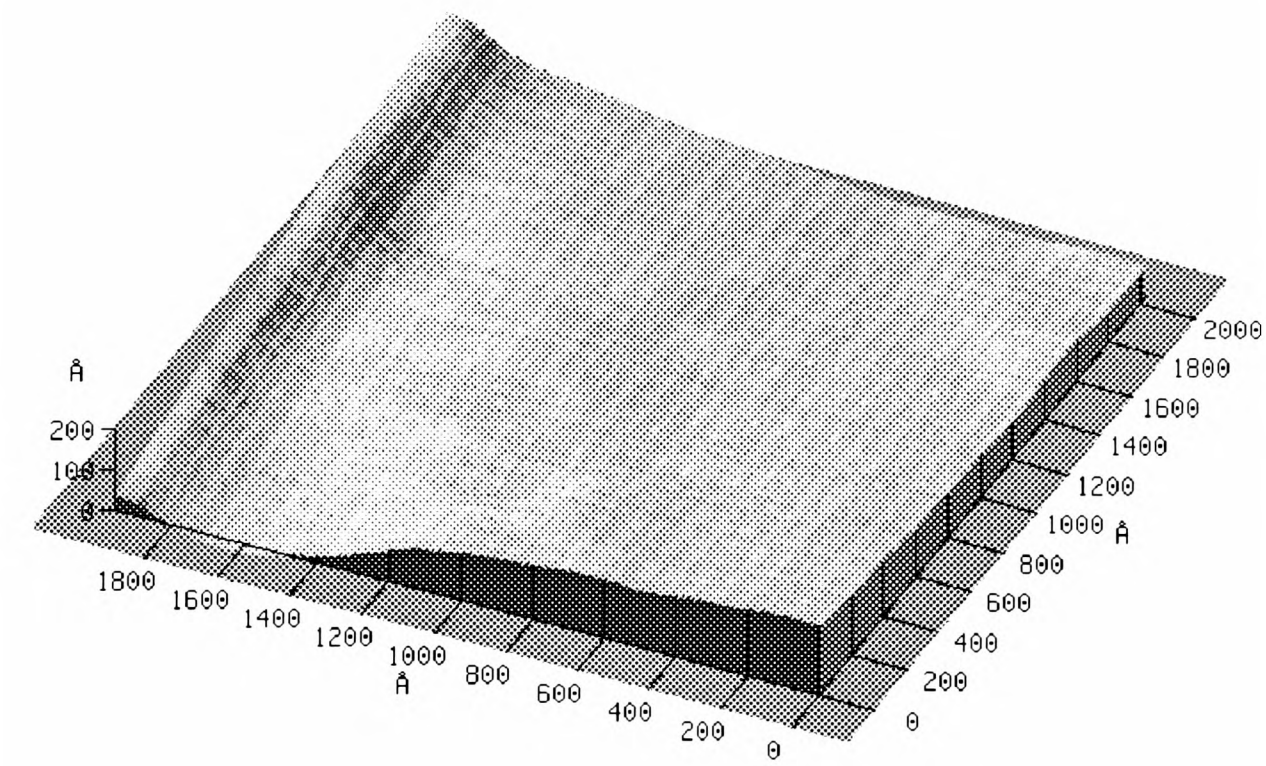
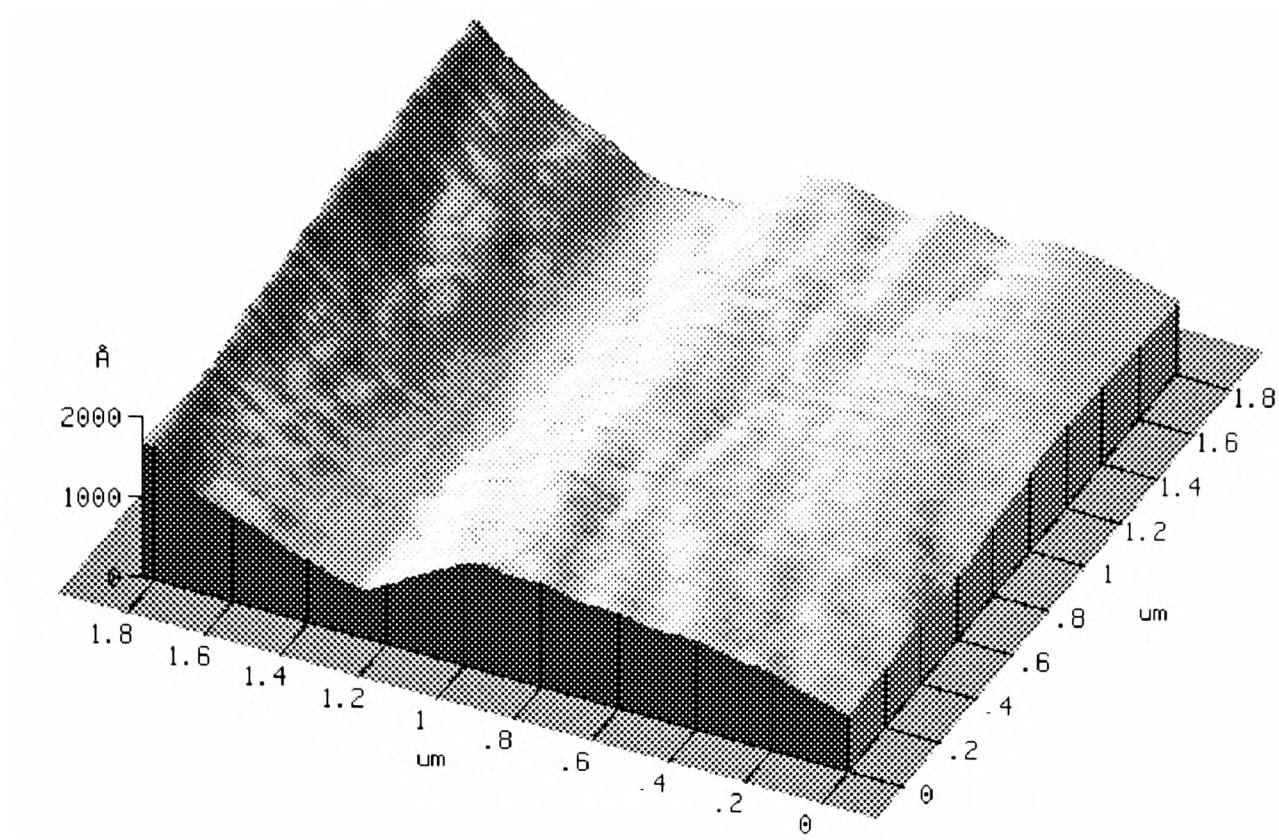


Fig. 17 Ex situ AFM micrographs of unmercerised cotton fibre dyed in 1.0 mM DCl, 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4 solution at a flow rate of 4.4 x 10^{-2} cm^3 s^{-1}. (a) 1.2 x 1.2 μm, (b) 0.2 x 0.2 μm.

a



b

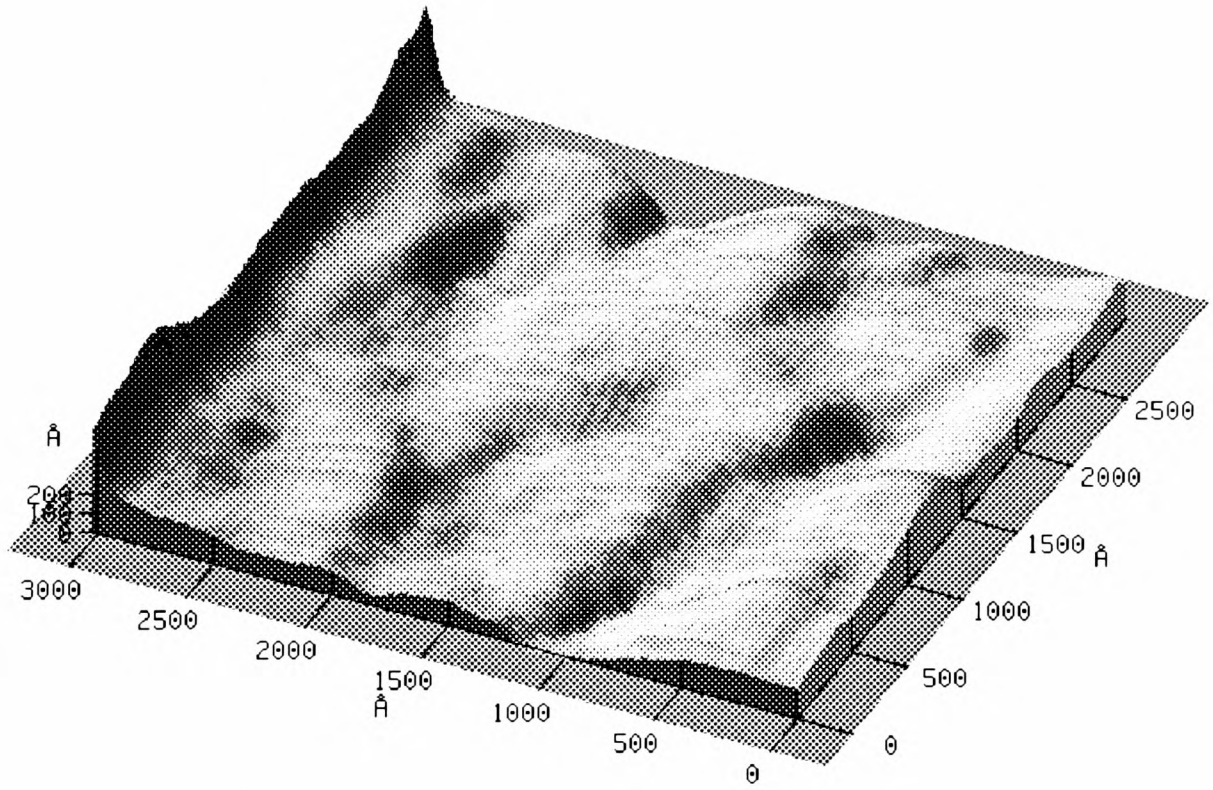


Fig. 18 Ex situ AFM micrographs of mercerised cotton fibre dyed in 1.0 mM DCl, 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4 solution at a flow rate of 4.4 x 10^{-2} cm^3 s^{-1}. (a) 1.8x 1.8 μm, (b) 0.3 x 0.3 μm.

agreement with literature reports.[2,15,31] It is anticipated that the disorder of the mercerised surface provides more accessible sites for dye fixation, this is consistent with the observed 10% increase in k_{DCl} (see table 3). Figs. 17 and 18 depict, respectively, the AFM images of unmercerised and mercerised fibres dyed in a channel flow cell using 1.0 mM DCl, 0.10 M Na_2CO_3 and 0.35 M Na_2SO_4 solution, at a flow rate of 4.4 x 10^{-2} cm^3 s^{-1}. It can be seen that the dyed fibres exhibit similar morphological features to the undyed samples (figs. 17 & 18).

5. Concluding remarks

A universal method based on a spectrochemical channel flow cell has been developed to study dyeing processes. The proposed method was exemplified by the reactive dyeing of PROCION Blue MX-R on knitted cotton fabrics. The reactive dyeing is complicated by the simultaneous hydrolysis of the dye molecules and the physical binding of the hydrolysed form onto the fabric. All these processes were taken into account in the reaction model evolved. In particular, the kinetic results indicate that the dye fixation to the fabric is controlled by a solid-liquid interfacial process which is first order with respect to the surface concentration of dye. However, the rate of this reaction is governed by the availability sites for the adsorption of dye molecules on the fabric surface. It is found that mercerisation pretreatment enhances the dye uptake rate whilst the presence of supporting electrolyte in a high pH environment are indispensable in the enhancement of the dye uptake rate. Microscopic studies suggest that the mercerisation pretreatment provides a disordered fibre surface which may offer additional sites for dye adsorption.

6. References

1. Kroschwitz J.I.; Howe-Grant M. (eds.) *Kirk-Othmer Encyclopedia of Chemical Technology*; *4th Ed., Vol. 8*; Wiley; New York, **1993**, .

2. Madaras G.W.; Parish G.J.; Shore J. *Batchwise Dyeing of Woven Cellulosic Fabrics*; Society of Dyers and Colourists; Bradford, **1993**.

3. Zollinger H. *J. Soc. Dyers and Colourists*, **1994**, *110*, 209.

4. Rusznak I.; Zobor J.; Levai G. *Dyes and Pigments*, **1981**, *2*, 285.

5. Sada E.; Kumazawa H. *J. Appl. Polym. Sci.*, **1982**, *27*, 2987.

6. Motomura H.; Morita Z. *J. Appl. Polym. Sci.*, **1977**, *21*, 487.

7. Motomura H.; Morita Z. *Bull. Chem. Soc. Japan*, **1978**, *51*, 1332.

8. Johnson A. (ed) *The Theory of Coloration of Textiles*; *Chap. 5*; Society of Dyers and Colourists; Bradford, **1989**.

9. Compton R.G.; Harding M.S.; Atherton J.H.; Brennan C.M. *J. Phys. Chem.*, **1993**, *97*, 4677.

10. Macpherson J.V.; Unwin P.R. *Progress Reaction Kinetics*, **1995**, *20*, 185.

11. Compton R.G.; Wilson M. *J. Appl. Electrochem.*, **1990**, *20*, 793,

12. Compton R.G.; Unwin P.R.; Wilson M. *Chem. & Ind.*, **1990**, *7*, 234.

13. Tam K.Y.; Wang. R.L.; Lee C.W.; Compton R.G. *Electroanalysis*, in press.

14. Wang R.L.; Tam K.Y.; Marken F.; Compton R.G. *Electroanalysis*, in press.

15. Pashley P. *J. Soc. Dyers and Colourists*, **1993**, *109*, 379.

16. Compton R.G.; Pilkington M.B.G.; Stearn G.M. *J. Chem. Soc. Faraday Trans. I*, **1988**, *84*, 2155.

17. Stephenson G. *Mathematical Methods for Science Students*; Longman; Harlow, **1973**.

18. Tam K.Y.; Chau F.T. *J. Auto. Chem.*, **1992**, *14*, 157.

19. Walsh G.R. *Methods of Optimization*; Wiley; London, **1975**.

20. Yuan Y. *IMA J. Numerical Analysis*, **1991**, *11*, 325.

21. Smith E.R. *Part II Thesis*; Physical and Theoretical Chemistry Laboratory; Oxford, **1994**.

22. Wilke C.R.; Chang P. *A.I.Ch.E.*, **1955**, *1*, 264.

23. Gooding J.J. *D.Phil. Thesis*; Oxford, **1994**.

24. Gooding J.J.; Compton R.G.; Atherton J.H.; Brennan C.M. *J. Colloid Interface Sci.*, **1996**, *180*, 605.

25. Atkins P.W. *Physical Chemistry*, Oxford University Press; Oxford, **1986**.

26. Freund J.E. *Modern Elementary Statistics*; 7th ed.; Prentice-Hall; Englewood Cliffs, **1988**.

27. Brennan C.M. unpublished results.

28. Motomura H.; Morita Z. *J. Appl. Polym. Sci.*, **1979**, *24*, 1747.

29. Bentley T.W.; Ratcliff J.; Renfrew A.H.M.; Taylor J.A. *J. Soc. Dyers and Colourists*, **1995**, *111*, 289.

30. Markova V.G.; Goncharov V.V.; Yashkichev V.J. *Russ. J. Phys. Chem.*, **1974**, *48*, 1266.

31. Nevell T.P.; Zeronian S.H. (ed.) *Cellulose Chemistry and Its Application*; Ellis Horwood Ltd., West Sussex, **1985**.

Appendix One

The Thomas Algorithm

The Thomas algorithm is an efficient method for the solution of tridiagonal matrix systems. The matrix equation to be solved is written as

$$\{d\} = \{T\}\{u\}$$

The algorithm operates by factorising the tridiagonal matrix {T} into two bidiagonal matrices $\{T_L\}$ and $\{T_U\}$:

$$\{T\} = \{T_L\}\{T_U\}$$

A solution is then found for the vector {f} in

$$\{T_L\}\{f\} = \{d\}$$

and {f} is used to give a final solution

$$\{T_U\}\{u\} = \{f\}$$

Since

$$\{T_L\}^{-1}\{d\} = \{f\} \qquad \{T_U\}\{u\} = \{T_L\}^{-1}\{d\}$$

and thus

$$(\{T_L\}\{T_U\})\{u\} = \{d\}$$

The matrix {T} of (J-1) x (J-1) elements can be written as

$$\{T\} = \begin{bmatrix} b_1 & c_1 & 0 & & & & & \\ a_2 & b_2 & c_2 & 0 & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & 0 & a_j & b_j & c_j & 0 & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & 0 & a_{J-2} & b_{J-2} & c_{J-2} \\ & & & & & & 0 & a_{J-1} & b_{J-1} \end{bmatrix}$$

On factorisation {T} takes the form

$$\{T\} = \{T_L\}\{T_U\} = \begin{bmatrix} \alpha_1 & & & & & 0 \\ a_2 & \alpha_2 & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & a_{J-2} & \alpha_{J-2} & \\ 0 & & & & a_{J-1} & \alpha_{J-1} \end{bmatrix} \begin{bmatrix} 1 & \beta_1 & & & & 0 \\ & 1 & \beta_2 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & 1 & \beta_{J-2} \\ 0 & & & & & 1 \end{bmatrix}$$

where α_j and β_j are to be determined. By equating the left and right hand sides of the equation $\{T\} = \{T_L\}\{T_U\}$, the following relations are obtained

$$\alpha_1 = b_1$$

$$\beta_1 = \frac{c_1}{\alpha_1}$$

$$\alpha_j = b_j - a_j \beta_{j-1} \qquad j = 2,3 \ldots J-1$$

$$\beta_j = \frac{c_j}{\alpha_j} \qquad j = 2,3 \ldots J-2 \quad (\alpha_j \neq 0)$$

After obtaining α_j and β_j, the equation for {f} is solved

$$\{T_L\}\{f\} = \{d\}$$

where the elements f_j of {f} are given by

$$f_1 = \frac{d_1}{\alpha_1}$$

$$f_j = \frac{(d_j - a_j f_{j-1})}{\alpha_j} \qquad j = 2,3...J-1$$

{f} is then used to determine the elements of u_j of {u}

$$u_{J-1} = f_{J-1}$$

$$u_j = f_j - \beta_j u_{j+1} \qquad j = 1,2...J-2$$

By this procedure the matrix equation {T} {u} = {d} is solved for {u}.

Appendix Two

Computer programs

```
      program calbg

c     Program to calculate the background resistance of supporting
c     electrolyte using SPICE. The program requires the cell geometry,
c     conductivity of the electrolyte and calc. Rx, Ry, Cx & Cy.
c
c     Declaration
c
c     Physical Parameters
c
c     2h - height of channel, cm
c     xe - electrode length, cm
c     we - width of electrode, cm
c     cond - conductivity of supporting electrolyte

      integer lendir,lenbnam,l,sflag

      double precision rbg(4),cond,conc,er,h,xe,we

      character*4   endstr
      character*9   bname
      character*40  dire
      character*55  fname,fname1,fname2,fname3
      character*133 buff
      character*150 spargum,othargum

c     Physical parameters

      h=0.094/2.0
      xe=0.396
      we=0.394
      cond=144.9
      conc=0.1886e-3

c     Cal. background R & C

      rx=(xe/(2.0*h*we))/(cond*conc)
      ry=((8.0*h)/(we*xe))/(cond*conc)
      er=78.54*8.85e-14
      cx=2.0*er*we*h/xe
      cy=xe*er*we/(8.0*h)

      print*, 'Rx = ', rx
      print*, 'Ry = ', ry
      print*, 'Cx = ', cx
      print*, 'Cy = ', cy

c     Establish the circuit file names

      dire='/home/joule/rgc/tam/Spice/'
      lendir=26
      bname='calbg.'
      lenbnam=6

      fname2(1:lendir)=dire
      fname2(lendir+1:lendir+lenbnam+1)=bname
      j=lendir+lenbnam
      fname3=fname2
      fname1=dire
      fname2(j+1:j+3)='cir'
      fname3(j+1:j+3)='out'
      fname1(lendir+1:lendir+11)='netbase.cir'

c     Create a background circuit file with the designated Rx, Ry, Cx, Cy

      endstr='.END'
      open(unit=9, file=fname1, status='old')
      open(unit=10, file=fname2, status='unknown')

      sflag=1
200   read(9,'(A)', err=220) buff
      l=133
210   continue
      if ((l.gt.1).and.(buff(l:l).eq.' ')) then
          l=l-1
          goto 210
      end if
      if (buff(1:4).eq.endstr) then
           sflag=0
      end if
      if ((sflag.eq.1).and.(buff(1:6).eq.'R1 1 2')) then
            write(10,500) 'R1', 1, 2, ry
            print 500, 'R1', 1, 2, ry
       else if ((sflag.eq.1).and.(buff(1:6).eq.'R2 3 4')) then
            write(10,500) 'R2', 3, 4, ry
            print 500, 'R2', 3, 4, ry
       else if ((sflag.eq.1).and.(buff(1:6).eq.'R3 2 3')) then
            write(10,500) 'R3', 2, 3, rx
            print 500, 'R3', 2, 3, rx
       else if ((sflag.eq.1).and.(buff(1:6).eq.'R4 1 4')) then
            write(10,500) 'R4', 1, 4, rx
            print 500, 'R4', 1, 4, rx
       else if ((sflag.eq.1).and.(buff(1:6).eq.'C1 1 2')) then
            write(10,510) 'C1', 1, 2, cy*(1e12), 'pF'
            print 510, 'C1', 1, 2, cy*(1e12), 'pF'
       else if ((sflag.eq.1).and.(buff(1:6).eq.'C2 3 4')) then
            write(10,510) 'C2', 3, 4, cy*(1e12), 'pF'
            print 510, 'C2', 3, 4, cy*(1e12), 'pF'
       else if ((sflag.eq.1).and.(buff(1:6).eq.'C3 2 3')) then
            write(10,510) 'C3', 2, 3, cx*(1e12), 'pF'
            print 510, 'C3', 2, 3, cx*(1e12), 'pF'
       else if ((sflag.eq.1).and.(buff(1:6).eq.'C4 1 4')) then
            write(10,510) 'C4', 1, 4, cx*(1e12), 'pF'
            print 510, 'C4', 1, 4, cx*(1e12), 'pF'
       else
            write(10,'(A)') buff(1:l)
      end if
      goto 200
220   write(10,'(A)') endstr
      close(unit=9)
      close(unit=10)

c     Remove the SPICE o/p circuit files first

      othargum='rm -f'
      othargum(7:j+10)=fname3
      call system(othargum)

c     Invoke SPICE via SYSTEM call to calc. background resistance

      spargum='/home/joule/bac/bin/spice -b '
      i=29+j+3
      spargum(30:i+1)=fname2
      spargum(i+1:i+4)=' > '
      spargum(i+5:i+j+8)=fname3
      call system(spargum)
```

```
c     Extract numerical part from SPICE o/p file

      fname2(j+1:j+3)='   '
      fname1=fname(fname2,j,1)

      call fzrealmin(fname3, fname1, rbg)
      print*, 'Background resistance (min. Z(imag))        = ', rbg(2)
      print*, 'Background resistance (min. 2nd derivative) = ', rbg(1)
      print*, 'Background resistance (reg, 200-40000Hz)    = ', rbg(3)
      print*, '   correlation factor = ', rbg(4)

500   format(a2,i2,i2,f12.2)
510   format(a2,i2,i2,f8.4,a2)

      end

      character*55 function fname(name,in,extn)

c     Function to combine file extension (integer) with file name (char)

      integer extn, i, in, j, k, l, l1, l2

      character*55 name
      character*3 ext

      fname(1:in)=name
      ext='   '
      k=1
      l=1
      l2=extn
      do i=3, 1, -1
         l1=int(10**(i-1))-1
         if ((l2.gt.l1).or.(l.eq.0)) then
            j=l2/(l1+1)
            l2=l2-j*(l1+1)
            l=1
            if ((l2.lt.10).and.(i.eq.3)) then
               l=0
            end if
            if ((l2.eq.0).and.(i.le.2)) then
               l=0
            end if
            j=j+48
            ext(k:k)=char(j)
            k=k+1
         end if
      end do
      i=in+1
      j=i+k-1
      fname(i:j)=ext
      do i=j+1, 55
         fname(i:i)=' '
      end do

      return
      end

      subroutine fzrealmin(spiceout, zplot,
     _                                zmin)

c     Extract numerical part of SPICE o/p file & locate Z' at min Z''

      character*255 buff
      character*55 spiceout, zplot
      double precision zmin(3),dummy,zreal(1000),zimag(1000),yp(1000),
     _          ypp(1000),frx(1000),fry(1000),a,b,r,freq(1000)
      integer ng, i, fsmax, fsmin

      open(unit=9, file=spiceout, status='old')
      open(unit=10, file=zplot, status='unknown')

90    read(9, '(A)', err=92) buff
      if ((ichar(buff(1:1)).le.57).and.(ichar(buff(1:1)).ge.48)) then
         write(10, '(A)') buff
      end if
      goto 90
92    close(unit=9)
      close(unit=10)

      open(unit=9, file=zplot, status='old')

      ng=1
94    read(9,*, err=96), dummy,freq(ng),dummy,zreal(ng),zimag(ng)
         zreal(ng)=abs(zreal(ng))
         zimag(ng)=abs(zimag(ng))
         ng=ng+1
      goto 94
96    close(unit=9)
      ng=ng-1

c     Locate the Z'

c     1st derivative

      do i=2, ng
         yp(i)=(zimag(i)-zimag(i-1))/(zreal(i)-zreal(i-1))
      end do

c     2nd derivative

      do i=ng-1, 2, -1
         ypp(i)=(yp(i+1)-yp(i))/(zreal(i+1)-zreal(i))
      end do

c     find Z(real) at min. 2nd derivative

      i=1
98    i=i+1
      if (ypp(i+1).gt.ypp(i)) then
         goto 99
      end if
      goto 98
99    continue
      zmin(1)=zreal(i)
      fsmax=i

c     find Z(real) at min. Z(imag)

      i=0
109   i=i+1
      if (zimag(i).lt.zimag(i+1)) then
         goto 119
      end if
      goto 109
119   continue
      zmin(2)=zreal(i)


c     selected data between 200-40000Hz

      i=0
50    i=i+1
      if (freq(i).ge.200.0) then
         goto 52
      end if
      goto 50
52    fsmin=i

54    i=i+1
      if (freq(i).ge.40000.0) then
         goto 56
      end if
      goto 54
56    fsmax=i

      do i=fsmin, fsmax
         ng=i-fsmin+1
         frx(ng)=zreal(i)
```

```
      fry(ng)=zimag(i)
    end do
    ng=abs(fsmin-fsmax)+1

    call lsfit(frx, fry, ng, a, b, r)

    zmin(3)=-1.0*b/a
    zmin(4)=r

    return
    end


    subroutine lsfit(x, y, nfq,
   -                 a, b, r)

c   compute least squares straight line and the
c   regression coefficient

    integer  madat, nfq, j
    parameter(madat=1000)
    double precision x(madat), y(madat)
    double precision a, b, r, sumx, sumy, sumx2, sumy2, sumxy
    double precision z1, z2, z3

c   reset variables

    sumx = 0.0
    sumy = 0.0
    sumx2 = 0.0
    sumy2 = 0.0
    sumxy  = 0.0

c   compute the summations

    do j=1, nfq
            sumx = sumx + x(j)
            sumy = sumy + y(j)
            sumx2 = sumx2 + x(j) * x(j)
            sumy2 = sumy2 + y(j) * y(j)
            sumxy = sumxy + x(j) * y(j)
    end do

c   compute the coefficients

c   least squares straight line

    z1 = sumxy - sumx*sumy / nfq
    z2 = sumx2 - sumx*sumx / nfq
    a = z1 / z2
    b = (sumy - a * sumx) / nfq

c   regression coefficient

    z3 = sqrt((sumx2 - sumx*sumx / nfq) * (sumy2 - sumy*sumy / nfq))
    r = z1 / z3

    return
    end
```

