Numerical Algorithms for Differential Equations with Periodicity

Hadrien Montanelli

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2017

To my grandfather.

Acknowledgements

I would like to thank my supervisor Nick Trefethen for his guidance, support and encouragement over the last four years. He is a fantastic mentor, and his view of numerical analysis and his writing style have greatly influenced mine.

I have discussed and exchanged many ideas with my collaborators Grady Wright, Mohsin Javed, Nikola Gushterov, Niall Bootland, and Yuji Nakatsukasa. It has been a pleasure working with and learning from them.

I thank my fellow Chebfun team members from over the years—Jared Aurentz, Ásgeir Birkisson, Toby Driscoll, Nick Hale, Behnam Hashemi, Hrothgar, Olivier Sète, Mikael Slevinsky, Alex Townsend, and Kuan Xu—for providing a highly stimulating intellectual environment.

I also thank my officemates Anthony Austin and Rodrigo Mendoza–Smith for countless hours of interesting discussions on topics both mathematical and otherwise.

My friends Arthur Roullier, Adrien Boiron, Paul Aubin, Léandre Ostino, Teddy Dresco, Sylvain Gaydou, and Charlie Jacquelin have supported me throughout and visited me several times. They play a very important role in my life and I am infinitely grateful to them.

I wish to thank my friend Mahdi Mahmoud. He has been my mentor and model at Oxford.

Abstract

This thesis presents new numerical methods for solving differential equations with periodicity.

Spectral methods for solving linear and nonlinear ODEs, linear ODE eigenvalue problems and linear time-dependent PDEs on a periodic interval are reviewed, and a novel approach for computing multiplication matrices is presented.

Choreographies, periodic solutions of the n-body problem that share a common orbit, are computed for the first time to high accuracy using an algorithm based on approximation by trigonometric polynomials and optimization techniques with exact gradient and exact Hessian matrix. New choreographies in spaces of constant curvature are found.

Exponential integrators for solving periodic semilinear stiff PDEs in 1D, 2D and 3D periodic domains are reviewed, and 30 exponential integrators are compared on 11 PDEs. It is shown that the complicated fifth-, sixth- and seventh-order methods do not really outperform one of the simplest exponential integrators, the fourth-order ETDRK4 scheme of Cox and Matthews.

Finally, algorithms for solving semilinear stiff PDEs on the sphere with spectral accuracy in space and fourth-order accuracy in time are proposed. These are based on a new variant of the double Fourier sphere method in coefficient space and standard implicit-explicit time-stepping schemes. A comparison is made against exponential integrators and it is found that implicit-explicit schemes perform better.

The algorithms described in each chapter of this thesis have been implemented in MATLAB and made available as part of Chebfun.

Contents

Chapter 1

Introduction

A large number of phenomena in natural and social sciences exhibit periodicity. Examples include day and night cycles, seasons, ocean waves, heartbeat and breathing, economic and political cycles, and many others. These phenomena vary in time but recur at intervals: these are examples of *temporal periodicity*. Another type of periodicity is *spatial periodicity*: even if the phenomenon we are looking at is not periodic in time, it might occur in a periodic domain, such as rows of turbine blades or Earth's surface. In both cases, periodicity is a special feature that we can take advantage of when solving differential equations that model such phenomena. We present in this thesis a collection of algorithms for solving linear and nonlinear differential equations with periodicity in time or in space. These have been implemented in MATLAB and made available as part of Chebfun [53].[1] Each chapter contains new contributions to numerical analysis and scientific computing.

In Chapter 2, we review trigonometric interpolation for approximating periodic functions and spectral methods for solving linear and nonlinear ODEs, linear ODE eigenvalue problems and linear time-dependent PDEs on a periodic interval. We review the main theoretical results and algorithms, and introduce the key concepts and notations that will be used throughout the thesis. We also present a new approach for computing multiplication matrices, which will be crucial in Chapter 5 for the discretization of the Laplacian operator on the sphere.

Chapter 3 explores the computation of choreographies, periodic solutions in time of the n-body problem that share a common orbit. These are computed for the first time to high accuracy using an algorithm based on approximation by trigonometric polynomials and optimization techniques with exact gradient and exact Hessian matrix. We also present new choreographies on the sphere and on the Poincaré disk.

[1] We use MATLAB R2015b and Chebfun v5.6.0.

Chapter 4 investigates exponential integrators for solving periodic semilinear stiff PDEs in 1D, 2D and 3D periodic domains. We compare 30 exponential integrators applied to 11 PDEs and show that the complicated fifth-, sixth- and seventh-order methods do not significantly outperform one of the simplest exponential integrators, the fourth-order ETDRK4 scheme of Cox and Matthews.

Chapter 5 develops numerical algorithms for solving stiff PDEs on the sphere with spectral accuracy in space and fourth-order accuracy in time. These are based on a new variant of the double Fourier sphere method in coefficient space and standard implicit-explicit time-stepping schemes. A comparison is made against exponential integrators and it is found that implicit-explicit schemes perform better.

Chapter 2

Solving differential equations in 1D periodic domains*

2.1 Introduction

In this chapter trigonometric interpolation for approximating functions and spectral methods for solving linear and nonlinear ODEs, linear ODE eigenvalue problems and linear time-dependent PDEs on a periodic interval are discussed. We review the main theoretical results and the standard algorithms, which we have implemented in Chebfun, and present a new approach for computing multiplication matrices.

The theory of periodic differential equations goes back to the work of Floquet [59], Mathieu [105] and Poincaré [133] in the late 19th century. Although it is absent from many books, good treatments can be found in [39], [57] and [158]. As to the mathematics and the algorithms, it is much like that of the familiar nonperiodic case expect for the differences that nongeneric configurations corresponding to eigenmodes have to be avoided to ensure existence and uniqueness, and that boundary conditions are automatically satisfied if working with expansions in the Fourier basis.

Throughout this chapter, and more generally throughout this thesis, the interval $[0, 2\pi]$ should be understood periodically: $\theta = 0$ and $\theta = 2\pi$ are identified and any smoothness assumption applies across this point in the same way as for $\theta \in (0, 2\pi)$ [85, Chap. 1]. Therefore, functions on the periodic interval $[0, 2\pi]$ are periodic *by construction*, and we shall assume that they are complex-valued as well, unless otherwise stated. Let us emphasise that the variable θ might be a space variable (for spatial periodicity), as well as a time variable (for temporal periodicity).

*This chapter is adapted from a paper with Grady Wright, Mohsin Javed and Nick Trefethen [174]. My contribution to the paper is mostly the section about periodic ODEs, operator exponentials and eigenvalue problems, and the implementation of the algorithms for solving such problems.

2.2 Approximating periodic functions

2.2.1 Theory

Fourier series are a powerful tool for the approximation of periodic functions. Using finite Fourier expansions or (closely related) trigonometric interpolants, it is possible to construct approximants to a function. This raises the following question, which has attracted the attention of generations of mathematicians: will finite Fourier expansions and trigonometric interpolants converge as the number of terms is increased, and how fast? The convergence of finite Fourier expansions is of vital importance from a theoretical point of view and will be addressed first, while the convergence of trigonometric interpolants is crucial in practice and will be discussed second.

Let $L^p([0, 2\pi])$ denote the Banach space of functions f whose $L^p([0, 2\pi])$-norm

$$\|f\|_p = \left(\int_0^{2\pi} |f|^p \right)^{1/p} \tag{2.1}$$

is finite. If f is in $L^p([0, 2\pi])$, $1 < p < \infty$, then the *partial sums*

$$f_n(\theta) = \sum_{k=-n}^{n} \hat{f}_k e^{ik\theta} \tag{2.2}$$

with coefficients

$$\hat{f}_k = \frac{1}{2\pi} \int_0^{2\pi} f(\theta) e^{-ik\theta} d\theta \tag{2.3}$$

converge to f in the $L^p([0, 2\pi])$-norm (2.1) [85, Th. 1.11 of Chap. 3], and we write

$$f(\theta) = \sum_{k=-\infty}^{\infty} \hat{f}_k e^{ik\theta}. \tag{2.4}$$

The partial sum (2.2) is a *trigonometric polynomial of degree* n, i.e., a linear combination of complex exponentials $\{1, e^{\pm i\theta}, \ldots, e^{\pm in\theta}\}$. The coefficients (2.3) are the *Fourier coefficients* of f and (2.4) is its *Fourier series*. When f is real-valued the coefficients (2.3) satisfy $\hat{f}_{-k} = \bar{\hat{f}}_k$ for all k.

For pointwise convergence, continuity is not enough: there exist continuous functions whose Fourier series diverge at a point [176, Th. 1.1 of Chap. 8]. However, continuous functions of bounded variation have a unique and uniformly convergent Fourier series [176, Th. 8.1 of Chap. 2], while Lipschitz continuous functions have a unique, uniformly *and* absolutely convergent Fourier series [176, Th. 3.6 of Chap. 6]. Let us summarize the last two results in our first theorem, which uses the notion

of functions f of *bounded variation*, functions whose *total variation* V—the supremum of all sums $\sum_{i=1}^{n} |f(x_i) - f(x_{i-1})|$ with ordered points $\{x_i\}$ in $[0, 2\pi]$ satisfying $x_0 = x_n$—is finite.

Theorem 1.1. *If f is a continuous functions of bounded variation on $[0, 2\pi]$, then it has a unique and uniformly convergent Fourier series* (2.4) *with Fourier coefficients* (2.3)*. If f is Lipschitz continuous, the convergence is uniform and absolute.*

Proof. See [176, Th. 8.1 of Chap. 2] and [176, Th. 3.6 of Chap. 6]. ∎

We now state a theorem about the decay of Fourier coefficients of differentiable and analytic functions.

Theorem 1.2. *For an integer $\nu \geq 0$, let f and its derivatives through $f^{(\nu-1)}$ be absolutely continuous on $[0, 2\pi]$ if $\nu \geq 1$, and suppose the νth derivative $f^{(\nu)}$ is of bounded variation V. Then for each $k \neq 0$, the k-th Fourier coeffcient $\hat{f}_k$ of f satisfies*

$$|\hat{f}_k| \leq \frac{V}{2\pi k^{\nu+1}}. \tag{2.5}$$

If f is analytic in a strip of half-width α around the real axis in the complex plane with $|f(\theta)| \leq M$, then for each $k \neq 0$

$$|\hat{f}_k| \leq M e^{-\alpha|k|}. \tag{2.6}$$

Proof. Suppose $\nu = 0$. The function f is merely of bounded variation V on $[0, 2\pi]$, so we cannot integrate by parts (2.3) in the classical sense. We have to use the theory of Riemann–Stieltjes inegration; the proof can be found in [176, Th. 4.12 of Chap. 2].

Now suppose that $\nu > 0$. Then f is $\nu - 1$ times continuously differentiable, and integration by parts gives

$$\hat{f}_k = \frac{1}{2\pi(ik)^{\nu-1}} \int_0^{2\pi} f^{(\nu-1)}(\theta) e^{-ik\theta} d\theta. \tag{2.7}$$

The function $f^{(\nu-1)}$ is absolutely continuous with derivative $f^{(\nu)}$ of bounded variation. By the fundamental theorem of calculus for absolutely continuous functions, we obtain

$$\hat{f}_k = \frac{1}{2\pi(ik)^{\nu}} \int_0^{2\pi} f^{(\nu)}(\theta) e^{-ik\theta} d\theta. \tag{2.8}$$

We finally apply the case $\nu = 0$ for $f^{(\nu)}$, which yields

$$|\hat{f}_k| \leq \frac{V}{2\pi k^{\nu+1}}. \tag{2.9}$$

If f is analytic, the claimed result follows from shifting the interval $[0, 2\pi]$ downward in the complex plane for $k > 0$, or upward for $k < 0$, by a distance arbitrarily close to α. ∎

It follows from (2.5) that the Fourier coefficients $\hat{f}_k$ of a function that is infinitely differentiable on $[0, 2\pi]$ decay faster than any negative power of k. This property is called *spectral accuracy*, *infinite-order accuracy*, or *superalgebraic accuracy*. For analytic functions, the decay is exponential and the property (2.6) is called *exponential accuracy*. The convergence rate improves further to *superexponential* if the function is entire.

Let us now use Theorem 1.2 to prove a theorem about how fast Fourier series converge.

Theorem 1.3. *For an integer $\nu \geq 1$, let f and its derivatives through $f^{(\nu-1)}$ be absolutely continuous on $[0, 2\pi]$, and suppose the νth derivative $f^{(\nu)}$ is of bounded variation V. Then the partial sums of its Fourier series satisfy*

$$\|f - f_n\|_\infty \leq \frac{V}{\pi\nu n^\nu}. \tag{2.10}$$

If f is analytic in a strip of half-width α around the real axis in the complex plane with $|f(\theta)| \leq M$, then they satisfy

$$\|f - f_n\|_\infty \leq \frac{2Me^{-\alpha n}}{e^\alpha - 1}. \tag{2.11}$$

Proof. The inequalities (2.10) and (2.11) follow from bounding the tail

$$f(\theta) - f_n(\theta) = \sum_{|k|>n} \hat{f}_k e^{ikt} \tag{2.12}$$

using (2.5) and (2.6). ∎

For approximating a function f on $[0, 2\pi]$ by Fourier series, there are two standard options: *projection* or *interpolation*. For an odd number $N = 2n + 1$, the *degree n trigonometric projection* of f is the the partial sum (2.2) and corresponds to the least-squares approximant to f in the N-dimensional space of such polynomials. Note that this approximation depends on the values of $f(\theta)$ for all $\theta \in [0, 2\pi]$. The *degree n trigonometric interpolant* is a computationally simpler approximation of f, which only depends on the values at certain points—therefore, we shall focus in this thesis on interpolants. The interpolant is the unique trigonometric polynomial of degree n that interpolates f at the N *trigonometric points*

$$\theta_j = \frac{2\pi j}{N}, \quad 0 \leq j \leq N - 1. \tag{2.13}$$

If we write it in the form

$$p_n(\theta) = \sum_{k=-n}^{n} \tilde{f}_k e^{ik\theta}, \quad n = (N-1)/2, \tag{2.14}$$

its coefficients are equal to those of f approximated by the trapezoidal rule [166], i.e.,

$$\tilde{f}_k = \frac{1}{N}\sum_{j=0}^{N} f(\theta_j)e^{-ik\theta_j} \approx \hat{f}_k. \tag{2.15}$$

These can be computed in $\mathcal{O}(N \log N)$ operations using the FFT. Note that the coefficients of p_n and those of f are related by the *aliasing formula* (or *Poisson's summation formula*),

$$\tilde{f}_k = \sum_{j=-\infty}^{\infty} \hat{f}_{k+jN}. \tag{2.16}$$

For an even number of points $N = 2n$, uniqueness of trigonometric interpolants is guaranteed if we impose the additional condition $\tilde{f}_{-n} = \tilde{f}_n$. In this case, we write p_n as

$$p_n(\theta) = \sum_{k=-n}^{n}{}' \ \tilde{f}_k e^{ik\theta}, \quad n = N/2, \tag{2.17}$$

where the prime on the summation sign signifies that the terms $k = \pm N$ are halved, and the formulas (2.15)–(2.16) still hold.

We state now a theorem, analogous to Theorem 1.3 for the partial sums, about the convergence of trigonometric interpolants. This theorem is crucial from the perspective of numerical computing with trigonometric interpolants: the smoother a function is, the faster the convergence of its trigonometric interpolant.

Theorem 1.4. *For an integer $\nu \geq 1$, let f and its derivatives through $f^{(\nu-1)}$ be absolutely continuous on $[0, 2\pi]$, and suppose the νth derivative $f^{(\nu)}$ is of bounded variation V. Then its degree n trigonometric interpolant satisfies*

$$\|f - p_n\|_\infty \leq \frac{2V}{\pi\nu n^\nu}. \tag{2.18}$$

If f is analytic in a strip of half-width α around the real axis in the complex plane with $|f(\theta)| \leq M$, then it satisfies

$$\|f - p_n\|_\infty \leq \frac{4Me^{-\alpha n}}{e^\alpha - 1}. \tag{2.19}$$

Proof. The inequalities (2.18) and (2.19) follow from bounding $f(\theta) - p_n(\theta)$ using (2.5) and (2.6) combined with the aliasing formula (2.16). ∎

For more details about Fourier series and trigonometric interpolation, we recommend the books of Henrici [71], Katznelson [85] and Zygmund [176].

2.2.2 Algorithms

The key algorithms for numerical computing with trigonometric interpolants include the FFT algorithm to compute Fourier coefficients [41], the trapezoidal rule to compute integrals [166], Horner's rule to evaluate trigonometric interpolants [80], and the companion matrix and related algorithms to compute roots of trigonometric interpolants [23].

An alternative to Horner's rule to evaluate trigonometric interpolants is the *barycentric formula* of Henrici [70], given for N odd by

$$p_n(\theta) = \sum_{j=0}^{N-1} (-1)^j f(\theta_j) \csc\Big(\frac{\theta - \theta_j}{2}\Big) \Big/ \sum_{j=0}^{N-1} (-1)^j \csc\Big(\frac{\theta - \theta_j}{2}\Big). \tag{2.20}$$

Like Horner's rule, this only requires $\mathcal{O}(N)$ operations, and stability has been established (after a minor modification) in [12].

Let us end this section by mentioning that Javed and Trefethen have recently extended the Remez algorithm to periodic problems, to compute best trigonometric approximation of periodic functions in the L^∞-norm [82].

2.3 Solving linear and nonlinear ODEs in 1D periodic domains

2.3.1 Theory

Let us consider the second-order linear ODE

$$\mathcal{L}u(\theta) = u''(\theta) + a(\theta)u'(\theta) + b(\theta)u(\theta) = f(\theta), \quad \theta \in [0, 2\pi], \tag{2.21}$$

with periodic boundary conditions and continuous functions a, b and f. We look for a twice continuously differentiable solution u. As we mentioned in the introduction of this chapter, the theory of existence and uniqueness of solutions to periodic ODEs goes back to Floquet [59], Mathieu [105] and Poincaré [133] in the late 19th century. We state the main theorem.

Theorem 1.5. *Let Δ be the Hill discriminant associated with* (2.21)*, defined by*

$$\Delta = \frac{c(2\pi) + s'(2\pi)}{2}, \tag{2.22}$$

where $c(\theta)$ and $s(\theta)$ are the solutions of the homogeneous version of (2.21) *corresponding to the initial conditions $c(0) = 1$, $c'(0) = 0$ and $s(0) = 0$, $s'(0) = 1$. The equation* (2.21) *has a unique periodic solution if and only if $\Delta \neq 1$.*

Proof. The key point is to avoid nongeneric configurations corresponding to eigenmodes; see [57, Th. 1.3.2]. ∎

We present now *Fourier spectral methods* in coefficient space for solving (2.21), and more generally for solving linear ODEs of order n and nonlinear ODEs of the form $\mathcal{N}(u) = 0$ for some nonlinear differential operator $\mathcal{N}$. Fourier spectral methods in value space are presented in detail in [164].

2.3.2 Algorithms

ODEs of the form (2.21) can be solved numerically by any standard finite-difference scheme, which would generate sparse linear systems that can be solved in linear time and give algebraic accuracy. Note that, in this case, periodicity would be imposed by the boundary conditions $u(0) = u(2\pi)$ and $u'(0) = u'(2\pi)$. Another approach is to use a Fourier spectral method in coefficient space, which has three advantages. First, this leads to spectral accuracy for smooth variable coefficients and right-hand sides. Second, no boundary conditions have to be imposed since the basis functions (complex exponentials) are periodic. Third, this also generates sparse linear systems when the variable coefficients are smooth.

Spectral methods were developed by Orszag in a series of papers from 1969 [120, 121, 122, 123, 124]; precursors include Arakawa [7], Lanczos [96] and Philips [130, 131]. Spectral methods have remarkable sability and accuracy properties [60, 91, 92, 155, 156]; there are today many reference books [22, 31, 61, 73, 129, 148, 164]. We review in this section Fourier spectral methods for solving linear and nonlinear periodic ODEs in 1D periodic domains. In Sections 2.4 and 2.5, we will apply these methods to the solution of linear ODE eigenvalue problems and linear time-dependent PDEs.

Let us assume that a, b and f in (2.21) are continuous *and* of bounded variation, and that we are looking for a twice continuously differentiable solution u with second derivative u'' of bounded variation. (These assumptions ensure that we can use Fourier series.)

If we write a, b, f and u as Fourier series (2.4) with Fourier coefficients (2.3) $\hat{a}_k$, $\hat{b}_k$, $\hat{f}_k$ and $\hat{u}_k$, then the Fourier series of u'', au', and bu are given by

$$u''(\theta) = \sum_{k=-\infty}^{\infty} (-k^2)\hat{u}_k e^{ik\theta}, \quad a(\theta)u'(\theta) = \sum_{k=-\infty}^{\infty} \left(\sum_{l=-\infty}^{\infty} il\hat{u}_l \hat{a}_{k-l} \right) e^{ik\theta},$$
$$b(\theta)u(\theta) = \sum_{k=-\infty}^{\infty} \left(\sum_{l=-\infty}^{\infty} \hat{u}_l \hat{b}_{k-l} \right) e^{ik\theta}. \tag{2.23}$$

Fourier spectral methods in coefficient space require that u satisfies (2.21) in the weak sense

$$\frac{1}{2\pi}\int_0^{2\pi} \mathcal{L}u(\theta)e^{-ik\theta}d\theta = \frac{1}{2\pi}\int_0^{2\pi} f(\theta)e^{-ik\theta}d\theta, \quad \forall k, \tag{2.24}$$

which yields

$$-k^2\hat{u}_k + \sum_{l=-\infty}^{\infty} il\hat{u}_l\hat{a}_{k-l} + \sum_{l=-\infty}^{\infty} \hat{u}_l\hat{b}_{k-l} = \hat{f}_k, \quad \forall k. \tag{2.25}$$

Let $\hat{u}$ and $\hat{f}$ denote the infinite vector of Fourier coefficients of u and f. The equation (2.25) can be rewritten using operators,

$$\Big(\mathcal{D}^{(2)} + \mathcal{M}[a]\mathcal{D} + \mathcal{M}[b]\Big)\hat{u} = \hat{f}, \tag{2.26}$$

where

$$\mathcal{D} = \begin{pmatrix} \ddots & & & & & & & \\ & -2i & & & & & & \\ & & -i & & & & & \\ & & & 0 & & & & \\ & & & & i & & & \\ & & & & & 2i & & \\ & & & & & & \ddots \end{pmatrix}, \quad \mathcal{D}^{(2)} = \begin{pmatrix} \ddots & & & & & & & \\ & -4 & & & & & & \\ & & -1 & & & & & \\ & & & 0 & & & & \\ & & & & -1 & & & \\ & & & & & -4 & & \\ & & & & & & \ddots \end{pmatrix} \tag{2.27}$$

are the first and second derivative operators, which map the coefficients of u to those of u' and u'', and

$$\mathcal{M}[a] = \begin{pmatrix} \ddots & \ddots & & & & & \\ \ddots & \ddots & \ddots & & & & \\ \dots & \hat{a}_1 & \hat{a}_0 & \hat{a}_{-1} & \dots & & \\ & \dots & \hat{a}_1 & \hat{a}_0 & \hat{a}_{-1} & \dots & \\ & & \dots & \hat{a}_1 & \hat{a}_0 & \hat{a}_{-1} & \dots \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \ddots & \ddots \end{pmatrix} \tag{2.28}$$

is the multiplication by a operator, which maps the coefficients of u to those of au (with an analogous expression for $\mathcal{M}[b]$).

The operators $\mathcal{D}$ and $\mathcal{D}^{(2)}$ map, e.g., the spaces $C^1([0,2\pi])$ and $C^2([0,2\pi])$ of complex-valued continuously differentiable and twice continuously differentiable functions on $[0,2\pi]$ to the space $C([0,2\pi])$ of complex-valued continuous functions on $[0,2\pi]$. These are examples of an *unbounded* operator,[1] a linear map $\mathcal{T} : D(\mathcal{T}) \subset X \to Y$ from a linear subspace $D(\mathcal{T})$ of X to the space Y.

[1]For details about functional analysis, we recommend the book of Brézis [24].

Let us now replace a, b, f and u by their trigonometric interpolants (2.17) at an even number $N = 2n$ of trigonometric points (2.13), with coefficients $\tilde{a}_k$, $\tilde{b}_k$, $\tilde{f}_k$ and $\tilde{u}_k$, $-N/2 \le k \le N/2-1$. (We shall only consider the even case; there are analogous formulas for N odd.) The vector of unknowns is the vector of $N+1$ Fourier coefficients

$$\left(\frac{\tilde{u}_{-N/2}}{2}, \tilde{u}_{-N/2+1}, \ldots, \tilde{u}_{N/2-1}, \frac{\tilde{u}_{N/2}}{2} = \frac{\tilde{u}_{-N/2}}{2}\right)^T. \tag{2.29}$$

However, since FFT codes only store N coefficients, we shall define the vector of coefficients as

$$\tilde{u} = \left(\frac{\tilde{u}_{-N/2}}{2} + \frac{\tilde{u}_{N/2}}{2} = \tilde{u}_{-N/2}, \tilde{u}_{-N/2+1}, \ldots, \tilde{u}_{N/2-1}\right)^T. \tag{2.30}$$

To compute derivatives, we map the vector of N Fourier coefficients (2.30) to the representation (2.29) with $N+1$ coefficients, differentiate, and then map back to N coefficients. Thus, the first-order differentiation matrix $\mathbf{D}_N$ can be written as the product of three matrices,

$$\mathbf{D}_N = \mathbf{Q}\mathbf{D}_{N+1}\mathbf{P}, \tag{2.31}$$

where the $(N+1) \times N$ matrix $\mathbf{P}$ maps (2.30) to (2.29),

$$\mathbf{P} = \begin{pmatrix} \frac{1}{2} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ \frac{1}{2} & & & 0 \end{pmatrix}, \tag{2.32}$$

$\mathbf{D}_{N+1}$ is the $(N+1) \times (N+1)$ first-order Fourier differentiation matrix,

$$\mathrm{diag}(\mathbf{D}_{N+1}) = i\Big(-N/2, -N/2+1, \ldots, N/2\Big)^T, \tag{2.33}$$

and $\mathbf{Q}$ is the $N \times (N+1)$ matrix that maps back to N coefficients,

$$\mathbf{Q} = \begin{pmatrix} 1 & & & & 1 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & 0 \end{pmatrix}. \tag{2.34}$$

This gives a diagonal matrix with entries given by

$$\mathrm{diag}(\mathbf{D}_N) = i\Big(0, -N/2+1, -N/2+2, \ldots, -1, 0, 1, \ldots, N/2-1\Big)^T. \tag{2.35}$$

Note the zero in the the first entry of (2.35) to cancel the mode $(-iN/2)e^{-iN\theta/2}$ that would lead to complex values for real data. Similarly we get

$$\mathrm{diag}(\mathbf{D}_N^{(2)}) = \Big(-(N/2)^2, -(N/2-1)^2, \ldots, -1, 0, -1, \ldots, -(N/2-1)^2 \Big)^T, \quad (2.36)$$

and more generally

$$\mathrm{diag}(\mathbf{D}_N^{(\nu)}) = i^\nu \Big((-N/2)^\nu, (-N/2+1)^\nu, \ldots, (-1)^\nu, 0, 1, \ldots, (N/2-1)^\nu \Big)^T, \quad (2.37)$$

with the first entry set to zero when the differentiation order ν is odd.

The same point of view can be adopted for multiplication matrices, with the difference that multiplying by a or b will increase the length of the representation by a factor of N since the products will now be trigonometric polynomials of degree $2N$, i.e., linear combinations of complex exponentials $\{1, e^{\pm i\theta}, \ldots, e^{\pm iN\theta}\}$. Therefore, to multiply by, e.g., a, we map (2.30) to (2.29), multiply by a with an $(2N+1)\times(N+1)$ matrix, and then truncate and map back to N coefficients. The resulting matrix for multiplication by a, which we denote by $\mathbf{T}[a]$, is given by

$$\mathbf{T}[a] = \mathbf{Q}\mathbf{M}[a](:, N/2+1 : 3N/2+1)\mathbf{P}, \quad (2.38)$$

where $\mathbf{M}[a]$ is the $(2N+1)\times(2N+1)$ matrix defined by

$$\mathbf{M}[a] = \begin{pmatrix} \tilde{a}_0 & \tilde{a}_{-1} & \tilde{a}_{-2} & \ldots & \tilde{a}_{-2N} \\ \tilde{a}_1 & \ddots & \ddots & \ddots & \vdots \\ \tilde{a}_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \tilde{a}_{2N} & \ldots & \ldots & \ldots & \tilde{a}_0 \end{pmatrix}, \quad (2.39)$$

$\mathbf{P}$ is defined as before and $\mathbf{Q}$ is the following $N \times (2N+1)$ matrix,

$$\mathbf{Q} = \begin{pmatrix} 0 & \ldots & 0 & 1 & & & & 1 & 0 & \ldots & 0 \\ & & & & 1 & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 1 & 0 & \ldots & \ldots & 0 \end{pmatrix}. \quad (2.40)$$

(There are $N/2$ zeros at the beginning and at the end of the first line.) We have used MATLAB notation in (2.38): $\mathbf{M}[a](:, N/2+1 : 3N/2+1)$ is obtained from (2.39) by removing the first and last $N/2$ columns. This approach for computing multiplication matrices seems to be new and will be crucial in Chapter 5 for the discretization of the Laplacian operator on the sphere.