SPICE program: netbase.cir

```
CONDUCTIVITY A.C. :  $
.WIDTH  IN=132 OUT=133
IDR 1 0 0 AC 1 0
VS 1 2 0
.AC DEC 10 1 1E8
.PRINT AC VR(2) VI(2)
*
*     2-----R3-----3
*     !            !
*     R1           R2
*     !            !
*     1-----R4-----4
*
*     2-----C3-----3
*     !            !
*     C1           C2
*     !            !
*     1-----C4-----4
.SUBCKT SLR 1 2 3 4
R1 1 2 81066.88
R2 3 4 81066.88
R3 2 3 430505.66
R4 1 4 430505.66
*capacitance calculated from dielectric const. of H20 = 78.54
C1 1 2 3.1471pF
C2 3 4 3.1471pF
C3 2 3 0.5926pF
C4 1 4 0.5926pF
.ENDS
.SUBCKT ERC 1 2
*resistance and capacitance of electrode per node
*Comment out R1 if no Faradaic resistance
*R1 1 2 0.1
*6uF/cm2 capacitors
C1 1 2 0.023226UF
.ENDS
.SUBCKT STR 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
+ 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
X1 1 2 23 22 SLR
X2 2 3 24 23 SLR
X3 3 4 25 24 SLR
X4 4 5 26 25 SLR
X5 5 6 27 26 SLR
X6 6 7 28 27 SLR
X7 7 8 29 28 SLR
X8 8 9 30 29 SLR
X9 9 10 31 30 SLR
X10 10 11 32 31 SLR
X11 11 12 33 32 SLR
X12 12 13 34 33 SLR
X13 13 14 35 34 SLR
X14 14 15 36 35 SLR
X15 15 16 37 36 SLR
X16 16 17 38 37 SLR
X17 17 18 39 38 SLR
X18 18 19 40 39 SLR
X19 19 20 41 40 SLR
X20 20 21 42 41 SLR
.ENDS
.SUBCKT EL 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
+22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
XR1 1 2 ERC
XR2 1 3 ERC
```

```
XR3 1 4 ERC
XR4 1 5 ERC
XR5 1 6 ERC
XR6 1 7 ERC
XR7 1 8 ERC
XR8 1 9 ERC
XR9 1 10 ERC
XR10 1 11 ERC
XR11 1 12 ERC
XR12 1 13 ERC
XR13 1 14 ERC
XR14 1 15 ERC
XR15 1 16 ERC
XR16 1 17 ERC
XR17 1 18 ERC
XR18 1 19 ERC
XR19 1 20 ERC
XR20 1 21 ERC
XR21 1 22 ERC
XR22 1 23 ERC
XR23 1 24 ERC
XR24 1 25 ERC
XR25 1 26 ERC
XR26 1 27 ERC
XR27 1 28 ERC
XR28 1 29 ERC
XR29 1 30 ERC
XR30 1 31 ERC
XR31 1 32 ERC
XR32 1 33 ERC
XR33 1 34 ERC
XR34 1 35 ERC
XR35 1 36 ERC
XR36 1 37 ERC
XR37 1 38 ERC
XR38 1 39 ERC
XR39 1 40 ERC
XR40 1 41 ERC
.ENDS
*actual circuit
*Earthing resistors needed if no Faradaic resistance in ERC
RGND 0 5100 1E10
RTOP 0 2 1E10
*net ends
X1 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020
+ 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 STR
X2 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120
+ 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 STR
X3 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220
+ 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 STR
X4 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320
+ 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 STR
X5 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420
+ 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 STR
X6 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520
+ 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 STR
X7 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620
+ 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 STR
X8 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720
+ 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 STR
X9 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820
+ 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 STR
X10 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920
+ 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 STR
X11 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020
+ 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 STR
X12 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120
```

+ 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 STR
X13 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220
+ 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 STR
X14 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320
+ 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 STR
X15 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420
+ 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 STR
X16 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520
+ 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 STR
X17 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620
+ 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 STR
X18 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720
+ 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820 STR
X19 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820
+ 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920 STR
X20 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920
+ 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 STR
*electrode starts on 3100
X21 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020
+ 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119 3120 STR
X22 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119 3120
+ 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 STR
X23 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220
+ 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320 STR
X24 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320
+ 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 STR
X25 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420
+ 3500 3501 3502 3503 3504 3505 3506 3507 3508 3509 3510 3511 3512 3513 3514 3515 3516 3517 3518 3519 3520 STR
X26 3500 3501 3502 3503 3504 3505 3506 3507 3508 3509 3510 3511 3512 3513 3514 3515 3516 3517 3518 3519 3520
+ 3600 3601 3602 3603 3604 3605 3606 3607 3608 3609 3610 3611 3612 3613 3614 3615 3616 3617 3618 3619 3620 STR
X27 3600 3601 3602 3603 3604 3605 3606 3607 3608 3609 3610 3611 3612 3613 3614 3615 3616 3617 3618 3619 3620
+ 3700 3701 3702 3703 3704 3705 3706 3707 3708 3709 3710 3711 3712 3713 3714 3715 3716 3717 3718 3719 3720 STR
X28 3700 3701 3702 3703 3704 3705 3706 3707 3708 3709 3710 3711 3712 3713 3714 3715 3716 3717 3718 3719 3720
+ 3800 3801 3802 3803 3804 3805 3806 3807 3808 3809 3810 3811 3812 3813 3814 3815 3816 3817 3818 3819 3820 STR
X29 3800 3801 3802 3803 3804 3805 3806 3807 3808 3809 3810 3811 3812 3813 3814 3815 3816 3817 3818 3819 3820
+ 3900 3901 3902 3903 3904 3905 3906 3907 3908 3909 3910 3911 3912 3913 3914 3915 3916 3917 3918 3919 3920 STR
X30 3900 3901 3902 3903 3904 3905 3906 3907 3908 3909 3910 3911 3912 3913 3914 3915 3916 3917 3918 3919 3920
+ 4000 4001 4002 4003 4004 4005 4006 4007 4008 4009 4010 4011 4012 4013 4014 4015 4016 4017 4018 4019 4020 STR
X31 4000 4001 4002 4003 4004 4005 4006 4007 4008 4009 4010 4011 4012 4013 4014 4015 4016 4017 4018 4019 4020
+ 4100 4101 4102 4103 4104 4105 4106 4107 4108 4109 4110 4111 4112 4113 4114 4115 4116 4117 4118 4119 4120 STR
X32 4100 4101 4102 4103 4104 4105 4106 4107 4108 4109 4110 4111 4112 4113 4114 4115 4116 4117 4118 4119 4120
+ 4200 4201 4202 4203 4204 4205 4206 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 STR
X33 4200 4201 4202 4203 4204 4205 4206 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220
+ 4300 4301 4302 4303 4304 4305 4306 4307 4308 4309 4310 4311 4312 4313 4314 4315 4316 4317 4318 4319 4320 STR
X34 4300 4301 4302 4303 4304 4305 4306 4307 4308 4309 4310 4311 4312 4313 4314 4315 4316 4317 4318 4319 4320
+ 4400 4401 4402 4403 4404 4405 4406 4407 4408 4409 4410 4411 4412 4413 4414 4415 4416 4417 4418 4419 4420 STR
X35 4400 4401 4402 4403 4404 4405 4406 4407 4408 4409 4410 4411 4412 4413 4414 4415 4416 4417 4418 4419 4420
+ 4500 4501 4502 4503 4504 4505 4506 4507 4508 4509 4510 4511 4512 4513 4514 4515 4516 4517 4518 4519 4520 STR
X36 4500 4501 4502 4503 4504 4505 4506 4507 4508 4509 4510 4511 4512 4513 4514 4515 4516 4517 4518 4519 4520
+ 4600 4601 4602 4603 4604 4605 4606 4607 4608 4609 4610 4611 4612 4613 4614 4615 4616 4617 4618 4619 4620 STR
X37 4600 4601 4602 4603 4604 4605 4606 4607 4608 4609 4610 4611 4612 4613 4614 4615 4616 4617 4618 4619 4620
+ 4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719 4720 STR
X38 4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719 4720
+ 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810 4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 STR
X39 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810 4811 4812 4813 4814 4815 4816 4817 4818 4819 4820
+ 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 STR
X40 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920
+ 5000 5001 5002 5003 5004 5005 5006 5007 5008 5009 5010 5011 5012 5013 5014 5015 5016 5017 5018 5019 5020 STR
X41 5000 5001 5002 5003 5004 5005 5006 5007 5008 5009 5010 5011 5012 5013 5014 5015 5016 5017 5018 5019 5020
+ 5100 5101 5102 5103 5104 5105 5106 5107 5108 5109 5110 5111 5112 5113 5114 5115 5116 5117 5118 5119 5120 STR
X42 5100 5101 5102 5103 5104 5105 5106 5107 5108 5109 5110 5111 5112 5113 5114 5115 5116 5117 5118 5119 5120
+ 5200 5201 5202 5203 5204 5205 5206 5207 5208 5209 5210 5211 5212 5213 5214 5215 5216 5217 5218 5219 5220 STR
X43 5200 5201 5202 5203 5204 5205 5206 5207 5208 5209 5210 5211 5212 5213 5214 5215 5216 5217 5218 5219 5220
+ 5300 5301 5302 5303 5304 5305 5306 5307 5308 5309 5310 5311 5312 5313 5314 5315 5316 5317 5318 5319 5320 STR
X44 5300 5301 5302 5303 5304 5305 5306 5307 5308 5309 5310 5311 5312 5313 5314 5315 5316 5317 5318 5319 5320
+ 5400 5401 5402 5403 5404 5405 5406 5407 5408 5409 5410 5411 5412 5413 5414 5415 5416 5417 5418 5419 5420 STR
X45 5400 5401 5402 5403 5404 5405 5406 5407 5408 5409 5410 5411 5412 5413 5414 5415 5416 5417 5418 5419 5420