Using these matrices, the infinite system of equations (2.26) is discretized by the following linear system of size N:

$$\Big(\mathbf{D}_N^{(2)} + \mathbf{T}[a]\mathbf{D}_N + \mathbf{T}[b]\Big)\tilde{u} = \tilde{f}. \tag{2.41}$$

When the functions a and b are analytic or many times differentiable, they can be approximated to high accuracy by trigonometric interpolants of the form (2.17) with a small number of Fourier coefficients $M \ll N$. In these cases, the multiplication matrices $\mathbf{T}[a]$ and $\mathbf{T}[a]$ are banded with bandwidth M. Hence, for a given N, the system (2.41) can be solved in $\mathcal{O}(M^2N)$ operations using a sparse direct solver [43]. To determine the grid size N that resolves u to high accuracy, a straightforward approach is to solve (2.41) on grids of sizes $N = 32$, $N = 64$, and so on until the coefficients $\tilde{u}$ are small enough. Another approach is to use the the adaptive QR algorithm of Olver and Townsend [119], which adaptively finds the right N and has a total cost of $\mathcal{O}(M^2N)$ operations.

To solve nonlinear ODEs of the form $\mathcal{N}(u) = 0$, one can use Newton's method in function space, where instead of the standard Jacobian matrices, the derivatives involve Fréchet derivatives, as described in [16]. The idea is to compute a sequence of approximate solutions $u_0, u_1, u_2, \ldots$ that converge to the true solution u. The iteration is as follows:

$$u_{k+1} = u_k - (\mathcal{N}'(u_k))^{-1}\mathcal{N}(u_k), \tag{2.42}$$

where $\mathcal{N}'(u_k)$ is the Fréchet derivative of $\mathcal{N}$ at u_k. Therefore, solving nonlinear ODEs reduces to solving a sequence of linear ODEs.

2.3.3 Numerical examples

Chebfun has an automatic solver for linear and nonlinear ODEs based on Chebyshev spectral methods, implemented in the `chebop` class [52] and based on MATLAB's backslash command and Newton's method in function space. We have extended it to periodic ODEs using a Fourier spectral method in coefficient space that uses the matrices described in the previous section.

Let us consider the second-order linear ODE

$$\mathcal{L}u(\theta) = u''(\theta) + \sin(\theta)u'(\theta) + 1000\cos(2\theta)u(\theta) = 100, \quad \theta \in [0, 2\pi], \tag{2.43}$$

with periodic boundary conditions. We can solve it using a Fourier spectral method in coefficient space with a few lines of code:[2]

[2]For periodic problems, Chebfun uses by default a Fourier spectral method in value space. To overwrite this, one can type `cheboppref.setDefaults('discretization','coeffs')`.

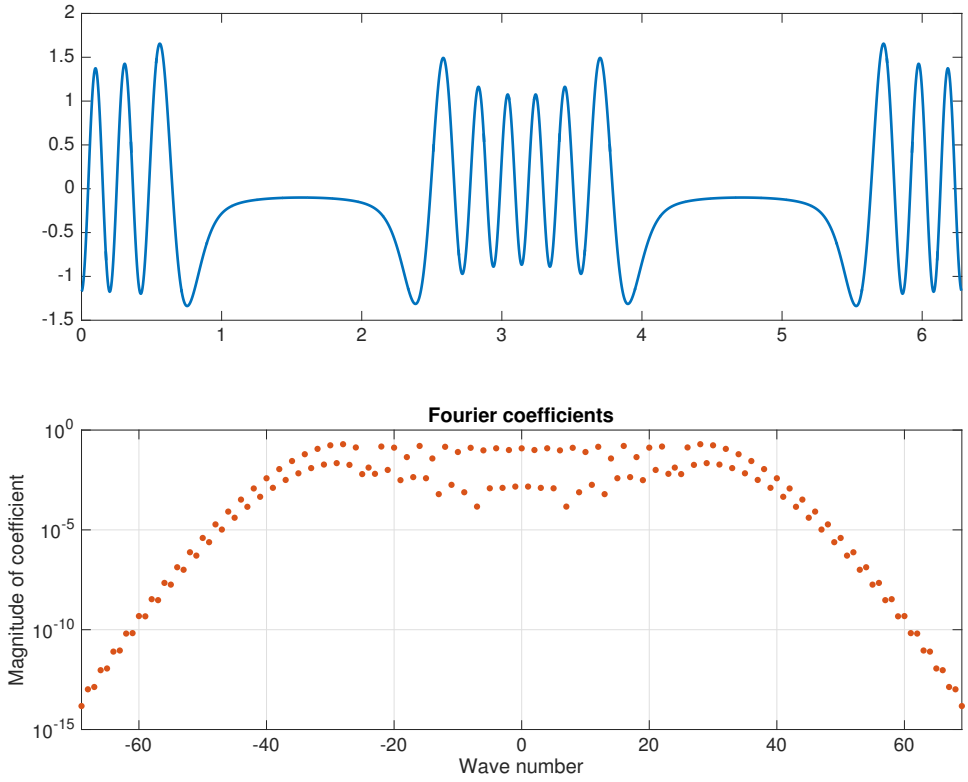

Figure 2.1: *Solution of the linear periodic ODE* (2.43) *as a trigonometric interpolant of degree* $n = N/2 = 69$ (*top*) *and its Fourier coefficients* (*bottom*).

```
dom = [0 2*pi];                                          % domain
a = chebfun(@(x) sin(x), dom);                           % coefficient a
b = chebfun(@(x) 1000*cos(2*x), dom);                    % coeffiicient b
L = chebop(@(u) diff(u,2) + a.*diff(u) + b.*u, dom); % operator
L.bc = 'periodic';                                       % boundary cond.
f = 100;                                                 % right-hand side
u = L\f;                                                 % solve
```

The numerical solution of (2.43) and its Fourier coefficients are shown in Figure 2.1. It satisfies the differential equation to high accuracy: the relative L^2-norm of the residual (relative L^2-error) $\|\mathcal{L}u - 100\|_2/\|u\|_2$ is 1.0×10^{-10}.

Suppose now that we want to solve the nonlinear ODE

$$\mathcal{N}(u(\theta)) = u''(\theta) + u(\theta)u'(\theta) + 1000e^{\sin(4\theta)}u(\theta) = f(\theta), \quad \theta \in [0, 2\pi], \tag{2.44}$$

with right-hand side $f(\theta) = 1000\cos(10\theta)$ and periodic boundary conditions. After five steps of Newton's method, the following Chebfun code produces the result shown in Figure 2.2 with relative L^2-error $\|\mathcal{N}(u) - f\|_2/\|u\|_2$ equal to 3.2×10^{-9}:

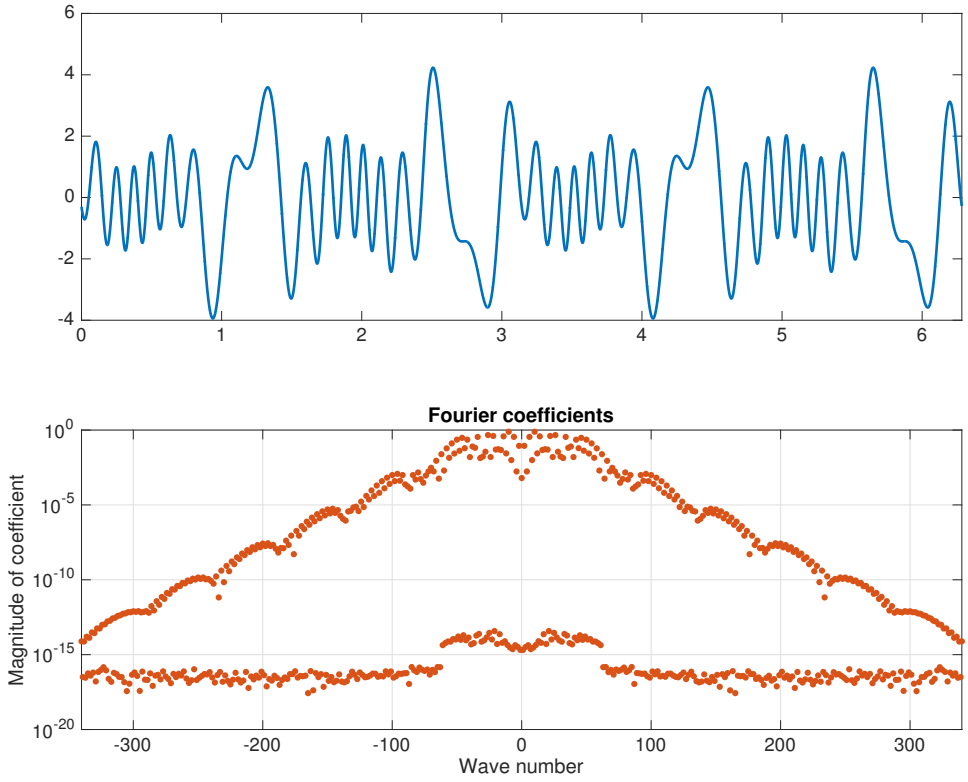

Figure 2.2: *Solution of the nonlinear periodic ODE* (2.44) *as a trigonometric interpolant of degree* $n = N/2 = 340$ (*top*) *and its Fourier coefficients* (*bottom*).

```
dom = [0 2*pi];                                          % domain
b = chebfun(@(x) 1000*exp(sin(4*x)), dom);               % coeffiicient b
N = chebop(@(u) diff(u,2) + u.*diff(u) + b.*u, dom); % operator
N.bc = 'periodic';                                       % boundary cond.
f = chebfun(@(x) 1000*cos(10*x), dom);                   % right-hand side
u = N\f;                                                 % solve
```

2.4 Solving linear ODE eigenvalue problems in 1D periodic domains

2.4.1 Theory

Let us consider the *Sturm–Liouville eigenvalue problem*

$$-\frac{d}{d\theta}\left[p(\theta)\frac{du}{d\theta}\right] + q(\theta)u(\theta) = \lambda w(\theta)u(\theta), \quad \theta \in [0, 2\pi], \tag{2.45}$$

with periodic boundary conditions. We assume that $w > 0$ and q are real-valued continuous functions and that $p > 0$ is real-valued and continuously differentiable. We look for complex eigenvalues λ and twice continuously differentiable real-valued

functions u. The main theorem states that (2.45) is diagonalisable in the basis of eigenfunctions with real eigenvalues.

Theorem 1.6. *There exists a basis of real-valued continuous functions on* $[0, 2\pi]$ *that consists of eigenfunctions* $u_n(x)$ *of the periodic Sturm–Liouville eigenvalue problem* (2.45)*. They are orthonormal with respect to the inner product*

$$\int_0^{2\pi} \overline{u_m(\theta)} u_n(\theta) w(\theta) d\theta, \tag{2.46}$$

and have real eigenvalues λ_n *of multiplicity one for* $n = 0$ *and at most two for* $n \geq 1$*, with* $\lambda_n \to \infty$ *as* $n \to \infty$*. Moreover, let* $\Delta(\lambda)$ *be the Hill discriminant associated with* (2.45) *and defined by*

$$\Delta(\lambda) = \frac{c_\lambda(2\pi) + p(2\pi) s'_\lambda(2\pi)}{2}, \tag{2.47}$$

where $c_\lambda(\theta)$ *and* $s_\lambda(\theta)$ *are the solutions of* (2.45) *for a given* λ *and initial conditions* $c_\lambda(0) = 1$*,* $p(0)c'_\lambda(0) = 0$ *and* $s_\lambda(0) = 0$*,* $p(0)s'_\lambda(0) = 1$*. The eigenvalues* λ_n *are precisely those numbers* λ *for which* $\Delta(\lambda)$ *takes the value* 1*.*

Proof. Let $C([0, 2\pi])$ and $C^2([0, 2\pi])$ denote the spaces of real-valued continuous and twice continuously differentiable functions on $[0, 2\pi]$, and let $\mathcal{L}$ be the symmetric *Sturm–Liouville operator* defined by

$$\mathcal{L}u = \frac{1}{w(\theta)} \Big(- \frac{d}{d\theta} \Big[p(\theta) \frac{du}{d\theta} \Big] + q(\theta) u \Big), \tag{2.48}$$

with $\mathcal{L} : C^2([0, 2\pi]) \to C([0, 2\pi])$. The operator $\mathcal{L}$ is not compact for the operator norm associated with (2.46). However, its resolvent $R_\lambda = (\mathcal{L} - \lambda \mathcal{I})^{-1}$ is compact, and symmetric if $\lambda \in \mathbb{R}$, and hence has real eigenvalues $\alpha_n \to 0$ with orthonormal eigenfunctions $u_n(\theta)$ if $\lambda \in \mathbb{R}$. From $R_\lambda u_n = \alpha_n u_n$ we get $\mathcal{L} u_n = \lambda_n u_n$, with eigenvalues $\lambda_n = \lambda + \frac{1}{\alpha_n} \in \mathbb{R}$, and $\lambda_n \to \infty$ as $n \to \infty$. The proofs that they have multiplicity one for $n = 0$ and at most two for $n \geq 1$, and that they correspond to $\Delta(\lambda) = 1$, can be found in [158, Th. 5.28] or [39, Th. 3.1 of Chap. 8]. ∎

2.4.2 Algorithms

Spectral methods for eigenvalue problems were introduced by Orszag to compute the eigenvalues of the Orr–Sommerfeld equation [122], using expansions in Chebyshev polynomials.

We assume that all the functions we deal with (w, q, p' and u'') satisfy the assumptions given below (2.45) *and* are of bounded variation. To solve the periodic

Sturm–Liouville eigenvalue problem (2.45), we replace p, q, w and u by their trigonometric interpolants and use the Fourier matrices of Section 2.3.2 on a grid of size N. The problem (2.45) is rewritten as a generalised (matrix) eigenvalue problem,

$$\Big(-\mathbf{D}_N \mathbf{T}[p] \mathbf{D}_N + \mathbf{T}[q] \Big) \tilde{u} = \lambda \mathbf{T}[w] \tilde{u}. \tag{2.49}$$

As in the section about ODEs, we emphasise that if p, q and w are smooth functions that can be resolved to high accuracy with $M \ll N$ Fourier coefficients, then the matrices $\mathbf{T}[p]$, $\mathbf{T}[q]$ and $\mathbf{T}[w]$ are banded. Note that since the matrix $\mathbf{T}[w]$ is Hermitian positive definite[3] a possible approach to solve (2.49) is to compute the Cholesky factorization of $\mathbf{T}[w]$ (in $\mathcal{O}(M^2 N)$ operations) to reduce (2.49) to a standard eigenvalue problem, and then use the QR algorithm, as in LAPACK [6]. The total cost would be $\mathcal{O}(N^3)$ operations.[4] Another approach is to use the shift-invert Arnoldi method, as in ARPACK [98]. The cost would be $\mathcal{O}(nLM^2N)$ operations to find L eigenvalues, n being the average number of Arnoldi iterations per eigenvalue, which is typically a modest constant. In both cases, the process is repeated on grids of sizes 32, 64, and so on until the coefficients $\tilde{u}$ are small enough.

2.4.3 Numerical examples

Chebfun can solve linear ODE eigenvalue problems via the command `eigs` [52], which uses the MATLAB command `eig` on the discrete systems of equations (2.49). (Note that MATLAB's command `eig`, for eigenvalue and generalised eigenvalue problems, is based on the QR algorithm of LAPACK.) We have extended `eigs` to periodic linear ODE eigenvalue problems using a Fourier spectral method in coefficient space.

If $p(\theta) = w(\theta) = 1$ and $q(\theta) = 0$ in (2.45), we obtain

$$-u''(\theta) = \lambda u(\theta), \quad \theta \in [0, 2\pi], \tag{2.50}$$

with eigenfunctions $A\cos(\sqrt{\lambda_n}x) + B\sin(\sqrt{\lambda_n}x)$, and discrete and real eigenvalues $\lambda_n = n^2$, simple for $n = 0$ and double for $n \geq 1$. Let us compute six eigenvalues and eigenfunctions of the ODE eigenvalue problem (2.50) with Chebfun:

```
dom = [0 2*pi];                    % domain
L = chebop(@(u) -diff(u,2), dom); % operator
L.bc = 'periodic';                 % boundary cond.
[V, D] = eigs(L);                  % solve (6 by default)
```

[3] It is Hermitian because w is real-valued and positive definite because $w > 0$.

[4] In the special case where $\mathbf{T}[w]$ is the identity matrix, the cost of the QR algorithm would be $\mathcal{O}(M^2N) + \mathcal{O}(N^2)$, $\mathcal{O}(M^2N)$ for reducing to tridiagonal form and $\mathcal{O}(N^2)$ for solving the tridiagonal problem.

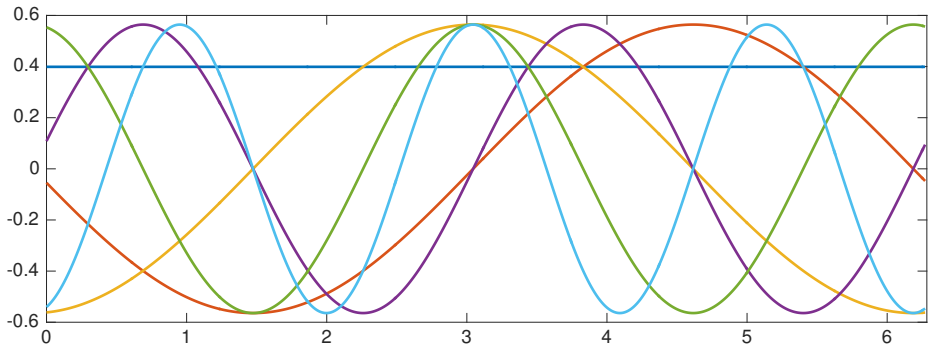

Figure 2.3: *Six eigenfunctions of the linear periodic ODE eigenvalue problem* (2.50). *These are linear combinations of sines and cosines.*

The eigenfunctions are shown in Figure 2.3. The maximum absolute error between the computed eigenvalues and the exact ones $(1, 2, 2, 4, 4, 9)^T$ is 1.5×10^{-13}, while the maximum relative L^2-error $\max_u \|u'' + \lambda u\|_2 / \|u\|_2$ is 2.1×10^{-13}.

If $p(\theta) = w(\theta) = 1$ and $q(\theta) = 2q\cos(2\theta)$, we obtain the *Mathieu equation*

$$-u''(\theta) + 2q\cos(2\theta)u(\theta) = \lambda u(\theta), \quad \theta \in [0, 2\pi]. \tag{2.51}$$

This equation was derived by Émile Mathieu to model vibrations of elliptical drumheads [105]. Given $q \neq 0$, it can be shown that there exists a countably infinite set of real eigenvalues $\lambda_n(q)$, simple for $n = 0$ and double for $n \geq 1$, with elliptic cosine and sine eigenfunctions, the *Mathieu functions*. The first six eigenvalues are given by [1, Chap. 20]:

$$\begin{aligned} \lambda_0 &= -1.513956885056520, \quad \lambda_1 = -1.390676501225323, \\ \lambda_2 &= 2.379199880488686, \quad \lambda_3 = 3.672232706497191, \\ \lambda_4 &= 5.172665133358294, \quad \lambda_5 = 9.140627737766440. \end{aligned} \tag{2.52}$$

The following code computes approximations of (2.52) and produces Figure 2.4:

```
dom = [0 2*pi];                                          % domain
L = chebop(@(x,u) -diff(u,2) + 4*cos(2*x).*u, dom); % operator
L.bc = 'periodic';                                       % boundary cond.
[V, D] = eigs(L);                                        % solve
```

The maximum error between the computed and exact eigenvalues is 1.6×10^{-13}, while the maximum relative L^2-error is 7.2×10^{-12}.

Figure 2.4: *Six eigenfunctions of the Mathieu eigenvalue problem* (2.51). *These are linear combinations of the Mathieu elliptic sine and cosine functions.*

2.5 Solving linear time-dependent PDEs in 1D periodic domains

2.5.1 Theory

Let $\mathcal{L}$ be an unbounded linear operator $\mathcal{L} : D(\mathcal{L}) \subset X \to X$ in a function space X of real-valued functions of one real variable,[5] and let us consider the linear time-dependent PDE

$$u_t(t,\theta) = \mathcal{L}u(t,\theta), \quad u(t=0,\theta) = u_0(\theta) \in D(\mathcal{L}), \tag{2.53}$$

on $[0,T]\times[0,2\pi]$ with periodic boundary conditions. The problem (2.53) is *well-posed* if, for every initial condition $u_0(\theta) \in D(\mathcal{L})$ and each time $0 < t \leq T$, there exists a unique solution $u(t,\theta) \in X$ and this solution depends continuously on $u_0(\theta)$.

A special case of well-posed problems corresponds to *maximal monotone* operators in Hilbert spaces. Let $X = H$ be a Hilbert space on $[0,2\pi]$ with inner product $(\cdot\,,\cdot)$ and norm $\|\cdot\|$. The operator $\mathcal{L}$ is said to be maximal monotone if $(\mathcal{L}u, u) \geq 0$, $\forall u \in D(\mathcal{L})$, and for every $f \in H$, there exists $u \in D(\mathcal{L})$ such that $\mathcal{L}u + u = f$. In this case, the *Hille–Yosida theorem* [24, Th. 7.4] can be used to prove existence, uniqueness, and continuous dependence on u_0.

Theorem 1.7. *Let $\mathcal{L}$ be a maximal monotone operator in a Hilbert space H. Then, given any initial condition $u_0(\theta) \in D(\mathcal{L})$, there exists a unique function*

$$u(t,\theta) \in C^1([0,\infty);\, H) \cap C([0,\infty);\, D(\mathcal{L})) \tag{2.54}$$

[5]The results we review in this section can be generalised to complex-valued functions of several real variables, as can the algorithms we shall present in the next section.

that is a solution of (2.53). *Moreover,*

$$\|u(t,\theta)\| \leq \|u_0(\theta)\|, \quad \|u_t(t,\theta)\| = \|\mathcal{L}u(t,\theta)\| \leq \|\mathcal{L}u_0(\theta)\|, \quad \forall t \geq 0. \tag{2.55}$$

Proof. See [24, Th. 7.4]. ∎

The constant coefficient operators

$$\mathcal{L} : H^1([0,2\pi]) \to L^2([0,2\pi]), \quad \mathcal{L} = a\frac{\partial}{\partial\theta}, \quad a \in \mathbb{R}, \tag{2.56}$$

and

$$\mathcal{L} : H^2([0,2\pi]) \to L^2([0,2\pi]), \quad \mathcal{L} = b\frac{\partial^2}{\partial\theta^2}, \quad b > 0, \tag{2.57}$$

with standard Sobolev spaces $H^1([0,2\pi])$ and $H^2([0,2\pi])$, are examples of maximal monotone operators; see [24, Th. 10.1] and [24, Th. 10.7]. These correspond to the *linear convection* and *heat* operators. On the other hand, the variants of (2.56) and (2.57) with variable coefficients are not maximal monotone. More sophisticated tools have to be employed to ensure existence, uniqueness and continuous dependence on the initial condition; see [24, Th. 10.9], [24, Th. 10.14] and the references therein.

Another case of well-posed problems corresponds to *m-accretive* operators in Banach spaces. Let $X = B$ be a Banach space on $[0,2\pi]$ with norm $\|\cdot\|$ and operator norm $\|\cdot\|_B$. The operator $\mathcal{L}$ is said to be m-accretive if $\overline{D(\mathcal{L})} = B$ and for every $\lambda > 0$, $\mathcal{I} + \lambda\mathcal{L}$ is bijective from $D(\mathcal{L})$ onto B with $\|(\mathcal{I} + \lambda\mathcal{L})^{-1}\|_B \leq 1$. The Hille–Yosida theorem is still valid [24, Th. 7.8] and the unique solution of (2.53) is given by the *operator exponential* [24, Th. 7.9].

Theorem 1.8. *Let $\mathcal{L}$ be an m-accretive operator in a Banach space B. Then, given any initial condition $u_0(\theta) \in D(\mathcal{L})$, there exists a unique function*

$$u(t,\theta) \in C^1([0,\infty);\, B) \cap C([0,\infty);\, D(\mathcal{L})) \tag{2.58}$$

that is a solution of (2.53). *Moreover,*

$$\|u(t,\theta)\| \leq \|u_0(\theta)\|, \quad \|u_t(t,\theta)\| = \|\mathcal{L}u(t,\theta)\| \leq \|\mathcal{L}u_0(\theta)\|, \quad \forall t \geq 0, \tag{2.59}$$

and the solution is given for each $0 < t \leq T$ and $\theta \in [0,2\pi]$ by the exponential formula

$$u(t,\theta) = e^{\mathcal{L}t}u_0(\theta) = \lim_{n\to+\infty}\left[\left(\mathcal{I} + \frac{t}{n}\mathcal{L}\right)^{-1}\right]^n u_0(\theta). \tag{2.60}$$

Proof. See [24, Th. 7.8], [24, Th. 7.9] and the references therein. ∎

2.5.2 Algorithms

To solve (2.53) numerically, we can use the *matrix exponential* on the discretized version of (2.53) obtained using the Fourier matrices of Section 2.3.2. For example, let us consider the linear convection equation

$$u_t(t,\theta) = a(\theta)u_\theta, \quad u(t=0,\theta) = u_0(\theta), \tag{2.61}$$

on $[0,T] \times [0,2\pi]$. We discretize space with $N = 2n$ trigonometric points and obtain

$$\tilde{u}' = \mathbf{T}[a]\mathbf{D}_N\tilde{u}, \quad \tilde{u}(0) = \tilde{u}_0, \tag{2.62}$$

where $\tilde{u}(t)$ are the Fourier coefficients of the trigonometric interpolant $p_n(t,\theta)$ of $u(t,\theta)$ at time t defined by

$$p_n(t,\theta) = \sum_{k=-N/2}^{N/2}{}' \, \tilde{u}_k(t)e^{ik\theta}, \tag{2.63}$$

and $\tilde{u}_0$ are the Fourier coefficients of the trigonometric interpolant of $u_0(\theta)$. Using the matrix exponential we get

$$\tilde{u}(t) = e^{t\mathbf{T}[a]\mathbf{D}_N}\tilde{u}_0. \tag{2.64}$$

This matrix exponential can be computed numerically by, e.g., the scaling and squaring method in $\mathcal{O}(N^3)$ operations [3, 74]. Note that the fact that the matrix $\mathbf{T}[a]$ might be banded with bandwidth M does not help. (However, if all the eigenvalues of the matrix are real, e.g., for the heat equation $u_t(t,\theta) = u_{\theta\theta}$, one can use the Carathéodory–Fejér method to compute the action of the matrix exponential, which reduces the cost to $\mathcal{O}(MN)$ operations [143].) As previously, the process is repeated on grids of sizes 32, 64, and so on until the coefficients $\tilde{u}$ are small enough.

Let us emphasise that no time-stepping is needed, i.e., the solution at large time t can be directly computed from the initial condition. For a general linear operator $\mathcal{L}$, explicit and implicit time-stepping schemes would cost $\mathcal{O}(N^2)$ per time-step[6] but many time-steps might be needed to compute the solution at large t. For the convection equation (2.61) with matrix $\mathbf{T}[a]$ of bandwidth M, implicit and explicit schemes might be competitive, since their cost would reduce to $\mathcal{O}(MN)$ operations per time-step. We shall come back to this discussion in Chapter 5.

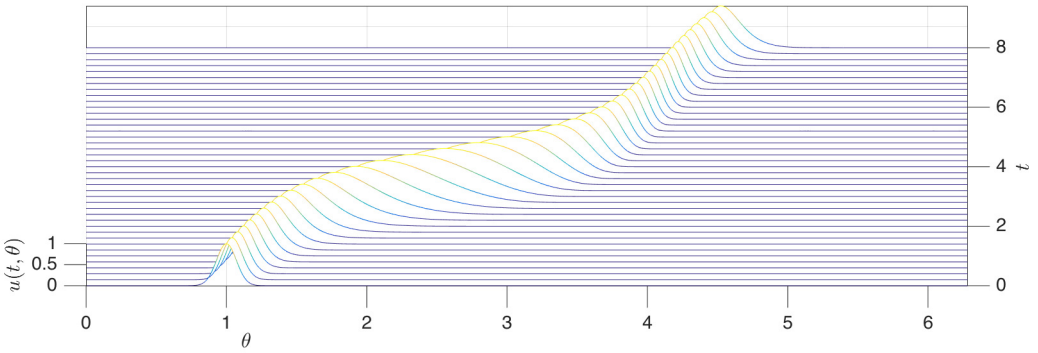

Figure 2.5: *Solution of the linear periodic time-dependent PDE* (2.65)*. This example from* [164] *shows the propagation at variable speed of the initial condition, which remains coherent and clean.*

2.5.3 Numerical examples

Chebfun can solve linear time-dependent PDEs with its command `expm` [52], which applies the MATLAB command `expm` on the differentiation and multiplication matrices, as in (2.64). (Note that MATLAB's `expm` is based on the scaling and squaring method.) We have extended this functionality to periodic linear time-dependent PDEs using a Fourier spectral method in coefficient space.

Let us solve

$$u_t = \Big(-\frac{1}{5} - \sin^2(\theta - 1) \Big) u_\theta, \quad u(0,\theta) = e^{-100(\theta-1)^2} \tag{2.65}$$

on $[0, 8] \times [0, 2\pi]$. The following code produces the result shown in Figure 2.5:

```
dom = [0 2*pi];                                % domain
T = 8; dt = 0.2; t = 0:dt:T;                   % time interval
c = chebfun(@(x) -(1/5 + sin(x-1).^2), dom); % variable coefficient
L = chebop(@(x,u) c.*diff(u), dom);            % operator
L.bc = 'periodic';                             % boundary cond.
u0 = chebfun(@(x) exp(-100*(x-1).^2), dom);  % initial cond.
u = expm(L, t, u0);                            % solve
```

2.6 Discussion

We reviewed in this chapter spectral methods for solving linear and nonlinear ODEs, linear ODE eigenvalue problems and linear time-dependent PDEs on a periodic interval. Our main contribution is a novel approach—described in Section 2.3.2—for computing Fourier multiplication matrices.

[6] Implicit schemes require a $\mathcal{O}(N^3)$ precomputation step to compute the LU factorization of the discrete version of $\mathcal{L}$.

Let us conclude this chapter by mentioning that it is possible to extend the theory of m-accretive operators to *semilinear* PDEs

$$u_t = \mathcal{L}u + \mathcal{N}(u), \tag{2.66}$$

where $\mathcal{N} : B \to B$ is a nonlinear operator; see, e.g., [104]. Algorithms for solving PDEs of the form (2.66) in several space dimensions will be the core of Chapters 4 and 5.

Chapter 3

Computing choreographies*

3.1 Introduction

We present in this chapter an application of trigonometric interpolation: the computation of *choreographies*. Choreographies are periodic solutions in time of the two-dimensional n-body problem, in which the bodies share a common orbit and are uniformly spread along it. The study of choreographies with unit masses is an old one. For the two-body problem, the only choreography is a circle, and for the three-body problem, the first one was found by Lagrange in 1772 [95], and is also a circle. The second choreography of the three-body problem, a figure-eight, was discovered numerically more than two centuries later by Moore in 1993 [115], while Chenciner and Montgomery proved the existence of this class of orbits a few years later [36]. In the early 2000s, many new choreographies for various n and unit masses were found by Simó using a combination of numerical methods [149]. In 2008, Chen proved the existence of infinitely many choreographies of the three-body problem with a certain topological type for various choices of masses [35].

Following the work of Chernoivan and Mamev [38] and Kilin [87], there has been a growing interest in the n-body problem in spaces of constant curvature, led by Borisov and collaborators [18, 19, 20], Diacu and collaborators [44, 46, 47, 48, 49, 50, 128] and others [33, 146]. Despite this interest, only a few noncircular choreographies have been found previously in spaces of constant curvature [20, 45].

We present an algorithm to compute choreographies to high accuracy in the plane and in two-dimensional spaces of constant curvature. It is based on approximation by

*This chapter is adapted from two papers [111, 113]. In the paper with Gushterov [111], I derived the closed-form expressions for the gradients and the Hessian matrices, designed and implemented the algorithms and wrote the paper. Gushterov suggested to me to look at choreographies on the sphere and helped me understand the physics of the n-body problem in spaces of constant curvature.

trigonometric polynomials, stereographic projection and optimization methods with exact gradient and exact Hessian matrix. The algorithm proceeds in two phases: first BFGS quasi-Newton iteration to get close to a solution, then Newton iteration for high accuracy. We have recomputed the planar choreographies of Simó to higher accuracy and have found many new choreographies in spaces of constant curvature. As the curvature of those spaces goes to zero, these new solutions converge to the planar ones.

3.2 Planar choreographies

Let $z_j(t) \in \mathbb{C}$, $0 \le j \le n-1$, denote the positions of n bodies with unit masses in the complex plane. The planar n-body problem describes the motion of these bodies under the action of Newton's law of gravitation, through the nonlinear coupled system of ODEs

$$z_j''(t) - \sum_{\substack{i=0 \\ i \neq j}}^{n-1} \frac{z_i(t) - z_j(t)}{\big|z_i(t) - z_j(t)\big|^3} = 0, \quad 0 \le j \le n-1. \tag{3.1}$$

We are interested in periodic solutions of (3.1) in which the bodies share a single orbit and are uniformly spread along it, that is, solutions $z_j(t)$ such that

$$z_j(t) = q\Big(t + \frac{2\pi j}{n}\Big), \quad 0 \le j \le n-1, \tag{3.2}$$

for some 2π-periodic function $q : [0, 2\pi] \to \mathbb{C}$. Such solutions were named choreographies by Simó, the n bodies being "seen to dance in a somewhat complicated way" [149]. We shall refer to them as *planar choreographies*. The period can be chosen equal to 2π because if $q(t)$ is a T-periodic solution of (3.1), then $\lambda^{-2/3} q(\lambda t)$, $\lambda = T/(2\pi)$, is a 2π-periodic solution. It has been well known since Poincaré [133, 134] that the *principle of least action*, first introduced by Maupertuis in 1744 [106], can be used to characterize periodic solutions of (3.1): choreographies (3.2) are minima of the *action functional*, or simply *action*, defined as the integral over one period of the kinetic minus the potential energy,

$$A = \int_0^{2\pi} \big(K(t) - U(t)\big)\, dt, \tag{3.3}$$

with kinetic energy

$$K(t) = \frac{1}{2} \sum_{j=0}^{n-1} \big|z_j'(t)\big|^2 = \frac{1}{2} \sum_{j=0}^{n-1} \Big|q'\Big(t + \frac{2\pi j}{n}\Big)\Big|^2 \tag{3.4}$$

and potential energy

$$U(t) = -\sum_{j=0}^{n-1}\sum_{i=0}^{j-1}\left|z_i(t) - z_j(t)\right|^{-1} = -\sum_{j=0}^{n-1}\sum_{i=0}^{j-1}\left|q\Big(t+\frac{2\pi i}{n}\Big) - q\Big(t+\frac{2\pi j}{n}\Big)\right|^{-1}. \quad (3.5)$$

Note that the action (5.57) depends on $q(t)$ via $U(t)$ and on $q'(t)$ via $K(t)$. Since the integral of (3.4) does not depend on j and the integral of (3.5) only depends on $i-j$, the action functional can be rewritten as

$$A = \frac{n}{2}\int_0^{2\pi}\left|q'(t)\right|^2 dt + \frac{n}{2}\sum_{j=1}^{n-1}\int_0^{2\pi}\left|q(t) - q\Big(t+\frac{2\pi j}{n}\Big)\right|^{-1} dt. \quad (3.6)$$

Planar choreographies correspond to functions $q(t)$ that minimize (3.6).

We are also interested in solutions of (3.1) in which the bodies share a single orbit $q(t)$ rotating with angular velocity ω relative to an inertial reference frame, i.e.,

$$z_j(t) = e^{i\omega t}q\Big(t+\frac{2\pi j}{n}\Big), \quad 0 \le j \le n-1. \quad (3.7)$$

Choreographies of the form (3.7) are said to be *relative*, as opposed to the *absolute* choreographies (3.2). The action associated with relative planar choreographies is

$$A = \frac{n}{2}\int_0^{2\pi}\left|q'(t) + i\omega q(t)\right|^2 dt + \frac{n}{2}\sum_{j=1}^{n-1}\int_0^{2\pi}\left|q(t) - q\Big(t+\frac{2\pi j}{n}\Big)\right|^{-1} dt. \quad (3.8)$$

Note that (3.6) is the special case of (3.8) with $\omega = 0$.

3.3 Computing planar choreographies

Our method for computing planar choreographies is based on the minimization of the action (3.8) and uses two key ingredients.

Ingredient 1. *Trigonometric interpolation.* The function $q(t)$ is represented by its trigonometric interpolant (2.14) in the $\exp(ikt)$ basis. The optimization variables are the real and imaginary parts of its Fourier coefficients. The action (3.8) is then computed with the exponentially accurate trapezoidal rule [166].

Ingredient 2. *Closed-form expressions for the gradient and the Hessian.* Formulas for the gradient and the Hessian matrix of (3.8) with respect to the optimization variables are derived explicitly and used in the optimization algorithms.

The numerical optimization of the action is then carried out in two phases.

Phase 1. *Quasi-Newton optimization methods.* Numerical optimization methods based on the exact gradient and approximations of the Hessian are employed with a

small number of optimization variables. The accuracy of the solution at this stage is from one to four digits. This step is computationally very cheap.

Phase 2. Newton's method. Once an approximation to a choreography has been computed via a quasi-Newton method, one can improve the accuracy to typically ten digits with a few steps of Newton's method with exact Hessian, and a larger number of optimization variables. This step is computationally more expensive.

3.3.1 Discretization with trigonometric interpolation

The approach used by Simó is to decompose the function $q(t)$ into real and imaginary parts, and to represent each of them by a trigonometric interpolant in the $\sin(kt)$ and $\cos(kt)$ basis [149]. We use instead a trigonometric interpolant of the complex function $q(t)$ itself in the $\exp(ikt)$ basis with an odd number of points N, as in (2.14). (The trigonometric interpolant problem goes back at least to the young Gauss's calculations of the orbit of the asteroid Ceres in 1801—it seems that planetary orbits and trigonometric interpolation share a long and ongoing relationship.) Throughout this chapter, the number of grid points N will always be odd; all our results have analogues for N even.

If we replace $q(t)$ by its trigonometric interpolant $p_n(t)$ with $N = 2n + 1$ Fourier coefficients $\tilde{q}_k = \tilde{u}_k + i\tilde{v}_k$, (3.8) becomes a function of the $2N$ real variables $\{\tilde{u}_k, \tilde{v}_k\}$. For example, the first term in (3.8), which corresponds to the kinetic energy and that we denote by A_K, becomes (using Parseval's identity)

$$A_K = \frac{n}{2} \int_0^{2\pi} \left| p_n'(t) + i\omega p_n(t) \right|^2 dt = \pi n \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} (k+\omega)^2 (\tilde{u}_k^2 + \tilde{v}_k^2). \tag{3.9}$$

The second term in (3.8) is discretized with the trapezoidal rule, which converges exponentially since the integrands are analytic (we are looking for solutions $q(t)$ without collisions).

Let us now say more about the second ingredient. We derive formulas for the exact gradient and exact Hessian matrix in Appendix A. Using these formulas, the gradient and the Hessian matrix can be computed in $\mathcal{O}(nN^2)$ and $\mathcal{O}(nN^3)$ operations. For example, the first derivatives of (3.9) with respect to the optimization variables are given by

$$\frac{\partial A_K}{\partial \tilde{u}_k} = 2\pi n (k+\omega)^2 \tilde{u}_k, \quad \frac{\partial A_K}{\partial \tilde{v}_k} = 2\pi n (k+\omega)^2 \tilde{v}_k, \tag{3.10}$$

while the second derivatives are given by

$$\frac{\partial^2 A_K}{\partial \tilde{u}_l \partial \tilde{u}_k} = \frac{\partial^2 A_K}{\partial \tilde{v}_l \partial \tilde{v}_k} = 2\pi n k^2 \delta_{kl}, \quad \frac{\partial^2 A_K}{\partial \tilde{u}_l \partial \tilde{v}_k} = \frac{\partial^2 A_K}{\partial \tilde{v}_k \partial \tilde{u}_l} = 0, \tag{3.11}$$

with Kronecker delta δ_{kl}.

3.3.2 Details about the optimization algorithm

The numerical optimization of the action is carried out in two phases. First, we apply a quasi-Newton method [117, Chap. 6] with the exact gradient and a small number of Fourier coefficients ($N = 55$ and 75 in our experiments). Quasi-Newton methods are based on the approximation of the Hessian matrix (or its inverse) using rank-one or rank-two updates specified by gradient evaluations; among them, we choose the BFGS algorithm [145]. We take $\mathcal{O}(N)$ steps of the BFGS algorithm. The cost of this first phase is thus $\mathcal{O}(nN^3)$ since, at each step, BFGS computes the gradient in $\mathcal{O}(nN^2)$ operations and matrix-vector products in $\mathcal{O}(N^2)$ operations.

Second, we perform a small number $s = \mathcal{O}(1)$ of iterations of an approximate Newton method with exact Hessian and M Fourier coefficients, $M > N$. The starting point of the approximate Newton method is the output of the BFGS algorithm, padded with $M - N$ zeros. An exact Newton method would have an $\mathcal{O}(snM^3)$ cost, since, at each iteration, it requires the computation of the exact Hessian ($\mathcal{O}(nM^3)$ operations) and the solution a linear system ($\mathcal{O}(M^3)$ operations). To reduce this cost, we use an approximate Newton method. We compute the exact Hessian at the first iteration only, and compute its LDL^T decomposition ($\mathcal{O}(M^3)$ operations, a generalization of Cholesky decomposition for symmetric matrices that are not positive definitive, typically half as expensive as LU factorization). The subsequent iterations of Newton's method do not recompute the Hessian but use this factorization instead. The solution of the linear system can then be computed in $\mathcal{O}(M^2)$ operations at each iteration. The total cost of this second phase is then $\mathcal{O}(nM^3)$, and the total cost of the optimization is $\mathcal{O}(n(N^3 + M^3))$. The computational costs of the two phases are summarized in Table 3.1.

Let us add four comments about this optimization process. First, since the initial guess of Newton's method—the output of BFGS—is a good approximation of a choreography, the Hessian matrix does not vary significantly from one iteration to another. As a consequence, using the LDL^T factorization of the Hessian of the first iterate at each iteration does not affect the convergence very much. Second, at a minimum of the action, i.e., a choreography, the Hessian is positive definite, so we

Table 3.1: *Computational costs of the two phases of the optimization process for n bodies, N Fourier coefficients for BFGS and $M > N$ Fourier coefficients for Newton's method. The main cost of the first phase is the computation of the exact gradient ($\mathcal{O}(nN^2)$ operations, at every step) while the dominant cost of the second phase is the precomputation of the exact Hessian matrix ($\mathcal{O}(nM^3)$, at the first step only.).*

	Phase 1: BFGS	**Phase 2: Newton**
Number of coefficients	$55 \leq N \leq 75$	$145 \leq M \leq 455$
Precomputation cost	–	$\mathcal{O}(nM^3)$
Cost per iteration	$\mathcal{O}(nN^2)$	$\mathcal{O}(M^2)$
Number of iterations	$\mathcal{O}(N)$	$\mathcal{O}(1)$
Total cost	$\mathcal{O}(nN^3)$	$\mathcal{O}(nM^3)$
Accuracy	one to four digits	10 digits

could in principle use the Cholesky decomposition instead of the LDL^T decomposition. However, in practice, because the Hessian is computed at an approximation of a choreography, it often has some small negative eigenvalues. Third, we do not use Newton's method with exact Hessian from the beginning because it only converges for initial guesses close enough to the solution. Fourth, another option would be to use $M > N$ coefficients with BFGS, and not use Newton's method at all. However, we found in practice that BFGS alone (with M coefficients) typically achieves an accuracy of six digits at most, while BFGS (with N coefficients) plus Newton's method (with $M > N$ coefficients) achieves an accuracy of 10 digits.

For both steps of the optimization, the accuracy is defined as the relative L^2-norm of the residual of (3.1). We also check that the l^2-norm of the gradient divided by the l^2-norm of the gradient of the initial guess (relative l^2-norm) is close to zero, and that the Fourier coefficients of the solution decay to sufficiently small values.

3.3.3 Numerical results

The famous figure-eight, with action $A \approx 24.371926$ [36], is plotted in Figure 3.1 and its Fourier coefficients are plotted in Figure 3.2. These are obtained by running the code of Figure 3.3. The code uses the `actiongradeval` and `gradhesseval` functions, which compute the action and the gradient, and the gradient and the Hessian. These functions can be found in the Chebfun `cheb.choreo` code, an interactive tool to find

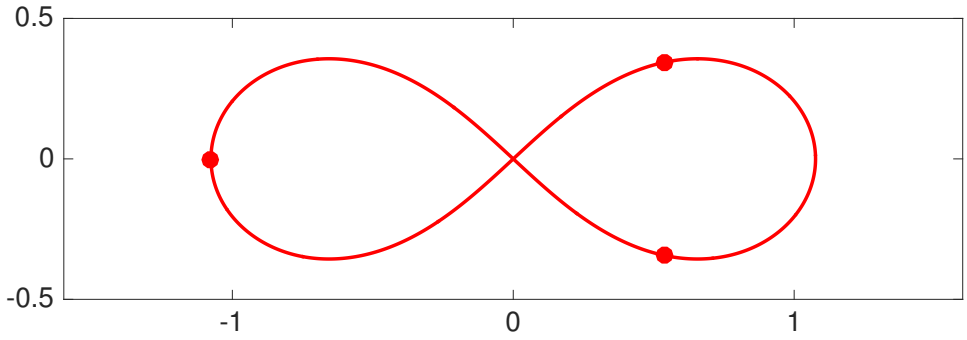

Figure 3.1: *The figure-eight solution of the planar three-body problem, obtained by running the code of Figure* 3.3. *The dots show the bodies at time* $t = 0$. *The action* $A = 24.371926476242809$ *agrees with the eight digits given in* [36].

Table 3.2: *Computation of the figure-eight choreography of Figure* 3.1.

	Phase 1: BFGS	**Phase 2: Newton**
Action	24.371926476245442	24.371926476242809
Number of coefficients	55	145
Computer time (s)	0.19	0.27
Number of iterations	51	2
Relative l^2-norm of gradient	2.9e-07	3.9e-16
Smallest coefficient	3.2e-08	2.1e-18
Relative L^2-norm of residual	2.1e-05	2.2e-11

choreographies starting with hand-drawn initial guesses. It is easy to use, fast, and enjoyable—the reader is encouraged to try it!

Table 3.2 shows some numbers pertaining to the computation of the figure-eight, including the relative norms of the solution and the gradient, and the amplitude of the smallest (numerically nonzero) Fourier coefficient. After 51 iterations of the BFGS algorithm, the solution is accurate to four digits, and two iterations of Newton's method gives six extra correct digits. The solutions found by BFGS and BFGS plus Newton look the same to the eye; if they were plotted on the same graph, they would be perfectly superimposed. The difference is visible in coefficient space, see Figure 3.2. Since choreographies are analytic functions, their Fourier coefficients decay geometrically. The solution obtained by the BFGS algorithm only uses 55 coefficients (wavenumbers $|k| \leq 27$), which decay to about 10^{-8}. The solution obtained by Newton's method uses 145 Fourier coefficients ($|k| \leq 72$), which decay to machine precision.

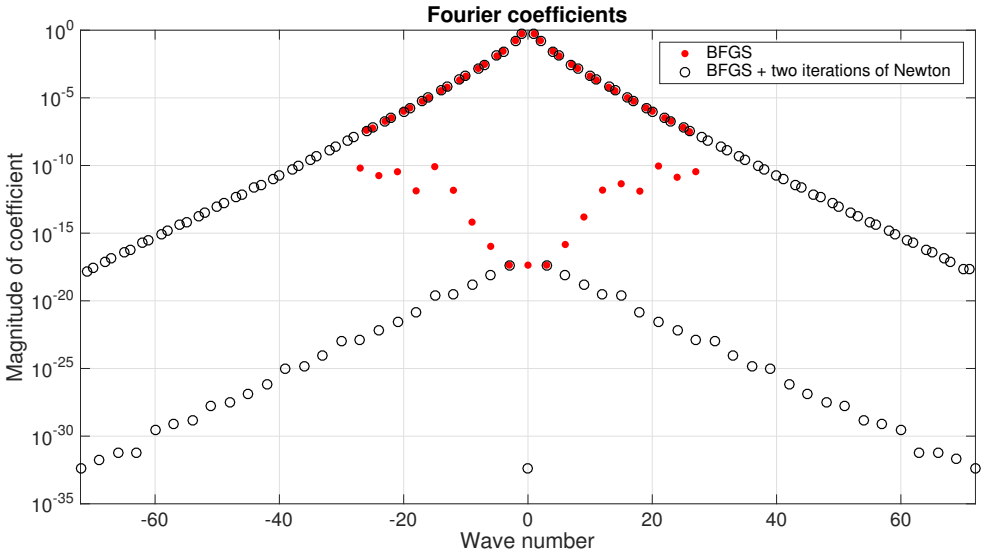

Figure 3.2: *Absolute values of the Fourier coefficients of Figure* 3.1*, obtained by BFGS (red dots) and BFGS followed by two iterations of Newton's method (black circles).*

```
% Initial guess:
  n = 3; N = 55; M = 145;
  q0 = chebfun(@(t)cos(t)+1i*sin(2*t),[0 2*pi],N,'trig');
  c0 = trigcoeffs(q0);

% BFGS algorithm:
  options = optimoptions('fminunc'); options.GradObj = 'on';
  options.Algorithm = 'quasi-newton'; options.HessUpdate = 'bfgs';
  c = fminunc(@(x)actiongradeval(x,n),[real(c0);imag(c0)],options);

% Two iterations of an approximate Newton method:
  c = [zeros((M-N)/2,1);c(1:N);zeros(M-N,1);c(N+1:end);zeros((M-N)/2,1)];
  mid = 1 + floor(M/2);
  [G, H] = gradhesseval(c,n); [L, D] = ldl(H);
  for k = 1:2
    s = L'\(D\(L\(-G)));
    cnew = c + [s(1:mid-1);0;s(mid:M+mid-2);0;s(M+mid-1:end)]; c = cnew;
    G = gradhesseval(c,n);
  end

% Reconstruct solution:
  c = c(1:M) + 1i*c(M+1:2*M);
  q = chebfun(c,[0 2*pi],'coeffs','trig');
```

Figure 3.3: MATLAB *code to compute Figure* 3.1*. This code gives results correct to* 10 *digits of accuracy in less than half a second on a* 2.7 GHz *Intel i7 machine.*

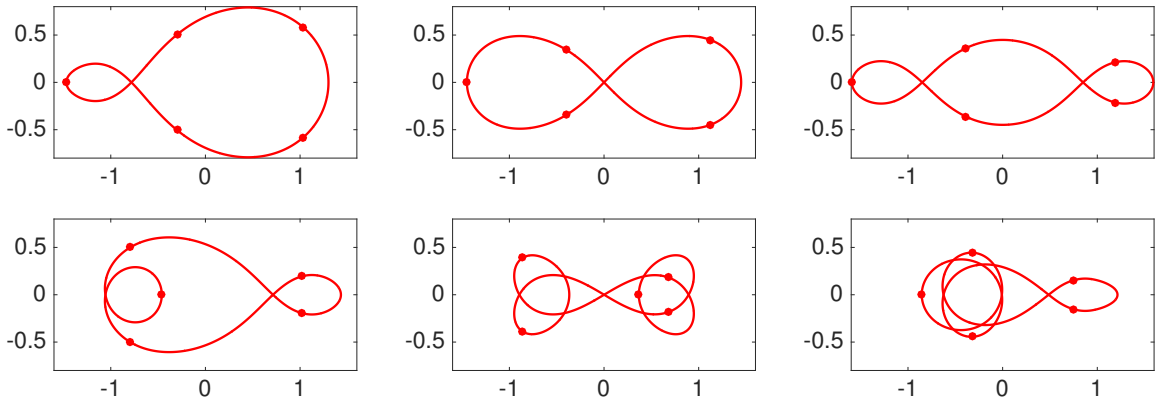

Figure 3.4: *Planar choreographies of the five-body problem. The dots show the bodies at* $t = 0$. *From top-left corner to bottom-right corner, these correspond to choreographies* 1, 2, 4, 7, 16 *and* 18 *of* [149, Fig. 2].

Table 3.3: *Computation of the absolute planar choreographies of Figure* 3.4 *with the BFGS algorithm* (*phase* 1).

	1	**2**	**4**	**7**	**16**	**18**
Number of coefficients	75	75	75	75	75	75
Computer time (s)	0.89	0.63	0.59	0.62	1.73	0.83
Number of iterations	98	68	69	136	373	166
Relative l^2-norm of gradient	3.3e-08	6.7e-08	3.4e-07	4.9e-09	3.2e-10	2.9e-09
Smallest coefficient	1.3e-05	6.1e-09	5.9e-06	8.1e-06	3.6e-05	6.5e-05
Relative L^2-norm of residual	3.5e-02	1.1e-05	6.7e-03	2.2e-02	2.8e-01	3.6e-01

All the planar absolute choreographies of the five-body problem found by Simó [149, Fig. 2] can be computed with this algorithm. We plot six of them in Figure 3.4. Tables 3.3 and 3.4 show some numbers pertaining to their computation. The BFGS algorithm leads to results accurate to a few digits, and two to six iterations of Newton's method lead to about 10 digits of accuracy.

The same method can be used to compute relative choreographies to high accuracy. We plot three relative planar choreographies of the seven-body problem in Figure 3.5.

Let us conclude this section with a few words about the number of choreographies for a given n. This number is not known, but there is an interesting result, due to Simó [149, Prop. 5.1], about the (smaller) number of choreographies that consist of a concatenation of "bubbles," such as the ones in the first row of Figure 3.4. For $n \geq 3$, there are $2^{n-3} + 2^{\lfloor (n-3)/2 \rfloor}$ such choreographies.

Table 3.4: *Computation of the absolute planar choreographies of Figure* 3.4 *with Newton's method using the outputs of BFGS as initial guesses (phase* 2*).*

	1	**2**	**4**	**7**	**16**	**18**
Number of coefficients	335	145	245	285	445	455
Computer time (s)	3.49	0.48	1.68	2.33	7.26	7.85
Number of iterations	4	2	3	4	5	6
Relative l^2-norm of gradient	6.9e-16	1.9e-16	1.7e-13	1.5e-16	3.7e-14	3.8e-14
Smallest coefficient	3.4e-15	8.8e-16	8.0e-17	2.1e-14	1.3e-16	1.4e-14
Relative L^2-norm of residual	8.3e-10	5.7e-11	7.3e-10	8.3e-10	1.5e-09	2.6e-09

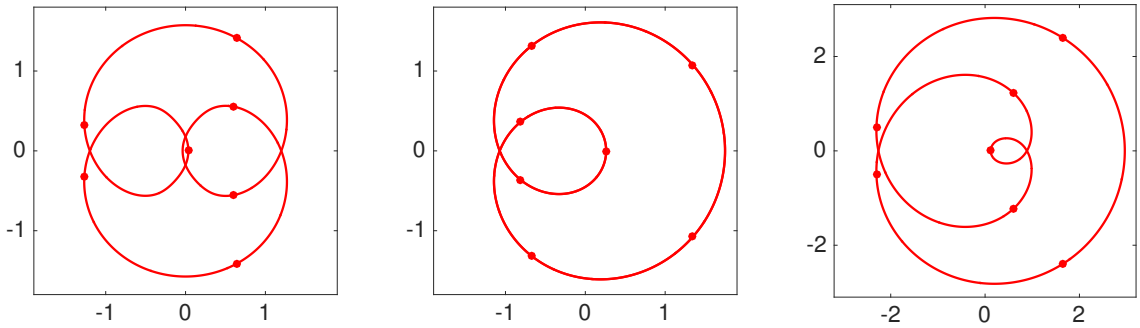

Figure 3.5: *Relative planar choreographies of the seven-body problem with angular velocity* 2.8 *(left),* -2.9 *(center) and* 2.31 *(right). The dots show the bodies at* $t = 0$*. They can be computed to ten digits of accuracy with about* 400 *Fourier coefficients.*

3.4 Spherical choreographies

Let $X_j(t) \in \mathbb{R}^3$, $0 \le j \le n-1$, denote the Cartesian coordinates of n bodies with unit mass on the sphere $\mathbb{S}^2_R = \{X \in \mathbb{R}^3, \|X\| = R\}$, where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^3$. The n-body problem on the sphere in a cotangent potential describes the motion of these bodies via the n coupled nonlinear ODEs

$$X_j''(t) - \sum_{\substack{i=0 \\ i\neq j}}^{n-1} \frac{R^3 X_i(t) - R(X_i(t)\cdot X_j(t))X_j(t)}{\left[R^4 - (X_i(t)\cdot X_j(t))^2\right]^{3/2}} + R^{-2}\left\|X_j'(t)\right\|^2 X_j(t) = 0, \qquad (3.12)$$

for $0 \le j \le n-1$. See [49] for details about the derivation of these equations. Note that the potential associated with (3.12) is no longer the Newtonian potential (3.5). It is a cotangent potential, a generalization of the Newtonian potential on the sphere, and dates back to the 1820s with the work of Bolyai and Lobachevsky. The reader can find a detailed history of the problem in Diacu's 2012 book [44].