+ 5500 5501 5502 5503 5504 5505 5506 5507 5508 5509 5510 5511 5512 5513 5514 5515 5516 5517 5518 5519 5520 STR
X46 5500 5501 5502 5503 5504 5505 5506 5507 5508 5509 5510 5511 5512 5513 5514 5515 5516 5517 5518 5519 5520
+ 5600 5601 5602 5603 5604 5605 5606 5607 5608 5609 5610 5611 5612 5613 5614 5615 5616 5617 5618 5619 5620 STR
X47 5600 5601 5602 5603 5604 5605 5606 5607 5608 5609 5610 5611 5612 5613 5614 5615 5616 5617 5618 5619 5620
+ 5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710 5711 5712 5713 5714 5715 5716 5717 5718 5719 5720 STR
X48 5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710 5711 5712 5713 5714 5715 5716 5717 5718 5719 5720
+ 5800 5801 5802 5803 5804 5805 5806 5807 5808 5809 5810 5811 5812 5813 5814 5815 5816 5817 5818 5819 5820 STR
X49 5800 5801 5802 5803 5804 5805 5806 5807 5808 5809 5810 5811 5812 5813 5814 5815 5816 5817 5818 5819 5820
+ 5900 5901 5902 5903 5904 5905 5906 5907 5908 5909 5910 5911 5912 5913 5914 5915 5916 5917 5918 5919 5920 STR
X50 5900 5901 5902 5903 5904 5905 5906 5907 5908 5909 5910 5911 5912 5913 5914 5915 5916 5917 5918 5919 5920
+ 6000 6001 6002 6003 6004 6005 6006 6007 6008 6009 6010 6011 6012 6013 6014 6015 6016 6017 6018 6019 6020 STR
X51 6000 6001 6002 6003 6004 6005 6006 6007 6008 6009 6010 6011 6012 6013 6014 6015 6016 6017 6018 6019 6020
+ 6100 6101 6102 6103 6104 6105 6106 6107 6108 6109 6110 6111 6112 6113 6114 6115 6116 6117 6118 6119 6120 STR
X52 6100 6101 6102 6103 6104 6105 6106 6107 6108 6109 6110 6111 6112 6113 6114 6115 6116 6117 6118 6119 6120
+ 6200 6201 6202 6203 6204 6205 6206 6207 6208 6209 6210 6211 6212 6213 6214 6215 6216 6217 6218 6219 6220 STR
X53 6200 6201 6202 6203 6204 6205 6206 6207 6208 6209 6210 6211 6212 6213 6214 6215 6216 6217 6218 6219 6220
+ 6300 6301 6302 6303 6304 6305 6306 6307 6308 6309 6310 6311 6312 6313 6314 6315 6316 6317 6318 6319 6320 STR
X54 6300 6301 6302 6303 6304 6305 6306 6307 6308 6309 6310 6311 6312 6313 6314 6315 6316 6317 6318 6319 6320
+ 6400 6401 6402 6403 6404 6405 6406 6407 6408 6409 6410 6411 6412 6413 6414 6415 6416 6417 6418 6419 6420 STR
X55 6400 6401 6402 6403 6404 6405 6406 6407 6408 6409 6410 6411 6412 6413 6414 6415 6416 6417 6418 6419 6420
+ 6500 6501 6502 6503 6504 6505 6506 6507 6508 6509 6510 6511 6512 6513 6514 6515 6516 6517 6518 6519 6520 STR
X56 6500 6501 6502 6503 6504 6505 6506 6507 6508 6509 6510 6511 6512 6513 6514 6515 6516 6517 6518 6519 6520
+ 6600 6601 6602 6603 6604 6605 6606 6607 6608 6609 6610 6611 6612 6613 6614 6615 6616 6617 6618 6619 6620 STR
X57 6600 6601 6602 6603 6604 6605 6606 6607 6608 6609 6610 6611 6612 6613 6614 6615 6616 6617 6618 6619 6620
+ 6700 6701 6702 6703 6704 6705 6706 6707 6708 6709 6710 6711 6712 6713 6714 6715 6716 6717 6718 6719 6720 STR
X58 6700 6701 6702 6703 6704 6705 6706 6707 6708 6709 6710 6711 6712 6713 6714 6715 6716 6717 6718 6719 6720
+ 6800 6801 6802 6803 6804 6805 6806 6807 6808 6809 6810 6811 6812 6813 6814 6815 6816 6817 6818 6819 6820 STR
X59 6800 6801 6802 6803 6804 6805 6806 6807 6808 6809 6810 6811 6812 6813 6814 6815 6816 6817 6818 6819 6820
+ 6900 6901 6902 6903 6904 6905 6906 6907 6908 6909 6910 6911 6912 6913 6914 6915 6916 6917 6918 6919 6920 STR
X60 6900 6901 6902 6903 6904 6905 6906 6907 6908 6909 6910 6911 6912 6913 6914 6915 6916 6917 6918 6919 6920
+ 7000 7001 7002 7003 7004 7005 7006 7007 7008 7009 7010 7011 7012 7013 7014 7015 7016 7017 7018 7019 7020 STR
X61 7000 7001 7002 7003 7004 7005 7006 7007 7008 7009 7010 7011 7012 7013 7014 7015 7016 7017 7018 7019 7020
+ 7100 7101 7102 7103 7104 7105 7106 7107 7108 7109 7110 7111 7112 7113 7114 7115 7116 7117 7118 7119 7120 STR
*electrode ends - last node 7000
X62 7100 7101 7102 7103 7104 7105 7106 7107 7108 7109 7110 7111 7112 7113 7114 7115 7116 7117 7118 7119 7120
+ 7200 7201 7202 7203 7204 7205 7206 7207 7208 7209 7210 7211 7212 7213 7214 7215 7216 7217 7218 7219 7220 STR
X63 7200 7201 7202 7203 7204 7205 7206 7207 7208 7209 7210 7211 7212 7213 7214 7215 7216 7217 7218 7219 7220
+ 7300 7301 7302 7303 7304 7305 7306 7307 7308 7309 7310 7311 7312 7313 7314 7315 7316 7317 7318 7319 7320 STR
X64 7300 7301 7302 7303 7304 7305 7306 7307 7308 7309 7310 7311 7312 7313 7314 7315 7316 7317 7318 7319 7320
+ 7400 7401 7402 7403 7404 7405 7406 7407 7408 7409 7410 7411 7412 7413 7414 7415 7416 7417 7418 7419 7420 STR
X65 7400 7401 7402 7403 7404 7405 7406 7407 7408 7409 7410 7411 7412 7413 7414 7415 7416 7417 7418 7419 7420
+ 7500 7501 7502 7503 7504 7505 7506 7507 7508 7509 7510 7511 7512 7513 7514 7515 7516 7517 7518 7519 7520 STR
X66 7500 7501 7502 7503 7504 7505 7506 7507 7508 7509 7510 7511 7512 7513 7514 7515 7516 7517 7518 7519 7520
+ 7600 7601 7602 7603 7604 7605 7606 7607 7608 7609 7610 7611 7612 7613 7614 7615 7616 7617 7618 7619 7620 STR
X67 7600 7601 7602 7603 7604 7605 7606 7607 7608 7609 7610 7611 7612 7613 7614 7615 7616 7617 7618 7619 7620
+ 7700 7701 7702 7703 7704 7705 7706 7707 7708 7709 7710 7711 7712 7713 7714 7715 7716 7717 7718 7719 7720 STR
X68 7700 7701 7702 7703 7704 7705 7706 7707 7708 7709 7710 7711 7712 7713 7714 7715 7716 7717 7718 7719 7720
+ 7800 7801 7802 7803 7804 7805 7806 7807 7808 7809 7810 7811 7812 7813 7814 7815 7816 7817 7818 7819 7820 STR
X69 7800 7801 7802 7803 7804 7805 7806 7807 7808 7809 7810 7811 7812 7813 7814 7815 7816 7817 7818 7819 7820
+ 7900 7901 7902 7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913 7914 7915 7916 7917 7918 7919 7920 STR
X70 7900 7901 7902 7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913 7914 7915 7916 7917 7918 7919 7920
+ 8000 8001 8002 8003 8004 8005 8006 8007 8008 8009 8010 8011 8012 8013 8014 8015 8016 8017 8018 8019 8020 STR
X71 8000 8001 8002 8003 8004 8005 8006 8007 8008 8009 8010 8011 8012 8013 8014 8015 8016 8017 8018 8019 8020
+ 8100 8101 8102 8103 8104 8105 8106 8107 8108 8109 8110 8111 8112 8113 8114 8115 8116 8117 8118 8119 8120 STR
X72 8100 8101 8102 8103 8104 8105 8106 8107 8108 8109 8110 8111 8112 8113 8114 8115 8116 8117 8118 8119 8120
+ 8200 8201 8202 8203 8204 8205 8206 8207 8208 8209 8210 8211 8212 8213 8214 8215 8216 8217 8218 8219 8220 STR
X73 8200 8201 8202 8203 8204 8205 8206 8207 8208 8209 8210 8211 8212 8213 8214 8215 8216 8217 8218 8219 8220
+ 8300 8301 8302 8303 8304 8305 8306 8307 8308 8309 8310 8311 8312 8313 8314 8315 8316 8317 8318 8319 8320 STR
X74 8300 8301 8302 8303 8304 8305 8306 8307 8308 8309 8310 8311 8312 8313 8314 8315 8316 8317 8318 8319 8320
+ 8400 8401 8402 8403 8404 8405 8406 8407 8408 8409 8410 8411 8412 8413 8414 8415 8416 8417 8418 8419 8420 STR
X75 8400 8401 8402 8403 8404 8405 8406 8407 8408 8409 8410 8411 8412 8413 8414 8415 8416 8417 8418 8419 8420
+ 8500 8501 8502 8503 8504 8505 8506 8507 8508 8509 8510 8511 8512 8513 8514 8515 8516 8517 8518 8519 8520 STR
X76 8500 8501 8502 8503 8504 8505 8506 8507 8508 8509 8510 8511 8512 8513 8514 8515 8516 8517 8518 8519 8520
+ 8600 8601 8602 8603 8604 8605 8606 8607 8608 8609 8610 8611 8612 8613 8614 8615 8616 8617 8618 8619 8620 STR
X77 8600 8601 8602 8603 8604 8605 8606 8607 8608 8609 8610 8611 8612 8613 8614 8615 8616 8617 8618 8619 8620
+ 8700 8701 8702 8703 8704 8705 8706 8707 8708 8709 8710 8711 8712 8713 8714 8715 8716 8717 8718 8719 8720 STR
X78 8700 8701 8702 8703 8704 8705 8706 8707 8708 8709 8710 8711 8712 8713 8714 8715 8716 8717 8718 8719 8720

```
+   8800 8801 8802 8803 8804 8805 8806 8807 8808 8809 8810 8811 8812 8813 8814 8815 8816 8817 8818 8819 8820 STR
X79 8800 8801 8802 8803 8804 8805 8806 8807 8808 8809 8810 8811 8812 8813 8814 8815 8816 8817 8818 8819 8820
+   8900 8901 8902 8903 8904 8905 8906 8907 8908 8909 8910 8911 8912 8913 8914 8915 8916 8917 8918 8919 8920 STR
X80 8900 8901 8902 8903 8904 8905 8906 8907 8908 8909 8910 8911 8912 8913 8914 8915 8916 8917 8918 8919 8920
+   9000 9001 9002 9003 9004 9005 9006 9007 9008 9009 9010 9011 9012 9013 9014 9015 9016 9017 9018 9019 9020 STR
X81 9000 9001 9002 9003 9004 9005 9006 9007 9008 9009 9010 9011 9012 9013 9014 9015 9016 9017 9018 9019 9020
+   9100 9101 9102 9103 9104 9105 9106 9107 9108 9109 9110 9111 9112 9113 9114 9115 9116 9117 9118 9119 9120 STR
*lower electrode
X100 0 3100 3200 3300 3400 3500 3600 3700 3800 3900 4000 4100 4200 4300 4400 4500 4600 4700 4800 4900 5000 5100
+ 5200 5300 5400 5500 5600 5700 5800 5900 6000 6100 6200 6300 6400 6500 6600 6700 6800 6900 7000 EL
*top electrode
X101 2 3120 3220 3320 3420 3520 3620 3720 3820 3920 4020 4120 4220 4320 4420 4520 4620 4720 4820 4920 5020 5120
+ 5220 5320 5420 5520 5620 5720 5820 5920 6020 6120 6220 6320 6420 6520 6620 6720 6820 6920 7020 EL
*Restore the .end statement if this circuit is to be run stand-alone
*.END
*
```

```
      program optimp

c     Program to optimise kdis using NAG Library routine (E04JAF)
c     -- quasi-Newton method -- for the heterogeneous reaction model
c     The subroutine funct1 calcs. theoretical conc. profiles of Cl
c     & H, overlay the equivalent circuit in a backbone circuit file
c     (by BAC & JLP) then invokes SPICE via SYSTEM calls to deduce
c     the soln. impedance. The program minimises the deviation, as a
c     function of kdis, between theoretical & expt'al impedance which
c     will be return by funct1
c
c
c     J(mol/cm2s),          = -kdis
c               Cl/H, y=0
c
c     TPMCl(s) + H2O(l)  ----->  TPMOH(s) + Cl-(aq) + H+(aq)
c      A1                  A2       A3       A4
c
c     The following files should be presented
c
c     Backbone resistance circuit file:     netbase.cir
c
c     Exp. data is stored in ASCII data file
c
c     format in data file '*.prn'
c     1st col. - vf (ml/s)
c     2nd col. - dR (ohm)
c
c     Declaration
c
c     Physical Parameters
c
c     2h - height of channel, cm
c     xe - electrode length, cm
c     xp - reactive solid length, cm
c     xg - length of gap between electrode & solid, cm
c     we - width of electrode, cm
c     diffa3 - diffusion coefficient of A3 (chloride), cm2/s
c     diffa4 - diffusion coefficient of A4 (hydrogen ion), cm2/s
c     vf - volume flow rate, cm3/s
c     condcl - conductivity of chloride (scm2/mol)
c     condh - conductivity of hydrogen ion (scm2/mol)
c     bcond - conductivity of background electrolyte (scm2/mol)
c     bconc - conc. of background electrolyte  (mol/ml)

c     Indexing variables
c
c     dy - y grid width
c     dx - x grid width
c     nj - no. grid in y-direction
c     nk - no. grid in x-direction
c     k0  -\
c          grid no. designate reactive solid
c     k1  -/
c     k2e - gap for edge effect
c          grid no. designate gap
c     k2  -/
c          grid no. designate Pt electrode
c     k3  -/
c     k3e - gap for edge effect
c     tnsx - total no. of SPICE box (x-directiom)
c     nsx - no. of SPICE box for detector (x-direction)
c     nsy - no. of SPICE box for channel height (y-direction)
c     nrk - no. of BI box (x-direction) for one SPICE box
c     nrj - no. of Bl box (y-firection) for one SPICE box
c     nbg - no. of x   BI box assigned for the edge effect gap
c     nbe - no. of x - BI box assigned for Pt electrode
c
c
c     Kinetic parameter
c
c     kdis - const. flux for chloride/hydrogen ion

      integer nk,nj,i,j,k0,k1,k2e,k2,k3,k3e,nbg,nbe,nd,ng,np,nbg1,
     _        nbe1,nrj,nrk,tnsx,nsx,nsy,ng,lendir,lennam,lenbnam,
     _        liw,lw,ibound,ifail

      parameter(ng=1, np=1, liw=80, lw=80)

      double precision vfarr(20),drth(20),drex(20),rth(20),rbg
      double precision h,d,xe,xp,xg,we,diffa3,diffa4,
     _   dy,dx,kdis,condcl,condh,x(np),optk(ng),bl(np),bu(np),
     _   x(np),mf,ww(lw),guess(ng),er,brx,bry,bcx,bcy,bconc,bcond

      integer iw(liw), sflag

      character*4 endstr
      character*9 name, bname
      character*40 dire
      character*55 fname,fname1,fname2,fname3,fnameop
      character*133 buff
      character*150 spargum,othargum

      common /param1/ dx,dy,h,d,diffa3,diffa4,xe,we,condcl,condh
      common /param2/ nj,nk,k0,k1,k2e,k2,k3,k3e,nd,nrj,nrk,nsy
      common /param3/ bcond,bconc
      common /param4/ lennam,lendir
      common /result/ drth, rth, drex, vfarr, rbg
      common /charat/ dire,name
```

```
c           assign numerical/string constants in the computation

      dire='/home/joule/rgc/tam/Spice/'
      lendir=26
      name='cond36.'
      lennam=7
      bname='netbg36.'
      lenbnam=8

c     output file name

      fnameop(1:lendir)=dire
      fnameop(lendir+1:lendir+13)='optcond36.dat'

      tnsx=80
      nsx=40
      nsy=20
      nrj=15
      nrk=4

      xg=0.897
      xp=0.817
      h=0.094/2.0

      xe=0.396
      we=0.394
      d=0.608

      condcl=76.3
      condh=349.8
      diffa3=2.00e-5
      diffa4=8.98e-5

c     Conductivity of background electrolyte, KNO3 = 144.9 s cm2/mol

      bcond=144.9
      bconc=2.2632e-6

      nj=nrj*nsy
      nbe1=nrk*nsx
      nbg1=nrk*(tnsx-nsx)/2

      dy=2.0*h/nj
      dx=(xe/2.0)/nbg1
      nk=int((xe+xp+xg+(xe/2.0))/dx)
      k0=1
      k1=int(xp/dx)
      k2=int((xp+xg)/dx)
      k3=int((xe+xp+xg)/dx)
      k3e=nk
      nbe=k3-k2
      nbg=k3e-k3
      k2e=k2-nbg

      print*,' k0  & dist. = ', k0, k0*dx
      print*,' k1  & dist. = ', k1, k1*dx
      print*,' k2e & dist. = ', k2e, k2e*dx
      print*,' k2  & dist. = ', k2, k2*dx
      print*,' k3  & dist. = ', k3, k3*dx
      print*,' k3e & dist. = ', k3e, k3e*dx

      print*, 'xp       = ', xp
      print*, 'xp+xg    = ', xp+xg
      print*, 'xp+xg+xe = ', xp+xg+xe
      print*, 'xe       = ', xe
      print*, 'we       = ', we

      print*,' nbe1 = ', nbe1
      print*,' nbe  = ', nbe
      print*,' nbg1 = ', nbg1
      print*,' nbg  = ', nbg
      print*,' nj   = ', nj
      print*,' nk   = ', nk
```

```
c     Get data file

      fname1(1:lendir)=dire
      fname1(lendir+1:lendir+lennam+1)=name
      j=lendir+lennam
      fname1(j+1:j+3)='prn'

      open(unit=13, file=fname1, status='old')

      j=1
1     read(13,*,err=2) vfarr(j), drex(j)
         j=j+1
         goto 1
2     close(unit=13)
      nd=j-1

c     Establish the circuit file names

      fname2(1:lendir)=dire
      fname2(lendir+1:lendir+lenbnam+1)=bname
      j=lendir+lenbnam
      fname3=fname2
      fname1=dire
      fname2(j+1:j+3)='cir'
      fname3(j+1:j+3)='out'
      fname1(lendir+1:lendir+11)='netbase.cir'

c     Cal. background R & C

      brx=(xe/(2.0*h*we))/(bcond*bconc)
      bry=((8.0*h)/(we*xe))/(bcond*bconc)
      er=78.54*8.85e-14
      bcx=2.0*er*we*h/xe
      bcy=xe*er*we/(8.0*h)

      print*, 'Rx = ', brx
      print*, 'Ry = ', bry
      print*, 'Cx = ', bcx
      print*, 'Cy = ', bcy

c     Create a background circuit file with the designated Rx, Ry, Cx, Cy

      endstr='.END'
      open(unit=9, file=fname1, status='old')
      open(unit=10, file=fname2, status='unknown')

      sflag=1
200   read(9,'(A)', err=220) buff
      l=133
210   continue
      if ((l.gt.1).and.(buff(l:l).eq.' ')) then
           l=l-1
           goto 210
      end if
      if (buff(1:4).eq.endstr) then
            sflag=0
      end if
      if ((sflag.eq.1).and.(buff(1:6).eq.'R1 1 2')) then
            write(10,500) 'R1', 1, 2, bry
            print 500, 'R1', 1, 2, bry
        else if ((sflag.eq.1).and.(buff(1:6).eq.'R2 3 4')) then
            write(10,500) 'R2', 3, 4, bry
            print 500, 'R2', 3, 4, bry
        else if ((sflag.eq.1).and.(buff(1:6).eq.'R3 2 3')) then
            write(10,500) 'R3', 2, 3, brx
            print 500, 'R3', 2, 3, brx
        else if ((sflag.eq.1).and.(buff(1:6).eq.'R4 1 4')) then
            write(10,500) 'R4', 1, 4, brx
            print 500, 'R4', 1, 4, brx
        else if ((sflag.eq.1).and.(buff(1:6).eq.'C1 1 2')) then
            write(10,510) 'C1', 1, 2, bcy*(1e12), 'pF'
            print 510, 'C1', 1, 2, bcy*(1e12), 'pF'
        else if ((sflag.eq.1).and.(buff(1:6).eq.'C2 3 4')) then
            write(10,510) 'C2', 3, 4, bcy*(1e12), 'pF'
```

```
          print 510, 'C2', 3, 4, bcy*(1e12), 'pF'
       else if ((sflag.eq.1).and.(buff(1:6).eq.'C3 2 3')) then
          write(10,510) 'C3', 2, 3, bcx*(1e12), 'pF'
          print 510, 'C3', 2, 3, bcx*(1e12), 'pF'
       else if ((sflag.eq.1).and.(buff(1:6).eq.'C4 1 4')) then
          write(10,510) 'C4', 1, 4, bcx*(1e12), 'pF'
          print 510, 'C4', 1, 4, bcx*(1e12), 'pF'
       else
          write(10,'(A)') buff(1:l)
       end if
       goto 200
220    write(10,'(A)') endstr
       close(unit=9)
       close(unit=10)

c      Remove the SPICE o/p circuit file first

       othargum='rm -f'
       othargum(7:j+10)=fname3
       call system(othargum)

c      Invoke SPICE via SYSTEM call to calc. background resistance

       spargum='/home/joule/bac/bin/spice -b '
       i=29+j+3
       spargum(30:i+1)=fname2
       spargum(i+1:i+4)=' > '
       spargum(i+5:i+j+8)=fname3
       call system(spargum)

c      Extract numerical part from SPICE o/p file

       fname2(j+1:j+3)='  '
       fname1=fname(fname2,j,1)
       call fzrealmin(fname3, fname1, rbg)
       print*, 'Background resistance = ', rbg

c             set constraints for kdis

       ibound=0
       bl(1)=2.0e-9
       bu(1)=5.0e-8

       print*, 'Output file name: ', fnameop
       print*, 'Const. chloride flux model '
       print*, 'Upper bound of parameter = ', bu(1)
       print*, 'Lower bound of parameter = ', bl(1)

       kdis=0.0

       do i=1, ng

       print*, 'No. of iteration ', i

c             x - initial guess of kdis

       guess(i)=2.37e-8
       x(1)=guess(i)

c      invoke the optimiser via NAG numerical library call (E04JAF)

       ifail=-1
       call e04jaf(np,ibound,bl,bu,x,mf,iw,liw,ww,lw,ifail)

c      assign optimised results to the corresponding variables

       optk(i)=x(1)
       kdis=kdis+optk(i)

       print*, 'Function value  ', mf
       print*, 'Optimised kdis  ', optk(i)
       print*, 'Initial guess ', guess(i)
       print*, 'IFAIL ', ifail
       print*
       end do

       kdis=kdis/ng
       print*, 'Mean kdis ', kdis
       print*
       print*, 'From mean kdis '
       call funct1(np, kdis, mf)

       open(unit=13, file=fnameop, status='unknown')
       print*, '      Vf          dR(expt)     dR(opt)'
       do i=1, nd
          write (13, 990) vfarr(i), drth(i)
          print 1000, vfarr(i), drex(i), drth(i)
       end do
       close(unit=13)
       print*, 'RMSD ', mf

500    format(a2,i2,i2,f12.2)
510    format(a2,i2,i2,f8.4,a2)
990    format(1x,e13.4,2x,e13.4)
1000   format(1x,e13.4,2x,e13.4,2x,e13.4)

       end

       subroutine funct1(np, x, fc)

       integer nk,nj,xdimens,dimens,i,j,k,k0,k1,k2e,k2,k3,k3e,ii,nd,np,
      _        nrj,nrk,nsy,l,xstep,ng,lendir,lennam

       parameter(dimens=300,xdimens=1000)
       double precision aja3(dimens),bja3(dimens),cja3(dimens),
      _                 aja4(dimens),bja4(dimens),cja4(dimens),
      _                 dj(dimens),uj(dimens),lama3(dimens),
      _                 lama4(dimens),ga3j(dimens),ga4j(dimens),
      _                 ch(20,80),ccl(20,80),x(np),fc,rmsd
       double precision vfarr(20), drth(20), drex(20), rth(20), rbg
       double precision dummy,dummy1,h,d,xe,we,diffa3,diffa4,vf,
      _                 dy,dx,kdis,condcl,condh,bconc,bcond
       character*9 name
       character*40 dire
       character*55 fname,fname1,fname2,fname3
       character*150 spargum,othargum

       common /param1/ dx,dy,h,d,diffa3,diffa4,xe,we,condcl,condh
       common /param2/ nj,nk,k0,k1,k2e,k2,k3,k3e,nd,nrj,nrk,nsy
       common /param3/ bcond,bconc
       common /param4/ lennam,lendir
       common /result/ drth, rth, drex, vfarr, rbg
       common /charat/ dire,name

       kdis=abs(x(1))
       rmsd=0.0

       print*, 'kdis = ', kdis
       print*, 'x(1) = ', x(1)

c      Calc. resistance values for diff. Vf

       do ii=1, nd
       vf=vfarr(ii)

c      Set up tridiagonal, {d} & {u} matrices

c      1. Calc. lamda & assign initial ga3j, ga4j

       dummy=(dx*d*(2.0*h)**3)/(6.0*vf*(dy**3))
       do j=1, nj-1
                dummy1=dummy/(j*(2.0*h-j*dy))
                lama3(j)=diffa3*dummy1
                lama4(j)=diffa4*dummy1
                ga3j(j)=0.0
                ga4j(j)=0.0
       end do
```

```
c    2. Calc. aj, bj & cj for A3 & A4

     do j=2, nj-1
                aja3(j-1)=-1.0*lama3(j)
                bja3(j)=2.0*lama3(j)+1.0
                cja3(j-1)=-1.0*lama3(j-1)
                aja4(j-1)=-1.0*lama4(j)
                bja4(j)=2.0*lama4(j)+1.0
                cja4(j-1)=-1.0*lama4(j-1)
     end do
     bja3(nj-1)=lama3(nj-1)+1.0
     bja4(nj-1)=lama4(nj-1)+1.0

c    3. Calc. ga3j(k) & ga4j(k) over pellet surface

     i=1
     do k=k0, k1

c         For species A3

               do j=1, nj-1
                        dj(j)=ga3j(j)
               end do
               bja3(1)=1.0+lama3(1)
               dj(1)=ga3j(1)+lama3(1)*dy*kdis/diffa3
               call thomas(nj,aja3,bja3,cja3,dj,uj)
               do j=1, nj-1
                        ga3j(j)=uj(j)
               end do

c         For species A4

               do j=1, nj-1
                        dj(j)=ga4j(j)
               end do
               bja4(1)=1.0+lama4(1)
               dj(1)=ga4j(1)+lama4(1)*dy*kdis/diffa4
               call thomas(nj,aja4,bja4,cja4,dj,uj)
               do j=1, nj-1
                        ga4j(j)=uj(j)
               end do

     end do


c    4. Calc. ga3j(k) & ga4j(k) over gap (exclude SPICE region)

     do k=k1+1, k2e-1

c         For species A3

               do j=1, nj-1
                        dj(j)=ga3j(j)
               end do
               bja3(1)=1.0+lama3(1)
               call thomas(nj,aja3,bja3,cja3,dj,uj)
               do j=1, nj-1
                        ga3j(j)=uj(j)
               end do

c         For species A4

               do j=1, nj-1
                        dj(j)=ga4j(j)
               end do
               bja4(1)=1.0+lama4(1)
               call thomas(nj,aja4,bja4,cja4,dj,uj)
               do j=1, nj-1
                        ga4j(j)=uj(j)
               end do

     end do
```

```
c    5. Calc. ga3j(k) & ga4j(k) over gap & Pt electrode (SPICE
c       region only)

     xstep=0
     dummy=nrj*nrk
     do l=1, nsy
               ch(l,1)=0.0
               ccl(l,1)=0.0
     end do
     do k=k2e, k3e

c         For species A3

               do j=1, nj-1
                        dj(j)=ga3j(j)
               end do
               bja3(1)=1.0+lama3(1)
               call thomas(nj,aja3,bja3,cja3,dj,uj)
               do j=1, nj-1
                        ga3j(j)=uj(j)
               end do

c         For species A4

               do j=1, nj-1
                        dj(j)=ga4j(j)
               end do
               bja4(1)=1.0+lama4(1)
               call thomas(nj,aja4,bja4,cja4,dj,uj)
               do j=1, nj-1
                        ga4j(j)=uj(j)
               end do

c         Write average conc. (20x80) to data matrics for SPICE

               xstep=xstep+1
               if (xstep.gt.nrk) then
                        do l=1, nsy
                                 ch(l,i)=ch(l,i)/dummy
                                 ccl(l,i)=ccl(l,i)/dummy
                           ch(l,i+1)=0.0
                           ccl(l,i+1)=0.0
                                 end do
                     i=i+1
                                 xstep=1
                      end if
                      do l=1, nsy
                                 ng=(l-1)*nrj+1
                                 do j=ng, ng+nrj-1
                                          ccl(l,i)=ccl(l,i)+ga3j(j)
                                          ch(l,i)=ch(l,i)+ga4j(j)
                                 end do
                      end do
     end do

c     Establish circuit file names

     fname1(1:lendir)=dire
     fname1(lendir+1:lendir+lennam+1)=name
     j=lendir+lennam
     fname2=fname1
     fname1(j+1:j+3)='cir'
     fname2(j+1:j+3)='out'

c     Remove the SPICE o/p file first

     othargum='rm -f'
     othargum(7:j+10)=fname2
     call system(othargum)

c     Convert conc. matrcs to SPICE matrix

     call bi2spic(ccl,ch,condcl,condh,we,xe,h,bconc,bcond,
    _            dire,lendir,fname1)
```

```
c     Invoke SPICE via SYSTEM call to calc. resistance

      spargum='/home/joule/bac/bin/spice -b '
      i=29+j+3
      spargum(30:i+1)=fname1
      spargum(i+1:i+4)=' > '
      spargum(i+5:i+j+8)=fname2
      call system(spargum)

c     Extract numerical part from SPICE o/p file

      fname1(j+1:j+3)='  '
      fname3=fname(fname1,j,ii)
      call fzrealmin(fname2, fname3, dummy)
      rth(ii)=dummy
      drth(ii)=rbg-rth(ii)

      rmsd=rmsd+(drex(ii)-drth(ii))**2

      print*, vfarr(ii), drex(ii), drth(ii)

      end do

      rmsd=sqrt(rmsd/nd)
      fc=rmsd

      print*, 'rmsd = ', fc
      print*, ' '

      return
      end

      include 'thomas.for'

      character*55 function fname(sname,in,extn)

c     Function to combine file extension (integer) with file name (char)

      integer extn, i, in, j, k, l, l1, l2

      character*55 sname
      character*3 ext

      fname(1:in)=sname
      ext='   '
      k=1
      l=1
      l2=extn
      do i=3, 1, -1
         l1=int(10**(i-1))-1
         if ((l2.gt.l1).or.(l.eq.0)) then
            j=l2/(l1+1)
            l2=l2-j*(l1+1)
            l=1
            if ((l2.lt.10).and.(i.eq.3)) then
               l=0
            end if
            if ((l2.eq.0).and.(i.le.2)) then
               l=0
            end if
            j=j+48
            ext(k:k)=char(j)
            k=k+1
         end if
      end do
      i=in+1
      j=i+k-1
      fname(i:j)=ext
      do i=j+1, 55
        fname(i:i)=' '
      end do

      return
      end

      subroutine fzrealmin(spiceout, zplot,
     _                     zmin)

c     Extract numerical part of SPICE o/p file & locate Z' at min Z''

      character*255 buff
      character*55 spiceout, zplot
      double precision zmin, dummy, zreal(1000), zimag(1000)
      integer ng, i

      open(unit=11, file=spiceout, status='old')
      open(unit=10, file=zplot, status='unknown')

90    read(11, '(A)', err=92) buff
      if ((ichar(buff(1:1)).le.57).and.(ichar(buff(1:1)).ge.48)) then
         write(10, '(A)') buff
      end if
      goto 90
92    close(unit=11)
      close(unit=10)

      open(unit=10, file=zplot, status='old')

      ng=1
94    read(10,*, err=96), dummy,dummy,dummy,zreal(ng),zimag(ng)
         ng=ng+1
      goto 94
96    close(unit=10)
      ng=ng-1

c     Locate the Z' at min Z''

      i=0
98    i=i+1
      if (abs(zimag(i)).lt.abs(zimag(i+1))) then
         goto 99
      end if
      goto 98
99    continue
      zmin=abs(zreal(i))

      return
      end


      subroutine bi2spic(ccl,ch,condcl,condh,we,xe,h,bconc,bcond,
     _                   direc,id,fout)

c     adapted from bi2spic2.f (written by Dr. B.A. Coles)
c     Converts Bl 20x80 concentration array into SPICE
c     resistors to be attached to a basic .cir file
c     resx=(8*h/we*xe)/(condy*conc), resy=(xe/2*we*h)/(condy*conc)
c     where condy=ionic conductivity cms2/mol
c     and concn=bidata(i,j). No resistors are generated if
c     rho=1/(concdy*conc) is greater than 7e5.
c
c              2---resx---3
c              |          |
c         resy           resy
c              |          |
c              1---resx---4
c

c     Declaration

      double precision we,xe,h,ccl(20,80),ch(20,80),condcl,condh,
     _                 rx,ry,resx,resy,rho,bidata(20,80),condy,
     _                 er,brx,bry,bcx,bcy,bcond,bconc
      character*133 buff
      character*55 fout,spcir
      character*26 direc
      character*1 rstr
      character*4 endstr
      integer i,id,idxr,j,k,l,node1,node2,node3,node4,sflag
```

```
      rx=xe/(2*we*h)
      ry=8*h/(we*xe)

c             String const.

      rstr='R'
      endstr='.END'

      spcir=direc
      spcir(id+1:id+1+11)='netbase.cir'

c     Cal. background R & C

      brx=(xe/(2.0*h*we))/(bcond*bconc)
      bry=((8.0*h)/(we*xe))/(bcond*bconc)
      er=78.54*8.85e-14
      bcx=2.0*er*we*h/xe
      bcy=xe*er*we/(8.0*h)

      open(unit=9, file=spcir, status='old')
      open(unit=10, file=fout, status='unknown')

      sflag=1
100   read(9,'(A)',err=120) buff
      l=133
110   continue
      if ((l.gt.1).and.(buff(l:l).eq.' ')) then
          l=l-1
          goto 110
      end if
      if (buff(1:4).eq.endstr) then
           sflag=0
      end if
      if ((sflag.eq.1).and.(buff(1:6).eq.'R1 1 2')) then
           write(10,600) 'R1', 1, 2, bry
        else if ((sflag.eq.1).and.(buff(1:6).eq.'R2 3 4')) then
           write(10,600) 'R2', 3, 4, bry
        else if ((sflag.eq.1).and.(buff(1:6).eq.'R3 2 3')) then
           write(10,600) 'R3', 2, 3, brx
        else if ((sflag.eq.1).and.(buff(1:6).eq.'R4 1 4')) then
           write(10,600) 'R4', 1, 4, brx
        else if ((sflag.eq.1).and.(buff(1:6).eq.'C1 1 2')) then
           write(10,610) 'C1', 1, 2, bcy*(1e12), 'pF'
        else if ((sflag.eq.1).and.(buff(1:6).eq.'C2 3 4')) then
           write(10,610) 'C2', 3, 4, bcy*(1e12), 'pF'
        else if ((sflag.eq.1).and.(buff(1:6).eq.'C3 2 3')) then
           write(10,610) 'C3', 2, 3, bcx*(1e12), 'pF'
        else if ((sflag.eq.1).and.(buff(1:6).eq.'C4 1 4')) then
           write(10,610) 'C4', 1, 4, bcx*(1e12), 'pF'
        else
           write(10,'(A)') buff(1:l)
      end if
      goto 100
120   close(unit=9)


c             Initialise the Rxxxx index

      idxr=1000
      k=1

c             Converting BI data to SPICE statements

150   do j=1, 80
        do i=1, 20
          if (k.eq.1) then
             bidata(i,j)=ccl(i,j)
           else
             bidata(i,j)=ch(i,j)
          end if
        end do
      end do
      if (k.eq.1) then
         condy=condcl
      else
         condy=condh
      end if

      do j=1, 80
        do i=1, 20
          node1=(j+9)*100-1+i
          node2=node1+1
          node4=node1+100
          node3=node2+100
             if (bidata(i,j).eq.0) then
            goto 201
          end if
             rho=1.0/(bidata(i,j)*condy)
             if (rho.lt.700000.0) then
                 resx=rx*rho
                 resy=ry*rho
            if ((idxr+3).gt.9999) then
              write(10,221) rstr,idxr,node1,node2,resy
                   idxr=idxr+1
                   write(10,221) rstr,idxr,node4,node3,resy
              idxr=idxr+1
                   write(10,221) rstr,idxr,node1,node4,resx
              idxr=idxr+1
                   write(10,221) rstr,idxr,node2,node3,resx
                   idxr=idxr+1
             else
              write(10,220) rstr,idxr,node1,node2,resy
                   idxr=idxr+1
                   write(10,220) rstr,idxr,node4,node3,resy
              idxr=idxr+1
                   write(10,220) rstr,idxr,node1,node4,resx
              idxr=idxr+1
                   write(10,220) rstr,idxr,node2,node3,resx
                   idxr=idxr+1
            end if
          end if
201       continue
        end do
      end do

c             Slice 81 using same data as slice 80

      do i=1, 20
           node1=8999+i
           node2=node1+1
           node4=node1+100
           node3=node2+100
           if (bidata(i,80).eq.0) then
          goto 215
        end if
        rho=1.0/(bidata(i,80)*condy)
        if (rho.lt.700000.0) then
                 resx=rx*rho
                 resy=ry*rho
          if ((idxr+3).gt.9999) then
            write(10,221) rstr,idxr,node1,node2,resy
                 idxr=idxr+1
                 write(10,221) rstr,idxr,node4,node3,resy
            idxr=idxr+1
                 write(10,221) rstr,idxr,node1,node4,resx
            idxr=idxr+1
                 write(10,221) rstr,idxr,node2,node3,resx
                 idxr=idxr+1
           else
            write(10,220) rstr,idxr,node1,node2,resy
                 idxr=idxr+1
                 write(10,220) rstr,idxr,node4,node3,resy
            idxr=idxr+1
                 write(10,220) rstr,idxr,node1,node4,resx
            idxr=idxr+1
                      write(10,220) rstr,idxr,node2,node3,resx
                 idxr=idxr+1
          end if
```

```
        end if
215       continue
    end do
    k=k+1
    if (k.le.2) then
       goto  150
    end if

c           Terminate output file

    WRITE(10,'(A)') endstr

220  format(a1,i4,2i8,f14.3)
221  format(a1,i5,2i8,f14.3)
600  format(a2,i2,i2,f12.2)
610  format(a2,i2,i2,f8.4,a2)

    close(unit=10)

    return
    end



    subroutine thomas(nj,aj,bj,cj,dj,
   _                                uj)

c   Subr. to solve uj by applying Thomas algorithm on the
c   tridigonal matrix

c   Declaration

    integer nj, j, dims

    parameter(dims=3000)
    double precision alpha(dims), beta(dims), fx(dims)
    double precision aj(dims),bj(dims),cj(dims),dj(dims),uj(dims)

    alpha(1)=bj(1)
    beta(1)=cj(1)/alpha(1)
    fx(1)=dj(1)/alpha(1)
    do j=2, nj-2
            alpha(j)=bj(j)-aj(j-1)*beta(j-1)
            beta(j)=cj(j)/alpha(j)
            fx(j)=(dj(j)-aj(j-1)*fx(j-1))/alpha(j)
    end do
    j=nj-1
    alpha(j)=bj(j)-aj(j-1)*beta(j-1)
    fx(j)=(dj(j)-aj(j-1)*fx(j-1))/alpha(j)
    uj(nj-1)=fx(nj-1)
    do j=nj-2, 1, -1
            uj(j)=fx(j)-beta(j)*uj(j+1)
    end do

    return
    end



    program gamatch

c   Program to index TPMCl triclinic crystal faces by using
c   standard GAs to maximise the 1/deviation of the measured &
c   observed interplanar angle

c   Declaration
c
c   popsize - popoulation size
c   pc - probability of crossover
c   pm - probability of mutation at current generation
c   pmi - initial mutation probability
c   dpm - increase in mutation probability throughout the evolution
c   m - length of chromosome
c   index   position where the chromosome split to different faces
c   v - chromosome for the current generation
c   fv - fitness value for the current generation
c   vp - chromosome for the next generation
c   ncount - generation number
c   mcount - max. generation number
c   error - accepted fitness value
c   bigf - total population fitness value
c   rwf, rwp - probability vector/matrix
c   nv - no. of faces to represent the chromosome
c   pos - position to crossover (2 pts.)
c   pe - probability to inject the best old string to the population
c   prf - no. of faces to be specified
c   pr - position (start from the beginning of the chromosome) of the
c        string to be protected from any change (= prf * 5)
c   face - MIs assigened to the faces
c   prnt - print flag (=1, print immediate results in crt)
c   disk - write flag (=1, save the immediate results in disk)
c   rmsflag - use rmsd% (spoil) = 1 or rmsd

    integer popsize, m, mcount, nv
    parameter(popsize=200, m=15, mcount=3000, nv=3)

    integer v(popsize,m),vp(popsize,m),mv(m),mg,i,j,k,l,
   _        ncount,cp(popsize),s(m),ind(20),nd,pr,disk,
   _        plane(50,2),lookup(32,4),x(50,3),pos(2),lendir,
   _        face(10,3),face1(3),binno(5),prf,prnt,rmsflag

    double precision pc,pm,fv(popsize),mfv,error,pmi,dpm,
   _                 rwf(popsize),rwp(m),ps(popsize),z,pe,
   _                 qi(popsize),dummy,obsang(50),phi(50),
   _                 as,bs,cs,alps,bets,gams,scale,penality,
   _                 spoil,rmsd

    character*40 dire
    character*55 fnamein, fnameout

c   Assign operational parameters

    pc = 0.1
    pmi = 0.15
    pm = pmi
    dpm = 0.05
    pe = 0.1
    scale = 100.0
    penality = 0.8
    error = scale/0.4
    rmsflag = 0

c   Assign file names

    dire = '/home/joule/rgc/tam/GAs/'
    lendir = 24
    fnamein(1:lendir) = dire
    fnameout(1:lendir) = dire

    fnamein(lendir+1:lendir+11) = '2xtbfdh.dat'
    fnameout(lendir+1:lendir+6) = '2a.dat'

c   Assign print/write flag

    prnt = 1
    disk = 1

c   Assign protected face(s)

    prf = 0

    face(1,1) = 0
    face(1,2) = 1
    face(1,3) = 0

    face(2,1) = -1
```

```
      face(2,2) = 1
      face(2,3) = 0

      face(3,1) = 1
      face(3,2) = 0
      face(3,3) = 0

      pr = 5 * prf

c     Get data file of observed angles & faces

      open(unit=13, file=fnamein, status='old')

      j = 1
10     read(13, *, err=20) plane(j,1),plane(j,2),obsang(j)
         j = j + 1
         goto 10
20     close(unit = 13)
      nd = j   1

c     Create data file to record the evolution process

      open(unit=14, file=fnameout, status='unknown')

c     Set up the lookup table for difference faces

      do i = 0, 31
        lookup(i+1, 1) = i
      end do
      lookup(1, 2) = -1
      lookup(1, 3) = 1
      lookup(1, 4) = 1
      lookup(2, 2) = 1
      lookup(2, 3) = -1
      lookup(2, 4) = -1
      lookup(3, 2) = -1
      lookup(3, 3) = 1
      lookup(3, 4) = 0
      lookup(4, 2) = 1
      lookup(4, 3) = -1
      lookup(4, 4) = 0
      lookup(5, 2) = 0
      lookup(5, 3) = 1
      lookup(5, 4) = 0
      lookup(6, 2) = 0
      lookup(6, 3) = -1
      lookup(6, 4) = 0
      lookup(7, 2) = 1
      lookup(7, 3) = 0
      lookup(7, 4) = 0
      lookup(8, 2) = -1
      lookup(8, 3) = 0
      lookup(8, 4) = 0
      lookup(9, 2) = 0
      lookup(9, 3) = 1
      lookup(9, 4) = 0
      lookup(10, 2) = 0
      lookup(10, 3) = -1
      lookup(10, 4) = 0
      lookup(11, 2) = 0
      lookup(11, 3) = 0
      lookup(11, 4) = 1
      lookup(12, 2) = 0
      lookup(12, 3) = 0
      lookup(12, 4) = -1
      lookup(13, 2) = 0
      lookup(13, 3) = 1
      lookup(13, 4) = 1
      lookup(14, 2) = 0
      lookup(14, 3) = -1
      lookup(14, 4) = -1
      lookup(15, 2) = 1
      lookup(15, 3) = 0
      lookup(15, 4) = 1
      lookup(16, 2) = -1
      lookup(16, 3) = 0
      lookup(16, 4) = -1
      lookup(17, 2) = 1
      lookup(17, 3) = 1
      lookup(17, 4) = 0
      lookup(18, 2) = -1
      lookup(18, 3) = -1
      lookup(18, 4) = 0
      lookup(19, 2) = 0
      lookup(19, 3) = -1
      lookup(19, 4) = 1
      lookup(20, 2) = 0
      lookup(20, 3) = 1
      lookup(20, 4) = -1
      lookup(21, 2) = -1
      lookup(21, 3) = 0
      lookup(21, 4) = 1
      lookup(22, 2) = 1
      lookup(22, 3) = 0
      lookup(22, 4) = -1
      lookup(23, 2) = -1
      lookup(23, 3) = 1
      lookup(23, 4) = 0
      lookup(24, 2) = 1
      lookup(24, 3) = -1
      lookup(24, 4) = 0
      lookup(25, 2) = 1
      lookup(25, 3) = 1
      lookup(25, 4) = 1
      lookup(26, 2) = -1
      lookup(26, 3) = -1
      lookup(26, 4) = -1
      lookup(27, 2) = -1
      lookup(27, 3) = 1
      lookup(27, 4) = 1
      lookup(28, 2) = 1
      lookup(28, 3) = -1
      lookup(28, 4) = -1
      lookup(29, 2) = -1
      lookup(29, 3) = -1
      lookup(29, 4) = 1
      lookup(30, 2) = 1
      lookup(30, 3) = 1
      lookup(30, 4) = -1
      lookup(31, 2) = -1
      lookup(31, 3) = 1
      lookup(31, 4) = -1
      lookup(32, 2) = 1
      lookup(32, 3) = -1
      lookup(32, 4) = 1

      j = 0
      do i = 5, 5*(nv - 1), 5
        j = j + 1
        ind(j) = i
      end do
      call calpar(as,bs,cs,alps,bets,gams)

c     Protect the gene(s)

      do j = 1, prf
        do k = 1, 3
          face1(k) = face(j, k)
        end do
        call face2bin(lookup, face1,
     -                          binno)
        do i = 1, popsize
          do l = (j - 1) * 5 + 1, j * 5
            v(i, l) = binno(l - (j - 1) * 5)
            vp(i, l) = v(i, l)
          end do
        end do
      end do
```

```
c     Initialise the initial state of the random number generator

      call g05ccf

c     Establish initial population

      ncount = 0
      mg = 0
      do i = 1, popsize
        do j = pr + 1, m
          call random(v(i,j), dummy)
        end do
      end do

c     Start evolution

      if (prnt.eq.1) then
        print*. '   Gen.  fitness  Total fitness'
      end if
      if (disk.eq.1) then
        write(14,*)'   Gen.  fitness  Total fitness'
      end if

100   continue
        ncount = ncount + 1
        bigf = 0.0
        do i = 1, popsize
          do j = 1, m
            s(j) = v(i, j)
          end do
          call angle(s,m,ind,nv,nd,as,bs,cs,alps,bets,
     _          gams,obsang,plane,lookup,scale,penality,
     _          spoil,rmsd,x,phi)
          if (rmsflag.eq.1) then
            fv(i) = spoil
          else
            fv(i) = rmsd
          end if
          bigf = bigf + fv(i)
          call random(j, rwf(i))
        end do
        call fmax(popsize, fv,
     _                    dummy, i)
        if ((ncount.eq.1).or.(dummy.gt.mfv)) then
          mfv = dummy
          do j = 1, m
            mv(j) = v(i,j)
          end do
          mg = ncount
        end if

c     Stochastic remainder selection

c       qi - scaled fitness
c       ps - remainder

        dummy = bigf/popsize
        j = 0
        do i = 1, popsize
          qi(i) = fv(i)/dummy
          pos(1) = int(qi(i))
          ps(i) = qi(i) - pos(1)
          do k = 1, pos(1)
            j = j + 1
            if (j.gt.popsize) then
              goto 130
            end if
            do l = 1, m
              vp(j, l) = v(i, l)
            end do
          end do
        end do
        do i = j, popsize
125       continue
          call random(k, dummy)
          cp(1) = int((popsize - 1) * dummy) + 1
          call random(k, dummy)
          if (dummy.lt.ps(cp(1))) then
            do l = 1, m
              vp(i, l) = v(cp(1), l)
            end do
            ps(cp(1)) = 0.0
          else
            goto 125
          end if
        end do

130     continue

c     Crossover

        j = 0
        do i = 1, popsize
          call random(k, rwf(i))
          if (rwf(i).lt.pc) then
            j = j + 1
            cp(j) = i
          end if
        end do
        if (real(j/2).ne.(j/2.0)) then
          j = j + 1
          call random(k, dummy)
          cp(j) = int((popsize - 1) * dummy) + 1
        end if
        do i = 2, j, 2
          do l = 1, 2
            call random(k, dummy)
            pos(l) = int((m - 1 - pr) * dummy) + 1 + pr
          end do

c           2 points crossover

          if (pos(1).lt.pos(2)) then
            do k = pos(1), pos(2)
              s(k) = vp(cp(i - 1), k)
              vp(cp(i   1), k) = vp(cp(i), k)
              vp(cp(i), k) = s(k)
            end do
          else
            do k = pr + 1, pos(2)
              s(k) = vp(cp(i - 1), k)
              vp(cp(i   1), k) = vp(cp(i), k)
              vp(cp(i), k) = s(k)
            end do
            do k = pos(1), m
              s(k) = vp(cp(i - 1), k)
              vp(cp(i   1), k) = vp(cp(i), k)
              vp(cp(i), k) = s(k)
            end do
          end if
        end do

c     Elitism

c     Inject the best old chromosome into the population

        call random(k, dummy)
        if (dummy.lt.pe) then
          do i = 1, popsize
            do j = 1, m
              s(j) = vp(i, j)
            end do
            call angle(s,m,ind,nv,nd,as,bs,cs,alps,bets,
     _          gams,obsang,plane,lookup,scale,penality,
     _          spoil,rmsd,x,phi)
            if (rmsflag.eq.1) then
              fv(i) = spoil
            else
```

```
            fv(i) = rmsd
          end if
        end do
        call fmax(popsize, fv,
     -                    z, l)
        if (mfv.ge.z) then
          call fmin(popzise, fv, dummy, j)
          do k = 1, m
            vp(j, k) = mv(k)
          end do
        else
          mfv = z
          do k = 1, m
            mv(k) = vp(l, k)
          end do
          mg = ncount
        end if
      end if

c     Mutation

      pm = pm + dpm / mcount
      do i = 1, popsize