We are looking for periodic solutions of (3.12) moving along the same orbit, i.e., solutions $X_j(t)$ such that

$$X_j(t) = Q\Big(t + \frac{2\pi j}{n}\Big), \quad 0 \le j \le n-1, \tag{3.13}$$

for some 2π-periodic function $Q : [0, 2\pi] \to \mathbb{S}^2_R \subset \mathbb{R}^3$. Again, the period can be chosen equal to 2π because if $Q(t)$ is a T-periodic solution of (3.12) on the sphere of radius R, then $\lambda^{-2/3}Q(\lambda t)$, $\lambda = T/(2\pi)$, is a 2π-periodic solution on the sphere of radius $\lambda^{-2/3}R$. We call these solutions *spherical choreographies*. They are minima of the action associated with (3.12), defined again as the integral over one period of the kinetic minus the potential energy, with kinetic energy

$$K(t) = \frac{1}{2}\sum_{j=0}^{n-1} \big\|X_j'(t)\big\|^2 = \frac{1}{2}\sum_{j=0}^{n-1} \Big\|Q'\Big(t + \frac{2\pi j}{n}\Big)\Big\|^2 \tag{3.14}$$

and potential energy

$$U(t) = -\frac{1}{R}\sum_{j=0}^{n-1}\sum_{i=0}^{j-1} \cot\frac{\hat{d}(X_i(t), X_j(t))}{R}, \tag{3.15}$$

where

$$\hat{d}(X_i(t), X_j(t)) = R\arccos\frac{X_i(t)\cdot X_j(t)}{R^2} \tag{3.16}$$

is the great-circle distance between $X_i(t)$ and $X_j(t)$ on $\mathbb{S}^2_R$. The potential (3.15) is the cotangent of the (rescaled) distance on the sphere. Using the trigonometric identity $\cot(\arccos(x)) = x/\sqrt{1-x^2}$, the potential energy can be rewritten

$$U(t) = -\frac{1}{R}\sum_{j=0}^{n-1}\sum_{i=0}^{j-1} \frac{X_i(t)\cdot X_j(t)}{\sqrt{R^4 - (X_i(t)\cdot X_j(t))^2}}. \tag{3.17}$$

The action is then given by

$$A = \frac{n}{2}\int_0^{2\pi} \big\|Q'(t)\big\|^2 dt + \frac{n}{2R}\sum_{j=1}^{n-1}\int_0^{2\pi} \frac{Q(t)\cdot Q\big(t+\frac{2\pi j}{n}\big)}{\sqrt{R^4 - \big(Q(t)\cdot Q\big(t+\frac{2\pi j}{n}\big)\big)^2}}\, dt. \tag{3.18}$$

Spherical choreographies correspond to functions $Q(t)$ that minimize (3.18). Note that since the cotangent potential (3.15) is singular not only when the distance between two bodies is zero but also for antipodal configurations. See [50] for details about the singularities of the n-body problem in a cotangent potential.

As in the plane, we are also interested in solutions of (3.12) in which the bodies share a single orbit $Q(t)$ that is rotating with angular velocity ω along the z-axis relative to an inertial reference frame, i.e.,

$$X_j(t) = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} Q\Big(t + \frac{2\pi j}{n}\Big), \quad 0 \le j \le n-1. \tag{3.19}$$

Let $R_\omega(t)$ denote the rotation matrix in (3.19). The kinetic part of the action associated with *relative* spherical choreographies is

$$\frac{n}{2}\int_0^{2\pi} \big\|R_\omega(t)Q'(t) + R'_\omega(t)Q(t)\big\|^2 dt. \tag{3.20}$$

3.5 Computing spherical choreographies

3.5.1 Methodology

Our method for computing spherical choreographies is based on *stereographic projection* and on the algorithm used for planar choreographies. Points $X = (x_1, x_2, x_3)^T$ on the sphere $\mathbb{S}^2_R$ are mapped to points $z = P_R(X)$ in the complex plane $\mathbb{C}$ via

$$z = P_R(X) = \frac{Rx_1 + iRx_2}{R - x_3}. \tag{3.21}$$

The inverse mapping is given by

$$X = P_R^{-1}(z) = \frac{1}{R^2 + |z|^2}(2R^2\mathrm{Re}(z), 2R^2\mathrm{Im}(z), -R^3 + R|z|^2)^T. \tag{3.22}$$

The Euclidean distance $d(X,Y) = \|X - Y\|$ between two points on the sphere is transformed into the distance $d(z,\xi)$ between their projections $z = P_R(X)$ and $\xi = P_R(Y)$ defined by

$$d(z,\xi) = \frac{2R^2|z-\xi|}{\sqrt{(R^2+|z|^2)(R^2+|\xi|^2)}}, \tag{3.23}$$

and the great-circle distance (3.16) is transformed into

$$\hat{d}(z,\xi) = 2R\arcsin\frac{d(z,\xi)}{2R}. \tag{3.24}$$

The complex plane endowed with the distance (3.24) is called the *spherical plane*. Let $q(t) = P_R(Q(t))$ denote the projection of the curve $Q(t)$ onto $\mathbb{C}$, and let

$$z_j(t) = P_R(X_j(t)) = P_R\Big(Q\Big(t + \frac{2\pi j}{n}\Big)\Big) = q\Big(t + \frac{2\pi j}{n}\Big), \quad 0 \le j \le n-1, \tag{3.25}$$

denote the projections of the n bodies $X_j(t)$. The action (3.18)–(3.20) can be then reformulated as

$$A = \frac{n}{2}\int_0^{2\pi}\left(\frac{2R^2|q'(t)+i\omega q(t)|}{R^2+|q(t)|^2}\right)^2 dt + \frac{n}{2R}\sum_{j=1}^{n-1}\int_0^{2\pi}\frac{2R^2-D_j(t)^2}{D_j(t)\sqrt{4R^2-D_j(t)^2}}\,dt, \quad (3.26)$$

with $D_j(t) = d\big(q(t), q\big(t+\frac{2\pi j}{n}\big)\big)$. Pérez–Chavela and Reyes–Victoria [128, Th. 2.3] showed the equivalence of the formulations (3.18)–(3.20) and (3.26) when $\omega = 0$.

Once the problem is reformulated in the spherical plane, we apply the two key ingredients of Section 3.3 and also proceed in two phases. Formulas for the gradient and the Hessian matrix of the action (3.26) are derived in Appendix A. As in the plane, at convergence, we check that the relative l^2-norm of the gradient of the action is close to zero, that the Fourier coefficients of the solution decay to sufficiently small values, and that the solution satisfies (3.12) projected into the plane (by computing the relative L^2-error). The projected equations of motion were first given by Pérez–Chavela and Reyes–Victoria in 2012 [128, Lemma 2.1] and can be written as

$$z_j''(t) = \frac{2\bar{z}_j(t)z_j'^2(t)}{R^2+|z_j(t)|^2} + \frac{4R}{\lambda_j(t)}\sum_{\substack{i=0\\ i\neq j}}^{n-1}\frac{P_{j,i}(t)}{\Theta_{j,i}(t)^{3/2}}, \quad 0\leq j\leq n-1, \quad (3.27)$$

where $\lambda_j(t) = 4R^4/(R^2+|z_j(t)|^2)^2$ is the conformal factor that appears in the kinetic part of (3.26), while $P_{j,i}(t)$ and $\Theta_{j,i}(t)$ are defined by

$$P_{j,i}(t) = \big[R^2+|z_j(t)|^2\big]\big[R^2+|z_i(t)|^2\big]^2\big[R^2+\bar{z}_i(t)z_j(t)\big]\big[z_i(t)-z_j(t)\big], \quad (3.28)$$

and

$$\begin{aligned}\Theta_{j,i}(t) = &-\big[2R^2z_j(t)\bar{z}_i(t)+2R^2z_i(t)\bar{z}_j(t)+(|z_j(t)|^2-R^2)(|z_i(t)|^2-R^2)\big]^2\\ &+\big[R^2+|z_j(t)|^2\big]^2\big[R^2+|z_i(t)|^2\big]^2.\end{aligned} \quad (3.29)$$

3.5.2 Numerical examples

As we mentioned in the introduction, only a few noncircular spherical choreographies have been found previously. Diacu and collaborators [49], and Pérez–Chavela and Reyes–Victoria [128] also characterized the solutions of the spherical n-body problem, in which the bodies move along the same circle (such as Figure 3.6) or along different ones—the *relative equilibria*.

We now present new noncircular spherical choreographies. The first one is the *spherical figure-eight*, a solution of the three-body problem on the sphere of radius

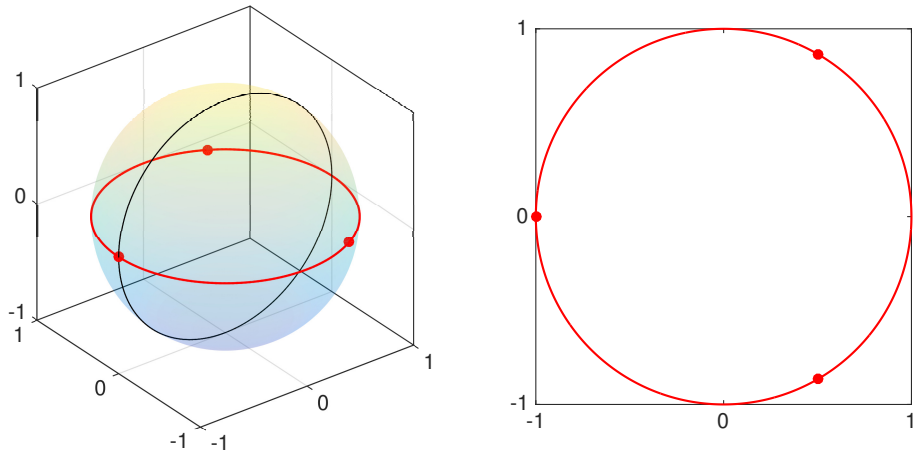

Figure 3.6: *A circular choreography of the spherical three-body problem on the unit sphere (left) and its projection in the plane (right). The dots show the bodies at* $t = 0$.

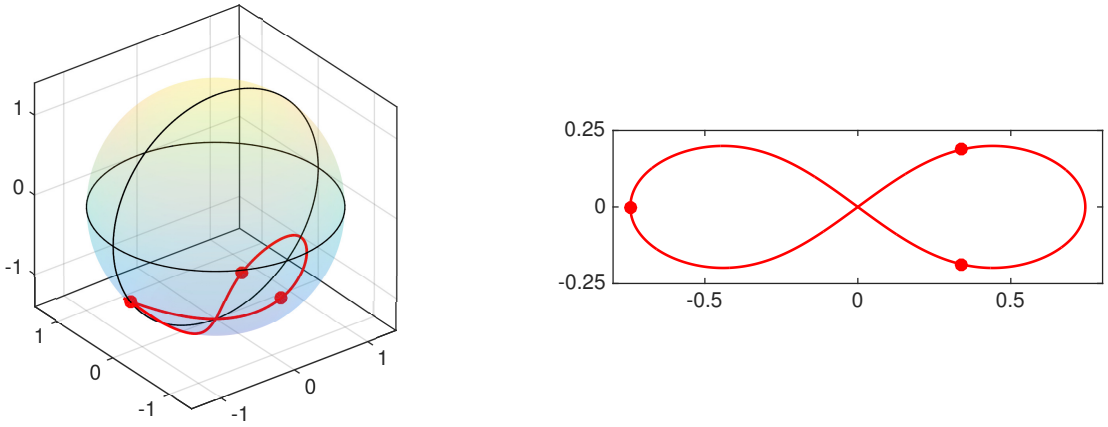

Figure 3.7: *Spherical figure-eight choreography on the sphere of radius* $R = 1.4$ *(left) and its projection in the plane (right). The dots show the bodies at* $t = 0$.

Table 3.5: *Computation of the spherical figure-eight choreography of Figure* 3.7.

	Phase 1: BFGS	**Phase 2: Newton**
Action	18.948304138530286	18.948304135957898
Number of coefficients	55	195
Computer time (s)	0.51	3.19
Number of iterations	72	2
Relative l^2-norm of gradient	5.2e-07	1.1e-13
Smallest coefficient	2.5e-07	1.1e-17
Relative L^2-norm of residual	4.1e-04	8.8e-13

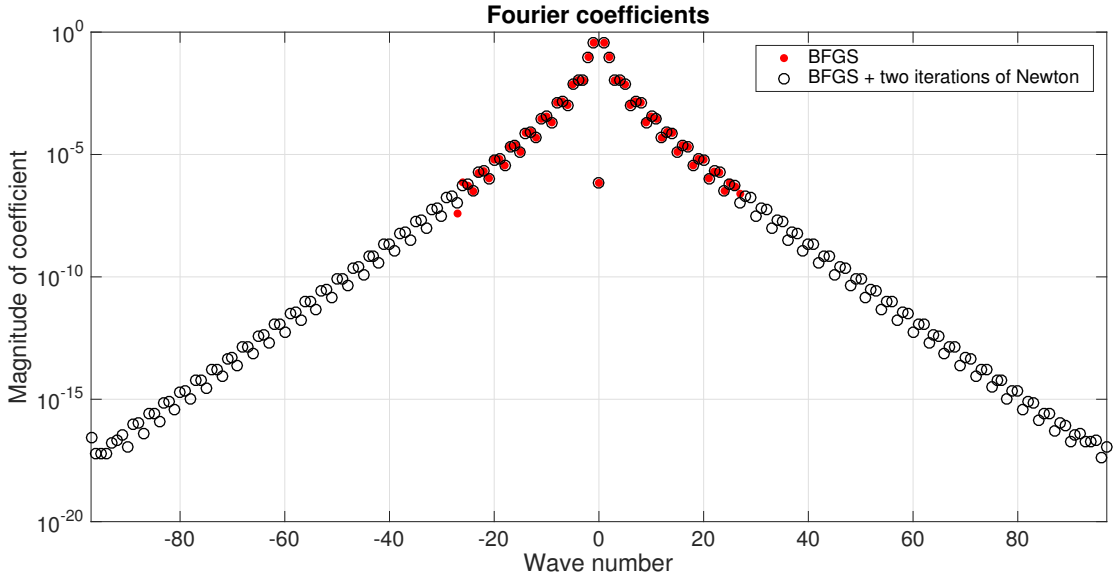

Figure 3.8: *Absolute values of the Fourier coefficients of Figure* 3.7*, obtained by BFGS* (*red dots*) *and BFGS followed by one step of Newton's method* (*black circles*).

R, shown in Figure 3.7 for $R = 1.4$. Table 3.5 shows some numbers pertaining to its computation. Combining the BFGS algorithm with Newton's method leads to 12 digits of accuracy. We plot the geometrically decaying Fourier coefficients of the outputs of BFGS and Newton's method in Figure 3.8. After BFGS, they decay to 10^{-7}, and after two iterations of Newton's method, they decay to machine precision. Numerically, we found that the spherical figure-eight exists on spheres of radius $R \geq 1.32$. Below this value, it cannot fit into a single hemisphere and would therefore lead to antipodal singularities.[1]

Many new spherical choreographies can be found with our algorithm. We show in Figure 3.9 three spherical choreographies of the five-body problem on the sphere of radius 2. These are curved versions of the choreographies of Figure 3.4. Table 3.6 shows some numbers pertaining to their computation. We get one to four digits of accuracy with the BFGS algorithm, and applying Newton's method with the outputs of BFGS as initial guesses leads to eight to 12 digits of accuracy.

Relative spherical choreographies can also be computed with this method. We plot three relative spherical choreographies of the seven-body problem on the sphere of radius 2.5 in Figure 3.10. These are curved versions of the relative choreographies of Figure 3.5.

[1]The solution of Figure 3.7 exists for radii $R < 1.32$ but with a period $T < 2\pi$ that can be obtained from the scaling invariance $R^3/T^2 = (1.4)^3/(2\pi)^2$ described below (3.13).

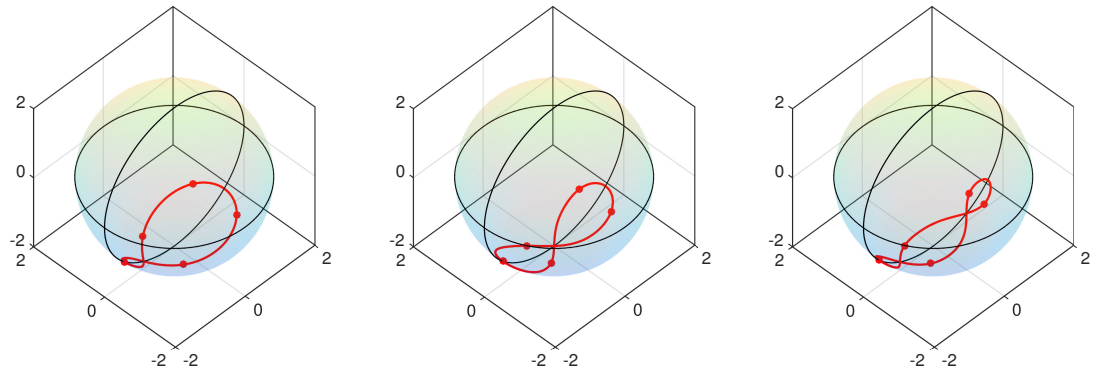

Figure 3.9: *Spherical choreographies of the five-body problem on the sphere of radius* $R = 2$*, analogous to the planar choreographies of Figure* 3.4*.*

Table 3.6: *Computation of the spherical choreographies of Figure* 3.9*.*

	Left		**Center**		**Right**	
	Phase 1	**Phase 2**	**Phase 1**	**Phase 2**	**Phase 1**	**Phase 2**
Number of coefficients	75	375	75	205	75	245
Computer time (s)	2.35	33.13	1.05	6.08	2.14	9.44
Number of iterations	142	6	65	3	115	4
Relative l^2-norm of gradient	3.6e-05	9.8e-11	3.5e-07	7.6e-14	2.0e-07	5.7e-14
Smallest coefficient	1.9e-05	2.9e-14	4.5e-08	1.2e-17	6.4e-06	2.5e-15
Relative L^2-norm of residual	4.3e-02	1.7e-09	5.6e-05	7.3e-13	7.1e-03	9.3e-09

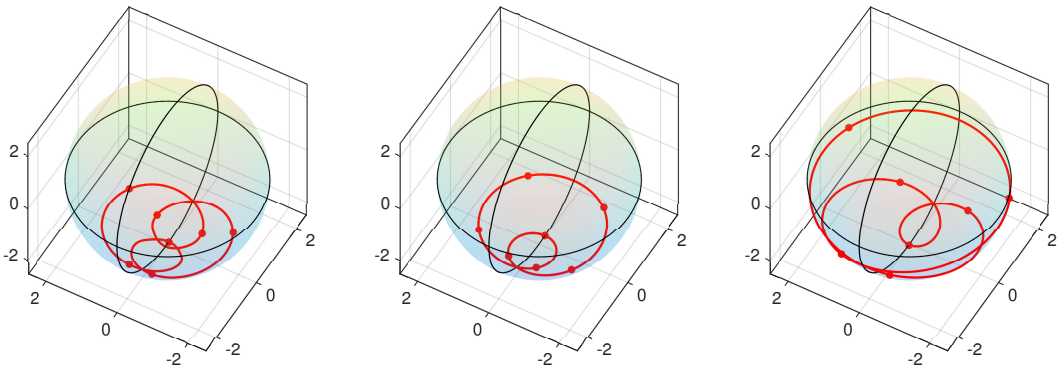

Figure 3.10: *Relative spherical choreographies of the seven-body problem on the sphere of radius* $R = 2.5$ *with angular velocities* 2.8 (*left*)*,* -2.9 (*center*) *and* 2.31 (*right*)*, analogous to the relative choreographies of Figure* 3.5*.*

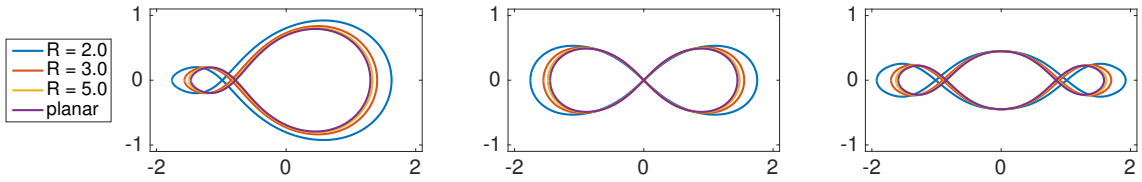

Figure 3.11: *Spherical choreographies of Figure* 3.9 (*multiplied by a factor* 2) *for different values of* R*, together with their planar analogues of Figure* 3.4*. As* R *increases, the spherical choreographies converge to the planar ones.*

Table 3.7: *Convergence of spherical to planar choreographies in Figure* 3.11.

	$R=2$	3	5	10	100	1000
Left	3.3e-01	1.1e-01	3.4e-02	8.1e-03	8.0e-05	8.0e-07
Middle	3.1e-01	1.0e-01	3.4e-02	8.2e-03	8.1e-05	8.0e-07
Right	3.3e-01	1.1e-01	3.6e-02	8.8e-03	8.7e-05	8.7e-07

An interactive tool to compute spherical choreographies using hand-drawn initial guesses, `cheb.choreosphere`, is also available in Chebfun.

3.5.3 Limit of infinitely large radius

As its radius R gets bigger, the sphere gets flatter, and in the limit $R \to \infty$, it converges to the complex plane. Equivalently, the spherical plane converges to the complex plane. The distances (3.23) and (3.24) converge to twice the absolute value, and the action (3.26) on the sphere converges to four times the action (3.8) in the plane, since it involves squares of distances. We might then expect that twice the spherical choreographies converge to the planar choreographies as $R \to \infty$, and this is indeed the case.[2] In Figures 3.11 and 3.12, we plot the spherical choreographies of Figures 3.9 and 3.10 (multiplied by a factor 2) for increasing values of R and plot them together with their planar analogues. Tables 3.7 and 3.8 report the ∞-norm of the difference between analogous spherical and planar choreographies as R increases. It is clear from the tables that spherical choreographies converge to their planar analogues at a rate proportional to the curvature $1/R^2$.

[2]We are studying the convergence with a fixed period 2π. Similarly, T-periodic spherical choreographies converge to T-periodic planar choreographies.

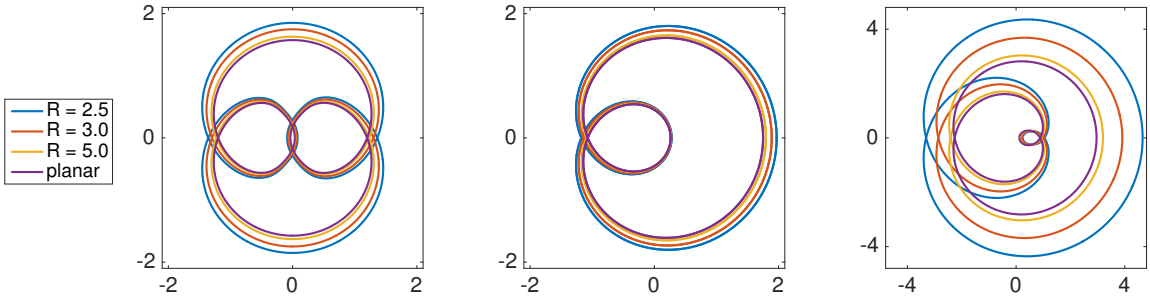

Figure 3.12: *Spherical choreographies of Figure* 3.10 (*multiplied by a factor* 2) *for different values of R, together with their planar analogues of Figure* 3.5. *As R increases, the spherical choreographies converge to the planar ones.*

Table 3.8: *Convergence of spherical to planar choreographies in Figure* 3.12.

	$R = 2.5$	3	5	10	100	1000
Left	2.8e-01	1.7e-01	5.5e-02	1.3e-02	1.3e-04	1.3e-06
Middle	2.2e-01	1.4e-01	4.5e-02	1.1e-02	1.1e-04	1.1e-06
Right	1.7e+00	9.5e-01	2.4e-01	5.2e-02	5.0e-04	5.0e-06

3.6 Hyperbolic choreographies

While there really is only one model of two-dimensional spherical geometry (the sphere $\mathbb{S}^2_R$ with the great-circle distance), there are several models of hyperbolic geometry, including the Beltrami–Klein disk, the Poincaré disk, the Poincaré half-plane and the Lorentz hyperboloid models, with appropriate geodesic distances. In this chapter, we first use the last of these and then we reformulate the problem on the Poincaré disk using stereographic projection, following [48].

To describe hyperbolic geometry, the Lorentz model uses the forward sheet of a two-sheeted hyperboloid, defined as

$$\mathbb{L}^2_R = \{X = (x_1, x_2, x_3)^T \in \mathbb{R}^3,\ X \odot X = -R^2,\ x_3 > 0\}, \quad R > 0, \tag{3.30}$$

with Lorentz inner product

$$X \odot Y = x_1 y_1 + x_2 y_2 - x_3 y_3 \tag{3.31}$$

and Lorentz distance

$$d(X, Y) = \sqrt{(X - Y) \odot (X - Y)} \tag{3.32}$$

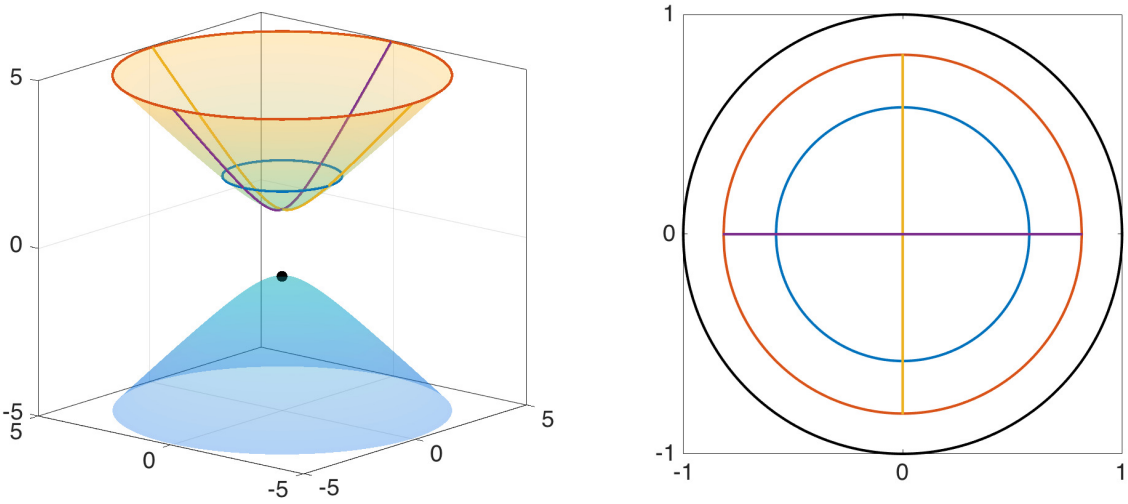

Figure 3.13: *On the left, a two-sheeted hyperboloid with* $R = 1$. *The space* $\mathbb{L}^2_R$ *corresponds to the forward sheet only. The north pole* $(0, 0, -R)$ *of the backward sheet is indicated by a black dot. On the right, the Poincaré disk* $|z| < 1$, *obtained using stereographic projection from the north pole of the backward sheet. The curves correspond to the projections of the curves of the same colour on the forward sheet. The black line, corresponding to* $|z| = 1$, *is the projection of the point at infinity.*

for $X = (x_1, x_2, x_3)^T$ and $Y = (y_1, y_2, y_3)^T$ on $\mathbb{L}^2_R$. A two-sheeted hyperboloid with $R = 1$ is shown in Figure 3.13 (left). The geodesic distance between X and Y on $\mathbb{L}^2_R$ is defined as

$$\hat{d}(X, Y) = R\,\mathrm{acosh}\Big(-\frac{X \odot Y}{R^2}\Big). \tag{3.33}$$

The forward sheet $\mathbb{L}^2_R$ together with the geodesic distance (3.33) is called the *Lorentz hyperboloid model*.

The n-body problem on $\mathbb{L}^2_R$ describes the motion of n bodies on $\mathbb{L}^2_R$ with Cartesian coordinates $X_j(t) \in \mathbb{R}^3$, $0 \le j \le n-1$, via the n coupled nonlinear ODEs

$$X_j''(t) - \sum_{\substack{i=0 \\ i\neq j}}^{n-1} \frac{R^3 X_i(t) + R(X_i(t) \odot X_j(t))X_j(t)}{\big[(X_i(t) \odot X_j(t))^2 - R^4\big]^{3/2}} - R^{-2}\big(X_j'(t) \odot X_j'(t)\big)X_j(t) = 0, \tag{3.34}$$

for $0 \le j \le n-1$. The potential associated with (3.34) is a hyperbolic cotangent potential. It is a generalization of the Newtonian potential and dates back to the 19th century with the work of Bolyai, Lobachevsky and Killing.

We are looking for *hyperbolic choreographies*, i.e., solutions $X_j(t)$ such that

$$X_j(t) = Q\Big(t + \frac{2\pi j}{n}\Big), \quad 0 \le j \le n-1, \tag{3.35}$$

for some 2π-periodic function $Q : [0, 2\pi] \to \mathbb{L}^2_R$. Again, we can choose the period equal to 2π since if $Q(t)$ is a T-periodic of (3.34) on $\mathbb{L}^2_R$ then $\lambda^{-2/3}Q(\lambda t)$, $\lambda = T/(2\pi)$, is a 2π-periodic solution in $\mathbb{L}^2_{R'}$ with $R' = \lambda^{-2/3}R$. As in the plane and on the sphere, they correspond to minima of the action associated with (3.34), defined as the integral over one period of the kinetic minus potential energy,

$$A = \int_0^{2\pi} \big(K(t) - U(t)\big)dt, \tag{3.36}$$

with kinetic energy

$$K(t) = \frac{1}{2}\sum_{j=0}^{n-1} X_j'(t) \odot X_j'(t) = \frac{1}{2}\sum_{j=0}^{n-1} Q'\Big(t + \frac{2\pi j}{n}\Big) \odot Q'\Big(t + \frac{2\pi j}{n}\Big) \tag{3.37}$$

and potential energy

$$U(t) = -\frac{1}{R}\sum_{j=0}^{n-1}\sum_{i=0}^{j-1} \coth \frac{\hat{d}(X_i(t), X_j(t))}{R}. \tag{3.38}$$

Using the trigonometric identity $\cot(\operatorname{acosh}(x)) = x/\sqrt{x^2-1}$, the potential energy can be rewritten

$$U(t) = \frac{1}{R}\sum_{j=0}^{n-1}\sum_{i=0}^{j-1} \frac{X_i(t) \cdot X_j(t)}{\sqrt{(X_i(t) \odot X_j(t))^2 - R^4}}. \tag{3.39}$$

Since the integral of (3.37) does not depend on j and the integral of (3.39) only depends on $i - j$, the action is given by

$$A = \frac{n}{2}\int_0^{2\pi} Q'(t) \odot Q'(t)\, dt - \frac{n}{2R}\sum_{j=1}^{n-1}\int_0^{2\pi} \frac{Q(t) \odot Q\big(t + \frac{2\pi j}{n}\big)}{\sqrt{\big(Q(t) \odot Q\big(t + \frac{2\pi j}{n}\big)\big)^2 - R^4}}\, dt. \tag{3.40}$$

Hyperbolic choreographies correspond to functions $Q(t)$ that minimize (3.40).

We are also looking for *relative* hyperbolic choreographies,

$$X_j(t) = R_\omega(t)\, Q\Big(t + \frac{2\pi j}{n}\Big),\ R_\omega(t) = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix},\ 0 \le j \le n-1, \tag{3.41}$$

i.e., choreographies rotating with angular velocity ω along the z-axis. In this case, the kinetic part of (3.40) is

$$\frac{n}{2}\int_0^{2\pi} \big(R_\omega(t)Q'(t) + R_\omega'(t)Q(t)\big) \odot \big(R_\omega(t)Q'(t) + R_\omega'(t)Q(t)\big)\, dt. \tag{3.42}$$

3.7 Computing hyperbolic choreographies

3.7.1 Methodology

Our method for computing hyperbolic choreographies is based on the reformulation of the minimization problem (3.40)–(3.42) on the Poincaré disk using stereographic projection. Points $X = (x_1, x_2, x_3)^T$ on $\mathbb{L}^2_R$ are mapped to points $z = P_R(X)$ on the Poincaré disk $\mathbb{D}^2_R = \{z \in \mathbb{C},\ |z| < R\}$ via

$$z = P_R(X) = \frac{Rx_1 + iRx_2}{R + x_3}. \tag{3.43}$$

The inverse mapping is given by

$$X = P_R^{-1}(z) = \frac{1}{R^2 - |z|^2}(2R^2\mathrm{Re}(z), 2R^2\mathrm{Im}(z), R^3 + R|z|^2)^T. \tag{3.44}$$

Note that (3.43) is a stereographic projection from the north pole $(0, 0, -R)$ of the backward sheet of the hyperboloid—see Figure 3.13 (right) for an example of such a projection. The Lorentz distance (3.32) between two points on $\mathbb{L}^2_R$ is transformed into the distance $d(z, \xi)$ between their projections $z = P_R(X)$ and $\xi = P_R(Y)$ defined as

$$d(z, \xi) = \frac{2R^2|z - \xi|}{\sqrt{(R^2 - |z|^2)(R^2 - |\xi|^2)}}, \tag{3.45}$$

and the geodesic distance (3.33) into

$$\hat{d}(z, \xi) = 2R\,\mathrm{asinh}\frac{d(z, \xi)}{2R}. \tag{3.46}$$

The Poincaré disk $\mathbb{D}^2_R$ together with the geodesic distance (3.46) is called the *Poincaré disk model*. Let $q(t) = P_R(Q(t))$ denote the projection of $Q(t) \in \mathbb{L}^2_R$ onto $\mathbb{D}^2_R$, and let

$$z_j(t) = P_R(X_j(t)) = P_R\Big(Q\Big(t + \frac{2\pi j}{n}\Big)\Big) = q\Big(t + \frac{2\pi j}{n}\Big), \quad 0 \le j \le n - 1, \tag{3.47}$$

denote the projections of the n bodies. The kinetic part (3.42) of the action can be rewritten as

$$\frac{n}{2}\int_0^{2\pi}\left(\frac{2R^2|q'(t) + i\omega q(t)|}{R^2 - |q(t)|^2}\right)^2 dt \tag{3.48}$$

with conformal factor $4R^4/(R^2 - |q(t)|^2)^2$. To derive the formula for the potential part of (3.40) in intrinsic coordinates, let us come back to the potential energy (3.39). On the Poincaré disk $\mathbb{D}^2_R$, this is given by

$$U(t) = -\frac{1}{R}\sum_{j=0}^{n-1}\sum_{i=0}^{j-1}\coth\frac{\hat{d}(z_i(t), z_j(t))}{R} = -\frac{1}{R}\sum_{j=0}^{n-1}\sum_{i=0}^{j-1}\coth\left(2\,\mathrm{asinh}\frac{d(z, \xi)}{2R}\right). \tag{3.49}$$

Using the trigonometric identity $\coth(2\,\mathrm{asinh}(x/2)) = (x^2/2+1)/(x\sqrt{x^2/4+1})$ and integrating over one period, we find that the action is given by

$$A = \frac{n}{2}\int_0^{2\pi}\left(\frac{2R^2|q'(t)+i\omega q(t)|}{R^2-|q(t)|^2}\right)^2 dt + \frac{n}{2R}\sum_{j=1}^{n-1}\int_0^{2\pi}\frac{2R^2+D_j(t)^2}{D_j(t)\sqrt{4R^2+D_j(t)^2}}\,dt, \quad (3.50)$$

with $D_j(t) = d\big(q(t), q\big(t+\frac{2\pi j}{n}\big)\big)$.

To compute hyperbolic choreographies, we use the same strategy as before. Formulas for the gradient and the Hessian matrix of the action (3.50) are derived in Appendix A and used in the optimization algorithms.

Once a choreography has been computed by our algorithm, we check that it is a solution of the equations of motion (3.34) projected onto the Poincaré disk (by computing the relative L^2-error). These were first derived in [48] and are given by

$$z_j''(t) = -\frac{2\bar{z}_j(t)z_j'^2(t)}{R^2-|z_j(t)|^2} + \frac{4R}{\lambda_j(t)}\sum_{\substack{i=0\\ i\neq j}}^{n-1}\frac{P_{j,i}(t)}{\Theta_{j,i}(t)^{3/2}}, \quad 0\le j\le n-1, \quad (3.51)$$

where $\lambda_j(t) = 4R^4/(R^2-|z_j(t)|^2)^2$ is the conformal factor mentioned below (3.48), while $P_{j,i}(t)$ and $\Theta_{j,i}(t)$ are defined by

$$P_{j,i}(t) = \big[R^2-|z_j(t)|^2\big]\big[R^2-|z_i(t)|^2\big]^2\big[R^2-\bar{z}_i(t)z_j(t)\big]\big[z_i(t)-z_j(t)\big], \quad (3.52)$$

and

$$\begin{aligned}\Theta_{j,i}(t) = {}& \big[2R^2z_j(t)\bar{z}_i(t)+2R^2z_i(t)\bar{z}_j(t)-(|z_j(t)|^2+R^2)(|z_i(t)|^2+R^2)\big]^2\\ & -\big[R^2-|z_j(t)|^2\big]^2\big[R^2-|z_i(t)|^2\big]^2.\end{aligned} \quad (3.53)$$

We also check that the Fourier coefficients of the solution decay to sufficiently small values and that the gradient of the action (3.50) has a small relative l^2-norm.

3.7.2 Numerical examples

The first choreography that we present is the hyperbolic figure-eight of the three-body problem with $R = 1.5$, see Figure 3.14. Table 3.9 shows that, after 87 iterations of the first phase, the choreography satisfies (3.51) to six digits and after two iterations of the second phase it satisfies it to twelve digits. Figure 3.15 shows the Fourier coefficients of the solution, they decay to about 10^{-10} after the first phase and to about 10^{-16} after the second phase.

Many choreographies can be found with our algorithm. We show in Figure 3.16 three hyperbolic choreographies of the five-body problem with $R = 1.2$. These are

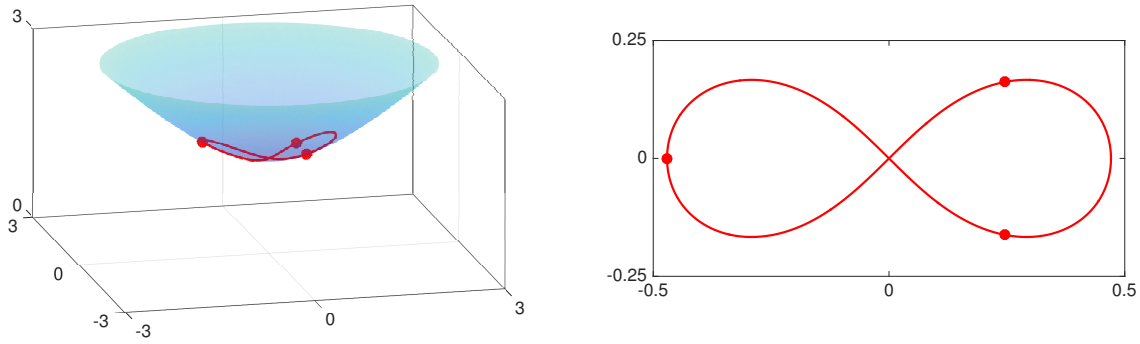

Figure 3.14: *Hyperbolic figure-eight choreography with* $R = 1.5$ (*left*) *and its projection on the Poincaré disk* (*right*). *The dots show the bodies at* $t = 0$.

Table 3.9: *Computation of the hyperbolic figure-eight choreography of Figure* 3.14.

	Phase 1: BFGS	**Phase 2: Newton**
Action	27.840867421590943	27.840867421590929
Number of coefficients	55	105
Computer time (s)	0.7734	0.6427
Number of iterations	87	2
Relative l^2-norm of gradient	8.1e-08	3.9e-15
Smallest coefficient	4.8e-10	1.0e-16
Relative L^2-norm of residual	9.4e-07	3.5e-13

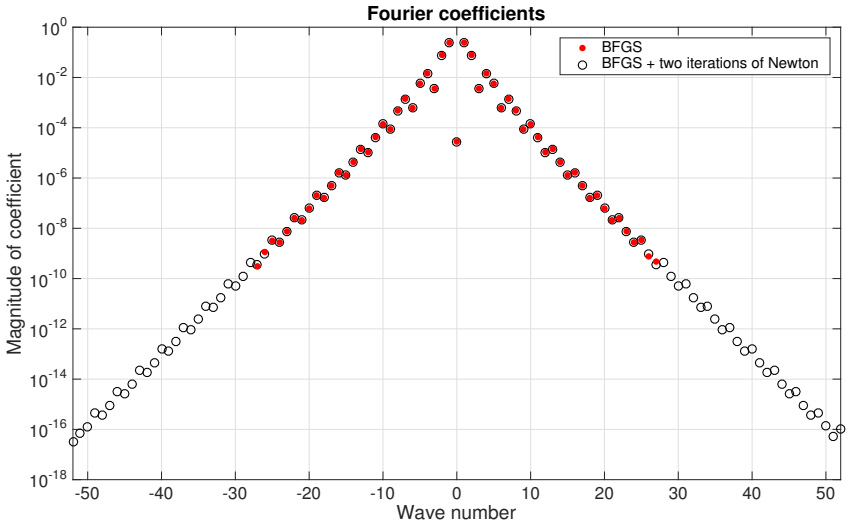

Figure 3.15: *Absolute values of the Fourier coefficients of Figure* 3.14, *obtained by BFGS* (*red dots*) *and BFGS followed by two steps of Newton's method* (*black circles*).

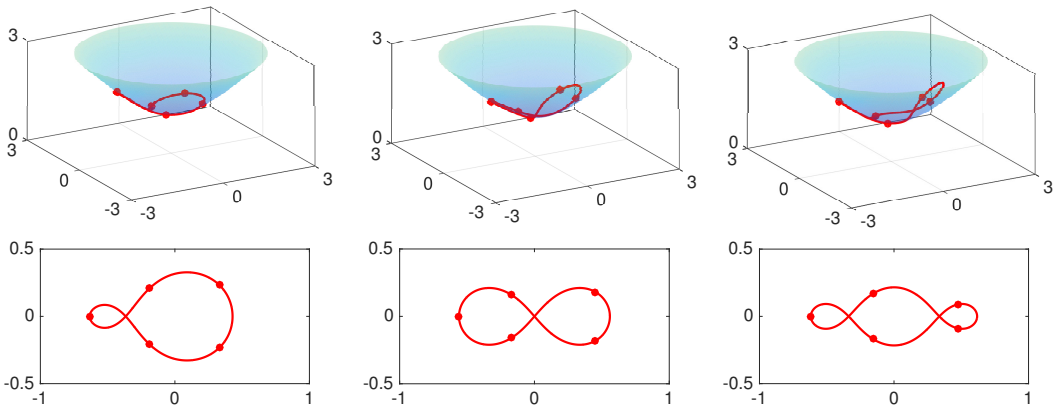

Figure 3.16: *Hyperbolic choreographies with* $R = 1.2$ *(top) and their projections on the Poincaré disk (bottom). The dots show the bodies at* $t = 0$.

Table 3.10: *Computation of the hyperbolic choreographies of Figure* 3.16.

	Left		**Center**		**Right**	
	Phase 1	**Phase 2**	**Phase 1**	**Phase 2**	**Phase 1**	**Phase 2**
Number of coefficients	75	305	55	155	65	245
Computer time (s)	2.52	17.54	0.75	3.02	1.03	10.88
Number of iterations	112	4	68	2	105	4
Relative l^2-norm of gradient	2.8e-08	5.8e-13	1.0e-07	8.3e-15	3.0e-09	9.9e-15
Smallest coefficient	1.6e-06	1.3e-17	1.1e-09	3.5e-18	3.9e-08	5.9e-18
Relative L^2-norm of residual	3.6e-03	3.5e-12	7.2e-06	3.6e-13	4.6e-03	2.0e-12

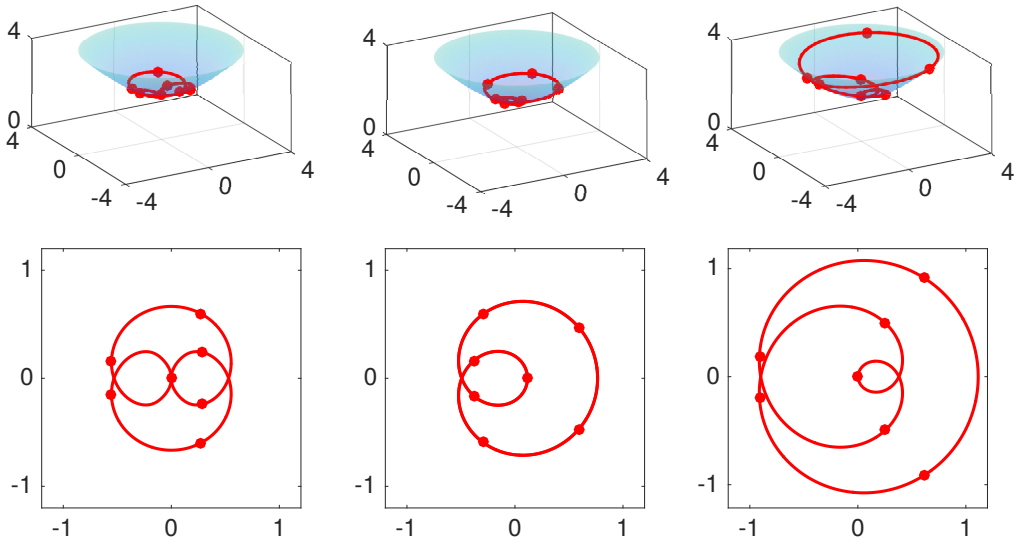

Figure 3.17: *Relative hyperbolic choreographies with* $R = 2$ *(top) and angular velocities* 2.8 *(left),* -2.9 *(center) and* 2.31 *(right), and their projections on the disk (bottom).*

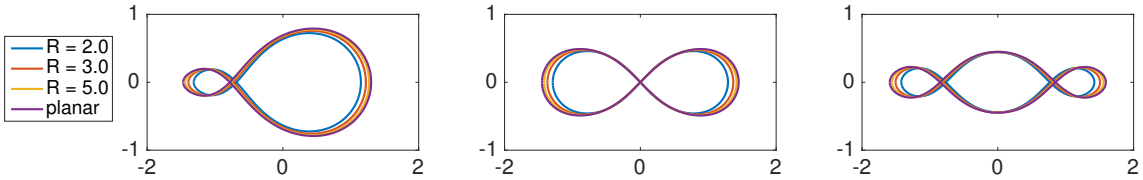

Figure 3.18: *Hyperbolic choreographies of Figure* 3.16 (*multiplied by a factor* 2) *for different values of R, together with their planar analogues of Figure* 3.4. *As R increases, the hyperbolic choreographies converge to the planar ones.*

Table 3.11: *Convergence of hyperbolic to planar choreographies in Figure* 3.18.

	$R = 2$	3	5	10	100	1000
Left	1.6e-01	7.7e-02	3.0e-02	7.9e-03	8.0e-05	8.0e-07
Middle	1.6e-01	8.0e-02	3.1e-02	8.0e-03	8.1e-05	8.1e-07
Right	1.7e-01	8.5e-02	3.3e-02	8.6e-03	8.7e-05	8.7e-07

curved versions of the choreographies of Figure 3.4. As shown in Table 3.10, they can be computed to high accuracy with a few hundred Fourier coefficients.

Relative choreographies can also be computed, see Figure 3.17. Again, a few hundred coefficients are enough to get about 10-digit accuracy.

3.7.3 Limit of infinitely large radius

In the limit $R \to \infty$, the Poincaré disk converges to the complex plane. The distances (3.45) and (3.46) converge to twice the absolute value and the action (3.50) converges to four times the action in the plane. As a consequence, twice the hyperbolic choreographies converge to the planar ones as $R \to \infty$, as shown in Figures 3.18 and 3.19. Tables 3.11 and 3.12 report the ∞-norm of the difference between analogous hyperbolic and planar choreographies as R increases. As on the sphere, the convergence appears to be at rate proportional to the absolute value of the curvature $1/R^2$.

3.8 Discussion

Choreographies are very special solutions of the n-body problem. They are not only periodic but also share a single orbit. We have shown numerical evidence that choreographies can be computed to high accuracy using trigonometric interpolation and

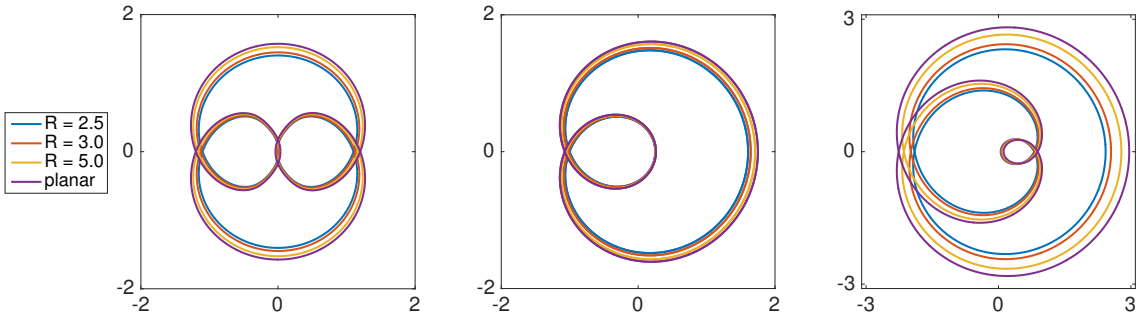

Figure 3.19: *Relative hyperbolic choreographies of Figure* 3.17 (*multiplied by a factor* 2) *for different values of R, together with their planar analogues of Figure* 3.5. *As R increases, the hyperbolic choreographies converge to the planar ones.*

Table 3.12: *Convergence of hyperbolic to planar choreographies in Figure* 3.19.

	$R = 2.5$	3	5	10	100	1000
Left	1.7e-01	1.3e-01	4.9e-02	1.3e-02	1.3e-04	1.3e-06
Middle	1.4e-01	1.1e-01	4.1e-02	1.1e-02	1.1e-04	1.1e-06
Right	5.3e-01	4.1e-01	1.8e-01	4.9e-02	5.0e-04	5.0e-06

optimization algorithms, that many choreographies exist on spaces of constant curvature for various n, and that curved choreographies converge to their planar analogues at a rate proportional to the curvature.

Stability properties of choreographies have not been discussed. In the plane, the only noncircular stable choreography is the figure-eight choreography of Figure 3.1. We have found numerical evidence that the spherical and hyperbolic figure-eight choreographies of Figure 3.7 and 3.14 are stable too. We have solved the curved three-body problem on the sphere and on the Lorentz hyperboloid, with initial conditions corresponding to the figure-eight choreographies. We ran them for a thousand full orbits, i.e., from $t = 0$ to $t = 2000\pi$, and the solutions did not lose their structure. All the other spherical and hyperbolic choreographies presented in this chapter lost their structure after only a few full orbits. A systematic approach to study the stability of choreographies would be to compute the eigenvalues of the derivatives of the associated Poincaré maps.

Chapter 4

Solving semilinear stiff PDEs in 1D, 2D and 3D periodic domains*

4.1 Introduction

Dozens of exponential integration formulas have been proposed for the high-accuracy solution of stiff PDEs such as the Allen–Cahn, Korteweg–de Vries and Ginzburg–Landau equations. We report in this chapter the results of extensive comparisons in MATLAB and Chebfun of such formulas in 1D, 2D and 3D periodic domains, focusing on fourth and higher order methods, and periodic semilinear stiff PDEs with constant coefficients.

More specifically, we are interested in computing smooth solutions of stiff PDEs of the form

$$u_t = \mathcal{S}(u) = \mathcal{L}u + \mathcal{N}(u), \quad u(0, X) = u_0(X), \quad \text{periodic boundary conditions,} \tag{4.1}$$

where $u(t, X)$ is a function of time t and space X, $\mathcal{L}$ is a linear differential operator with constant coefficients on a spatial domain in one, two or three periodic space dimensions and $\mathcal{N}$ is a nonlinear differential (or non-differential) operator of lower order with constant coefficients and on the same domain.[1] In applications, PDEs of this kind typically arise when two or more different physical processes are combined, and many PDEs of interest in science and engineering take this form. For example,

*This chapter is adapted from a paper with Niall Bootland [112]. I selected the integrators and the PDEs, ran the experiments, and wrote the paper. Earlier comparisons of exponential integrators in 1D/2D can be found in Bootland's thesis [17], though the experiments described here are more extensive and challenging. In particular, we consider parameters that make the PDEs particularly stiff, initial conditions that lead to more complicated solutions, and 3D as well as 1D/2D problems.

[1] X denotes a space variable in 1D, 2D or 3D. Throughout this chapter, we will use the variables x in 1D, (x, y) in 2D and (x, y, z) in 3D. The domain will be an interval in 1D, a square in 2D and a cube in 3D.

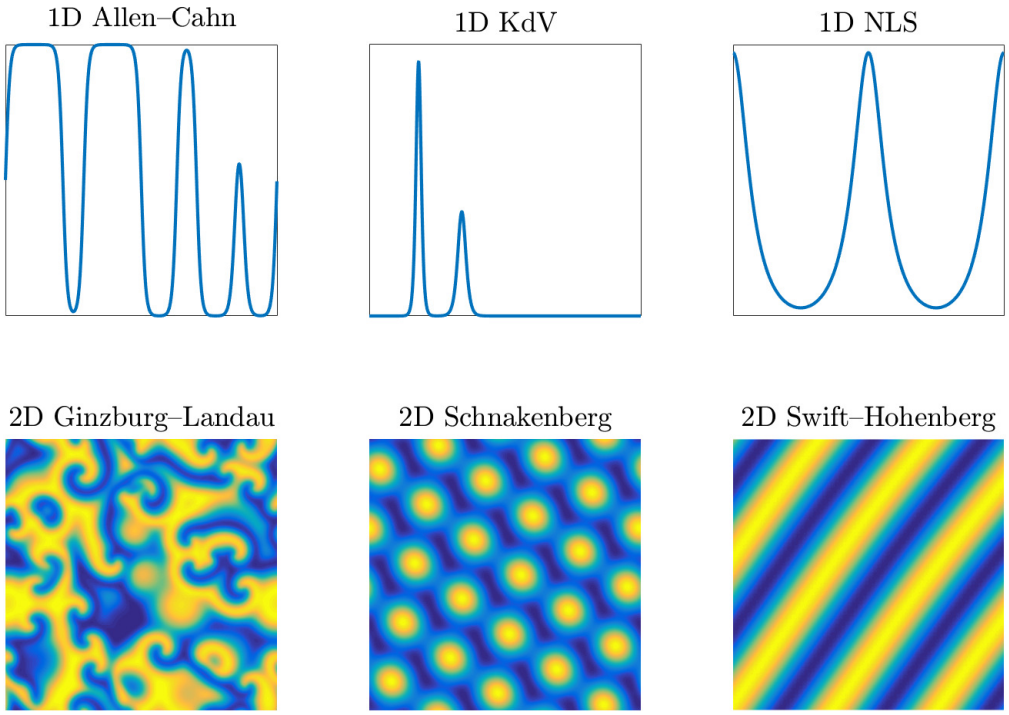

Figure 4.1: *First row (left to right): metastable solution of the Allen–Cahn equation* (4.25)*, two-soliton solution of the KdV equation* (4.31) *and breather solution of the NLS equation* (4.40)*. Second row (left to right): real part of a frozen state solution of the Ginzburg–Landau equation* (4.48)*, spot solution of the Schnakenberg equations* (4.51)*, and convection rolls of the Swift–Hohenberg equation* (4.54)*.*

the Korteweg–de Vries (KdV) equation $u_t = -u_{xxx} - uu_x$, the starting point of the study of nonlinear waves and solitons, couples third-order linear dispersion with first-order convection, and the Allen–Cahn equation $u_t = \epsilon u_{xx} + u - u^3$ couples second-order linear diffusion with a nondifferentiated cubic reaction term. Often a system of equations rather than a single scalar equation is involved, for example in the Gray–Scott and Schnakenberg equations, which involve two components coupled together. (The importance of coupling of nonequal diffusion constants in science was made famous by Alan Turing in the most highly-cited of all his papers [169].) Fourth-order terms also arise, for example in the Cahn–Hilliard equation, whose solutions describe structures of alloys, and in the Kuramoto–Sivashinsky equation, related to combustion problems among others, whose solutions are chaotic. Other examples of stiff PDEs include the Ginzburg–Landau, nonlinear Schrödinger (NLS) and Swift–Hohenberg equations. Figure 4.1 shows six examples of solutions of such PDEs.

Solving all these PDEs by generic numerical methods can be highly challenging.

One of the main issues is *stiffness*, characterised by the need for an explicit method to use small time-steps, much smaller than the condition required by accuracy. When too many steps are required, this can result in an infeasibly long computation. There are other important issues such as the numerical conservation of various properties (e.g., for the KdV and NLS equations)—we shall not discuss this here; see, e.g., [30].

This chapter describes and compares specialized methods that take advantage of two special features of (4.1). The first one is the periodic boundary conditions. This allows us to discretize the spatial component of (4.1) with a Fourier spectral method on N points, as described in Chapter 2; equation (4.1) becomes a system of N ODEs,

$$\hat{u}' = \mathbf{S}(\hat{u}) = \mathbf{L}\hat{u} + \mathbf{N}(\hat{u}), \quad \hat{u}(0) = \hat{u}_0, \tag{4.2}$$

where $\hat{u}(t)$ is the vector of N Fourier coefficients of the trigonometric interpolant of $u(t, X)$ at time t, as in (2.63).[2] Note that $\mathbf{L}$ (a $N \times N$ matrix) and $\mathbf{N}$ are the discretized versions of $\mathcal{L}$ and $\mathcal{N}$. (Stiffness is related to $\mathbf{L}$ having large eigenvalues since stability of spectral methods for time-dependent PDEs requires that the eigenvalues of $\mathbf{L}$, scaled by the time-step, lie in the stability region of the time-stepping formula [164, Chap. 10].)

The second special feature of (4.1) is that it is semilinear, i.e., the higher-order terms of the equation are linear. Exponential integrators are a class of numerical methods for systems of ODEs that are aimed at taking advantage of this. The linear part $\mathbf{L}$, responsible for the stiffness, is integrated exactly using the matrix exponential, while a numerical scheme is applied to $\mathbf{N}$.