        l = 0
        do j = pr + 1, m
          call random(k, rwp(j))
          if (rwp(j).lt.pm) then
            l = l + 1
            s(l) = j
          end if
        end do
        do j = 1, l
          call random(k, dummy)
          vp(i,s(j)) = k
        end do
      end do

c     Update the population

      do i = 1, popsize
        do j = 1, m
          v(i, j) = vp(i, j)
        end do
      end do
      if (prnt.eq.1) then
         print 1000, ncount, mfv, bigf
      end if
      if (disk.eq.1) then
         write(14,1000) ncount, mfv, bigf
      end if
      if ((ncount.lt.mcount).and.(mfv.lt.error)) then
         goto 100
      end if
200   continue

      do i = 1, m
        s(i) = mv(i)
      end do
      call angle(s,m,ind,nv,nd,as,bs,cs,alps,bets,gams,
     -           obsang,plane,lookup,scale,penality,
     -           spoil,rmsd,x,phi)
      if (rmsflag.eq.1) then
         dummy = spoil
       else
         dummy = rmsd
      end if
      if (prnt.eq.1) then
        print*, ' '
        print*, 'Minimised angular deviation/fitness value'
        print 1010, scale/rmsd, dummy
        print*, 'ncount (optm)'
        print*, ' '
        print*, mg
        print*, '     Assigned faces'
```

```
      end if

      write(14,*) ' '
      write(14,*) 'Minimised angular deviation/fitness value'
      write(14,1010) scale/rmsd, dummy
      write(14,*) 'ncount (optm)'
      write(14,*) mg
      write(14,*) ' '
      write(14,*) '     Assigned faces'
      do i = 1, nv
         if (prnt.eq.1) then
           print 1020, i, x(i,1), x(i,2), x(i,3)
         end if
         write(14,1020) i, x(i,1), x(i,2), x(i,3)
      end do
      if (prnt.eq.1) then
        print*, ' '
        print*, '     Faces     Obs.    Calc.'
      end if
      write(14,*) ' '
      write(14,*) '     Faces     Obs.    Calc.'
      do i = 1, nd
         if (prnt.eq.1) then
           print 1030,plane(i,1),plane(i,2),obsang(i),phi(i)
         end if
         write(14,1030) plane(i,1),plane(i,2),obsang(i),phi(i)
      end do

      write(14,*) ' '
      write(14,1040) 'Population size', popsize
      write(14,1040) 'Max. generation', mcount
      write(14,1050) 'Pr(crossover)  ', pc
      write(14,1050) 'Pr(mutation,0) ', pmi
      write(14,1050) 'd Pr(mutation) ', dpm
      write(14,1050) 'Pr(elitism)    ', pe
      write(14,1040) 'Protected face ', prf
      write(14,1050) 'Penalty        ', penality
      write(14,1060) 'Data file name ', fnamein
      write(14,1060) 'O/P file name  ', fnameout

      close(unit = 14)

1000  format(i5,2x,f10.2,2x,f10.2)
1010  format(f7.2,20x,f7.2)
1020  format(i5,i5,i5,i5)
1030  format(i5,i5,2x,f7.1,2x,f7.2)
1040  format(a20,2x,i7)
1050  format(a20,2x,f7.2)
1060  format(a20,a55)

      end

      subroutine angle(s,m,ind,nv,nd,as,bs,cs,alps,bets,gams,
     -                 obsang,plane,lookup,scale,penality,
     -                 spoil,rmsd,x,phi)

      integer s(1000),ind(50),m,ni,i,j,lookup(32,4),plane(50,2),
     -        ix(50),x(50,3),nv,nd
      integer bin2dec

      double precision as,bs,cs,alps,bets,gams,spoil,h(3),hp(3),
     -          rmsd,phi(50),obsang(50),penality,factor,scale
      double precision iang,rmill

c     s - chromosome in binary format
c     index - indicate the position to split s into different variables
c       e.g. [1,1,1,1,1,1,1,1,1,1] to 2 variables at position 5, ind = 5
c       [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] at 5 & 10 then ind = [5, 10]

      ni = nv - 1
      factor = 1.0

      ix(1) = bin2dec(s, 1, ind(1))
      do i = 2, ni
```

```
      j = 1 + ind(i - 1)
      ix(i) = bin2dec(s, j, ind(i))
    end do
    j = 1 + ind(ni)
    ix(ni+1) = bin2dec(s, j, m)
    do i = 1, nv - 1
      do j = i + 1, nv
        if (ix(i).eq.ix(j)) then
          factor = factor * penality
        else
          if ((ix(i).le.5).or.(ix(j).le.5)) then
            call chkeqgene(ix(i),ix(j),0,26,penality,
     _                     factor)
            call chkeqgene(ix(i),ix(j),1,27,penality,
     _                     factor)
            call chkeqgene(ix(i),ix(j),2,22,penality,
     _                     factor)
            call chkeqgene(ix(i),ix(j),3,23,penality,
     _                     factor)
            call chkeqgene(ix(i),ix(j),4,8,penality,
     _                     factor)
            call chkeqgene(ix(i),ix(j),5,9,penality,
     _                     factor)
          end if
        end if
      end do
    end do
    do i = 1, nv
      do j = 1, 32
        if (ix(i).eq.lookup(j,1)) then
          x(i,1) = lookup(j,2)
          x(i,2) = lookup(j,3)
          x(i,3) = lookup(j,4)
        end if
      end do
    end do

    spoil = 0.0
    rmsd = 0.0
    do i = 1, nd
      do j = 1, 3
        h(j)=rmill(x(plane(i,1),j))
        hp(j)=rmill(x(plane(i,2),j))
      end do
      phi(i) = iang(h,hp,as,bs,cs,alps,bets,gams)
      spoil = spoil + ((obsang(i) - phi(i))/obsang(i))**2
      rmsd = rmsd + (obsang(i) - phi(i))**2
    end do
    spoil = factor*scale/(sqrt(spoil/nd))
    rmsd = factor*scale/(sqrt(rmsd/nd))

    return
    end

    subroutine chkeqgene(ix1, ix2, a, b, penality,
     _                   factor)

c   check the repeated genes: 0,1,0; -1,1,1; -1,1,0 & their complement

    integer ix1, ix2, a, b
    double precision penality, factor

    if (((ix1.eq.a).and.(ix2.eq.b)).or.((ix2.eq.a).
     _                  and.(ix1.eq.b))) then
      factor = factor * penality
    end if

    return
    end

    integer function bin2dec(s, i1, i2)

c   convert binary number to integer
c   s - i/p binary number
c   i1 - start bit position
c   i2 - end bit position

    integer i1, i2, i, dummy, s(1000)

    dummy=0
    do i = i1, i2
      dummy = dummy + s(i)*2**(i2-i)
    end do
    bin2dec = dummy

    return
    end

    subroutine face2bin(lookup,face,
     _                  binno)

c   convert the face to the corresponding binary value in the lookup table

    integer i, lookup(32, 4), face(3), binno(5), j

    do i = 1, 32
      if (((lookup(i,2).eq.face(1)).and.(lookup(i,3).eq.face(2))).
     _    and.(lookup(i,4).eq.face(3))) then
        j = lookup(i, 1)
        goto 2000
      end if
    end do
2000  if (i.eq.33) then
      do i = 1, 5
        binno(i) = -1
      end do
      else
      do i = 0, 3
        dummy = j / 2
        j = j - 2 * dummy
        binno(5-i) = j
        j = dummy
      end do
      binno(1) = j
    end if

    return
    end

    subroutine random(numi, numd)

c   generate uniformly dirtributed random no. within 0 & 1
c   numi - integer o/p (0 or 1)
c   numd - double precision o/p

    integer numi
    double precision numd

    numd = 10.0*(g05daf(0.9,1.0)-0.9)
    if (numd.ge.0.5) then
      numi = 1
    else
      numi = 0
    end if

    return
    end

    subroutine fmax(np, y,
     _              maxi, pos)

c   find max. in an array

    integer i, np, pos
    double precision y(10000), maxi

    maxi = y(1)
```

```
      pos = 1
      do i = 2, np
        if (y(i).gt.maxi) then
           maxi = y(i)
           pos = i
        end if
      end do

      return
      end

      subroutine fmin(np, y,
     _                mini, i)

c     find min. in an array

      integer i, np, pos
      double precision y(10000), mini

      mini = y(1)
      pos = 1
      do i = 2, np
        if (y(i).lt.mini) then
           mini = y(i)
           pos = i
        end if
      end do

      return
      end

      subroutine calpar(as,bs,cs,alps,bets,gams)

c     calculate reciprocal lattice parameters for triclinic TPMCl
c
c     formular taken from:
c
c     International Table For X-ray Crystallography, vol II,
c       Kynoch Press, Brmingham, 1973, p106.
c
c     Declaration of variables
c
c     a, b, c, alp, bet, gam - unit cell parameters
c     v - unit cell volume
c     s - (alp + bet + gam)/2
c     as, bs, cs, alps, bets, gams - reciprocal lattice parameters

      double precision  a, b, c, alp, bet, gam, v, s, pi,
     _                  as, bs, cs, alps, bets, gams
      double precision invlen, invang

      pi=3.141593

c         Assign unit cell parameters
c         source:  B. Kahr & R.L. Carter, Mol. Cryst. Liq. Cryst.,
c                  219, 1992, p79.

      a = 14.1562
      b = 21.3190
      c = 13.0654
      alp = 99.9200
      bet = 92.6800
      gam = 106.1500

c         Conversion from degree to radian

      alp = alp * pi / 180.0
      bet = bet * pi / 180.0
      gam = gam * pi / 180.0

c         Calculate reciprocal lattice parameters

      s = (alp+bet+gam)/2.0
      v = 2.0*a*b*c*sqrt(sin(s)*sin(s-alp)*sin(s-bet)*sin(s-gam))

      as = invlen(b, c, alp, v)
      bs = invlen(c, a, bet, v)
      cs = invlen(a, b, gam, v)
      alps = invang(bet, gam, alp)
      bets = invang(gam, alp, bet)
      gams = invang(alp, bet, gam)

      return
      end

      double precision function invlen(x, y, ang, vol)

      double precision x, y, ang, vol

      invlen = x*y*sin(ang)/vol

      return
      end

      double precision function invang(x, y, z)
      double precision x, y, z

      invang = (cos(x)*cos(y)-cos(z))/(sin(x)*sin(y))

      return
      end

      double precision function quad(m, x, y, z, al, be, ga)

      double precision x, y, z, al, be, ga, m(3)

      quad = (m(1)*x)**2+(m(2)*y)**2+(m(3)*z)**2+2.0*m(2)*m(3)*
     _     y*z*al+2.0*m(3)*m(1)*z*x*be+2.0*m(1)*m(2)*x*y*ga

      return
      end

      double precision function iang(m1,m2,x,y,z,al,be,ga)

      double precision x,y,z,al,be,ga,q1,q2,m1(3),m2(3),pi
      double precision quad

      pi = 3.141593

      q1 = quad(m1,x,y,z,al,be,ga)
      q2 = quad(m2,x,y,z,al,be,ga)
      iang = (180.0/pi)*acos((m1(1)*m2(1)*x*x+m1(2)*m2(2)*y*y+
     _     m1(3)*m2(3)*z*z+(m1(2)*m2(3)+m1(3)*m2(2))*y*z*al+
     _     (m1(3)*m2(1)+m1(1)*m2(3))*z*x*be+
     _     (m1(1)*m2(2)+m1(2)*m2(1))*x*y*ga)/sqrt(q1*q2))

      return
      end

      double precision function rmill(a)

      integer a

      rmill=real(a)

      return
      end
```

```
      program gdis

c     Program to calculate the conc. of chloride for TPMCl reactive
c     dissolution in aq. soln. by assuming a const. flux model
c
c     J(mol/cm2s), TPMCl(s) = -kdis
c
c                kh
c     TPMCl(aq)   ====>   TPM+(aq)  +   Cl-(aq)
c       A1                 A2             A3
c
c     Declaration
c
c     Physical Parameters
c
c     nj - no. grid in y-direction
c     nk - no. grid in x-direction
c     k0 -\
c          grid no. designate reactive solid
c     k1 -/
c          grid no. designate gap
c     k2 -/
c          grid no. designate chloride electrode
c     k3 -/
c     2h - height of channel, cm
c     xe - electrode length, cm
c     xp - reactive solid length, cm
c     xg - length of gap between electrode & solid, cm
c     diffa1 - diffusion coefficient of A1, cm2/s
c     diffa3 - diffusion coefficient of A3, cm2/s
c     vf - volume flow rate, cm3/s
c     dy - y grid width
c     dx - x grid width
c     lnmean - detector response indicator (1=log mean, 0=arith. mean)
c     gemo - detector shape (1=rod, 0=foil)
c
c     Kinetic parameters
c
c     kh - homogenous hydrolysis rate constant
c     kdis - dissolution rate const. for trityl chloride

      integer nk,nj,xdimens,dimens,i,j,k,k0,k1,k2,k3,kem,lnmean,
     _        gemo,nd

      parameter(dimens=3000,xdimens=10000)
      double precision aja1(dimens),bja1(dimens),cja1(dimens),
     _                 aja3(dimens),bja3(dimens),cja3(dimens),
     _                 dj(dimens),uj(dimens),lama1(dimens),
     _                 lama3(dimens),ga3(xdimens),wdet(xdimens),
     _                 ga1j(dimens), ga3j(dimens),vfarr(20),
     _                 concarr(20)
      double precision dummy,h,d,xe,xp,xg,diffa1,diffa3,vf,dy,dx,
     _                 kh,kdis,conc

      print*, 'Enter no. of y-grid (<=3000) '
      read*, dummy
      nj=int(dummy)
      print*, 'Enter no. of x-grid (<=10000) '
      read*, dummy
      nk=int(dummy)

      h=0.102/2.0
      d=0.581
      xe=0.163
      xg=0.304
      xp=0.888
      diffa1=5.357e-6
      diffa3=1.959e-5
      lnmean=0
      gemo=1
      dy=2.0*h/nj
      dx=(xe+xp+xg)/nk
      k0=1
      k1=int(xp/dx)
      k2=int((xp+xg)/dx)
      k3=nk

      print*, 'Enter kh  (1/s) '
      read*, kh
      print*, 'Enter kdis (mol/cm2 s) '
      read*, kdis

      open(unit=9, file='/home/joule/rgc/tam/ch16.prn',
     _                                    status='old')
      j=1
10    read(9,*,err=12) vfarr(j), conc
          j=j+1
          goto 10
12    close(unit=9)
      nd=j-1

      open(unit=9,file='/home/joule/rgc/tam/conc.dat',
     _                           status='unknown')
      print*, '   Vf          Conc of Cl'

      do i=1, nd
      vf=vfarr(i)

c     Set up tridiagonal, {d} & {u} matrices

c     1. Calc. lamda & assign initial ga1j, ga3j

      dummy=(dx*d*(2.0*h)**3)/(6.0*vf*(dy**3))
      do j=1, nj-1
          lama1(j)=diffa1*dummy/(j*(2.0*h-j*dy))
          lama3(j)=diffa3*dummy/(j*(2.0*h-j*dy))
          ga1j(j)=0.0
          ga3j(j)=0.0
      end do

c     2. Calc. aj, bj & cj for A1 & A3

      do j=2, nj-1
          aja1(j-1)=-1.0*lama1(j)
          bja1(j)=2.0*lama1(j)+1.0
          cja1(j-1)=-1.0*lama1(j-1)
          aja3(j-1)=-1.0*lama3(j)
          bja3(j)=2.0*lama3(j)+1.0
          cja3(j-1)=-1.0*lama3(j-1)
      end do
      bja1(nj-1)=lama1(nj-1)+1.0
      bja3(nj-1)=lama3(nj-1)+1.0

c     3. Calc. ga1j(k) & ga3j(k) at pellet surface

      do k=k0, k1

c         For species A1

            do j=1, nj-1
              dj(j)=ga1j(j)-dy*dy*lama1(j)*kh*ga1j(j)/diffa1
            end do
          bja1(1)=1.0+lama1(1)
            dj(1)=dj(1)+lama1(1)*dy*kdis/diffa1
            call thomas(nj,aja1,bja1,cja1,dj,uj)

c         For species A3

            do j=1, nj-1
              dj(j)=ga3j(j)+dy*dy*lama3(j)*kh*ga1j(j)/diffa3
            end do
            bja3(1)=1.0+lama3(1)

c         updated ga1j as k+1 th

          do j=1, nj-1
              ga1j(j)=uj(j)
          end do
```

```
            call thomas(nj,aja3,bja3,cja3,dj,uj)
            do j=1, nj-1
              ga3j(j)=uj(j)
            end do
            ga3(k)=uj(1)

      end do

c     4. Calc. ga1j(k) & ga3j(k) at gap / electrode surface

      do k=k1+1, k3

c           For species A1

            do j=1, nj-1
              dj(j)=ga1j(j)-dy*dy*lama1(j)*kh*ga1j(j)/diffa1
            end do
          bja1(1)=1.0+lama1(1)
            call thomas(nj,aja1,bja1,cja1,dj,uj)

c           For species A3

            do j=1, nj-1
              dj(j)=ga3j(j)+dy*dy*lama3(j)*kh*ga1j(j)/diffa3
            end do
          bja3(1)=1.0+lama3(1)

c           update ga1j as k+1 th

            do j=1, nj-1
              ga1j(j)=uj(j)
            end do

            call thomas(nj,aja3,bja3,cja3,dj,uj)
            do j=1, nj-1
              ga3j(j)=uj(j)
            end do
            ga3(k)=uj(1)

      end do

      if (gemo.eq.1) then
        kem=int((k3+k2)/2)
        do k=k2, k3
          wdet(k-k2+1)=sqrt(abs((xe/2.0)**2-(dx*abs(kem-k))**2))
     _              /(xe/2.0)
        end do
      else
        do k=k2, k3
          wdet(k-k2+1)=1.0
        end do
      end if

      conc=0.0

      if (lnmean.eq.1) then
c
c       Negligible exchange current
          do k=k2, k3
            if (ga3(k).ne.0.0) then
                  conc=conc+log(abs(wdet(k-k2+1)*ga3(k)))
            end if
          end do
          conc=exp(conc/(k3-k2))
        else
c
c       Appreciable exchange current
          do k=k2, k3
            conc=conc+abs(wdet(k-k2+1)*ga3(k))
          end do
          conc=conc/(k3-k2)
      end if

c     convert mol/cm3 to molarity
      conc=conc*1000.0
      concarr(i)=conc
      write (9, 1000) vfarr(i), concarr(i)
      print 1000, vfarr(i), concarr(i)

      end do

      close(unit=9)

1000  format(1x,e12.4,2x,e12.4)

      end

      include 'thomas.for'


      program gsatd

c     Program to calculate the conc. of chloride for TPMCl reactive
c     dissolution in aq. soln. by assuming saturated surface model

c     The reactive surface is saturated with TPMCl(aq), csat,
c
c              kh
c     TPMCl(aq)  ====>  TPM+(aq) +   Cl-(aq)
c       A1              A2           A3

c     Declaration
c
c     Physical Parameters
c
c     nj - no. grid in y-direction
c     nk - no. grid in x-direction
c     k0 -\
c          grid no. designate reactive solid
c     k1 -/
c          grid no. designate gap
c     k2 -/
c          grid no. designate chloride electrode
c     k3 -/
c     2h - height of channel, cm
c     xe - electrode length, cm
c     xp - reactive solid length, cm
c     xg - length of gap between electrode & solid, cm
c     diffa1 - diffusion coefficient of A1, cm2/s
c     diffa3 - diffusion coefficient of A3, cm2/s
c     vf - volume flow rate, cm3/s
c     dy - y grid width
c     dx - x grid width
c     lnmean - detector response indicator (1 = lm, 0 = am)
c     gemo - detector shape (1 = rod, 0 = foil)
c
c     Kinetic parameters
c
c     kh - homogenous hydrolysis rate constant
c     csat - concentration of the sat'd reactive surface

      integer nk,nj,nd,dimens,kdim,i,j,k,k0,k1,k2,k3,kem,lnmean,
     _      gemo

      parameter(dimens=500, kdim=50000)
      double precision aja1(dimens),bja1(dimens),cja1(dimens),
     _            aja3(dimens),bja3(dimens),cja3(dimens),
     _            dj(dimens),uj(dimens),lama1(dimens),
     _            lama3(dimens),ga3(kdim),ga1j(dimens),
     _            ga3j(dimens),vfarr(30),expconc(30),
     _            wdet(kdim),rmsd
      double precision dummy,h,d,xe,xp,xg,diffa1,diffa3,vf,dy,dx,
     _            kh,csat,conc

c     No. of y-grid
      nj=500
c     No. of x-grid
      nk=60000
```

```
      h=0.102/2.0
      d=0.581

      xp=0.915
      xg=0.311
      xe=0.159
      diffa1=5.357e-6
      diffa3=1.959e-5
      lnmean=0
      gemo=1
      dy=2.0*h/nj
      dx=(xe+xp+xg)/nk
      k0=1
      k1=int(xp/dx)
      k2=int((xp+xg)/dx)
      k3=nk

      kh=80.0
      csat=8.64e-11

      open(unit=9, file='/home/joule/rgc/tam/Sattpmcl/ch17.prn',
     _     status='old')

      i=1
10     read(9, *, err=12) vfarr(i), expconc(i)
         i = i + 1
         goto 10
12     close(unit=9)
      nd = i - 1

      if (gemo.eq.1) then
        kem=int((k3+k2)/2)
        do k=k2, k3
          wdet(k-k2+1)=sqrt(abs((xe/2.0)**2-(dx*abs(kem-k))**2))
     _                /(xe/2.0)
        end do
       else
        do k=k2, k3
          wdet(k-k2+1)=1.0
        end do
      end if

      open(unit=9, file='/home/joule/rgc/tam/Sattpmcl/ch17.d9',
     _     status='unknown')

      print*, '   Vf          [Cl, expt]       [Cl, calc]'
      rmsd=0.0

      do i=1, nd
      vf=vfarr(i)

c     Set up tridiagonal, {d} & {u} matrices

c     1. Calc. lamda & assign initial ga1j, ga3j

      dummy=(dx*d*(2.0*h)**3)/(6.0*vf*(dy**3))
      do j=1, nj-1
                lama1(j)=diffa1*dummy/(j*(2.0*h-j*dy))
                lama3(j)=diffa3*dummy/(j*(2.0*h-j*dy))
                ga1j(j)=0.0
                ga3j(j)=0.0
      end do

c     2. Calc. aj, bj & cj for A1 & A3

      do j=2, nj-1
                aja1(j-1)=-1.0*lama1(j)
                bja1(j)=2.0*lama1(j)+1.0
                cja1(j-1)=-1.0*lama1(j-1)
                aja3(j-1)=-1.0*lama3(j)
                bja3(j)=2.0*lama3(j)+1.0
                cja3(j-1)=-1.0*lama3(j-1)
      end do
      bja1(nj-1)=lama1(nj-1)+1.0
      bja3(nj-1)=lama3(nj-1)+1.0

c     3. Calc. ga1j(k) & ga3j(k) at electrode surface

      do k=1, nk

c           For species A1

            do j=1, nj-1
                      dj(j)=ga1j(j)-dy*dy*lama1(j)*kh*ga1j(j)/diffa1
            end do
            if ((k.ge.k0).and.(k.le.k1)) then
              dj(1)=ga1j(1)-dy*dy*lama1(1)*kh*ga1j(1)/diffa1+
     _          lama1(1)*csat
              bja1(1)=1.0+2.0*lama1(1)
             else
                      bja1(1)=1.0+lama1(1)
            end if
            call thomas(nj,aja1,bja1,cja1,dj,uj)
            do j=1, nj-1
                      ga1j(j)=uj(j)
            end do

c           For species A3

            do j=1, nj-1
                      dj(j)=ga3j(j)+dy*dy*lama3(j)*kh*ga1j(j)/diffa3
            end do
            bja3(1)=1.0+lama3(1)
            call thomas(nj,aja3,bja3,cja3,dj,uj)
            do j=1, nj-1
                      ga3j(j)=uj(j)
            end do
            ga3(k)=uj(1)

      end do

      conc=0
      if (lnmean.eq.1) then
c
c     Negligible exchange current
        do k=k2, k3
          if (ga3(k).ne.0.0) then
            conc=conc+log(abs(wdet(k-k2+1)*ga3(k)))
           end if
        end do
        conc=exp(conc/(k3-k2))
       else
c
c     Appreciable exchange current
        do k=k2, k3
          conc=conc+abs(wdet(k-k2+1)*ga3(k))
        end do
        conc=conc/(k3-k2)
      end if
      conc=1000.0*conc

      rmsd=rmsd+(conc-expconc(i))**2

      write (9, 1000) vfarr(i), expconc(i), conc
      print 1000, vfarr(i), expconc(i), conc

      end do

      rmsd=sqrt(rmsd/nd)
      print*, 'RMSD    ', rmsd

      close(unit=9)

 1000  format(1x,e12.4,2x,e12.4,2x,e12.4)

      end

      include 'thomas.for'
```

```
      program gdiscl

c     Program to calculate the conc. of chloride for TPMCl reactive
c     dissolution in aq. soln. by assuming a heterogeneous reaction model

c     J(mol/cm2s),        = -kdis
c                 Cl, y=0
c
c     TPMCl(aq),        TPM+(aq),     Cl-(aq)
c       A1                A2            A3

c     Declaration
c
c     Physical Parameters
c
c     nj - no. grid in y-direction
c     nk - no. grid in x-direction
c     k0 -\
c          grid no. designate reactive solid
c     k1 -/