According to the 2005 review of Minchev and Wright [109], the first exponential integrators were constructed by Certaine in 1960 [34] and Pope in 1963 [136]. Subsequently, however, Hochbruck and Ostermann noted, in a comprehensive theoretical review of these schemes [77], that Hersch [72] had previously considered exponential integrators in 1958 in an effort to find schemes that are exact for linear problems with constant coefficients. The first use of the term *exponential integrator* was by Hochbruck, Lubich and Selhofer [75] in a seminal paper of 1998. The extensive use of these formulas for solving stiff PDEs seems to have been initiated by the papers by Cox and Matthews [42] and Kassam and Trefethen [84]. A striking unpublished paper by Kassam [83] showed how effective such methods can be also for PDEs in 2D and 3D. A software package for such computations called EXPINT was produced by Berland, Skaflestad and Wright [13].

[2] In this chapter we will assume that the grid size N is large enough so the Fourier coefficients $\hat{u}(t)$ of $u(t, X)$ and $\tilde{u}(t)$ of its trigonometric interpolant are indistinguishable.

One of the simplest exponential integrators, first derived by Pope in 1963 [136] and commonly known as the Exponential Time Differencing (ETD) Euler method, is given by[3]

$$\hat{u}^{n+1} = e^{h\mathbf{L}}\hat{u}^n + h\varphi_1(h\mathbf{L})\mathbf{N}(\hat{u}^n), \tag{4.3}$$

where $h = t_{n+1} - t_n$ is the time-step and

$$\varphi_1(z) = \frac{e^z - 1}{z}. \tag{4.4}$$

As Minchev and Wright [109] point out, this method has been rediscovered from many different viewpoints and has been known by several other names. It can be derived by considering the linearized version of (4.2) on $[t_n, t_{n+1}]$,

$$\hat{u}' = \mathbf{S}(\hat{u}^n) + \mathbf{S}_{\hat{u}}(\hat{u}^n)(\hat{u} - \hat{u}^n), \quad \hat{u}(t_n) = \hat{u}^n, \tag{4.5}$$

with exact solution at $t_{n+1} = t_n + h$,

$$\hat{u}^{n+1} = \hat{u}^n + h\varphi_1(h\mathbf{S}_{\hat{u}}(\hat{u}^n))\mathbf{S}(\hat{u}^n). \tag{4.6}$$

Approximating $\mathbf{S}_{\hat{u}}(\hat{u}^n)$ by $\mathbf{L}$ in (4.6) leads to (4.3). Note that (4.6) defines a time-stepping scheme too, known as the exponential Euler method. The problem with (4.6) is that the exact Jacobian $\mathbf{S}_{\hat{u}}$, and its value under the exponential-like function φ_1, need to be computed at each time-step. This would involve a high computational cost, so typically one either does not compute φ_1 but rather an approximation, such as a Padé approximation, or else one uses an approximation to the Jacobian as opposed to the exact Jacobian. The former approach contains the Rosenbrock methods [68, 69, 79, 101, 141, 172] and the Exponential Propagation Iterative methods of Runge-Kutta type (EPIRK) [137, 138, 159, 160, 162]. The latter approach is what we shall consider in this chapter.

As we just described, exponential integrators are characterised by the use of exponential and related functions of the matrix $\mathbf{L}$. Standard methods for computing the matrix exponential in the context of exponential integrators include the scaling and squaring method [4], the Carathéodory–Fejér method [143] and Krylov subspace methods [159]. There is recent work that shows that exponential integrators together with Krylov methods are competitive, for instance see Tokman and Loffeld [99, 161]. In our case we consider periodic problems with constant coefficients so the matrices are diagonal and the matrix exponential is trivial.

[3]Throughout this chapter and the next one, when introducing an exponential integrator such as (4.3), $\hat{u}^n$ will mean $\hat{u}(t)$ at $t = t_n$.

We compare in this chapter 30 exponential integrators of fourth and higher order on 11 model problems in 1D, 2D and 3D. Comparisons with other types of time-stepping schemes are out of the scope of this chapter; see, e.g., [66, 84, 88, 89, 99]. Let us emphasize that we are interested in determining if one of the high order integrators outperforms the others on a large class of problems. For a particular problem, it might be possible to design a very specific scheme, of possibly lower order than four, which performs extremely well. For example, Cano and Gonzáles–Pachón have recently shown that the low-order Lawson methods, combined with orthogonal projections onto some invariants, can be very competitive for the nonlinear Schrödinger equation [29, 30]. Let us also emphasize that since we only consider periodic problems, we do not expect to see any order reduction in the convergence of the exponential integrators, as already observed in, e.g., [84]. For different types of boundary conditions (e.g., homogeneous Dirichlet conditions), certain schemes (e.g., Lawson methods) do not satisfy the so-called *stiff order conditions* [77]—which guarantee a certain order of convergence independently of the considered problem—and can therefore exhibit a strong order reduction in practice.

This chapter is structured as follows. We present the 30 exponential integrators in Section 4.2 and the 11 model problems in Section 4.3. The numerical results are presented in Section 4.4 and show that it is hard to do much better than one of the simplest of these formulas, the ETDRK4 scheme of Cox and Matthews [42].

4.2 Thirty exponential integrators

4.2.1 Exponential general linear methods

We consider exponential integrators, based on the approximation of the Jacobian, that belong to the large class of exponential general linear methods, introduced by Minchev and Wright in 2005 [109]. This class contains the ETD Runge–Kutta (one-step), ETD Adams–Bashforth (multistep), Lawson and exponential predictor-corrector methods. For given starting values $\hat{u}^0, \hat{u}^1, \ldots, \hat{u}^{q-1}$ at times $t = 0, h, \ldots, (q-1)h$, the numerical approximation $\hat{u}^{n+1}$ at time $t_{n+1} = (n+1)h$, $n+1 \geq q$, is given by the formula

$$\hat{u}^{n+1} = e^{h\mathbf{L}}\hat{u}^n + h\sum_{i=1}^{s} B_i(h\mathbf{L})\mathbf{N}(\hat{v}^i) + h\sum_{i=1}^{q-1} V_i(h\mathbf{L})\mathbf{N}(\hat{u}^{n-i}), \tag{4.7}$$

with q *steps* $\hat{u}^{n-i}$ and s *stages* $\hat{v}^i$, with $\hat{v}^1 = \hat{u}^n$ and

$$\hat{v}^i = e^{C_i h\mathbf{L}}\hat{u}^n + h\sum_{j=1}^{i-1} A_{i,j}(h\mathbf{L})\mathbf{N}(\hat{v}^j) + h\sum_{j=1}^{q-1} U_{i,j}(h\mathbf{L})\mathbf{N}(\hat{u}^{n-j}), \quad 2 \leq i \leq s. \tag{4.8}$$

Table 4.1: *Butcher tableau of an exponential integrator with q steps and s stages.*

C_2	$A_{2,1}$				$U_{2,1}$	$\dots$	$U_{2,q-1}$
$\vdots$	$\vdots$	$\ddots$			$\vdots$		$\vdots$
C_s	$A_{s,1}$	$\dots$	$A_{s,s-1}$		$U_{s,1}$	$\dots$	$U_{s,q-1}$
	B_1	$\dots$	B_{s-1}	B_s	V_1	$\dots$	V_{q-1}

Each scheme is characterised by its coefficients A, B, C, U, and V, which can be conveniently listed in a Butcher tableau, as in Table 4.1. Note that these coefficients (except C) depend on $\mathbf{L}$—for instance, (4.3) uses one stage and one step, and its only non-zero coefficient is $B_1 = \varphi_1(h\mathbf{L})$. Note that, in practice, the nonlinear evaluations $\mathbf{N}(\hat{v}^i)$ and $\mathbf{N}(\hat{u}^{n-i})$ are carried out in value space, e.g., $\mathbf{N}(\hat{v}^i)$ means $\mathbf{F}\big(\mathbf{N}\big(\mathbf{F}^{-1}\hat{v}^i\big)\big)$, with discrete Fourier transform $\mathbf{F}$. Methods of the form (4.7)–(4.8) not only include purely one-step methods ($q = 1$, $s \geq 1$) and purely multistep methods ($q \geq 1$, $s = 1$), but also combinations of both.

The coefficients satisfy the following summation properties,[4]

$$\begin{aligned} B_1 &= \varphi_1(h\mathbf{L}) - \sum_{i=2}^{s} B_i(h\mathbf{L}) - \sum_{i=1}^{q-1} V_i(h\mathbf{L}), \\ A_{i,1} &= \psi_{1,i}(h\mathbf{L}) - \sum_{j=2}^{i-1} A_{i,j}(h\mathbf{L}) - \sum_{j=1}^{q-1} U_{i,j}(h\mathbf{L}), \quad 2 \leq i \leq s, \end{aligned} \tag{4.9}$$

where the φ- and ψ-functions are exponential and related functions that we shall define in the next subsection. As a consequence, it is notationally convenient to incorporate this condition by filling the corresponding entries of the Butcher tableau with a dot on the understanding that these method coefficients are given by (4.9). Note that exponential integrators of the form (4.7)–(4.8) do not include the EMAM4 scheme of Calvo and Palencia [28]. It has been shown in [17] that it often suffers from stability problems.

Let us finish this section with a few words about the computational cost per time-step. Since the matrices in (4.7)–(4.8) are diagonal, the matrix-vector products cost only $\mathcal{O}(N)$ operations. The dominant cost per time-step is then the cost of an FFT, i.e., $\mathcal{O}(N \log N)$ operations. For exponential integrators of the form (4.7)–(4.8), the

[4]There are two exceptions: the coefficients of the Lawson4 and ABLawson4 schemes do not satisfy the summation properties (4.9).

total cost of the step is therefore $\mathcal{O}(2sN \log N)$. As a consequence, purely multistep methods have a low computational cost per time-step.

4.2.2 Evaluating the φ-functions

The coefficients A, B, U, and V involve the φ and ψ-functions applied to $\mathbf{L}$. Because $\mathbf{L}$ is diagonal, $\varphi(\mathbf{L})$ and $\psi(\mathbf{L})$ reduce to φ and ψ applied to the diagonal elements λ of $\mathbf{L}$, so all we have to be able to is to compute $\varphi(\lambda)$ and $\psi(\lambda)$ for $\lambda \in \mathbb{C}$. The φ-functions are defined by the recurrence relation,

$$\varphi_{l+1}(z) = \frac{\varphi_l(z) - 1/l!}{z}, \quad l \geq 1, \tag{4.10}$$

with $\varphi_0(z) = e^z$. After φ_0, the first few φ-functions are (4.4) and

$$\varphi_2(z) = \frac{e^z - z - 1}{z^2}, \quad \varphi_3(z) = \frac{e^z - \frac{z^2}{2} - z - 1}{z^3}, \tag{4.11}$$

while the ψ-functions are defined via the φ-functions and the coefficients C,

$$\psi_{l,m}(z) = C_m^l \varphi_l(C_m z), \quad l \geq 0, \quad 1 \leq m \leq s. \tag{4.12}$$

Equations (4.10) and (4.12) can be implemented recursively, but the accurate evaluation of φ and ψ is not straightforward because it can suffer from cancellation error. Following the idea of Kassam and Trefethen in [84], to compute the functions at some $\lambda \in \mathbb{C}$, we use Cauchy's integral formula

$$\varphi(\lambda) = \frac{1}{2\pi i} \oint_\Gamma \frac{\varphi(z)}{z - \lambda} dz, \tag{4.13}$$

which can be approximated with exponential accuracy by the trapezoidal rule [166],

$$\varphi(\lambda) \approx \frac{1}{M} \sum_{k=1}^{M} \varphi\big(\lambda + e^{2\pi i(k-0.5)/M}\big), \tag{4.14}$$

taking Γ to be the circle of radius 1 centred at λ, oriented counter-clockwise, discretized with M equally spaced points. Note that the φ-functions satisfy $\varphi(\bar{z}) = \bar{\varphi}(z)$ for all $z \in \mathbb{C}$. As a consequence, when λ is on the real axis, we can take Γ to be the upper half of the circle of radius 1 centred at λ and take the real part of the result, i.e.,

$$\varphi(\lambda) = \frac{1}{\pi} \mathcal{R}\left(\int_0^\pi \varphi\big(\lambda + e^{i\theta}\big) d\theta \right), \tag{4.15}$$

which can be discretized by

$$\varphi(\lambda) \approx \frac{1}{M}\mathcal{R}\left(\sum_{k=1}^{M}\varphi\big(\lambda + e^{\pi i(k-0.5)/M}\big)\right). \tag{4.16}$$

If this symmetry is not explicitly used in the computation of the φ-functions when λ is real, rounding errors appear that lead to numerical instability. Note that the evaluation of the φ-functions using (4.14) or (4.16) requires $\mathcal{O}(M)$ operations per λ.

Let us emphasize that we can use circles or radius 1 around each eigenvalue λ of $\mathbf{L}$ because $\mathbf{L}$ is diagonal. When the matrix is not diagonal, one has to use a single contour that encloses all the eigenvalues, and the best possible contour depends on the problem. For example, for diffusive problems, all the eigenvalues are on the real negative axis and the best contour is a Hankel contour [143, 167].

Using contour integrals is not the only possible remedy for cancellation error. When Pope introduced (4.3), he suggested the use of Taylor series for small λ and the direct formula for large λ. The problem with this approach is that, for some intermediate values, neither method gives full precision, as noted by Cox and Matthews [42] and Kassam and Trefethen [84]. Another approach is to use Padé approximations, combined with a scaling and squaring technique [14]. This method is also effective, but the contour integral method is particularly appealing because of its greater generality for dealing with arbitrary functions.

4.2.3 Introducing the 30 exponential integrators

Table 4.2 lists the exponential integrators considered in this chapter. Their Butcher tableaux can be found in Appendix B.

ETD Adams–Bashforth The first category of exponential integrators is the *ETD Adams–Bashforth* schemes of order four to six. These are ETD (purely) multistep methods, which reduce to Adams–Bashforth schemes when $\mathbf{L} = 0$, and go back to Nørsett in 1969 [118]. Since $s = 1$, (4.7)–(4.8) takes the simpler form

$$\hat{u}^{n+1} = e^{h\mathbf{L}}\hat{u}^n + hB_1(h\mathbf{L})\mathbf{N}(\hat{u}^n) + h\sum_{i=1}^{q-1} V_i(h\mathbf{L})\mathbf{N}(\hat{u}^{n-i}), \tag{4.17}$$

i.e., the only non-zero coefficients are B_1 and those in V. Note that since these schemes are purely multistep, (4.17) only requires two FFTs per time-step. We label these methods as ABNørsett$q$, where $4 \le q \le 6$ is the order and also the number of steps. For more details on the derivation of ABNørsett methods see Minchev

Table 4.2: *A reference table for the exponential integrators considered in this chapter. Note that some methods do not appear explicitly in the references listed but can be derived using order conditions or recurrence formulas found there; these cases are marked by asterisks. The Butcher tableaux can be found in Appendix B.*

Method	**Type**	**Order**	**Stages** s	**Steps** q	**Ref.**
ABNørsett4	ETD Adams–Bashforth	4	1	4	[118]*
ABNørsett5	ETD Adams–Bashforth	5	1	5	[118]*
ABNørsett6	ETD Adams–Bashforth	6	1	6	[118]*
ETDRK4	ETD Runge–Kutta	4	4	1	[42]
Friedli	ETD Runge–Kutta	4	4	1	[65]
Krogstad	ETD Runge–Kutta	4	4	1	[93]
Minchev	ETD Runge–Kutta	4	4	1	[108]
Strehmel–Weiner	ETD Runge–Kutta	4	4	1	[153]
Hochbruck–Ostermann	ETD Runge–Kutta	4	5	1	[76]
EXPRK5S8	ETD Runge–Kutta	5	8	1	[100]
ABLawson4	Lawson	4	1	4	[97]
Lawson4	Lawson	4	4	1	[97]
GenLawson41	Gen. Lawson	4	4	1	[93]
GenLawson42	Gen. Lawson	4	4	2	[93]
GenLawson43	Gen. Lawson	4	4	3	[93]
GenLawson44	Gen. Lawson	5	4	4	[93]
GenLawson45	Gen. Lawson	6	4	5	[93]
ModGenLawson41	Mod. Gen. Lawson	4	4	1	[126]*
ModGenLawson42	Mod. Gen. Lawson	4	4	2	[126]*
ModGenLawson43	Mod. Gen. Lawson	4	4	3	[126]*
ModGenLawson44	Mod. Gen. Lawson	5	4	4	[126]*
ModGenLawson45	Mod. Gen. Lawson	6	4	5	[126]*
PEC423	Exp. Predictor-Corrector	4	2	3	[126]*
PECEC433	Exp. Predictor-Corrector	4	3	3	[126]*
PEC524	Exp. Predictor-Corrector	5	2	4	[126]*
PECEC534	Exp. Predictor-Corrector	5	3	4	[126]*
PEC625	Exp. Predictor-Corrector	6	2	5	[126]*
PECEC635	Exp. Predictor-Corrector	6	3	5	[126]*
PEC726	Exp. Predictor-Corrector	7	2	6	[126]*
PECEC736	Exp. Predictor-Corrector	7	3	6	[126]*

Table 4.3: *Butcher tableau for ETDRK4. Note the dots in the first column, which indicate that these coefficients are computed using* (4.9)*. These coefficients are* (*from top to bottom*)*:* $A_{2,1} = \psi_{1,2}$*,* $A_{3,1} = 0$*,* $A_{4,1} = \varphi_1 - 2\psi_{1,2}$ *and* $B_1 = \varphi_1 - 3\varphi_2 + 4\varphi_3$*.*

$\frac{1}{2}$	$A_{2,1} = \cdot$			
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \psi_{1,2}$		
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = 2\psi_{1,2}$	
	$B_1 = \cdot$	$B_2 = 2\varphi_2 - 4\varphi_3$	$B_3 = 2\varphi_2 - 4\varphi_3$	$B_4 = -\varphi_2 + 4\varphi_3$

and Wright [109], who also show a connection between these schemes and the IMEX schemes of Ascher, Ruuth and Wetton [8]. One can also derive methods based on Adams–Moulton methods, known as AMNørsett$q$ methods. These are implicit but can be used within predictor-corrector pairs, as we will see when introducing exponential predictor-corrector schemes (the last category in the table). A comprehensive look at both the Adams–Bashforth and Adams–Moulton exponential integrators can be found in the paper by Hochbruck and Ostermann [78].

ETD Runge–Kutta The second category is the *ETD Runge–Kutta* schemes of order four to five. These are (purely) one-step methods and go back to Friedli in 1978 [65] and Strehmel–Weiner in 1982 [153]. More recently, inspired by Cox and Matthews' ETDRK4 scheme [42], Minchev [108] in 2004 and Krogstad [93] and Hochbruck and Ostermann [76] in 2005 derived ETD Runge–Kutta schemes of order four. Luan and Ostermann proposed a scheme of order five (EXPRK5S8) in 2014 [100]. Overviews of ETD Runge–Kutta methods and some of their history can be found in the reviews of Hochbruck and Ostermann [77] and Minchev and Wright [109], where connections are described between ETD Runge–Kutta, generalised Runge–Kutta and semi-implicit methods. Since $q = 1$, (4.7)–(4.8) reduces to

$$\begin{aligned} \hat{u}^{n+1} &= e^{h\mathbf{L}}\hat{u}^n + h\sum_{i=1}^{s} B_i(h\mathbf{L})\mathbf{N}(\hat{v}^i), \\ \hat{v}^1 &= \hat{u}^n, \quad \hat{v}^i = e^{C_i h\mathbf{L}}\hat{u}^n + h\sum_{j=1}^{i-1} A_{i,j}(h\mathbf{L})\mathbf{N}(\hat{v}^j), \quad 2 \le i \le s. \end{aligned} \tag{4.18}$$

The only non-zero coefficients are those in A, B and C. The coefficients for the

ETDRK4 scheme can be found in Table 4.3 and correspond to the following formula:

$$\begin{aligned}
\hat{v}^1 &= \hat{u}^n, \\
\hat{v}^2 &= e^{\mathbf{L}h/2}\hat{u}^n + (h/2)\varphi_1(\mathbf{L}h/2)\mathbf{N}(\hat{v}^1), \\
\hat{v}^3 &= e^{\mathbf{L}h/2}\hat{u}^n + (h/2)\varphi_1(\mathbf{L}h/2)\mathbf{N}(\hat{v}^2), \\
\hat{v}^4 &= e^{\mathbf{L}h/2}\hat{v}^2 + (h/2)\varphi_1(\mathbf{L}h/2)[2\mathbf{N}(\hat{v}^3) - \mathbf{N}(\hat{v}^1)], \\
\hat{u}^{n+1} &= e^{\mathbf{L}h}\hat{u}^n + hB_1\mathbf{N}(\hat{v}^1) + hB_2[\mathbf{N}(\hat{v}^2) + \mathbf{N}(\hat{v}^3)] + hB_4\mathbf{N}(\hat{v}^4),
\end{aligned} \tag{4.19}$$

where $B_1 = B_1(h\mathbf{L}), \ldots, B_4 = B_4(h\mathbf{L})$.

Lawson The third category is the *Lawson* methods. First developed by Lawson in 1967 [97], and often known as integrating factor (IF) methods, the motivation behind Lawson methods is to use a change of variable in (4.2) to get rid of the stiff linear part, and then apply a numerical solver to the transformed equation. The Lawson transformation consists of the change of variables $\hat{v}(t) = e^{-\mathbf{L}t}\hat{u}(t)$. If we differentiate this and substitute into (4.2), the transformed equation is

$$\hat{v}' = e^{-\mathbf{L}t}\mathbf{N}(e^{\mathbf{L}t}\hat{v}), \quad \hat{v}(0) = \hat{u}_0. \tag{4.20}$$

The linear term is gone, and the transformed equation (4.20), while no longer stiff, now has rapidly varying coefficients. Once we have decided on a scheme to solve (4.20), we can transform back to $\hat{u}$. Lawson, in his 1967 paper, used the classical fourth order Runge–Kutta scheme on the transformed equation (4.20); we call this method Lawson4. Using the classical fourth order Adams–Bashforth scheme gives the ABLawson4 method. Ehle and Lawson observed in [58] that Runge–Kutta based Lawson methods only work well when the problem is moderately stiff. Another problem with Lawson methods, as indicated by Krogstad [93], is that they do not preserve fixed points of the differential equation.

Generalised Lawson Krogstad worked around these problems to derive *generalised Lawson methods,* also called generalised integrating factor (GIF) methods, the fourth category in the table. These are based on the transform

$$\hat{v}(t) = e^{-\mathbf{L}t}\hat{u}(t) - e^{-\mathbf{L}t}\sum_{l=1}^{q} t^l \varphi_l(t\mathbf{L})p_{l-1}, \tag{4.21}$$

where the p_l are the coefficients, in a (scaled) monomial basis, of the polynomial $P(t)$ of degree $q-1$ that interpolates the values $\{N(\hat{u}^{n-l})\}_{l=1}^{q}$ at the points $\{t_{n-l}\}_{l=1}^{q}$;

see [77, 109] for details. Differentiating this and substituting into (4.2) leads to the transformed equation

$$\hat{v}' = e^{-\mathbf{L}t}\Big(\mathbf{N}\Big(e^{\mathbf{L}t}\hat{v} + \sum_{l=1}^{q} t^l \varphi_l(t\mathbf{L})p_{l-1}\Big) - P(t)\Big), \quad \hat{v}(0) = \hat{u}_0. \tag{4.22}$$

Note that (4.20) is the special case of (4.22) with $P(t) = 0$. The idea of Krogstad is to apply, for various values of q, the classical fourth order Runge–Kutta scheme on (4.22), and then transform back to $\hat{u}$. It leads to methods with four stages and q steps, called the GenLawson4q methods.

Modified generalised Lawson As we increase q in the generalised Lawson methods we incorporate more of the nonlinearity and the methods have improved accuracy. However, this in part comes at the cost of stability, especially for dispersive problems, as was demonstrated by Krogstad [93]. A modification, based on satisfying order conditions, given by Ostermann, Thalhammer and Wright [126], significantly improves stability. The modification is given by the requirement that

$$\sum_{i=1}^{4} B_i(h\mathbf{L})\frac{c_i^j}{j!} + \sum_{i=1}^{q-1} V_i(h\mathbf{L})\frac{(-1)^j}{j!} = \varphi_{j+1}(h\mathbf{L}), \quad 0 \le j \le q-1, \tag{4.23}$$

where, as before, $q-1$ is the degree of the polynomial approximation. These are the *modified generalised Lawson* methods, labelled as ModGenLawson4q.

Exponential predictor-corrector Just as with the standard Adams–Bashforth and Adams–Moulton multistep methods, the exponential versions can be used in predictor-corrector pairs. These are the *exponential predictor-corrector* methods, the last category in the table. For instance, using ABNørsett3 for a predictor step and AMNørsett4 for the corrector step yields the fourth order method called PEC423 in the MATLAB package EXPINT [13]. (PEC stands for predict-evaluate-correct, four is the order, two is the number of stages and three is the number of steps.) One can evaluate and correct again, that is, use the corrector twice. The name PECEC433 is given in EXPINT for the fourth order method that uses ABNørsett3 for a predictor step and AMNørsett4 for two corrector steps.

4.3 Eleven model problems

In this section we describe the PDEs used in the comparisons of Section 4.4, including the initial conditions, the domains and the time intervals. There are five PDEs in 1D and three PDEs considered in both 2D and 3D; see Table 4.4.

Table 4.4: *The model problems we consider in this chapter. The linear operator of a diffusive PDE has real eigenvalues while it has purely imaginary eigenvalues for dispersive PDEs. Note that we take $A = 0$ for the Ginzburg–Landau equation* (4.48)*. For $A \neq 0$, the linear part would have complex eigenvalues.*

PDE	Dimension	Stiff linear Part
Allen–Cahn	1D	second-order diffusive
Cahn–Hilliard	1D	fourth-order diffusive
Korteweg–de Vries	1D	third-order dispersive
Kuramoto–Sivashinsky	1D	fourth-order diffusive
nonlinear Schrödinger	1D	second-order dispersive
Ginzburg–Landau	2D & 3D	second-order diffusive
Schnakenberg	2D & 3D	second-order diffusive
Swift–Hohenberg	2D & 3D	fourth-order diffusive

4.3.1 Model problems in 1D

In 1D we look for solutions of the form

$$u(t,x) \approx \sum_{k=-N/2}^{N/2}{}' \, \hat{u}_k(t) e^{ikx}, \quad x \in [0, 2\pi], \tag{4.24}$$

with appropriate rescaling for different domains, and we use the differentiation matrices of Section 2.3.2.

Allen–Cahn The *Allen–Cahn equation*, derived by Allen and Cahn in the 1970s, is a reaction-diffusion equation which describes the process of phase separation in iron alloys (see, e.g., [5]). It is given in one dimension as

$$u_t = \epsilon u_{xx} + u - u^3, \tag{4.25}$$

with linear diffusion ϵu_{xx} and a cubic reaction term $u - u^3$. The function u is the order parameter, a correlation function related to the positions of the different components of the alloy. The Allen–Cahn equation exhibits stable equilibria at $u = \pm 1$ while $u = 0$ is an unstable equilibrium. Solutions often display metastability where wells $u \approx -1$ compete with peaks $u \approx 1$, and structures remain almost unchanged for long periods of time before changing suddenly. This can be quantified: features with width

L persist for time scales on the order of $e^{L/\epsilon}$. In Fourier space with a grid of size N, (4.25) becomes

$$\hat{u}' = \epsilon \mathbf{D}_N^{(2)} \hat{u} + \hat{u} - \mathbf{F}\big(\big(\mathbf{F}^{-1}\hat{u}\big)^3\big). \tag{4.26}$$

We take $\epsilon = 5 \times 10^{-2}$,

$$u(0,x) = \frac{1}{3}\tanh(2\sin(x)) - e^{-23.5.(x-\pi/2)^2} + e^{-27(x-4.2)^2} + e^{-38(x-5.4)^2}, \tag{4.27}$$

with $x \in [0, 2\pi]$ and solve up to $t = 60$. This initial condition quickly converges to a set of wells $u \approx -1$ and peaks $u \approx 1$ (at around $t = 4$) and eventually to a two-plateau solution (at around $t = 500$). Figure 4.1 shows the solution at time $t = 113$, when the peak on the far right is switching to $u \approx -1$.

Cahn–Hilliard The *Cahn–Hilliard equation*,

$$u_t = \alpha(-u_{xx} - \gamma u_{xxxx} + (u^3)_{xx}), \tag{4.28}$$

is a fourth order reaction-diffusion problem which Cahn and Hilliard proposed in 1958 as a model for the process of phase separation in binary alloys [26]. It couples second-order destabilizing diffusion $-u_{xx}$ with fourth-order stabilizing diffusion $-u_{xxxx}$ and a differentiated cubic reaction term $(u^3)_{xx}$. The function u is defined as $u = 1 - 2c_A$ where $0 \leq c_A \leq 1$ and $c_B = 1 - c_A$ denote the concentrations of the two components A and B of the alloy, that is, $u = -1$ means pure A while $u = 1$ means pure B. The Cahn–Hilliard equation also exhibits metastable solutions. When quenched below a critical temperature, alloys described by (4.28) become unstable in the sense that small metastable pockets of relatively pure A and B may soon appear, corresponding to wells $u = -1$ and peaks $u = 1$. These pockets may coarsen into larger pockets at progressively larger times. In Fourier space, (4.28) becomes

$$\hat{u}' = \alpha(-\mathbf{D}_N^{(2)} - \gamma \mathbf{D}_N^{(4)})\hat{u} + \alpha \mathbf{D}_N^{(2)} \mathbf{F}\big(\big(\mathbf{F}^{-1}\hat{u}\big)^3\big). \tag{4.29}$$

We take $\alpha = 10^{-2}$, $\gamma = 10^{-3}$,

$$u(0,x) = \frac{1}{5}\sin(4\pi x)^5 - \frac{4}{5}\sin(\pi x), \quad x \in [-1, 1], \tag{4.30}$$

and solve up to $t = 12$. This initial condition evolves to a four-plateau solution (two wells $u \approx -1$, two peaks $u \approx 1$) at around $t = 12$ before switching to a two-plateau solution (one well, one peak) at around $t = 70$.

Korteweg–de Vries The *Korteweg–de Vries equation*,

$$u_t = -u_{xxx} - uu_x, \tag{4.31}$$

was derived by Korteweg and de Vries in 1895 to model the propagation of waves in shallow water [90]. It couples dispersion $-u_{xxx}$ with nonlinear convection $-uu_x$. Among the solutions of (4.31) are solitary waves or *solitons*. These are waves that maintain their shapes as they travel and are given by

$$u(t,x) = 3c\,\mathrm{sech}^2\Big(\frac{\sqrt{c}}{2}(x - x_0 - ct)\Big), \quad c > 0. \tag{4.32}$$

Waves of the form (4.32) have amplitude $3c$ and travel at constant speed c. This is contrast to solutions of linear wave equations $u_t = -cu_x$, which all travel at velocity c, regardless of their amplitudes. In Fourier space, (4.31) becomes

$$\hat{u}' = -\mathbf{D}_N^{(3)}\hat{u} - \frac{\mathbf{D}_N}{2}\mathbf{F}\big(\big(\mathbf{F}^{-1}\hat{u}\big)^2\big). \tag{4.33}$$

We take

$$u(0,x) = 3A^2\,\mathrm{sech}^2\Big(\frac{A}{2}(x+2)\Big) + 3B^2\,\mathrm{sech}^2\Big(\frac{B}{2}(x+1)\Big), \quad x \in [-\pi,\pi], \tag{4.34}$$

with $A = 25$ and $B = 16$, and solve up to $t = 10^{-2}$. This is a superposition of two solitons with speed A^2 and B^2 initially centred at $x = -2$ and $x = -1$, respectively. The stronger wave ($A = 25$) catches up with the weaker one ($B = 16$) at around $t = 10^{-3}$. Both waves remain unchanged after the interaction, the only nonlinear effect being a forward shift

$$\frac{1}{A^2}\log\Big(\frac{A^2+B^2}{A^2-B^2}\Big)^2 \tag{4.35}$$

for the stronger wave and a backward shift

$$-\frac{1}{B^2}\log\Big(\frac{A^2+B^2}{A^2-B^2}\Big)^2 \tag{4.36}$$

for the weaker one. The interaction ends at around $t = 3.5 \times 10^{-3}$. Figure 4.1 shows the initial condition.

Kuramoto–Sivashinsky The *Kuramoto–Sivashinsky equation*,

$$u_t = -u_{xx} - u_{xxxx} - uu_x, \tag{4.37}$$

dates to the mid-1970s with the work of Kuramoto [94] and Sivashinsky [150]. It couples destabilizing $-u_{xx}$ and stabilizing $-u_{xxxx}$ diffusions with nonlinear convection $-uu_x$. The nonlinear term shifts energy created at low wavenumbers by the

second-order term to high wavenumbers where the fourth-order term stabilises. The Kuramoto–Sivashinsky equation models various physical phenomena, from unstable drift waves in plasmas to thermal instabilities in laminar flame fronts. In the latter, the function u represents the perturbation of the flame front surface. The solutions of (4.37) can demonstrate a wide range of spatio-temporal dynamics, including chaos. In Fourier space, (4.37) becomes

$$\hat{u}' = (-\mathbf{D}_N^{(2)} - \mathbf{D}_N^{(4)})\hat{u} - \frac{\mathbf{D}_N}{2}\mathbf{F}\big((\mathbf{F}^{-1}\hat{u})^2\big). \tag{4.38}$$

We take

$$u(0,x) = \cos\left(\frac{x}{16}\right)\left(1+\sin\left(\frac{x}{16}\right)\right), \quad x \in [0, 32\pi], \tag{4.39}$$

and solve up to $t = 100$. This simple initial data progressively evolves into a much more complicated superposition of wavenumbers and, even though the solution looks quite complicated, a characteristic pattern emerges from $t \approx 50$.

Nonlinear Schrödinger The (*focusing*) *nonlinear Schrödinger equation*,

$$u_t = iu_{xx} + i|u|^2u, \tag{4.40}$$

models several physical phenomena, including the nonlinear propagation of light in optical fibres. A nonlinear variant of the Schrödinger equation, it couples dispersion iu_{xx} with a nonlinear potential $i|u|^2u$. Note that the wave function u is complex-valued. Among the solutions of (4.40) are *breathers*, given by

$$u(t,x) = A\left(\frac{2B^2\cosh(\theta) + 2iB\sqrt{2-B^2}\sinh(\theta)}{2\cosh(\theta) - \sqrt{2}\sqrt{2-B^2}\cos(ABx)} - 1\right)e^{iA^2t}, \tag{4.41}$$

with $\theta = A^2B\sqrt{2-B^2}t$ and $B \leq \sqrt{2}$. These are nonlinear waves in which energy concentrates in a localized and oscillatory fashion. In Fourier space, (4.40) becomes

$$\hat{u}' = i\mathbf{D}_N^{(2)}\hat{u} + i\mathbf{F}\Big(\big|\mathbf{F}^{-1}\hat{u}\big|^2\mathbf{F}^{-1}\hat{u}\Big). \tag{4.42}$$

We take

$$u(0,x) = \frac{2AB^2}{2 - \sqrt{2}\sqrt{2-B^2}\cos(ABx)} - A, \quad x \in [-\pi, \pi], \tag{4.43}$$

with $A = 2$ and $B = 1$, and solve up to $t = 2$. This is a breather whose amplitude oscillates in time around $A = 2$. Figure 4.1 shows the (real) initial condition.

4.3.2 Model problems in 2D and 3D

In 2D, we look for solutions of the form

$$u(t,x,y) \approx \sum_{k=-N_x/2}^{N_x/2}{}' \sum_{l=-N_y/2}^{N_y/2}{}' \hat{u}_{k,l}(t) e^{i(kx+ly)}, \quad (x,y) \in [0,2\pi]^2, \tag{4.44}$$

with N_x and N_y points in the x- and y-directions, and appropriate rescaling for different domains. The unknowns, at each time t, are the $N_x N_y$ Fourier coefficients $\hat{u}_{k,l}(t)$. In 3D, we look for solutions of the form

$$u(t,x,y,z) \approx \sum_{k=-N_x/2}^{N_x/2}{}' \sum_{l=-N_y/2}^{N_y/2}{}' \sum_{m=-N_z/2}^{N_a/2}{}' \hat{u}_{k,l,m}(t) e^{i(kx+ly+mz)}, \ (x,y,z) \in [0,2\pi]^3, \tag{4.45}$$

with $N_x N_y N_z$ Fourier coefficients $\hat{u}_{k,l,m}(t)$ at each time t. As in 1D, the primes on the summation signs in (4.44)–(4.45) signify that the extreme terms are halved.

To construct differentiation matrices in 2D and 3D, we use Kronecker products and the 1D Fourier differentiation matrices of Section 2.3.2. For example, the Laplacian operator in 2D,

$$\mathcal{L}u = \Delta u = u_{xx} + u_{yy}, \tag{4.46}$$

is discretized by the $N_x N_y \times N_x N_y$ matrix

$$\mathbf{L} = \mathbf{I}_{N_y} \otimes \mathbf{D}_{N_x}^{(2)} + \mathbf{D}_{N_y}^{(2)} \otimes \mathbf{I}_{N_x}, \tag{4.47}$$

where $\mathbf{I}_N$ denotes the identity matrix of size N.

Ginzburg–Landau The (*complex*) *Ginzburg–Landau equation*,

$$u_t = (1+iA)\Delta u + u - (1+iB)u|u|^2, \tag{4.48}$$

was first derived in 2D by Stewartson and Stuart in 1971 to study nonlinear instabilities in plane Poiseuille flow [152], using concepts from Ginzburg–Landau theory for superconductivity. The function u is the amplitude of a nonlinear perturbation wave for values of the Reynolds number close to the critical value, above which perturbations may grow. Equation (4.48) admits solutions known as *frozen states* which correspond to quasi-frozen spiral defects surrounded by shock lines. In this regime, $|u|$ is stationary in time. We take $A = 0$ and $B = 1.5$,

$$u(0,x,y) = e^{-0.1[(x-50)^2+(y-50)^2]}, \quad (x,y) \in [0,100]^2, \tag{4.49}$$

and

$$u(0,x,y,z) = e^{-0.1[(x-50)^2+(y-50)^2+(z-50)^2]}, \quad (x,y,z) \in [0,100]^3, \tag{4.50}$$

and solve up to $t = 10$ in both 2D and 3D. These two initial conditions generate spiral waves. Figure 4.1 shows an example of a 2D frozen state solution at $t = 30$ (real part), obtained with an initial condition of amplitude 0.1 involving random noise on the grid.

Schnakenberg The *Schnakenberg equations*,

$$\begin{cases} u_t = \epsilon_u \Delta u + \gamma(a - u + u^2 v), \\ \\ v_t = \epsilon_v \Delta v + \gamma(b - u^2 v), \end{cases} \tag{4.51}$$

are reaction-diffusion equations derived by Schnakenberg in 1979 to study limit cycle behaviours of two-component chemical reactions [144]. The system (4.51) models the chemical reaction $2U + V \to 3U$; $U \rightleftharpoons A$; $B \to V$; u and v are the concentrations of U and V, and A and B are another two chemical species, assumed to be maintained at constant concentrations a and b. We take $\epsilon_u = 1$, $\epsilon_v = 10$, $\gamma = 3$, $a = 0.1$ and $b = 0.9$. The initial conditions are

$$\begin{aligned} u(0,x,y) &= 1 - e^{-2[(x-G/2.15)^2+(y-G/2.15)^2]}, \\ v(0,x,y) &= \frac{0.9}{(0.1+0.9)^2} + e^{-2[(x-G/2)^2+2(y-G/2)^2]}, \end{aligned} \tag{4.52}$$

with $(x,y) \in [0,G]^2$ and $G = 30$ in 2D, and

$$\begin{aligned} u(0,x,y,z) &= 1 - e^{-2[(x-G/2.15)^2+(y-G/2.15)^2+(z-G/2.15)^2]}, \\ v(0,x,y,z) &= \frac{0.9}{(0.1+0.9)^2} + e^{-2[(x-G/2)^2+2(y-G/2)^2+2(z-G/2)^2]}, \end{aligned} \tag{4.53}$$

with $(x,y,z) \in [0,G]^3$ and $G = 30$ in 3D. We solve up to $t = 20$ in both 2D and 3D. Note that these initial conditions are small perturbations from the constant solution $(u,v) = (a+b, b/(a+b)^2)$. They lead to a set of spots at around $t = 500$ in 2D and $t = 300$ in 3D. Figure 4.1 shows the 2D solution at $t = 500$.

Swift–Hohenberg The *Swift–Hohenberg equation*,

$$u_t = ru - (1+\Delta)^2 u + gu^2 - u^3, \tag{4.54}$$

was first derived in 2D by Swift and Hohenberg in 1977 to study thermal fluctuations on a fluid near the Rayleigh–Bénard convective instability [154]. In 2D, the

function u is the temperature field in a plane horizontal layer of fluid heated from below. Equation (4.54) is another example of a PDE that exhibits pattern formation, including stripes, spots and spirals. We take $r = 0.1$, $g = 1$,

$$u(0, x, y) = \frac{1}{4}\Big(\sin\Big(\frac{\pi x}{10}\Big) + \sin\Big(\frac{\pi y}{10}\Big) + \sin\Big(\frac{\pi x}{2}\Big) \sin\Big(\frac{\pi y}{2}\Big)\Big), \tag{4.55}$$

with $(x, y) \in [0, 20]^2$ and

$$\begin{aligned} u(0, x, y, z) = \frac{1}{4}\Big(&\sin\Big(\frac{\pi x}{10}\Big) + \sin\Big(\frac{\pi y}{10}\Big) + \sin\Big(\frac{\pi z}{10}\Big) + \sin\Big(\frac{\pi x}{2}\Big) \sin\Big(\frac{\pi y}{2}\Big) \\ &+ \sin\Big(\frac{\pi x}{2}\Big) \sin\Big(\frac{\pi z}{2}\Big) + \sin\Big(\frac{\pi y}{2}\Big) \sin\Big(\frac{\pi z}{2}\Big)\Big), \end{aligned} \tag{4.56}$$

with $(x, y, z) \in [0, 20]^3$, and solve up to $t = 20$ in both 2D and 3D. Both of these two initial conditions lead to a set of spots. Figure 4.1 shows the 2D solution at $t = 1000$ obtained with $r = 0.1$, $g = 0$ and an initial condition of amplitude 0.1 involving random noise on the grid. This solution corresponds to the so-called *convection rolls*.

4.4 Numerical comparisons

4.4.1 Methodology

To compare exponential integrators, we follow the methodology of [84], though the experiments described here are far more extensive. We solve a given PDE up to $t = T$ for various time-steps h and a fixed number of grid points. We estimate the "exact" solution $u^{ex}(t = T, X)$ by using a "very small" time-step (half the smallest time-step h) and the PECEC736 scheme (one of the two seventh-order accurate schemes in Table 4.2). We then measure the relative L^2-error E at $t = T$ between the computed solution $u(t = T, X)$ and $u^{ex}(t = T, X)$, i.e.,

$$E = \frac{\|u(t = T, X) - u^{ex}(t = T, X)\|_2}{\|u^{ex}(t = T, X)\|_2}. \tag{4.57}$$

For both u and u^{ex} we use $N = 512$ grid points in 1D, $N_x = N_y = 128$ grid points in 2D and $N_x = N_y = N_z = 128$ grid points in 3D. (With these grid sizes, the error due to the spatial discretization is small compared to the error due to the time discretization; see Figure 4.2.) For the contour integrals, we use $M = 64$ points in 1D and $M = 32$ points in 2D and 3D. We plot (4.57) against relative time-steps h/T and computer times on a pair of graphs.[5] The former gives a measure of the accuracy of

[5]The precomputation of the coefficients of the exponential integrators and the starting phase of multistep methods are not included in the computing time. Timings were done on a 2.8 GHz Intel i7 machine with 16 GB of RAM.

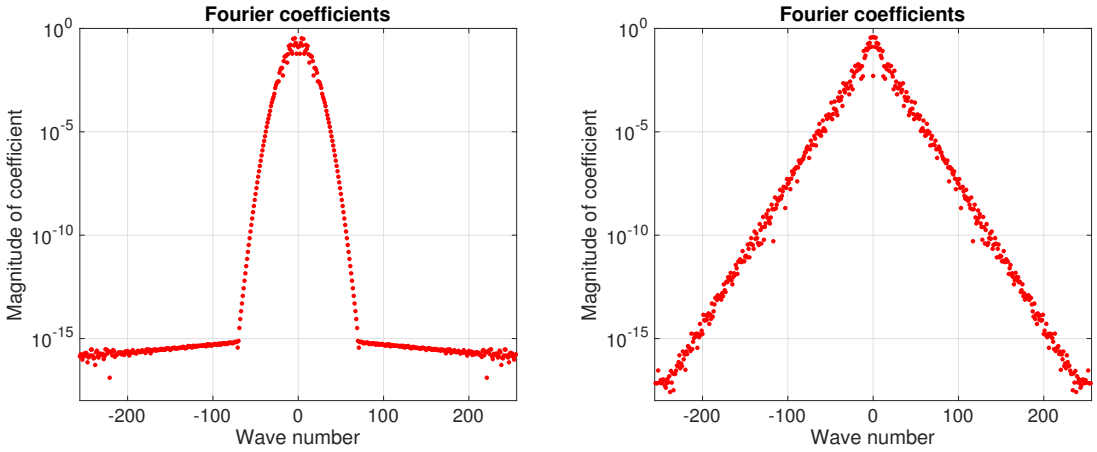

Figure 4.2: *Fourier coefficients of the initial condition* (4.27) *of the Allen–Cahn equation* (*left*) *and of the solution at* $t = 60$ *computed with time-step* $h = 10^{-2}$ (*right*). *With* $N = 512$ *grid points, the Fourier coefficients decay to about* 10^{-16}. *In our computations for the Allen–Cahn equation, the smallest error due to the time discretization is about* $10^{-12} \gg 10^{-16}$.

the exponential integrator for various time-steps or, equivalently, for various number of integration steps. (If the relative time-step is 10^{-3}, it means that the integrator performed 10^3 steps to reach $t = T$.) However, it is possible that each step is more costly, so it is the latter that ultimately matters. We compare different families on different pairs of graphs with curves for ETDRK4 included on all plots as a baseline. We have tested every integrator on every PDE, but we shall only show graphs that correspond to the characteristic behaviour of a family of integrators, or highlight notable features such as instability or particularly good/bad performance. The rest of the graphs can be found in Appendix C.

4.4.2 Starting multistep schemes

To start a multistep scheme with q steps, one needs q values: the initial condition $\hat{u}^0$ and $q-1$ extra values $\hat{u}^1, \ldots, \hat{u}^{q-1}$. It is suggested in [28] to use the following strategy: first, compute an approximation of $\hat{U} = (\hat{u}^1, \ldots, \hat{u}^{q-1})^T$ using a low-order exponential integrator (e.g., ETDRK2, the second-order version of ETDRK4 [42]), and second, use a fixed point iteration to refine this approximation. The fixed point iteration is applied to the following system of nonlinear equations,

$$\hat{u}^j = e^{jh\mathbf{L}}\hat{u}^0 + h\sum_{l=0}^{q-1} \gamma_l(j, h\mathbf{L})\Delta^l \mathbf{N}(\hat{u}^0), \quad 1 \le j \le q-1, \tag{4.58}$$

where Δ^l is the forward difference operator,

$$\Delta^0 \mathbf{N}(\hat{u}^0) = \hat{u}^0, \quad \Delta^l \mathbf{N}(\hat{u}^0) = \sum_{i=0}^{l} (-1)^i \binom{l}{i} \mathbf{N}(\hat{u}^{l-i}), \quad l \geq 1, \tag{4.59}$$

and the γ-functions are defined by the recurrence relation,

$$\begin{aligned}
\gamma_0(k,z) &= \frac{e^{kz}-1}{z}, \\
\gamma_j(k,z) &= \left[\left(\sum_{m=1}^{j} \frac{(-1)^{m-1}}{m} \gamma_{j-m}(k,z) \right) - \binom{k}{j} \right] \Bigg/ z, \quad 1 \leq j \leq k, \\
\gamma_j(k,z) &= \left(\sum_{m=1}^{j} \frac{(-1)^{m-1}}{m} \gamma_{j-m}(k,z) \right) \Bigg/ z, \quad j > k.
\end{aligned} \tag{4.60}$$

Note that, like the φ-functions, the γ-functions can be evaluated by contour integrals and satisfy the symmetry property $\gamma(\bar{z}) = \bar{\gamma}(z)$ for all $z \in \mathbb{C}$.

Let us write (4.58) as $\hat{U} = F(\hat{U})$. The fixed point iteration is then given by

$$\hat{U}_{[n+1]} = F(\hat{U}_{[n]}), \tag{4.61}$$

where $\hat{U}_{[n]} = (\hat{u}^1_{[n]}, \ldots, \hat{u}^{q-1}_{[n]})^T$ denotes the approximation obtained after n iterations ($\hat{U}_{[0]}$ corresponding to the approximation given by ETDRK2). The fixed point iteration (4.61) is carried out until the norm of the difference between two successive iterations is of the order of h^q, which guarantees convergence [28, Th. 1].

4.4.3 Results

We now report on the results of our numerical testing and present plots showing the typical behaviours and notable features we see from each set of methods.

Figure 4.3 shows results for the *ETD Adams–Bashforth* methods for the 1D Kuramoto–Sivashinsky equation. These formulas are often unstable for large time-steps but can be competitive at high accuracies. For the Cahn–Hilliard and KdV equations (Figures C.3 and C.5), we could not get them to work at all with the spatial discretization that we used.

Figure 4.4 shows results for the *ETD Runge–Kutta* methods for the 1D Cahn–Hilliard, 1D KdV and 2D Swift–Hohenberg equations. These formulas have good stability properties. The fourth-order methods have similar performance to ETDRK4. The fifth-order EXPRK5S8 integrator is impressively efficient in 1D, but exhibits

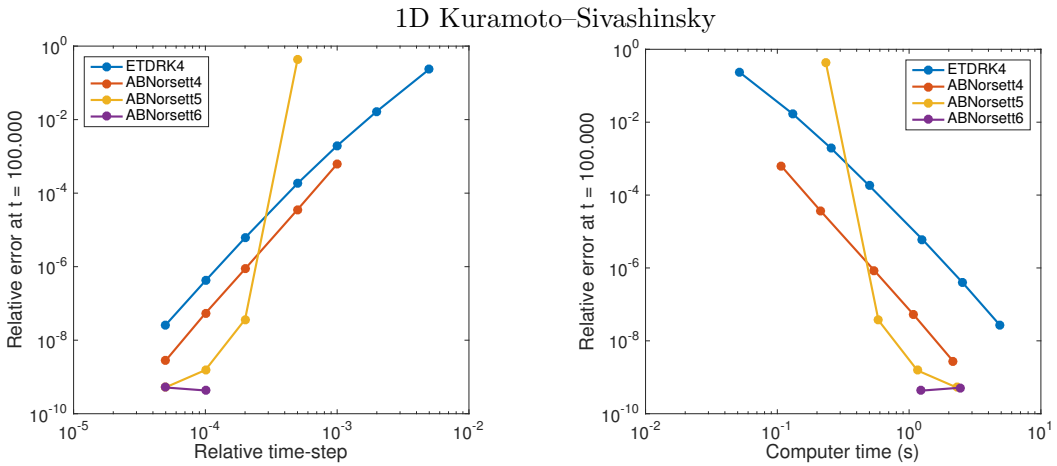

Figure 4.3: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth methods for the* 1*D Kuramoto–Sivashinsky equation. The order* 5 *and* 6 *methods are more efficient than ETDRK*4 *at high accuracies, but often unstable at lower accuracies, as reflected in dots missing from the curves.*

instability for the Cahn–Hilliard and KdV equations. In 2D and 3D, it is more accurate than the fourth-order methods for the six PDEs we have considered, and more efficient for the Ginzburg–Landau equation (Figures C.11 and C.17). However, for the Schnakenberg (Figures C.13 and C.19) and Swift–Hohenberg equations, it is not accurate enough to counterbalance its high cost per time-step.

Figure 4.5 shows results for the *Lawson* methods for the 1D NLS equation. These formulas are not accurate enough to be competitive. For the ABLawson4 formula, this lack of accuracy is partly compensated by its low computational cost per time-step (it is a purely multistep method).

Figure 4.6 shows results for the *generalised Lawson* methods for the 1D Kuramoto–Sivashinsky and 2D Ginzburg–Landau equations. In 1D, these formulas suffer from instabilities for most PDEs. In 2D and 3D, the GenLawson41 formula has virtually identical performance to ETDRK4, the GenLawson42 formula is always less efficient than ETDRK4, while the other variants with three to five steps perform well for the Ginzburg–Landau equation but are less efficient for the Schnakenberg and Swift–Hohenberg equations (Figures C.14, C.16, C.20 and C.22).

Figure 4.7 shows results for the *modified generalised Lawson* methods for the 1D Kuramoto–Sivashinsky, 1D KdV and 3D Swift–Hohenberg equations. In 1D, these formulas are much more stable than the generalised Lawson schemes and are quite efficient, but still suffer from instabilities for the 1D KdV equation. In 2D and 3D, we reach the same conclusions as for the generalised Lawson methods: the ModGen-

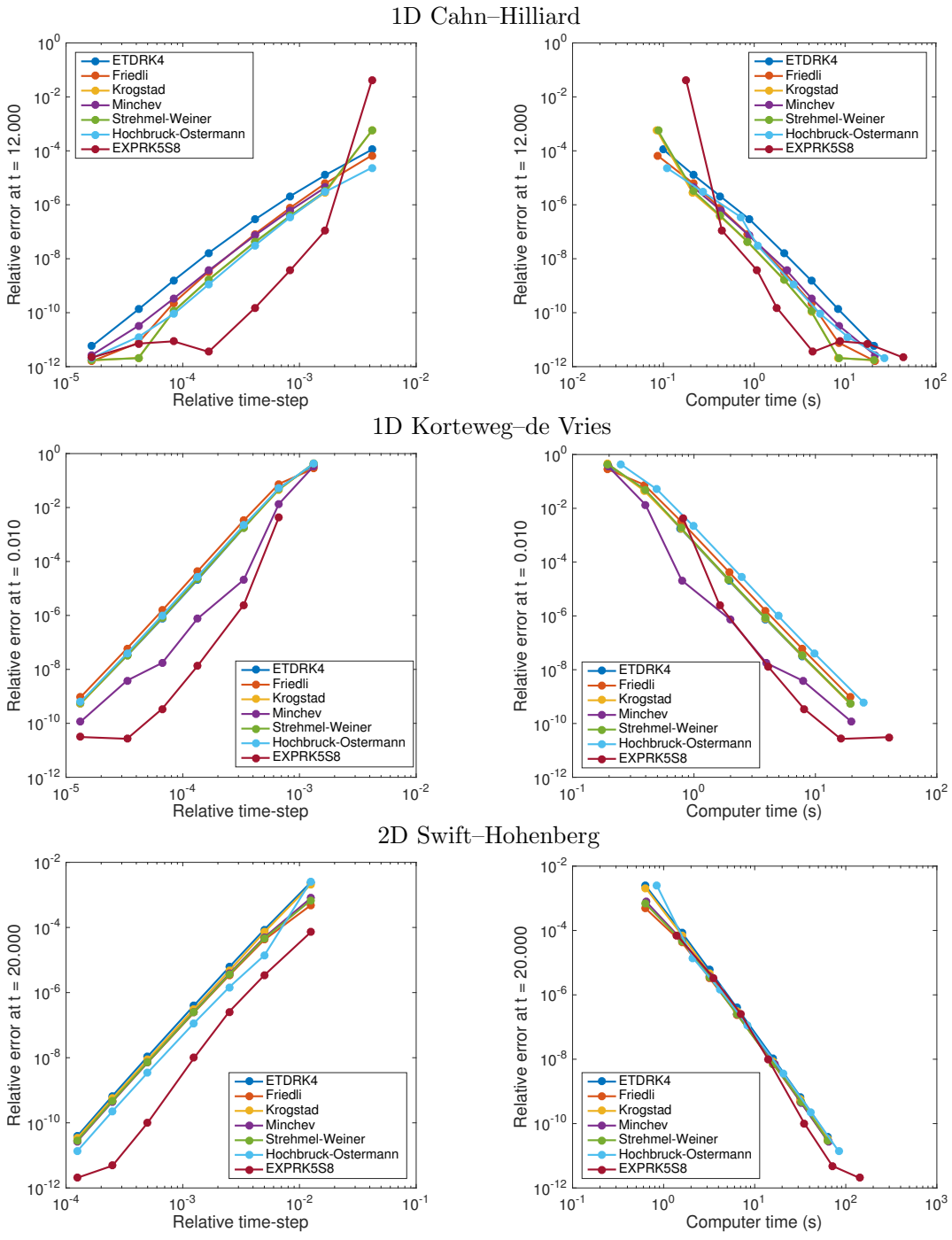

Figure 4.4: *Accuracy versus time-step and computer time for the ETD Runge–Kutta methods for the* 1*D Cahn–Hilliard* (*top*), 1*D KdV* (*centre*) *and* 2*D Swift–Hohenberg* (*bottom*) *equations. The EXPRK*5*S*8 *scheme is impressively efficient in* 1*D but is unstable at low accuracies for the Cahn–Hilliard and KdV equations. In* 2*D and* 3*D, it does not beat the fourth-order methods for the Schnakenberg* (*Figures* C.13 *and* C.19) *and Swift–Hohenberg equations.*

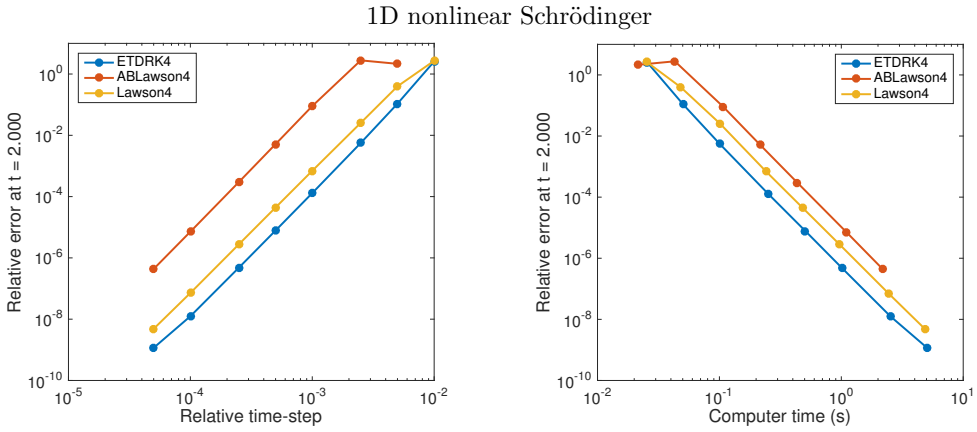

Figure 4.5: *Accuracy versus time-step and computer time for the Lawson methods for the* 1*D NLS equation. These formulas are too inaccurate to be competitive; the constants involved in the convergence bounds are too great.*