c          grid no. designate gap
c     k2 -/
c          grid no. designate chloride electrode
c     k3 -/
c     2h - height of channel, cm
c     xe - electrode length, cm
c     xp - reactive solid length, cm
c     xg - length of gap between electrode & solid, cm
c     diffa3 - diffusion coefficient of A3, cm2/s
c     vf - volume flow rate, cm3/s
c     dy - y grid width
c     dx - x grid width
c     lnmean - detector response indicator (1=log mean, 0=arith. mean)
c     gemo - detector shape (1=rod, 0=foil)
c
c     Kinetic parameters
c
c     kdis - const. flux for chloride

      integer nk,nj,xdimens,dimens,gemo,i,j,k,k0,k1,k2,k3,kem,
     _        lnmean,nd

      parameter(dimens=3000,xdimens=10000)
      double precision aja3(dimens),bja3(dimens),cja3(dimens),
     _                 dj(dimens),uj(dimens),lama3(dimens),
     _                 wdet(xdimens),ga3(xdimens),ga3j(dimens),
     _                 vfarr(20),concarr(20)
      double precision dummy,h,d,xe,xp,xg,diffa3,vf,dy,dx,kdis,conc

      print*, 'Input no. of y-grid (<=3000)'
      read*, dummy
      nj=int(dummy)
      print*, 'Input no. of x-grid (<=10000)'
      read*, dummy
      nk=int(dummy)
      h=0.102/2.0
      d=0.581
      xe=0.163
      xg=0.304
      xp=0.888
      diffa3=1.959e-5
      lnmean=0
      gemo=1
      dy=2.0*h/nj
      dx=(xe+xp+xg)/nk
      k0=1
      k1=int(xp/dx)
      k2=int((xp+xg)/dx)
      k3=nk
      print*, 'Input kdis (mol/cm2 s) '
      read*, kdis

      open(unit=9,file='/home/Joule/rgc/tam/ch16.prn',
     _                status='old')
      j=1
10    read(9,*,err=12) vfarr(j), conc
            j=j+1
            goto 10
12    close(unit=9)
      nd=j-1

      open(unit=9, file='conc.dat', status='unknown')
      print*, '   Vf          Conc of Cl'

      do i=1, nd
      vf=vfarr(i)

c     Set up tridiagonal, {d} & {u} matrices

c     1. Calc. lamda & assign initial ga3j

      dummy=(dx*d*(2.0*h)**3)/(6.0*vf*(dy**3))
      do j=1, nj-1
               lama3(j)=diffa3*dummy/(j*(2.0*h-j*dy))
               ga3j(j)=0.0
      end do

c     2. Calc. aj, bj & cj for A3

      do j=2, nj-1
               aja3(j-1)=-1.0*lama3(j)
               bja3(j)=2.0*lama3(j)+1.0
               cja3(j-1)=-1.0*lama3(j-1)
      end do
      bja3(nj-1)=lama3(nj-1)+1.0

c     3. Calc. ga3j(k) at electrode surface

      do k=1, nk

c           For species A3

                  do j=1, nj-1
                     dj(j)=ga3j(j)
                  end do
               bja3(1)=1.0+lama3(1)
                  if ((k.ge.k0).and.(k.le.k1)) then
                     dj(1)=ga3j(1)+lama3(1)*dy*kdis/diffa3
                  end if
                  call thomas(nj,aja3,bja3,cja3,dj,uj)
                  do j=1, nj-1
                     ga3j(j)=uj(j)
                  end do
                  ga3(k)=uj(1)

      end do

      if (gemo.eq.1) then
          kem=int((k3+k2)/2)
          do k=k2, k3
             wdet(k-k2+1)=sqrt(abs((xe/2.0)**2-(dx*abs(kem-k))**2))
     _                    /(xe/2.0)
          end do
       else
          do k=k2, k3
             wdet(k-k2+1)=1.0
          end do
      end if

      conc=0.0
      if (lnmean.eq.1) then
c
c     Negligible exchange current
          do k=k2, k3
             if (ga3(k).ne.0.0) then
                   conc=conc+log(abs(wdet(k-k2+1)*ga3(k)))
             end if
          end do
```

```
        conc=exp(conc/(k3-k2))
       else
c
c      Appreciable exchange current
        do k=k2, k3
           conc=conc+abs(wdet(k-k2+1)*ga3(k))
        end do
        conc=conc/(k3-k2)
      end if

c      convert conc in unit of mol/cm3 to molarity

      conc=conc*1000.0
      concarr(i)=conc
      write (9, 1000) vfarr(i), concarr(i)
      print 1000, vfarr(i), concarr(i)

      end do

      close(unit=9)

1000  format(1x,e12.4,2x,e12.4)

      end

      include 'thomas.for'


      program optdis

c     Program to optimise kdis using NAG Library routine (E04JAF)
c     -- quasi-Newton method -- for the heterogeneous reaction
c     model.
c
c     Exp. data is stored in ASCII data file
c
c     format in data file '*.prn'
c     1st col. - vf (ml/s)
c     2nd col. - [Cl] (M)

c     Assuming a const. flux of chloride release from TPMCl

c     J(mol/cm2s),        = -kdis
c                 Cl, y=0
c
c      TPMCl(aq),       TPM+(aq),      Cl-(aq)
c       A1               A2             A3

c     Declaration
c
c     Physical Parameters
c
c     nj - no. grid in y-direction
c     nk - no. grid in x-direction
c     k0 -\
c         grid no. designate reactive solid
c     k1 -/
c         grid no. designate gap
c     k2 -/
c         grid no. designate chloride electrode
c     k3 -/
c     2h - height of channel, cm
c     xe - electrode length, cm
c     xp - reactive solid length, cm
c     xg - length of gap between electrode & solid, cm
c     diffa3 - diffusion coefficient of A3, cm2/s
c     vf - volume flow rate, cm3/s
c     dy - y grid width
c     dx - x grid width
c     lnmean - detector response indicator (1=log mean, 0=arith. mean)
c     gemo - detector shape (1=rod, 0=foil)
c
c     Kinetic parameters
c     kdis - const. flux for chloride
c
c     ng - no. of repeative calc. in optimisation
c     guess - initial guess generated from random number generator
c     optk - optimised kdis in each calc.

      integer nk,nj,nd,np,ng,gemo,i,k0,k1,k2,k3,lnmean,liw,lw

      parameter(ng=3, np=1, liw=80, lw=80)

      integer ibound, iw(liw), ifail
      double precision h,d,xe,xp,xg,diffa3,dy,dx,kdis,optk(ng)
      double precision vfarr(30),expconc(30),calconc(30)
      double precision bl(np),bu(np),x(np),mf,ww(lw),guess(ng)

      common /param1/ dx,dy,h,d,diffa3,xe
      common /param2/ nj,nk,k0,k1,k2,k3,nd,lnmean,gemo
      common /result/  expconc, calconc, vfarr

c           assign numerical constant in the computation

c     No. of y-grid (<=3000)
      nj=300
c     No. of x-grid (<=3000)
      nk=500

      h=0.102/2.0
      d=0.581
      xe=0.163
      xg=0.304
      xp=0.888
      diffa3=1.959e-5
      lnmean=0
      gemo=1
      dy=2.0*h/nj
      dx=(xe+xp+xg)/nk
      k0=1
      k1=int(xp/dx)
      k2=int((xp+xg)/dx)
      k3=nk

      open(unit=9, file='/home/joule/rgc/tam/conc.dat',
     _           status='old')
      i=1
10    read(9, *, err=12) vfarr(i), expconc(i)
            i=i+1
            goto 10
12    close(unit=9)
      nd=i-1

c           set constraints for kdis

      ibound=0
      bl(1)=1.0e-10
      bu(1)=1.0e-4

      print*, 'Const. chloride flux model '
      print*, 'lnmean = ', lnmean
      print*, 'Gemo = ', gemo
      print*, 'nj = ', nj
      print*, 'nk = ', nk
      print*, 'Upper bound of parameter = ', bu(1)
      print*, 'Lower bound of parameter = ', bl(1)

      kdis=0.0

      do i=1, ng

      print*, 'No. of iteration ', i

c           x - initial guess of kdis

      guess(i)=g05daf(0.1,1.0)*(10**g05daf(log10(bl(1)),log10(bu(1))))
      x(1)=guess(i)
```

```
c    invoke the optimiser via NAG numerical library call (E04JAF)

     ifail=-1
     call e04jaf(np,ibound,bl,bu,x,mf,iw,liw,ww,lw,ifail)

c    assign optimised results to the corresponding variables

     optk(i)=x(1)
     kdis=kdis+optk(i)

     print*, 'Function value  ', mf
     print*, 'Optimised kdis  ', optk(i)
     print*, 'Initial guess ', guess(i)
     print*, 'IFAIL ', ifail
     print*

     end do

     kdis=kdis/ng
     print*, 'Mean kdis ', kdis
     print*
     print*, 'From mean kdis '
     call funct1(np, kdis, mf)
     open(unit=9, file='/home/joule/rgc/tam/result.dat',
    _        status='unknown')
     print*, '      Vf          Conc(expt)    Conc(opt)'
     do i=1, nd
        write (9, 990) vfarr(i), calconc(i)
        print 1000, vfarr(i), expconc(i), calconc(i)
     end do
     close(unit=9)
     print*, 'RMSD ', mf

990  format(1x,e13.4,2x,e13.4)
1000 format(1x,e13.4,2x,e13.4,2x,e13.4)

     end

     subroutine funct1(np, x, fc)

     integer nk,nj,dimens,nd,np,gemo,i,j,k,k0,k1,k2,k3,kem,lnmean

     parameter(dimens=3000)

     double precision aja3(dimens),bja3(dimens),cja3(dimens),
    _                 dj(dimens),uj(dimens),lama3(dimens),ga3(dimens),
    _                 ga3j(dimens),wdet(dimens)
     double precision dummy,h,d,diffa3,vf,dy,dx,kdis,conc,x(np),fc
     double precision vfarr(30),expconc(30),calconc(30),xe

     common /param1/ dx,dy,h,d,diffa3,xe
     common /param2/ nj,nk,k0,k1,k2,k3,nd,lnmean,gemo
     common /result/  expconc, calconc, vfarr

     kdis=x(1)
     rmsd=0.0

     do i=1, nd
     vf=vfarr(i)

c    Set up tridiagonal, {d} & {u} matrices

c    1. Calc. lamda & assign initial ga3j

     dummy=(dx*d*(2.0*h)**3)/(6.0*vf*(dy**3))
     do j=1, nj-1
              lama3(j)=diffa3*dummy/(j*(2.0*h-j*dy))
              ga3j(j)=0.0
     end do

c    2. Calc. aj, bj & cj for A3

     do j=2, nj-1
              aja3(j-1)=-1.0*lama3(j)
              bja3(j)=2.0*lama3(j)+1.0
              cja3(j-1)=-1.0*lama3(j-1)
     end do
     bja3(nj-1)=lama3(nj-1)+1.0

c    3. Calc. ga3j(k) at electrode surface

     do k=1, nk

c         For species A3

              do j=1, nj-1
                       dj(j)=ga3j(j)
              end do
                       bja3(1)=1.0+lama3(1)
              if ((k.ge.k0).and.(k.le.k1)) then
                       dj(1)=ga3j(1)+lama3(1)*dy*kdis/diffa3
              end if
              call thomas(nj,aja3,bja3,cja3,dj,uj)
              do j=1, nj-1
                       ga3j(j)=uj(j)
              end do
              ga3(k)=uj(1)

     end do

     if (gemo.eq.1) then
        kem=int((k3+k2)/2)
        do k=k2, k3
          wdet(k-k2+1)=sqrt(abs((xe/2.0)**2-(dx*abs(kem-k))**2))
    _                 /(xe/2.0)
        end do
      else
        do k=k2, k3
          wdet(k-k2+1)=1.0
        end do
     end if

     conc=0
     if (lnmean.eq.1) then
c
c    Negligible exchange current
        do k=k2, k3
           if (ga3(k).ne.0.0) then
                    conc=conc+log(abs(wdet(k-k2+1)*ga3(k)))
           end if
        end do
        conc=exp(conc/(k3-k2))
      else
c
c    Appreciable exchange current
        do k=k2, k3
           conc=conc+abs(wdet(k-k2+1)*ga3(k))
        end do
        conc=conc/(k3-k2)
     end if
c
c     change conc in unit of mol/cm3 to molarity

     conc=1000.0*conc

     calconc(i)=conc
     rmsd=rmsd+(conc-expconc(i))**2

     end do

     rmsd=sqrt(rmsd/nd)
     fc=rmsd

     return
     end

     include 'thomas.for'
```

```
/*  tmpd.c   program to optimise the extinction coefficient of TMPD  */

#include <stdio.h>
#include <malloc.h>
#include <math.h>
#include <string.h>

#define NS 10

char fs1[] = "  Cnt    Lin      mf          x(n)     Sigma(n)....\n";
char fs2[] = "%4i  %4i   %6.3e";
char fs3[] = "  %6.3e  %6.3e";
char dire[] = "/home/joule/rgc/tam/Tmpd/";
char name1[] = "tmpd11";
char ext1[] = ".prn";
char name2[] = "op11i5";
char ext2[] = ".dat";
int nj, nk, nd, k1, k2;
double c0, h, d, dx, dy, diff, xd, we, *ilim, *vf, *abso, *thabs;

double funfcn();
double fmax();
double fmin();
void fgrad();
void fmind();
void fdfpstep();
void bsort();
void sumx();
void fnmin();
void thomas();

void main()
{
      char filname[50];
      FILE *fp;

      double *x, *iexpt, tol, xe, xg, e, rmsd, dummy;
      int  *cnt, prnt, np, i;

cnt = (int *) malloc(sizeof(int));
x = (double *) malloc(NS*sizeof(double));
thabs = (double *) malloc(30*sizeof(double));
vf = (double *) malloc(30*sizeof(double));
abso = (double *) malloc(30*sizeof(double));
ilim = (double *) malloc(30*sizeof(double));
iexpt = (double *) malloc(30*sizeof(double));

d = 0.601;
xd = 0.407;
xg = 0.103;
xe = 0.394;
we = 0.402;
diff = 6.77e-6;

e = 12000000.0;

nj = 1000;
nk = 1000;

h = 0.0819/2.0;
c0 = 2.192e-6;

dy = 2.0*h/nj;
dx = (xd+xg+xe)/nk;
k1 = xe/dx;
k2 = (xe+xg)/dx;

strcpy(filname, dire);
strcat(filname, name1);
strcat(filname, ext1);
fp = fopen(filname, "r");
nd = 0;
while (!feof(fp))
   {
      fscanf(fp, "%lf %lf %lf\n", &vf[nd], &iexpt[nd], &abso[nd]);
      nd++;
     }

close(fp);

np = 1;
x[1] = e;
tol = 1.0e-6;
prnt = 1;
fmind(np, tol, prnt, x, cnt);
printf("No. count  %i\n", cnt[0]);
printf("Optimised e        %6.4e\n", x[1]);
printf("Initial guess      %6.4e\n", e);
rmsd = funfcn(x);
printf("function value     %6.4e\n", rmsd);
printf("Input file name    %s\n", filname);
strcpy(filname, dire);
strcat(filname, name2);
strcat(filname, ext2);
printf("Output file name   %s\n", filname);
fp = fopen(filname, "w");

for (i = 0; i < nd; i++)
        {
  fprintf(fp, "%6.2e  %11.5f  %11.3f   %11.4e   %11.4e\n", vf[i],
        thabs[i], abso[i], ilim[i], iexpt[i]);
  printf("%6.2e  %11.5f  %11.3f   %11.4e   %11.4e\n", vf[i],
        thabs[i], abso[i], ilim[i], iexpt[i]);
        }

close(fp);
free(cnt);
free(x);
free(iexpt);
}

double funfcn(x)
      double *x;
{
      int i, j, k, kem;
      double *aj, *bj, *cj, *dj, *uaj, *ubj, *lam, *fcdy, *wdet,
        dummy, e, rmsd, wxdx;

      e = x[1];

      aj = (double *) malloc(nj*sizeof(double));
      bj = (double *) malloc(nj*sizeof(double));
      cj = (double *) malloc(nj*sizeof(double));
      dj = (double *) malloc(nj*sizeof(double));
      uaj = (double *) malloc(nj*sizeof(double));
      ubj = (double *) malloc(nj*sizeof(double));
      lam = (double *) malloc(nj*sizeof(double));
      fcdy = (double *) malloc(nk*sizeof(double));
      wdet = (double *) malloc(nk*sizeof(double));

      rmsd = 0.0;
      for (i = 0; i < nd; i++)
          {

/*  Set up tridiagonal, {d} & {u} matrices  */

/*  1. Calc. lamda & assign initial dj      */

          dummy = (diff*dx*d*pow(2.0*h,3.0))/(6.0*vf[i]*pow(dy,3.0));
          for (j = 0; j < nj   1; j++)
            {
             lam[j] = dummy/((j+1.0)*(2.0*h-(j+1.0)*dy));
             uaj[j] = c0;
             ubj[j] = 0.0;
            }

/*  2. Calc. aj, bj & cj                    */
```

```
            for (j = 1; j < nj - 1; j++)
              {
                 aj[j   1] = -1.0*lam[j];
                 bj[j] = 2.0*lam[j]+1.0;
                 cj[j - 1] = -1.0*lam[j - 1];
              }
            bj[nj - 2] = lam[nj - 2] + 1.0;

/*  3. Calc. g(k) at electrode                  */

      ilim[i] = 0.0;

            for (k = 0; k < k1; k++)
              {
            bj[0] = 2.0*lam[0]+1.0;
                  for (j = 0; j < nj   1; j++)
                        dj[j] = uaj[j];
                  thomas(nj,aj,bj,cj,dj,uaj);
              ilim[i] += uaj[0];
            bj[0] = lam[0]+1.0;
            for (j = 0; j < nj - 1; j++)
               dj[j] = ubj[j];
            dj[0] += (lam[0]*uaj[0]);
            thomas(nj,aj,bj,cj,dj,ubj);
          }

       ilim[i] = ilim[i]*dx*diff*96480.0*we/dy;

/*  4. Calc. g(k) at gap & detector       */

      bj[0] = lam[0]+1.0;
      for (k = k1; k < k2; k++)
            {
          for (j = 0; j < nj - 1; j++)
             dj[j] = ubj[j];
          thomas(nj,aj,bj,cj,dj,ubj);
            }
      for (k = k2; k < nk; k++)
        {
          for (j = 0; j < nj - 1; j++)
             dj[j] = ubj[j];
          thomas(nj,aj,bj,cj,dj,ubj);
          fcdy[k] = 0.0;
          for (j = 0; j < nj - 1; j++)
             fcdy[k] += ubj[j];
          fcdy[k] = e*dy*(2.0*fcdy[k] - ubj[0] - ubj[nj   2])/2.0;
        }

/*    5. mean value of fcdy along x-direction  */

      thabs[i] = 0.0;
      wxdx = 0.0;

/*      5. Weighted average                 */

      kem = (nk - 1 + k2) / 2;
      for (k = k2; k < nk; k++)
        {
         wdet[k] = sqrt(fabs(pow(xd/2.0,2.0)-pow(dx*abs(kem-k),2.0)))
                  /(xd/2.0);
         wxdx += wdet[k];
         fcdy[k] = wdet[k]*fcdy[k];
        }
       wxdx = (dx/2.0)*(2.0*wxdx - wdet[k2] - wdet[nk - 1]);

/*      int / W(x) dx    */

      thabs[i] = thabs[i] / wxdx;

      rmsd += pow((thabs[i] - abso[i]), 2.0);
      }
      free(aj);
      free(bj);
      free(cj);
      free(dj);
      free(uaj);
      free(ubj);
      free(lam);
      free(fcdy);
      free(wdet);
      rmsd = pow((rmsd/nd), 0.5);
      return rmsd;

 }

void thomas(nj, aj, bj, cj, dj, uj)

      int nj;
      double *aj, *bj, *cj, *dj, *uj;
{
            int j;
            double *alpha, *beta, *fx;

            alpha = (double *) malloc(nj*sizeof(double));
            beta = (double *) malloc(nj*sizeof(double));
            fx = (double *) malloc(nj*sizeof(double));

            alpha[0] = bj[0];
            beta[0] = cj[0]/alpha[0];
            fx[0] = dj[0]/alpha[0];
            for (j = 1; j < nj - 2; j++)
              {
                 alpha[j] = bj[j] - aj[j   1]*beta[j - 1];
                 beta[j] = cj[j]/alpha[j];
                 fx[j] = (dj[j]-aj[j-1]*fx[j-1])/alpha[j];
              }
            j = nj - 2;
            alpha[j] = bj[j]-aj[j-1]*beta[j-1];
            fx[j] = (dj[j]-aj[j-1]*fx[j-1])/alpha[j];
            uj[nj - 2] = fx[nj   2];
            for (j = nj   3; j > -1; j--)
                 uj[j] = fx[j]-beta[j]*uj[j+1];
            free(fx);
            free(alpha);
            free(beta);
   }


void fmind(np, tol, prnt, x, cnt)

      double *x, tol;
      int  *cnt, prnt, np;

{

      double *grad, *grad1, *lamder, *mf, *ddd, *sigma, grama[NS],
            hk[NS][NS], result1[NS], result2, result3[NS][NS],
            result4a[NS][NS], result4[NS][NS], result5[NS][NS],dummy,
            result6[NS][NS], result7[NS][NS], denum, dtol, reltol, beta;
      int *lcnt, i, iterate, j, k, nan;

      grad = (double *) malloc(NS*sizeof(double));
      grad1 = (double *) malloc(NS*sizeof(double));
      lamder = (double *) malloc(sizeof(double));
      mf = (double *) malloc(sizeof(double));
      ddd = (double *) malloc(NS*sizeof(double));
      sigma = (double *) malloc(NS*sizeof(double));
      lcnt = (int *) malloc(sizeof(int));

      dtol = tol*1.0e-3;
      if (tol <= 1.0e-30)
         tol = 1.0e-30;
      cnt[0] = 0;
      iterate = 1;
      if (prnt == 1)
         printf(fs1);
      fgrad(np, x, dtol, grad);
```

```
for (i = 1; i < np+1; i++)
  for (j = 1; j < np+1; j++)
    if (i == j)
       hk[i][j] = 1.0;
      else
       hk[i][j] = 0.0;
 do
        {
      for (i = 1; i < np+1; i++)
            {
        ddd[i] = 0.0;
        for (j = 1; j < np+1; j++)
            ddd[i] -= hk[i][j]*grad[j];
                    }
      fdfpstep(np, x, ddd, lamder, mf, lcnt);
      for (i = 1; i < np+1; i++)
            {
         sigma[i] = lamder[0]*ddd[i];
         x[i] += sigma[i];
                    }
      reltol = tol * fabs(fmax(np,x));
      nan = 0;
      for (i = 1; i < np+1; i++)
            if (isnan(sigma[i]) == 1)
          nan = 1;
      if (cnt[0] > 0)
          dummy = fabs(fmin(np,ddd));
       else
          dummy = reltol + 1.0;
      if ((nan==1)||(dummy<reltol)||(fabs(fmin(np,sigma))<reltol))
        iterate = 0;
       else
        {
          fgrad(np, x, dtol, grad1);
          denum = 0.0;
          for (i = 1; i < np+1; i++)
            {
              grama[i] = grad1[i] - grad[i];
              grad[i] = grad1[i];
    denum += sigma[i]*grama[i];
  }
for (i = 1; i < np+1; i++)
  {
    result1[i] = 0.0;
    for (j = 1; j < np+1; j++)
        result1[i] += grama[j]*hk[j][i];
                }
result2 = 0.0;
for (i = 1; i < np+1; i++)
   result2 += result1[i]*grama[i];
beta = 1.0 + result2/denum;
for (i = 1; i < np+1; i++)
  for (j = 1; j < np+1; j++)
    {
      result3[i][j] = beta*sigma[i]*sigma[j];
      result4[i][j] = grama[i]*sigma[j];
      result4a[i][j] = sigma[i]*grama[j];
                    }
for (k = 1; k < np+1; k++)
  for (i = 1; i < np+1; i++)
    {
      result5[k][i] = 0.0;
      result6[k][i] = 0.0;
      for (j = 1; j < np+1; j++)
        {
          result5[k][i] += result4a[k][j]*hk[j][i];
          result6[k][i] += hk[k][j]*result4[j][i];
                        }
    }
for (i = 1; i < np+1; i++)
  for (j = 1; j < np+1; j++)
    {
      result4[i][j] = result3[i][j] - result5[i][j] -
                      result6[i][j];
                hk[i][j] +=result4[i][j]/denum;
              }
                    }
            cnt[0]++;
            if (prnt == 1)
                    {
              printf(fs2, cnt[0], lcnt[0], mf[0]);
              for (i = 1; i < np+1; i++)
                printf(fs3, x[i], sigma[i]);
              printf("\n");
                    }
          }
      while (iterate == 1);

      free(mf);
      free(grad);
      free(grad1);
      free(lamder);
      free(lcnt);
      free(ddd);
      free(sigma);
      }

void fgrad(np, x, tol, grad)

      int np;
      double *x, *grad, tol;

{
      int i, j;
      double *a, *b, h;

      a = (double *) malloc(NS*sizeof(double));
      b = (double *) malloc(NS*sizeof(double));

      for (i = 1; i < np+1; i++)
            {
        if ((fabs(x[i]) > 1.0e-30)&&(x[i] != 0.0))
          h = fabs(x[i])*tol;
          else
           h = tol;
        for (j = 1; j < np+1; j++)
          {
            a[j] = x[j];
            b[j] = x[j];
          }
        a[i] = x[i] + h;
        b[i] = x[i] - h;
        grad[i] = (funfcn(a) - funfcn(b))/(2.0*h);
        if ((x[i] == 0.0)||(grad[i] == 0.0))
          grad[i] = grad[i] + tol;
      }
    free(a);
    free(b);
}

void fdfpstep(np, x, ddd, lamder, mf, lcnt)

      int np, *lcnt;
      double *x, *lamder, *mf, *ddd;

{
      int  cnts, counter, i, idx, im, il, iterate, nl, *order,
          refine, mcount;
      double *dummy,*dummy1,f[NS], l[NS], m[NS], tf[NS], mf1, u1, u2,
           u3, sh, lh, ep, v1, a, b, c, lm, ym, ldiffer, mdiffer, z;

      order = (int *) malloc(NS*sizeof(int));
      dummy = (double *) malloc(NS*sizeof(double));
      dummy1 = (double *) malloc(NS*sizeof(double));

      mcount = 13;
      u1 = fabs(fmax(np, x));
      u2 = fabs(fmin(np, x));
```

```
if (u2 == 0.0)
   u2 = u1;
u3 = 0.0;
for (i = 1; i < np+1; i++)
    u3 += fabs(ddd[i]);
u3 /= np;
mf[0] = 0.0;
lamder[0] = 0.0;
sh = u2/(u3*33.3333);
lh = 10.0*u2/u3;
ep = u2/(u3*100000.0);
f[1] = funfcn(x);
l[1] = 0.0;
mf1 = f[1];
v1 = 0.0;
sumx(np, x, sh, ddd, dummy);
f[2] = funfcn(dummy);
l[2] = sh;
if (f[1] < f[2])
    l[3] = -1.0*sh;
  else
    l[3] = 2.0*sh;
sumx(np, x, l[3], ddd, dummy);
f[3] = funfcn(dummy);
counter = 1;
cnts = 1;
lcnt[0] = 3;
iterate = 1;
do
        {
    a = (pow(l[2],2.0)-pow(l[3],2.0))*f[1]+(pow(l[3],2.0)-
        pow(l[1],2.0))*f[2]+(pow(l[1],2.0)-pow(l[2],2.0))*f[3];
    b = (l[2]-l[3])*f[1]+(l[3]-l[1])*f[2]+(l[1]-l[2])*f[3];
    c = (l[1]-l[2])*(l[2]-l[3])*(l[3]-l[1]);
    if ((isnan(a)==1)||(isnan(b)==1)||(isnan(c)==1)||(b==0)||(c==0))
            {
        iterate = 0;
        l[4] = lm;
        sumx(np, x, lm, ddd, dummy);
        f[4] = funfcn(dummy);
        fnmin(f, l, mf, lamder);
      }
    else
        {
      lm = 0.5*a/b;
      ym = b/c;
      refine = 0;
      for (i = 1; i < 4; i++)
        dummy[i] = fabs(lm - l[i]);
      bsort(3, dummy, order);
      ldiffer = dummy[1];
      il = order[1];
      mdiffer = dummy[3];
      im = order[3];
      if ((ym > 0.0)||(ldiffer > lh))
        {
          if (ym > 0.0)
              idx = im;
            else
              idx = il;
          m[1] = l[idx] - lh;
          m[2] = l[idx] + lh;
          sumx(np, x, m[1], ddd, dummy);
          tf[1] = funfcn(dummy);
          sumx(np, x, m[2], ddd, dummy);
          tf[2] = funfcn(dummy);
          if (tf[1] > tf[2])
            {
              l[im] = m[2];
              f[im] = tf[2];
            }
           else
                        {
              l[im] = m[1];
              f[im] = tf[1];
                        }
        refine = 1;
              }
if ((refine==0)&&(ym<0.0)&&(ep>ldiffer))
              {
    lamder[0] = (lm + l[il])/2.0;
    z = lamder[0];
    sumx(np, x, z, ddd, dummy);
    mf[0] = funfcn(dummy);
    iterate = 0;
    refine = 1;
 }
if ((refine==0)&&(ym<0.0)&&(ldiffer>ep)&&(ldiffer<lh))
              {
    for (i = 1; i < 4; i++)
                  {
        dummy[i] = f[i];
        dummy1[i] = l[i];
      }
    dummy1[4] = lm;
    bsort(3, dummy, order);
    n1 = order[3];
    sumx(np, x, lm, ddd, dummy);
    tf[1] = funfcn(dummy);
    bsort(4, dummy1, order);
    if ((lm==dummy[1])||(lm==dummy1[4]))
      {
        l[n1] = lm;
        f[n1] = tf[1];
      }
    if (lm==dummy1[2])
      {
        dummy[1] = f[order[3]];
        dummy[2] = f[order[4]];
        if (dummy[1]>dummy[2])
          {
            l[order[3]] = lm;
            f[order[3]] = tf[1];
          }
        else
                        {
            l[order[4]] = lm;
            f[order[4]] = tf[1];
                        }
                      }
    if (lm==dummy1[3])
              {
        dummy[1] = f[order[1]];
        dummy[2] = f[order[2]];
        if (dummy[1]>dummy[2])
          {
            l[order[1]] = lm;
            f[order[1]] = tf[1];
          }
        else
                        {
            l[order[2]] = lm;
            f[order[2]] = tf[1];
          }
      }
            }
if (iterate == 1)
              {
    counter++;
    if (counter>3)
      {
        for (i = 1; i < 4; i++)
            dummy[i] = f[i];
        bsort(3, dummy, order);
        lamder[0] = l[order[1]];
        mf[0] = dummy[1];
        if (mf[0] > mf1)
                        {
```

```
                        if (v1!=0.0)
                          {
                             mf[0] = mf1;
                             lamder[0] = v1;
                          }
                         iterate = 0;
                                        }
                      else
                        {
                          mf1 = mf[0];
                          v1 = lamder[0];
                          counter = 1;
                        }
                                        }
                cnts++;
                lcnt[0] += 2;
                if (cnts>mcount)
                  {
                     iterate = 0;
                     l[4] = lm;
                     sumx(np, x, lm, ddd, dummy);
                     f[4] = funfcn(dummy);
                     fnmin(f, l, mf, lamder);
                                        }
                  }
        }
            }
    while (iterate == 1);

    free(order);
    free(dummy);
    free(dummy1);
}

void sumx(np, x, ah, ddd, dummy)

    int np;
    double *x, *ddd, *dummy, ah;

{
    int i;

    for (i = 1; i < np+1; i++)
      dummy[i] = x[i] + ah*ddd[i];
}

void fnmin(f, l, minf, minl)

    double f[NS], l[NS], *minf, *minl;

{
    int i;

    minf[0] = f[1];
    minl[0] = l[1];
    for (i = 2; i < 5; i++)
      if (f[i] < minf[0])
             {
          minf[0] = f[i];
          minl[0] = l[i];
              }
}

void bsort(n, num, order)

    int *order, n;
    double *num;

{
    int i, j, io;
    double dum;

    for (i = 1; i < n+1; i++)
      order[i] = i;

    for (i = 1; i < n+1; i++)
       for (j = i; j < n+1; j++)
          if (num[i] > num[j])
                        {
              dum = num[i];
              num[i] = num[j];
              num[j] = dum;
              io = order[i];
              order[i] = order[j];
              order[j] = io;
             }
}

double fmax(np, y)

    int np;
    double *y;

{
    int i;
    double dummy;

    if (np == 1)
        return y[1];
      else
        {
          dummy = y[1];
          for (i = 2; i < np+1; i++)
             if (y[i] > dummy)
                 dummy = y[i];
          return dummy;
              }
   }

double fmin(np, y)

    int np;
    double *y;

{
    int i;
    double dummy;

    if (np == 1)
       return y[1];
     else
             {
        dummy = y[1];
        for (i = 2; i < np+1; i++)
           if (y[i] < dummy)
               dummy = y[i];
        return dummy;
             }
  }
```

```
/*  mvcl2.c  program to calculate the conc. of MV+. & (MV+)2  */
/*  thomas() is the same as optmpd.c                          */

#include <stdio.h>
#include <malloc.h>
#include <string.h>
#include <math.h>

char dire[] = "/home/joule/rgc/tam/MvCl2/";
char name1[] = "mvcl26";
char ext1[] = ".prn";
char name2[] = "mvcl26b";
char ext2[] = ".dat";

void thomas();

void main ()
{

      char filname[50];
      FILE *fp;

      int     nk, nj, nd, i, j, k, k1, k2, kem, iterate;
      double  *aj, *bj, *cj, *dj, *uaj, *ubj, *ucj, *ub0j, *ub1j,
              *uc0j, *lam, *const, *fbdy, *fcdy;
      double  dummy, h, d, xd, xg, xe, w, diff, dy, dx, vf[30], abso[30],
              thabsb[30], thabsc[30], iexpt[30], c0, eb, ec, wdet, keqm,
              kf, errb, errc, temp, tol, reltol;

      d = 0.601;
      xd = 0.407;
      xg = 0.103;
      xe = 0.394;
      w = 0.402;
      diff = 7.13e-6;
      eb = 3.49e7;
      ec = 8.97e7;

      nj = 500;
      nk = 50000;

      h = 0.0831/2.0;
      c0 = 7.723e-6;

      keqm = 720000.0;
      kf = 1.0e5;
      reltol = 0.01;
      tol = 1.0e-50;

      dy = 2.0*h/nj;
      dx = (xd+xg+xe)/nk;
      k1 = xe/dx;
      k2 = (xe+xg)/dx;

      strcpy(filname, dire);
      strcat(filname, name1);
      strcat(filname, ext1);
      fp = fopen(filname, "r");
      nd = 0;
      while (!feof(fp))
         {
          fscanf(fp, "%lf %lf %lf\n", &vf[nd], &iexpt[nd], &abso[nd]);
          nd++;
          }

      close(fp);

      strcpy(filname, dire);
      strcat(filname, name2);
      strcat(filname, ext2);
      fp = fopen(filname, "w");
      aj = (double *) malloc(nj*sizeof(double));
      bj = (double *) malloc(nj*sizeof(double));
      cj = (double *) malloc(nj*sizeof(double));
      dj = (double *) malloc(nj*sizeof(double));
      uaj = (double *) malloc(nj*sizeof(double));
      ubj = (double *) malloc(nj*sizeof(double));
      ucj = (double *) malloc(nj*sizeof(double));
      ub0j = (double *) malloc(nj*sizeof(double));
      ub1j = (double *) malloc(nj*sizeof(double));
      uc0j = (double *) malloc(nj*sizeof(double));
      lam = (double *) malloc(nj*sizeof(double));
      const = (double *) malloc(nj*sizeof(double));
      fbdy = (double *) malloc(nk*sizeof(double));
      fcdy = (double *) malloc(nk*sizeof(double));

     for (i = 0; i < nd; i++)
            {

/*   Set up tridiagonal, {d} & {u} matrices  */

/*   1. Calc. lamda & assign initial dj      */

            dummy = (diff*dx*d*pow(2.0*h,3.0))/(6.0*vf[i]*pow(dy,3.0));
            for (j = 0; j < nj - 1; j++)
              {
            lam[j] = dummy/((j+1.0)*(2.0*h-(j+1.0)*dy));
            const[j] = dy*dy*lam[j]/diff;
            uaj[j] = c0;
              ubj[j] = 0.0;
            ucj[j] = 0.0;
               }

/*   2. Calc. aj, bj & cj                    */

            for (j = 1; j < nj - 1; j++)
              {
                 aj[j - 1] = -1.0*lam[j];
                 bj[j] = 2.0*lam[j]+1.0;
                 cj[j   1] = -1.0*lam[j - 1];
              }
            bj[nj - 2] = lam[nj - 2] + 1.0;

/*   3. Calc. g(k) at electrode              */

      dummy = 0.0;
            for (k = 0; k < k1; k++)
             {
              bj[0] = 2.0*lam[0]+1.0;
                for (j = 0; j < nj - 1; j++)
                          dj[j] = uaj[j];
                thomas(nj,aj,bj,cj,dj,uaj);
              dummy += uaj[0];
              bj[0] = lam[0]+1.0;
              for (j = 0; j < nj   1; j++)
                  ub0j[j] = ubj[j];
              iterate = 1;
              while (iterate)
             {
            for (j = 0; j < nj   1; j++)
              {
               ub1j[j] = ubj[j];
               uc0j[j] = ucj[j];
               dj[j] = ub0j[j]-2.0*const[j]*(kf*pow(ub0j[j],2.0)-
                       (kf/keqm)*ucj[j]);
                     }
              dj[0] += (lam[0]*uaj[0]);
              thomas(nj,aj,bj,cj,dj,ubj);
              for (j = 0; j < nj - 1; j++)
                dj[j] = ucj[j] + const[j]*(kf*pow(ubj[j],2.0)-
                       (kf/keqm)*ucj[j]);
              thomas(nj,aj,bj,cj,dj,ucj);
              errb = 0.0;
              errc = 0.0;
              for (j = 0; j < nj - 1; j++)
                {
                  if (fabs(ubj[j]) > tol)
                    {
```

```
                temp = fabs((ubj[j] - ub1j[j])/ubj[j]);
                if (temp > errb)
                  errb = temp;
                                }
              if (fabs(ucj[j]) > tol)
                                {
                temp = fabs((ucj[j] - uc0j[j])/ucj[j]);
                if (temp > errc)
                  errc = temp;
                                }
            }
          if ((errc < reltol) & (errb < reltol))
             iterate = 0;
          }
      }

    dummy = dummy * dx * diff * 96480.0 * w / dy;

/*  4. Calc. g(k) at gap & detector       */

    bj[0} = lam[0]+1.0;
    for (k = k1; k < k2; k++)
          {
        for (j = 0; j < nj - 1; j++)
           ub0j[j] = ubj[j];
        iterate = 1;
        while (iterate)
          {
          for (j = 0; j < nj   1; j++)
            {
             ub1j[j] = ubj[j];
             uc0j[j] = ucj[j];
             dj[j] = ub0j[j]-2.0*const[j]*(kf*pow(ub0j[j],2.0)-
                    (kf/keqm)*ucj[j]);
                   }
           thomas(nj,aj,bj,cj,dj,ubj);
           for (j = 0; j < nj - 1; j++)
             dj[j] = ucj[j] + const[j]*(kf*pow(ubj[j],2.0)-
                    (kf/keqm)*ucj[j]);
           thomas(nj,aj,bj,cj,dj,ucj);
           errb = 0.0;
           errc = 0.0;
           for (j = 0; j < nj   1; j++)
            {
              if (fabs(ubj[j]) > tol)
                {
                  temp = fabs((ubj[j] - ub1j[j])/ubj[j]);
                  if (temp > errb)
                    errb = temp;
                                }
              if (fabs(ucj[j]) > tol)
                                {
                 temp = fabs((ucj[j] - uc0j[j])/ucj[j]);
                 if (temp > errc)
                   errc = temp;
                                }
            }
           if ((errc < reltol) & (errb < reltol))
              iterate = 0;
           }
            }
    for (k = k2; k < nk; k++)
      {
        for (j = 0; j < nj - 1; j++)
           ub0j[j] = ubj[j];
        iterate = 1;
        while (iterate)
          {
          for (j = 0; j < nj - 1; j++)
            {
             ub1j[j] = ubj[j];
             uc0j[j] = ucj[j];
             dj[j] = ub0j[j]-2.0*const[j]*(kf*pow(ub0j[j],2.0)-
                    (kf/keqm)*ucj[j]);
                   }
           thomas(nj,aj,bj,cj,dj,ubj);
           for (j = 0; j < nj - 1; j++)
             dj[j] = ucj[j] + const[j]*(kf*pow(ubj[j],2.0)-
                    (kf/keqm)*ucj[j]);
           thomas(nj,aj,bj,cj,dj,ucj);
           errb = 0.0;
           errc = 0.0;
           for (j = 0; j < nj - 1; j++)
            {
              if (fabs(ubj[j]) > tol)
                {
                  temp = fabs((ubj[j] - ub1j[j])/ubj[j]);
                  if (temp > errb)
                    errb = temp;
                                }
              if (fabs(ucj[j]) > tol)
                                {
                 temp = fabs((ucj[j] - uc0j[j])/ucj[j]);
                 if (temp > errc)
                   errc = temp;
                                }
            }
           if ((errc < reltol) & (errb < reltol))
              iterate = 0;
           }
        fbdy[k] = 0.0;
        fcdy[k] = 0.0;
        for (j = 0; j < nj   1; j++)
          {
            fbdy[k] += ubj[j];
            fcdy[k] += ucj[j];
          }
        fbdy[k] = eb*dy*(2.0*fbdy[k] - ubj[0] - ubj[nj - 2])/2.0;
        fcdy[k] = ec*dy*(2.0*fcdy[k] - ucj[0] - ucj[nj - 2])/2.0;
      }

/*      5. mean value of fbdy & fcdy along x-direction  */

      thabsb[i] = 0.0;
      thabsc[i] = 0.0;

/*        5a. Weighted average                */

      kem = (nk - 1 + k2) / 2;
      for (k = k2; k < nk; k++)
       {
            wdet = sqrt(fabs(pow(xd/2.0,2.0)-pow(dx*abs(kem-k),2.0)))
                  /(xd/2.0);
           fbdy[k] = fbdy[k]*wdet;
           fcdy[k] = fcdy[k]*wdet;
         }

/*        5b. Arith. mean                     */

      for (k = k2; k < nk; k++)
            {
        thabsb[i] += fbdy[k];
        thabsc[i] += fcdy[k];
        }
      thabsb[i] = (dx/2.0)*(2.0*thabsb[i] - fbdy[k2] - fbdy[nk - 1])/xd;
      thabsc[i] = (dx/2.0)*(2.0*thabsc[i] - fcdy[k2] - fcdy[nk - 1])/xd;

      printf("%6.2e  %11.3e  %11.3e  %11.3e  %6.3f  %11.3e\n",
           vf[i], thabsb[i], thabsc[i], thabsc[i]/thabsb[i],
           abso[i], dummy);
      fprintf(fp, "%6.2e  %11.5e  %6.3f\n", vf[i], thabsc[i]/thabsb[i],
            abso[i]);

              }
      free(aj);
      free(bj);
      free(cj);
      free(dj);
```

```
        free(uaj);
        free(ubj);
        free(ucj);
        free(ub0j);
        free(ub1j);
        free(uc0j);
        free(lam);
        free(const);
        free(fbdy);
        free(fcdy);
        close(fp);
}


/*  optmvcl2.c  program to optimise the extinction coefficients of MV+
    & (MV+)2    */
/*  fmind() and the related procedures are the same as optmpd.c   */

#include <stdio.h>
#include <malloc.h>
#include <math.h>
#include <string.h>

#define NS 10

char fs1[] = "  Cnt   Lin     mf          x(n)    Sigma(n)....\n";
char fs2[] = "%4i  %4i   %6.3e";
char fs3[] = "   %6.3e  %6.3e";
char dire[] = "/home/joule/rgc/tam/MvCl2/";
char name1m[] = "5mm13";
char name1e[] = "mvcl23t";
char ext1[] = ".prn";
char name2[] = "op95e5mv23";
char ext2[] = ".dat";
int nd;
double *vf, *absonf, *absoss, *thabsnf, *thabsss, *concm, *concd, *scale;

double funfcn();
double fmax();
double fmin();
void fgrad();
void fmind();
void fdfpstep();
void bsort();
void sumx();
void fnmin();

void main()
{
        char filname[50];
        FILE *fp;
        double *x,tol,ebhd,echd,emnf,ednf,emss,edss,rmsd,dummy;
        int  *cnt, prnt, maxdat, np, i;

        ebhd = 3.49e7;
        echd = 8.97e7;

        maxdat = 30;

        cnt = (int *) malloc(sizeof(int));
        x = (double *) malloc(NS*sizeof(double));
        scale = (double *) malloc(2*sizeof(double));
        vf = (double *) malloc(maxdat*sizeof(double));
        absonf = (double *) malloc(maxdat*sizeof(double));
        absoss = (double *) malloc(maxdat*sizeof(double));
        thabsnf = (double *) malloc(maxdat*sizeof(double));
        thabsss = (double *) malloc(maxdat*sizeof(double));
        concm = (double *) malloc(maxdat*sizeof(double));
        concd = (double *) malloc(maxdat*sizeof(double));

/*      read exptal data   */

        strcpy(filname, dire);
        strcat(filname, name1e);
        strcat(filname, ext1);
        fp = fopen(filname, "r");
        nd = 0;
        while (!feof(fp))
          {
          fscanf(fp, "%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf\n",
                &dummy,&dummy,&dummy,&dummy,&absoss[nd],
                &dummy,&absonf[nd],&dummy,&dummy,&dummy,
                &dummy,&dummy);
          nd++;
            }

        close(fp);

/*      read model data   */

        strcpy(filname, dire);
        strcat(filname, name1m);
        strcat(filname, ext2);
        printf("%s\n", filname);
        fp = fopen(filname, "r");
        nd = 0;
        while (!feof(fp))
           {
           fscanf(fp, "%lf %lf %lf %lf %lf %lf\n",&vf[nd],&thabsnf[nd],
                  &thabsss[nd],&dummy,&dummy,&dummy);

/*           correct for int W(x) dx       */

             thabsnf[nd] = thabsnf[nd] * 0.407/0.3196;
             thabsss[nd] = thabsss[nd] * 0.407/0.3196;

/*           convert to concentration      */

             concm[nd] = thabsnf[nd]/ebhd;
             concd[nd] = thabsss[nd]/echd;
             nd++;
           }
        close(fp);

        np = 4;

/*      x1 = em,395; x2 = ed,395; x3 = em,366; x4 = ed,366  */

        scale[1] = 0.1;
        scale[2] = 0.1;
        emnf = 1.0e7;

/*      emnf = 3.5e7;   */

        ednf = emnf;
        edss = 6.5e7;

/*      edss = 7.0e7;   */

        emss = edss;
        x[1] = emnf;
        x[2] = ednf;
        x[3] = emss;
        x[4] = edss;

        tol = 1.0e-6;
        prnt = 1;
        fmind(np, tol, prnt, x, cnt);
        printf("No. count  %i\n", cnt[0]);
        printf("Optimised em,395   %6.4e\n", x[1]);
        printf("Initial guess      %6.4e\n", emnf);
        printf("Optimised ed,395   %6.4e\n", x[2]*scale[1]);
        printf("Initial guess      %6.4e\n", ednf*scale[1]);
        printf("Optimised em,366   %6.4e\n", x[3]*scale[2]);
        printf("Initial guess      %6.4e\n", emss*scale[2]);
        printf("Optimised ed,366   %6.4e\n", x[4]);
        printf("Initial guess      %6.4e\n", edss);
```

```
    rmsd = funfcn(x);
    printf("function value     %6.4e\n", rmsd);
    printf("Input file name    %s\n", filname);
    strcpy(filname, dire);
    strcat(filname, name2);
    strcat(filname, ext2);
    printf("Output file name   %s\n", filname);
    fp = fopen(filname, "w");

    for (i = 0; i < nd; i++)
          {
      fprintf(fp, "%6.2e  %11.5f  %11.5f   %11.3f   %11.3f\n", vf[i],
          thabsnf[i], thabsss[i], absonf[i], absoss[i]);
      printf("%6.2e  %11.5f  %11.5f   %11.3f   %11.3f\n", vf[i],
          thabsnf[i], thabsss[i], absonf[i], absoss[i]);
          }

    close(fp);
    free(cnt);
    free(x);
}

double funfcn(x)

    double *x;
{
    int i;
    double  emnf, ednf, emss, edss, rmsdnf, rmsdss, rmsd;

    emnf = x[1];
    ednf = x[2]*scale[1];
    emss = x[3]*scale[2];
    edss = x[4];

    rmsdnf = 0.0;
    rmsdss = 0.0;
    for (i = 0; i < nd; i++)
        {
      thabsnf[i] = emnf * concm[i] + ednf * concd[i];
      thabsss[i] = emss * concm[i] + edss * concd[i];
      rmsdnf += pow((absonf[i] - thabsnf[i]), 2.0);
      rmsdss += pow((absoss[i] - thabsss[i]), 2.0);
    }

    rmsd = pow((rmsdnf/nd), 0.5) + pow((rmsdss/nd), 0.5);
    return rmsd;

}

/*  optup.c  program to optimise the dyeing rate constant k_DOH
          Process is followd by using UV-visible spectroscopy
  thomas(), fmind() and related procedures are given in optmpd.c */

#include <stdio.h>
#include <malloc.h>
#include <math.h>
#include <string.h>

#define NS 10

char fs1[] = "  Cnt   Lin     mf          x(n)     Sigma(n)....\n";
char fs2[] = "%4i  %4i   %6.3e";
char fs3[] = "   %6.3e  %6.3e";
char dire[] = "/home/joule/rgc/tam/Pbmxr1/";
char name1[] = "ucsnh5";
char ext1[] = ".prn";
char name2[] = "optucsnh5";
char ext2[] = ".dat";
int nj, nk, nd, k1, k2;
double c0, e, h, d, dx, dy, diff, bgabs, xd, *vf, *abso, *thabs;

double funfcn();
double fmax();
double fmin();
void fgrad();
void fmind();
void fdfpstep();
void bsort();
void sumx();
void fnmin();
void thomas();

void main()
{
    char filname[50];
    FILE *fp;

    double *x, tol, xg, xc, khet, rmsd, dummy;
    int  *cnt, prnt, np, i;

    cnt = (int *) malloc(sizeof(int));
    x = (double *) malloc(NS*sizeof(double));
    thabs = (double *) malloc(30*sizeof(double));
    vf = (double *) malloc(30*sizeof(double));
    abso = (double *) malloc(30*sizeof(double));

    d = 0.617;
    xd = 0.302;

/*    DClOH            */
    diff = 4.20e-6;

/*    0.35 M Na2SO4, 0.1 M Na2CO3   */
/*      e = 6911854.0;*/
/*    0.35 M Na2SO4                 */
    e = 6665689.0;
/*    0.1 M Na2CO3                  */
/*      e = 7360065.0;*/

    nj = 3000;
    nk = 5000;

/*      khet = 4.5e-5;  */

    khet = 0.1e-5;

    h = 0.0664/2.0;
    c0 = 3.022e-6;
    xg = 0.151;
    xc = 1.992;

    dy = 2.0*h/nj;
    dx = (xd+xg+xc)/nk;
    k1 = xc/dx;
    k2 = (xc+xg)/dx;
    bgabs = 2.0*h*c0*e;

    strcpy(filname, dire);
    strcat(filname, name1);
    strcat(filname, ext1);
    fp = fopen(filname, "r");
    nd = 0;
    while (!feof(fp))
      {
        fscanf(fp, "%lf %lf %lf\n", &vf[nd], &abso[nd], &dummy);
        nd++;
      }

    close(fp);

    np = 1;
    x[1] = khet;
    tol = 1.0e-5;
    prnt = 1;
    fmind(np, tol, prnt, x, cnt);
    printf("No. count  %i\n", cnt[0]);
    printf("Optimised khet     %6.4e\n", x[1]);
    printf("Initial guess      %6.4e\n", khet);
```

```
        rmsd = funfcn(x);
        printf("function value     %6.4e\n", rmsd);
        printf("Input file name    %s\n", filname);
        strcpy(filname, dire);
        strcat(filname, name2);
        strcat(filname, ext2);
        printf("Output file name   %s\n", filname);
        fp = fopen(filname, "w");

        for (i = 0; i < nd; i++)
                {
           fprintf(fp, "%6.2e  %11.5f  %11.5f\n", vf[i], thabs[i], abso[i]);
           printf("%6.2e  %11.5f  %11.5f\n", vf[i], thabs[i], abso[i]);
                }

        close(fp);
        free(cnt);
        free(x);

   }

double funfcn(x)

        double *x;
{
        int i, j, k;
        double *aj, *bj, *cj, *dj, *uj, *lam, *fcdy, dummy, khet, rmsd;

        khet = x[1];

        aj = (double *) malloc(nj*sizeof(double));
        bj = (double *) malloc(nj*sizeof(double));
        cj = (double *) malloc(nj*sizeof(double));
        dj = (double *) malloc(nj*sizeof(double));
        uj = (double *) malloc(nj*sizeof(double));
        lam = (double *) malloc(nj*sizeof(double));
        fcdy = (double *) malloc(nk*sizeof(double));

        rmsd = 0.0;
        for (i = 0; i < nd; i++)
                {

/*   Set up tridiagonal, {d} & {u} matrices  */

/*   1. Calc. lamda & assign initial dj      */

                dummy = (diff*dx*d*pow(2.0*h,3.0))/(6.0*vf[i]*pow(dy,3.0));
                for (j = 0; j < nj - 1; j++)
                  {
                     lam[j] = dummy/((j+1.0)*(2.0*h-(j+1.0)*dy));
                     uj[j] = c0;
                  }

/*   2. Calc. aj, bj & cj                    */

                for (j = 1; j < nj - 1; j++)
                  {
                     aj[j   1] = -1.0*lam[j];
                     bj[j] = 2.0*lam[j]+1.0;
                     cj[j - 1] = -1.0*lam[j - 1];
                  }
                bj[nj   2] = lam[nj   2] + 1.0;

/*   3. Calc. g(k) at cloth                  */

        bj[0] = lam[0]*(2.0*khet*dy+diff)/(khet*dy+diff)+1.0;
                for (k = 0; k < k1; k++)
                  {
                     for (j = 0; j < nj - 1; j++)
                                dj[j] = uj[j];
                     thomas(nj,aj,bj,cj,dj,uj);
                  }

/*   4. Calc. g(k) at gap & detector         */

                bj[0] = 1.0+lam[0];
                for (k = k1; k < k2; k++)
                        {
                   for (j = 0; j < nj - 1; j++)
                        dj[j] = uj[j];
                   thomas(nj,aj,bj,cj,dj,uj);
                        }
                for (k = k2; k < nk; k++)
                  {
                     for (j = 0; j < nj - 1; j++)
                        dj[j] = uj[j];
                     thomas(nj,aj,bj,cj,dj,uj);
                     fcdy[k] = 0.0;
                     for (j = 0; j < nj - 1; j++)
                                fcdy[k] = fcdy[k] + uj[j];
                     fcdy[k] = e*dy*(2.0*fcdy[k] - uj[0] - uj[nj - 2])/2.0;
                  }

/*   5. mean value of fcdy along x-direction  */

                thabs[i] = 0.0;

/*      Arith. mean                          */

                for (k = k2; k < nk; k++)
                   thabs[i] = thabs[i] + fcdy[k];
                thabs[i] = (dx/2.0)*(2.0*thabs[i] - fcdy[k2] - fcdy[nk - 1])/xd;
                thabs[i] = bgabs - thabs[i];
                rmsd += pow((thabs[i] - abso[i]), 2.0);
                }
                free(aj);
                free(bj);
                free(cj);
                free(dj);
                free(uj);
                free(lam);
                free(fcdy);
                rmsd = pow((rmsd/nd), 0.5);
                return rmsd;
   }


/*  optuphyd.c  program to optimise the dyeing rate constant k_DCl
                Reaction is followed by using UV-vis spectroscopy
    thomas(), fmind() and related procedures are given in optmpd.c */


#include <stdio.h>
#include <malloc.h>
#include <math.h>
#include <string.h>

#define NS 10

char fs1[] = "  Cnt   Lin     mf          x(n)    Sigma(n)....\n";
char fs2[] = "%4i  %4i   %6.3e";
char fs3[] = "  %6.3e  %6.3e";
char dire[] = "/home/joule/rgc/tam/Pbmxr1/";
char name1[] = "s2sam6c";
char ext1[] = ".prn";
char name2[] = "opts2sam6c";
char ext2[] = ".dat";
int nj, nk, nd, k1, k2;
double c0,e,eoh,h,d,dx,dy,diffcl,diffoh,khomo,kup,bgabs,xd;
double *t,*ct,*vf,*abso,*thabs;

double funfcn();
double fmax();
double fmin();
void fgrad();
void fmind();
void fdfpstep();
void bsort();
```

```
void sumx();
void fnmin();
void thomas();

void main()
{
      char filname[50];
      FILE *fp;

      double *x, tol, xg, xc, khet, rmsd, dummy;
      int  *cnt, prnt, np, ndp, i;

       ndp = 30;
       cnt = (int *) malloc(sizeof(int));
       x = (double *) malloc(NS*sizeof(double));
       thabs = (double *) malloc(ndp*sizeof(double));
       vf = (double *) malloc(ndp*sizeof(double));
       abso = (double *) malloc(ndp*sizeof(double));
       t = (double *) malloc(ndp*sizeof(double));
       ct = (double *) malloc(ndp*sizeof(double));

       d = 0.617;
       xd = 0.302;

/*     DCl2              */
       diffcl = 4.14e-6;

/*     DClOH             */
       diffoh = 4.20e-6;

/*     0.35 M Na2SO4, 0.1 M Na2CO3   */
       e = 6911854.0;
/*     0.35 M Na2SO4                 */
/*       e = 6665689.0;*/
/*     0.1 M Na2CO3                  */
/*       e = 7360065.0;*/

       eoh = e;

       nj = 3000;
       nk = 5000;

/*     0.1 M Na2CO3                  */
/*       khomo = 8.34e-5;*/

/*     0.35 M Na2SO4, 0.1 M Na2CO3   */
       khomo = 9.13e-5;

       kup = 1.424e-5;

/*       khet = 4.5e-5;  */

       khet = 2.0e-5;

       h = 0.0717/2.0;
       c0 = 1.031e-6;
       xg = 0.152;
       xc = 2.011;

       dy = 2.0*h/nj;
       dx = (xd+xg+xc)/nk;
       k1 = xc/dx;
       k2 = (xc+xg)/dx;
       bgabs = 2.0*h*c0*e;
       strcpy(filname, dire);
       strcat(filname, name1);
       strcat(filname, ext1);

       printf("%s\n", filname);

       fp = fopen(filname, "r");
       nd = 0;
       while (!feof(fp))
        {
         fscanf(fp, "%lf %lf %lf %lf\n",&vf[nd],&abso[nd],
              &dummy,&t[nd]);
         nd++;
          }

       close(fp);

       np = 1;
       x[1] = khet;
       tol = 1.0e-5;
       prnt = 1;
       fmind(np, tol, prnt, x, cnt);
       printf("No. count  %i\n", cnt[0]);
       printf("Optimised khet     %6.4e\n", x[1]);
       printf("Initial guess      %6.4e\n", khet);
       rmsd = funfcn(x);
       printf("function value     %6.4e\n", rmsd);
       printf("Input file name    %s\n", filname);
       strcpy(filname, dire);
       strcat(filname, name2);
       strcat(filname, ext2);
       printf("Output file name   %s\n", filname);
       fp = fopen(filname, "w");

       for (i = 0; i < nd; i++)
             {
         fprintf(fp, "%6.2f    %10.3e    %6.2e  %11.5f  %11.5f\n", t[i],
                    ct[i], vf[i], thabs[i], abso[i]);
         printf("%6.2f     %10.3e     %6.2e  %11.5f  %11.5f\n", t[i],
                    ct[i], vf[i], thabs[i], abso[i]);
             }

       close(fp);
       free(cnt);
       free(x);

}

double funfcn(x)
       double *x;
{

/*     DCl  + OH-   ---khomo--->  D(OH) + Cl-        */
/*      A                          B                 */

       int i, j, k;
       double *aaj,*baj,*caj,*daj,*uaj,*lama,*facdy,*abj,*bbj,*cbj,
            *dbj,*ubj,*lamb,*fbcdy,coht,dummy,khet,rmsd,thabsoh;

       khet = x[1];

       aaj = (double *) malloc(nj*sizeof(double));
       baj = (double *) malloc(nj*sizeof(double));
       caj = (double *) malloc(nj*sizeof(double));
       daj = (double *) malloc(nj*sizeof(double));
       uaj = (double *) malloc(nj*sizeof(double));
       lama = (double *) malloc(nj*sizeof(double));
       facdy = (double *) malloc(nk*sizeof(double));

       abj = (double *) malloc(nj*sizeof(double));
       bbj = (double *) malloc(nj*sizeof(double));
       cbj = (double *) malloc(nj*sizeof(double));
       dbj = (double *) malloc(nj*sizeof(double));
       ubj = (double *) malloc(nj*sizeof(double));
       lamb = (double *) malloc(nj*sizeof(double));
       fbcdy = (double *) malloc(nk*sizeof(double));

       rmsd = 0.0;
       for (i = 0; i < nd; i++)
             {

       ct[i] = c0*exp(-1.0*khomo*t[i]*60.0);
       coht = c0 - ct[i];
```

```
/*  Set up tridiagonal, {d} & {u} matrices  */

/*  1. Calc. lamda & assign initial dj      */

        dummy = (dx*d*pow(2.0*h,3.0))/(6.0*vf[i]*pow(dy,3.0));
        for (j = 0; j < nj - 1; j++)
          {
            lama[j] = dummy/((j+1.0)*(2.0*h-(j+1.0)*dy));
        lamb[j] = diffoh*lama[j];
        lama[j] = diffcl*lama[j];
            uaj[j] = ct[i];
        ubj[j] = coht;
          }

/*  2. Calc. aj, bj & cj                    */

        for (j = 1; j < nj - 1; j++)
          {
            aaj[j - 1] = -1.0*lama[j];
            baj[j] = 2.0*lama[j]+1.0;
            caj[j - 1] = -1.0*lama[j   1];
            abj[j - 1] = -1.0*lamb[j];
            bbj[j] = 2.0*lamb[j]+1.0;
            cbj[j - 1] = -1.0*lamb[j   1];
          }
        baj[nj - 2] = lama[nj - 2] + 1.0;
        bbj[nj - 2] = lamb[nj - 2] + 1.0;

/*  3. Calc. g(k) at cloth                  */

    baj[0] = lama[0]*(2.0*khet*dy+diffcl)/(khet*dy+diffcl)+1.0;
    bbj[0] = lamb[0]*(2.0*kup*dy+diffoh)/(kup*dy+diffoh)+1.0;
        for (k = 0; k < k1; k++)
          {
            for (j = 0; j < nj - 1; j++)
        {
                    daj[j] = uaj[j];
          dbj[j] = ubj[j];
                    }
              thomas(nj,aaj,baj,caj,daj,uaj);
          thomas(nj,abj,bbj,cbj,dbj,ubj);
            }

/*  4. Calc. g(k) at gap & detector         */

    baj[0] = 1.0+lama[0];
    bbj[0] = 1.0+lamb[0];
    for (k = k1; k < k2; k++)
            {
      for (j = 0; j < nj   1; j++)
        {
          daj[j] = uaj[j];
          dbj[j] = ubj[j];
        }
      thomas(nj,aaj,baj,caj,daj,uaj);
      thomas(nj,abj,bbj,cbj,dbj,ubj);
            }
    for (k = k2; k < nk; k++)
     {
        for (j = 0; j < nj - 1; j++)
                    {
          daj[j] = uaj[j];
          dbj[j] = ubj[j];
                    }
        thomas(nj,aaj,baj,caj,daj,uaj);
        thomas(nj,abj,bbj,cbj,dbj,ubj);
        facdy[k] = 0.0;
        fbcdy[k] = 0.0;
        for (j = 0; j < nj - 1; j++)
          {
                    facdy[k] = facdy[k] + uaj[j];
          fbcdy[k] = fbcdy[k] + ubj[j];
                    }
        facdy[k] = e*dy*(2.0*facdy[k]-uaj[0]-uaj[nj-2])/2.0;
        fbcdy[k] = eoh*dy*(2.0*fbcdy[k]-ubj[0]-ubj[nj-2])/2.0;
      }

/*  5. mean value of fcdy along x-direction  */

    thabs[i] = 0.0;
    thabsoh = 0.0;

/*    Arith. mean                           */

    for (k = k2; k < nk; k++)
            {
      thabs[i] = thabs[i] + facdy[k];
      thabsoh = thabsoh + fbcdy[k];
            }
    thabs[i] = (dx/2.0)*(2.0*thabs[i]-facdy[k2]-facdy[nk-1])/xd;
    thabsoh = (dx/2.0)*(2.0*thabsoh-fbcdy[k2]-fbcdy[nk-1])/xd;
    thabs[i] = bgabs - (thabs[i] + thabsoh);
    rmsd += pow((thabs[i] - abso[i]), 2.0);
    }
    free(aaj);
    free(abj);
    free(baj);
    free(bbj);
    free(caj);
    free(cbj);
    free(daj);
    free(dbj);
    free(uaj);
    free(ubj);
    free(lama);
    free(lamb);
    free(facdy);
    free(fbcdy);
    rmsd = pow((rmsd/nd), 0.5);
    return rmsd;

}
```

```
/*  optuphydcl.c  program to optimise the dyeing rate constant k_DCl
                Reaction is followed by using a chloride ISE
    thomas(), fmind() and related procedures are given in optmpd.c */

#include <stdio.h>
#include <malloc.h>
#include <math.h>
#include <string.h>

#define NS 10

char fs1[] = "  Cnt   Lin     mf          x(n)    Sigma(n)....\n";
char fs2[] = "%4i  %4i   %6.3e";
char fs3[] = "  %6.3e  %6.3e";
/*char dire[] = "/home/joule/rgc/tam/Pbmxr1/";*/
char dire[] = "";
char name1[] = "cldet2";
char ext1[] = ".prn";
char name2[] = "opcldet2";
char ext2[] = ".dat";
int nj, nk, nd, k1, k2;
double c0,h,d,dx,dy,diffcl,diffoh,khomo,xd,wxdx;
double *t,*ct,*vf,*abso,*thabs,*wdet;

double funfcn();
double fmax();
double fmin();
void fgrad();
void fmind();
void fdfpstep();
void bsort();
void sumx();
void fnmin();
```

```
void thomas();

void main()
{
      char filname[50];
      FILE *fp;

      double *x, tol, xg, xc, khet, rmsd, dummy;
      int  *cnt, prnt, np, ndp, i, kem;

       nj = 3000;
       nk = 5000;

       ndp = 30;
       cnt = (int *) malloc(sizeof(int));
       x = (double *) malloc(NS*sizeof(double));
       thabs = (double *) malloc(ndp*sizeof(double));
       vf = (double *) malloc(ndp*sizeof(double));
       abso = (double *) malloc(ndp*sizeof(double));
       t = (double *) malloc(ndp*sizeof(double));
       ct = (double *) malloc(ndp*sizeof(double));
       wdet = (double *) malloc(nk*sizeof(double));

       d = 1.148;

/*     DCl2              */
       diffcl = 4.14e-6;

/*     Cl-               */
       diffoh = 2.4e-5;

/*     0.1 M Na2CO3              */
/*       khomo = 8.34e-5;*/

/*     0.35 M Na2SO4, 0.1 M Na2CO3   */
       khomo = 9.13e-5;

       khet = 4.7e-5;

       h = 0.0930/2.0;
       c0 = 1.017e-6;
       xd = 0.798;
       xg = 0.214;
       xc = 1.990;
       dy = 2.0*h/nj;
       dx = (xd+xg+xc)/nk;
       k1 = xc/dx;
       k2 = (xc+xg)/dx;

/*     Weighed function in x-direction        */

       wxdx = 0.0;
       kem = (nk - 1 + k2) / 2;
       for (i = k2; i < nk; i++)
         {
          wdet[i] = sqrt(fabs(pow(xd/2.0,2.0)-pow(dx*abs(kem-i),2.0)))
                    /(xd/2.0);
          wxdx += wdet[i];
          }
       wxdx = (dx/2.0)*(2.0*wxdx - wdet[k2] - wdet[nk - 1]);

       strcpy(filname, dire);
       strcat(filname, name1);
       strcat(filname, ext1);
       printf("%s\n", filname);

       fp = fopen(filname, "r");
       nd = 0;
       while (!feof(fp))
        {
         fscanf(fp, "%lf %lf %lf\n",&vf[nd],&abso[nd],&t[nd]);
         abso[nd] = abso[nd]/1000.0;
         nd++;
          }

       close(fp);

       np = 1;
       x[1] = khet;
       tol = 1.0e-5;
       prnt = 1;
       fmind(np, tol, prnt, x, cnt);
       printf("No. count  %i\n", cnt[0]);
       printf("Optimised khet     %6.4e\n", x[1]);
       printf("Initial guess      %6.4e\n", khet);
       rmsd = funfcn(x);
       printf("function value     %6.4e\n", rmsd);
       printf("Input file name    %s\n", filname);
       strcpy(filname, dire);
       strcat(filname, name2);
       strcat(filname, ext2);
       printf("Output file name   %s\n", filname);
       fp = fopen(filname, "w");

       for (i = 0; i < nd; i++)
             {
         fprintf(fp, "%6.2f    %10.3e    %6.2e %11.5e %11.5e\n", t[i],
                   ct[i], vf[i], thabs[i], abso[i]);
         printf("%6.2f     %10.3e     %6.2e %11.5e %11.5e\n", t[i],
                   ct[i], vf[i], thabs[i], abso[i]);
             }

       close(fp);
       free(cnt);
       free(x);

  }

double funfcn(x)

       double *x;
{

/*     DCl  + OH-   ---khomo--->   D(OH) + Cl-          */
/*     A                    B             */

       int i, j, k;
       double *aaj,*baj,*caj,*daj,*uaj,*lama,*facdy,*abj,*bbj,*cbj,
            *dbj,*ubj,*lamb,coht,dummy,khet,rmsd;

       khet = x[1];

       aaj = (double *) malloc(nj*sizeof(double));
       baj = (double *) malloc(nj*sizeof(double));
       caj = (double *) malloc(nj*sizeof(double));
       daj = (double *) malloc(nj*sizeof(double));
       uaj = (double *) malloc(nj*sizeof(double));
       lama = (double *) malloc(nj*sizeof(double));
       facdy = (double *) malloc(nk*sizeof(double));

       abj = (double *) malloc(nj*sizeof(double));
       bbj = (double *) malloc(nj*sizeof(double));
       cbj = (double *) malloc(nj*sizeof(double));
       dbj = (double *) malloc(nj*sizeof(double));
       ubj = (double *) malloc(nj*sizeof(double));
       lamb = (double *) malloc(nj*sizeof(double));

       rmsd = 0.0;
       for (i = 0; i < nd; i++)
             {

       ct[i] = c0*exp(-1.0*khomo*t[i]*60.0);
       coht = c0 - ct[i];

/*  Set up tridiagonal, {d} & {u} matrices  */

/*  1. Calc. lamda & assign initial dj       */

            dummy = (dx*d*pow(2.0*h,3.0))/(6.0*vf[i]*pow(dy,3.0));
```

```
        for (j = 0; j < nj   1; j++)
          {
           lama[j] = dummy/((j+1.0)*(2.0*h-(j+1.0)*dy));
           lamb[j] = diffoh*lama[j];
           lama[j] = diffcl*lama[j];
           uaj[j] = ct[i];
           ubj[j] = coht;
          }

/*   2. Calc. aj, bj & cj                    */

        for (j = 1; j < nj - 1; j++)
          {
             aaj[j - 1] = -1.0*lama[j];
             baj[j] = 2.0*lama[j]+1.0;
             caj[j   1] = -1.0*lama[j   1];
             abj[j - 1] = -1.0*lamb[j];
             bbj[j] = 2.0*lamb[j]+1.0;
             cbj[j - 1] = -1.0*lamb[j - 1];
          }
        baj[nj - 2] = lama[nj - 2] + 1.0;
        bbj[nj - 2] = lamb[nj - 2] + 1.0;

/*   3. Calc. g(k) at cloth                  */

      baj[0] = lama[0]*(2.0*khet*dy+diffcl)/(khet*dy+diffcl)+1.0;
      bbj[0] = lamb[0]+1.0;
        for (k = 0; k < k1; k++)
          {
             for (j = 0; j < nj   1; j++)
            {
             daj[j] = uaj[j];
             dbj[j] = ubj[j];
                    }
             thomas(nj,aaj,baj,caj,daj,uaj);
            dbj[0] += lamb[0]*((uaj[1]-uaj[0])*diffcl/diffoh);
            thomas(nj,abj,bbj,cbj,dbj,ubj);
            }

/*   4. Calc. g(k) at gap & detector      */

      bbj[0] = 1.0+lamb[0];
      for (k = k1; k < k2; k++)
        {
          for (j = 0; j < nj   1; j++)
              dbj[j] = ubj[j];
          thomas(nj,abj,bbj,cbj,dbj,ubj);
          }
      for (k = k2; k < nk; k++)
        {
           for (j = 0; j < nj - 1; j++)
               dbj[j] = ubj[j];
           thomas(nj,abj,bbj,cbj,dbj,ubj);
           facdy[k] = ubj[0];
         }

/*      5. mean value of fcdy along x-direction  */

      thabs[i] = 0.0;

/*        Arith. mean                   */

      for (k = k2; k < nk; k++)
        {
           facdy[k] = fabs(wdet[k]*facdy[k]);
           thabs[i] += facdy[k];
               }
      thabs[i] = (dx/2.0)*(2.0*thabs[i]-facdy[k2]-facdy[nk-1]);
      thabs[i] = thabs[i]/wxdx;
      rmsd += pow((thabs[i] - abso[i]), 2.0);

      }
      free(aaj);
      free(abj);
      free(baj);
      free(bbj);
      free(caj);
      free(cbj);
      free(daj);
      free(dbj);
      free(uaj);
      free(ubj);
      free(lama);
      free(lamb);
      free(facdy);
      rmsd = pow((rmsd/nd), 0.5);
      return rmsd;
}


/*  block.c  program to optimise the kinetic parameters of the surface
          blocking model in dyeing kinetics studies
    fmind() and related procedures are given in optmpd.c */

#include <stdio.h>
#include <malloc.h>
#include <math.h>
#include <string.h>

#define NS 10

char fs1[] = "  Cnt   Lin     mf          x(n)    Sigma(n)....\n";
char fs2[] = "%4i  %4i   %6.3e";
/*char fs3[] = "  %6.3e  %6.3e"; */
char fs3[] = "  %12.7e  %6.3e";
char dire[] = "";
char name1[] = "block1";
char ext1[] = ".prn";
char name2[] = "block1";
char ext2[] = ".dat";
int nd;
double  *ycalc,*yobs,*x1,*x2,*scale;

double funfcn();
double fmax();
double fmin();
void fgrad();
void fmind();
void fdfpstep();
void bsort();
void sumx();
void fnmin();

void main()
{
      char filname[50];
      FILE *fp;
      double *x,tol,rmsd,dummy;
      int  *cnt, prnt, maxdat, np, i;

      maxdat = 30;

      cnt = (int *) malloc(sizeof(int));
      x = (double *) malloc(NS*sizeof(double));
      scale = (double *) malloc(2*sizeof(double));
      ycalc = (double *) malloc(maxdat*sizeof(double));
      yobs = (double *) malloc(maxdat*sizeof(double));
      x1 = (double *) malloc(maxdat*sizeof(double));
      x2 = (double *) malloc(maxdat*sizeof(double));

/*      read exptal data   */

            strcpy(filname, dire);
            strcat(filname, name1);
      strcat(filname, ext1);
      fp = fopen(filname, "r");
      nd = 0;
      while (!feof(fp))
```

```
    {
       fscanf(fp, "%lf %lf %lf\n", &yobs[nd],&x1[nd],&x2[nd]);
       nd++;
     }

close(fp);

np = 2;

scale[1] = 1.0;
scale[2] = 1.0e4;

x[1] = 5000.0/scale[1];
x[2] = 14000000.0/scale[2];

tol = 1.0e-6;
prnt = 1;
fmind(np, tol, prnt, x, cnt);
printf("No. count  %i\n", cnt[0]);
printf("Optimised a        %6.4e\n", x[1]*scale[1]);
printf("Optimised b        %6.4e\n", x[2]*scale[2]);

rmsd = funfcn(x);
printf("function value     %6.4e\n", rmsd);
printf("Input file name    %s\n", filname);
strcpy(filname, dire);
strcat(filname, name2);
strcat(filname, ext2);
printf("Output file name   %s\n", filname);
fp = fopen(filname, "w");

for (i = 0; i < nd; i++)
        {
  fprintf(fp, "%%15.5f  %15.5f   %11.3e   %11.3e\n", ycalc[i],
        yobs[i], x1[i], x2[i]);
  printf("%15.5f  %15.5f   %11.3e   %11.3e\n", ycalc[i],
        yobs[i], x1[i], x2[i]);
        }
        close(fp);
        free(cnt);
        free(x);
   }

double funfcn(x)

        double *x;
{
        int i;
        double  a, b, rmsd;

        a = x[1]*scale[1];
        b = x[2]*scale[2];

        rmsd = 0.0;
        for (i = 0; i < nd; i++)
                {
           ycalc[i] = a*x1[i]+b*x2[i];
           rmsd += pow((ycalc[i] - yobs[i]), 2.0);
        }
        rmsd = pow((rmsd/nd), 0.5);
        return rmsd;
}
```