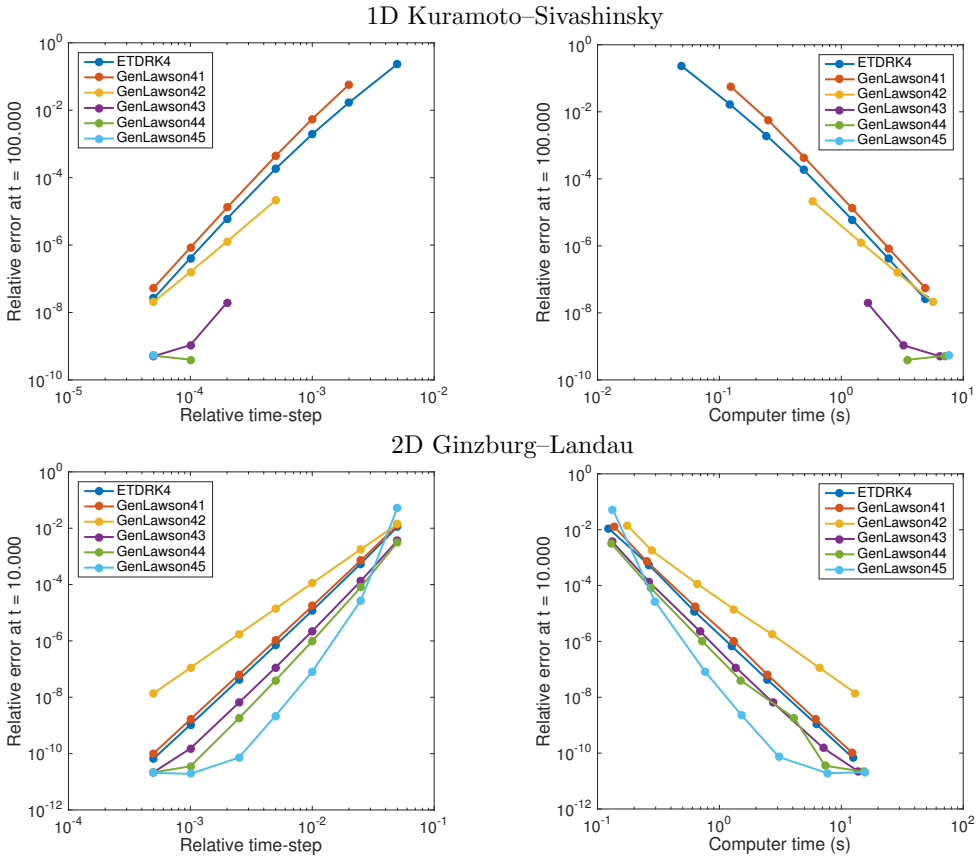

Figure 4.6: *Accuracy versus time-step and computer time for the generalised Lawson methods for the* 1*D Kuramoto–Sivashinsky (top) and* 2*D Ginzburg–Landau (bottom) equations. These methods are highly unstable for most PDEs in* 1*D. In* 2*D and* 3*D, the GenLawson*41 *formula has virtually identical performance to ETDRK*4 *while the other variants with three to five steps perform well for some problems (e.g.,* 2*D Ginzburg–Landau equation above) but are less efficient for others (e.g., Figure* C.14*).*

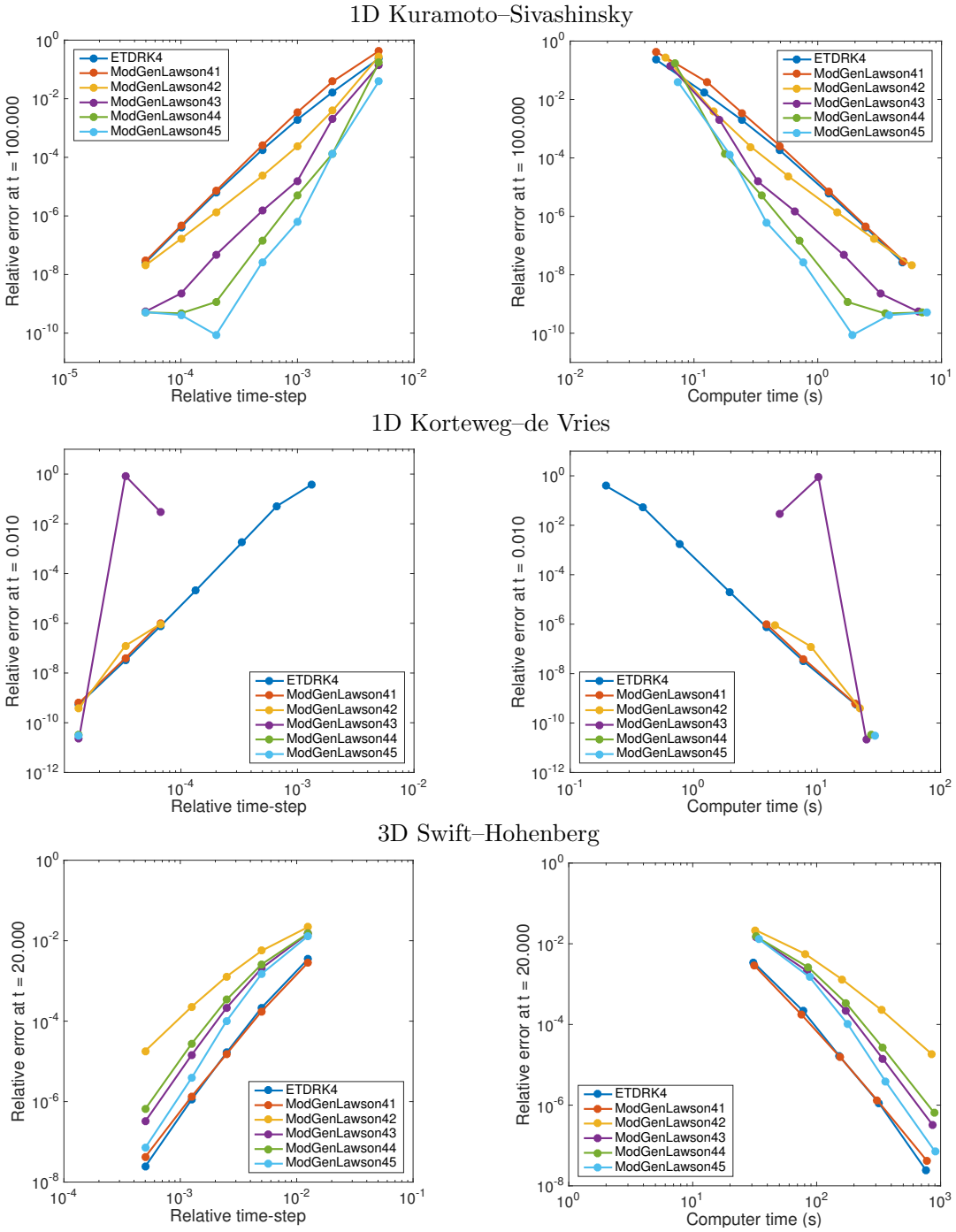

Figure 4.7: *Accuracy versus time-step and computer time for the modified generalised Lawson methods for the 1D Kuramoto–Sivashinsky (top), 1D KdV (centre) and 3D Swift–Hohenberg (bottom) equations. In 1D, these methods are much more stable than the generalised Lawson methods but are still highly unstable for the KdV equation. In 2D and 3D, they are very similar to the generalised Lawson methods, i.e., they perform well for some problems (e.g., Figure C.18) but are less efficient for others (e.g., 3D Swift–Hohenberg equation above).*

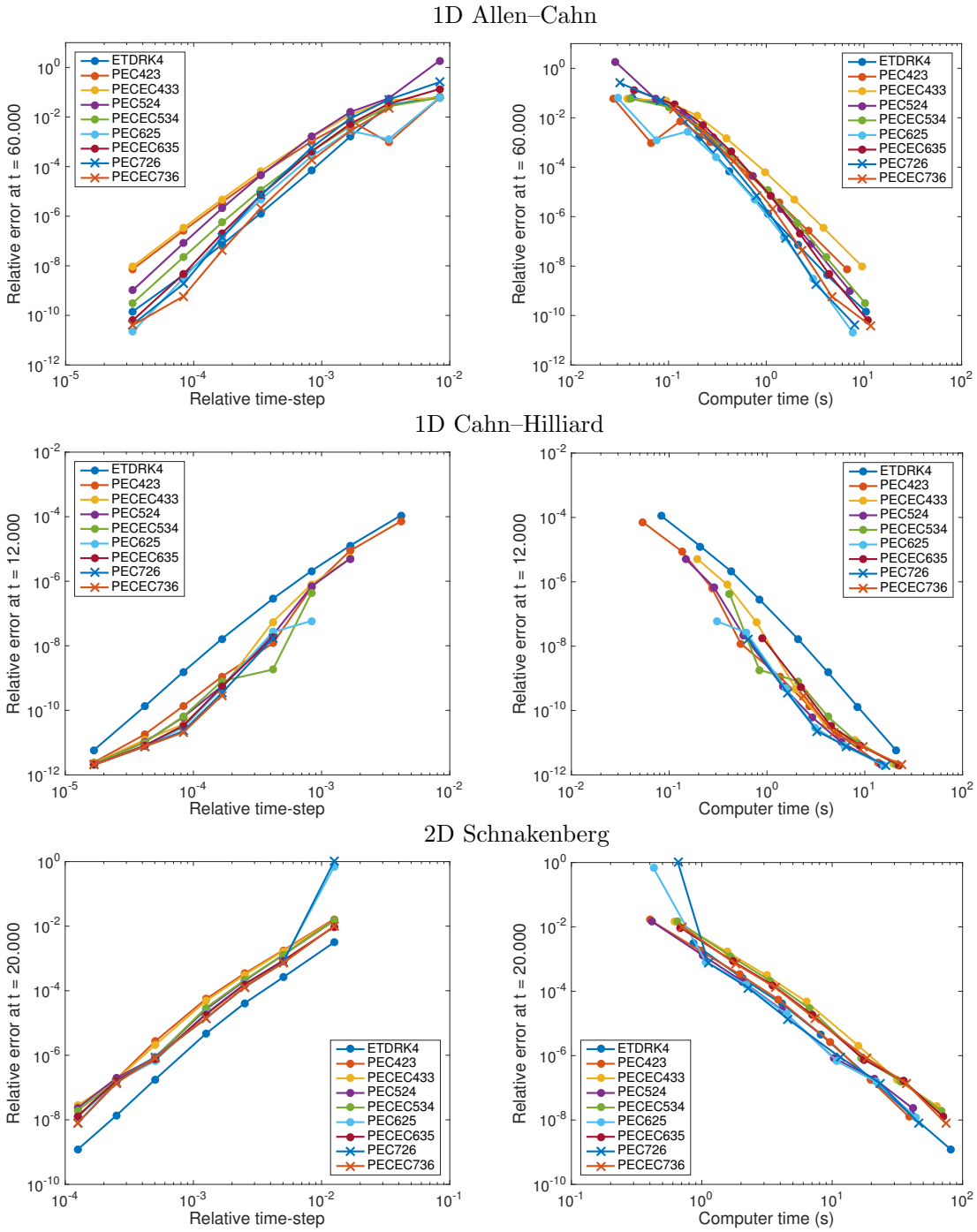

Figure 4.8: *Accuracy versus time-step and computer time for the predictor-corrector methods for the 1D Allen–Cahn (top), 1D Cahn–Hilliard (centre) and 2D Schnakenberg equations (bottom). In 1D, these schemes are efficient, but do not clearly outperform ETDRK4 for the Allen–Cahn equation and are unstable at low accuracies for the Cahn–Hilliard equation. In 2D and 3D, they beat ETDRK4 for the Ginzburg–Landau equation (Figures C.12 and C.18) but have similar performance for the other equations (e.g., 2D Schnakenberg equations above).*

Lawson41 formula has virtually identical performance to ETDRK4, the ModGenLawson42 formula is always the least efficient and the other variants perform well for some problems but are less efficient for others.

Figure 4.8 shows results for the *exponential predictor-corrector* methods for the 1D Allen–Cahn, 1D Cahn–Hilliard and 2D Schnakenberg equations. These formulas are particularly efficient in 1D, especially for the Kuramoto–Sivashinsky and NLS equations (Figures C.8 and C.10), but do not clearly outperform ETDRK4 for the Allen–Cahn equation. Most of them are unstable at low accuracies for the Cahn–Hilliard equation, especially the higher-order schemes, and some of them are also unstable at low accuracies for the KdV equation (Figure C.6). In 2D and 3D, they are more efficient than ETDRK4 for the Ginzburg–Landau equation (Figures C.12 and C.18) but most of them have similar performance to it for the other PDEs we have considered. Note that the higher-order schemes with two steps (PEC625 and PEC726) also beat ETDRK4 for the Swift–Hohenberg equations, but these are particularly unstable for the Cahn–Hilliard and KdV equations.

4.5 Discussion

We have tested 30 exponential integrators on 11 model problems in 1D, 2D and 3D, and have observed considerable differences in stability and efficiency. As expected, the schemes did not exhibit any order reduction (periodic boundary conditions). The main conclusion is that it is difficult to find a method that outperforms ETDRK4 for all the PDEs we have considered.

Our experiments show that the ETD Adams–Bashforth and the generalised Lawson methods are highly unstable while the Lawson methods are not accurate enough. Within the ETD Runge–Kutta methods, it is hard to do much better than ETDRK4. The fourth-order schemes are quite similar in terms of efficiency and stability. The fifth-order EXPRK5S8 integrator is more efficient than ETDRK4 for most PDEs in 1D, but is unstable at low accuracies for the KdV and Cahn–Hilliard equations. In 2D and 3D, it outperforms ETDRK4 only for the Ginzburg–Landau equation. Since it requires the precomputation of more than twice as many coefficients as ETDRK4, it makes it much more complicated to implement and probably less appealing to general users. The high-order modified generalised Lawson and exponential predictor-corrector methods are competitive stiff solvers for some PDEs, but for others do not outperform ETDRK4 or else suffer from instabilities.

Our numerical experiments were performed using MATLAB and have been embedded within Chebfun. More specifically, the `spin`, `spin2` and `spin3` codes implement a Fourier spectral method and exponential integrators to solve PDEs in 1D, 2D and 3D periodic domains. (Note that `spin` stands for **s**tiff **P**DE **in**tegrator.) These have been one of the most major additions to Chebfun in recent years from a user point of view. The simplest way to see `spin` in action is to type simply `spin('ks')` (for the Kuramoto–Sivashinsky equation) or `spin2('gl2')` (for the 2D Ginzburg–Landau equation) to invoke an example computation. It is also possible to define your own PDE using the `spinop` class. To produce the graphs of Section 4.4.3, we have used the `spincomp` code.

Let us mention three exciting articles that are particularly relevant to this chapter and that we have learned about recently. In the first one, the spectral deferred correction method of Dutt, Greengard and Rokhlin [56] is extended to exponential integrators [25]. It is found that these schemes are more efficient than ETDRK4 at high accuracy. The second one develops preconditioned implicit exponential integrators for problems with stiff $\mathcal{N}$ [102]. (The exponential integrators we have presented in this chapter are particularly efficient when the stiffness comes from $\mathcal{L}$, as opposed to $\mathcal{N}$.) Finally, the third paper investigates algorithms for solving PDEs with spectral accuracy in both space and time [103].

As we mentioned in the introduction of this chapter, comparisons of exponential integrators with other time-stepping schemes can be found in, e.g., [66, 84, 88, 89, 99]. For example, Kassam and Trefethen showed in [84] that exponential integrators outperform implicit-explicit algorithms for several stiff PDEs in 1D. We shall reach quite a different conclusion in the next chapter, which concerns the solution of semilinear stiff PDEs on the sphere.

Chapter 5

Solving semilinear stiff PDEs on the sphere*

5.1 Introduction

In this chapter, we are interested in the computation of smooth solutions of stiff PDEs on the unit sphere of the form

$$u_t = \mathcal{L}u + \mathcal{N}(u), \quad u(t = 0, x, y, z) = u_0(x, y, z), \tag{5.1}$$

where u is a function of time t and coordinates (x, y, z) with $x^2 + y^2 + z^2 = 1$. In Chapter 4, we considered 1D, 2D and 3D *flat* periodic domains; we consider here a two-dimensional *curved* periodic domain. The function u can be real or complex and (5.1) can be a single equation, as well as a system of equations. We restrict our attention to $\mathcal{L}u = \alpha\Delta u$ and to a nonlinear nondifferential operator $\mathcal{N}$ with constant coefficients, but the techniques we present can be applied to more general cases. As we mentioned in Chapter 4, a large number of PDEs of interest in science and engineering take this form. Examples on the sphere include the (diffusive) Allen–Cahn equation $u_t = \epsilon\Delta u + u - u^3$ with $\epsilon \ll 1$ [55], the (dispersive) focusing nonlinear Schrödinger equation $u_t = i\Delta u + iu|u|^2$ [157], the Gierer–Meinhardt [15], Ginzburg–Landau [142] and Brusselator [168] equations, and many others.

There are several methods to discretize the spatial part of (5.1) with spectral accuracy, including spherical harmonics [10], radial basis functions (RBFs) [63] and the double Fourier sphere (DFS) method [107, 125]. The DFS method is the only one

*This chapter is adapted from a paper with Yuji Nakatsukasa [114]. I derived the Fourier matrices, compared the time-stepping schemes from both a theoretical and practical point of view, implemented the algorithms and wrote most of the paper. Nakatsukasa proved the realness of the eigenvalues of the Laplacian matrix (Appendix D) and developed the numerical linear algebra of Sections 5.2.3 and 5.3.1 jointly with me.

that leads to a $\mathcal{O}(N \log N)$ complexity per time-step, where N is the total number of grid points in the spatial discretization of (5.1). For spherical harmonics, the cost per time-step is $\mathcal{O}(N^{3/2})$ since there are no effective "fast" spherical transforms,[1] and for global RBFs [63, Chap. 6], the cost per time-step is $\mathcal{O}(N^2)$ since these generate dense differentiation matrices.[2] We focus in this chapter on the DFS method and present a novel formulation operating in coefficient space.

Once the spatial part of (5.1) has been discretized by the DFS method on an $N = n \times m$ uniform longitude-latitude grid, it becomes a system of nm ODEs,

$$\hat{u}' = \mathbf{L}\hat{u} + \mathbf{N}(\hat{u}), \quad \hat{u}(0) = \hat{u}_0, \tag{5.2}$$

where $\hat{u}(t)$ is a vector of nm Fourier coefficients and $\mathbf{L}$ (an $nm \times nm$ matrix) and $\mathbf{N}$ are the discretized versions of $\mathcal{L}$ and $\mathcal{N}$. As in Chapter 4, solving (5.2) with generic explicit time-stepping schemes can be highly challenging because of stiffness. Exponential integrators—studied in detail in Chapter 4—and implicit-explicit (IMEX) schemes are two classes of numerical methods that are aimed at treating stiffness. For exponential integrators, the linear part $\mathbf{L}$ is integrated exactly using the matrix exponential while a numerical scheme is applied to the nonlinear part $\mathbf{N}$. For IMEX schemes, an explicit formula is used to advance $\mathbf{N}$ while an implicit scheme is used to advance $\mathbf{L}$. We show in this chapter that the DFS method combined with IMEX schemes leads to $\mathcal{O}(nm \log nm)$ per time-step algorithms for both diffusive and dispersive PDEs, that exponential integrators achieve this complexity for diffusive PDEs only, and that IMEX schemes outperform exponential integrators in both cases. For numerical comparisons, we consider two versions of the fourth-order ETDRK4 exponential integrator (4.19) and two fourth-order IMEX schemes, the IMEX-BDF4 [81] and LIRK4 [27] schemes.

By contrast, Kassam and Trefethen demonstrated in [84] that exponential integrators (ETDRK4) outperform IMEX schemes (IMEX-BDF4); this can be explained by two factors. First, they focused on problems with diagonal matrices $\mathbf{L}$. (IMEX-BDF4 performed better than ETDRK4 for the only non-diagonal problem they considered.) For diagonal problems, exponential integrators are particularly efficient since the computation of the matrix exponential is trivial and the matrix exponential is diagonal

[1] Fast $\mathcal{O}(N \log N)$ spherical transforms have received significant attention but require so far a $\mathcal{O}(N^2)$ precomputational cost [140, 170, 171]. Note that in a recent manuscript [151] Slevinsky proposed a new fast spherical transform based on conversions between spherical harmonics and bivariate Fourier series with a lower $O(N^{3/2})$ precomputational cost. We shall see in Section 5.3.2 that for implicit-explicit schemes with the DFS method, the precomputation is $\mathcal{O}(N)$.

[2] RBF-FD [63, Chap. 7] generate sparse matrices but only achieve algebraic orders of accuracy. Moreover, the solution time for these sparse matrices is not necessarily $\mathcal{O}(N)$.

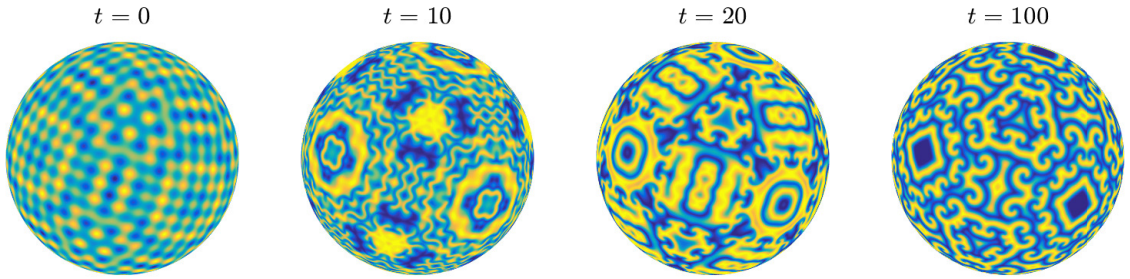

Figure 5.1: *Initial condition and real part of the solution at times* $t = 10, 20, 100$ *of the Ginzburg–Landau equation computed by the* `spinsphere` *code. This solution is oriented in a direction at a* $\pi/8$ *angle from the north-south axis, so the symmetry maintained in the computations is a reflection of global accuracy.*

too (hence, its action on vectors can trivially be computed in linear time). Second, IMEX-BDF4 is unstable for dispersive PDEs since it is based on the fourth-order backward differentiation formula, which is unstable for dispersive PDEs—this is why Kassam and Trefethen could not make it work for the KdV equation. In this chapter, the matrices we work with are not diagonal but have a sparsity structure that makes IMEX schemes particularly efficient, and we consider not only IMEX-BDF4 but also LIRK4, which is stable for dispersive PDEs.

There are libraries for solving time-dependent PDEs on the sphere, including SPHEREPACK [2] and FEniCS [139]. However, none of these are aimed at solving stiff PDEs and easily allow for computing in an integrated environment. The algorithms we shall describe in this chapter are aimed at solving stiff PDEs, and have been implemented in MATLAB and made available as part of Chebfun. The recent extension of Chebfun to the sphere [163] provides a very convenient framework for working with functions on the sphere. For example, the function

$$f(\lambda, \theta) = \cos(1 + \cos\lambda \sin(2\theta)) \tag{5.3}$$

can be approximated to machine precision by the following command:

```
f = spherefun(@(lam,th) cos(1 + cos(lam).*sin(2*th)));
```

Using `spherefun` objects as initial conditions, the `spinsphere` code allows one to solve stiff PDEs with a few lines of code (using IMEX-BDF4 for diffusive and LIRK4 for dispersive PDEs). Together with the `spin`/`spin2`/`spin3` codes for solving PDEs in 1D/2D/3D periodic domains, it is one of the most exciting additions to Chebfun in

recent years. For example, the following MATLAB code solves the complex Ginzburg–Landau equation[3] $u_t = 10^{-4}\Delta u + u - (1+1.5i)u|u|^2$ with 1024 grid points in longitude and latitude and a time-step $h = 10^{-1}$:

```
n = 1024;                                          % number of grid pts
h = 1e-1;                                          % time-step
tspan = [0 100];                                   % time interval
S = spinopsphere(tspan);                           % initialize operator
S.lin = @(u) 1e-4*lap(u);                          % linear part
S.nonlin = @(u) u-(1+1.5i)*u.*abs(u).^2;           % nonlinear part
u0 = @(x,y,z) 1/3*(cos(40*x)+cos(40*y)+cos(40*z)); % initial cond.
th = pi/8; c = cos(th); s = sin(th);               % pi/8 rotation
S.init = spherefun(@(x,y,z)u0(c*x-s*z,y,s*x+c*z)); % rotated initial cond.
u = spinsphere(S, n, h);                           % solve
```

The initial condition and the solution at times $t = 10, 20, 100$ are shown in Figure 5.1.

This chapter is structured as follows. In the next section, we review the DFS method and present a new Fourier spectral method in coefficient space, which, using multiplication matrices that differ from the usual ones, avoids the coordinate singularity, takes advantage of sparse direct solvers, and maintains smoothness at the poles, The time-stepping schemes are presented in Section 5.3 while Section 5.4 is dedicated to numerical comparisons on simple PDEs.

5.2 A Fourier spectral method in coefficient space

We present in this section a Fourier spectral method for the discretization of (5.1), based on the DFS method and the Fourier multiplication matrices of Section 2.3.2. The accuracy of the method is tested by solving the Poisson and heat equations.

5.2.1 The double Fourier sphere method

The DFS method uses the longitude-latitude coordinate transform,

$$x = \cos\lambda \sin\theta,\ y = \sin\lambda \sin\theta,\ z = \cos\theta, \tag{5.4}$$

with $(\lambda, \theta) \in [-\pi, \pi] \times [0, \pi]$. The azimuth angle λ corresponds to the longitude while the polar (or zenith) angle θ corresponds to the latitude.[4] A function $u(x, y, z)$ on

[3] As we mentioned in Chapter 4, the Ginzburg–Landau equation in 2D goes back to the 1970s with the work of Stewartson and Stuart [152]. Rubinstein and Sternberg studied it on the sphere [142], with applications in the study of liquid crystals [132] and nonequilibrium patterns [135].

[4] To be precise, θ is the colatitude, defined as "$\pi/2$ minus latitude" with latitude in $[-\pi/2, \pi/2]$. For simplicity, we will refer to it as latitude. Note that $\theta = 0$ corresponds to the north pole and $\theta = \pi$ to the south pole.

the sphere is written as $u(\lambda, \theta)$ using (5.4), i.e.,

$$u(\lambda, \theta) = u(\cos\lambda \sin\theta, \sin\lambda \sin\theta, \cos\theta), \quad (\lambda, \theta) \in [-\pi, \pi] \times [0, \pi], \tag{5.5}$$

and (5.1) with $\mathcal{L} = \alpha\Delta$ becomes

$$u_t = \alpha\Delta u + \mathcal{N}(u), \quad u(t = 0, \lambda, \theta) = u_0(\lambda, \theta), \quad (\lambda, \theta) \in [-\pi, \pi] \times [0, \pi]. \tag{5.6}$$

Note that the function $u(\lambda, \theta)$ in (5.5) is 2π-periodic in λ but not periodic in θ. The key idea of the DFS method—developed by Merilees [107] and further studied by Orszag [125] in the 1970s, and recently revisited by Townsend et al. with the use of low-rank approximations [163]—is to associate a function $\tilde{u}(\lambda, \theta)$ with $u(\lambda, \theta)$, 2π-periodic in both λ and θ, defined on $[-\pi, \pi] \times [-\pi, \pi]$, and constant along the lines $\theta = 0$ and $\theta = \pm\pi$ corresponding to the poles. Mathematically, the function $\tilde{u}(\lambda, \theta)$ is defined as

$$\tilde{u}(\lambda, \theta) = \begin{cases} u(\lambda, \theta), & (\lambda, \theta) \in [-\pi, \pi] \times [0, \pi], \\ u(\lambda + \pi, -\theta), & (\lambda, \theta) \in [-\pi, 0] \times [-\pi, 0], \\ u(\lambda - \pi, -\theta), & (\lambda, \theta) \in [0, \pi] \times [-\pi, 0]. \end{cases} \tag{5.7}$$

The function u is "doubled-up" in the θ-direction and flipped; see, e.g., [163, Fig. 1]. Since the function $\tilde{u}$ is 2π-periodic in both λ and θ, it can be approximated by its 2D trigonometric interpolant,[5]

$$\tilde{u}(\lambda, \theta) \approx \sum_{j=-m/2}^{m/2}{}' \sum_{k=-n/2}^{n/2}{}' \hat{u}_{jk} e^{i(j\theta + k\lambda)}. \tag{5.8}$$

The numbers n and m are assumed to be even (this will be the case throughout this chapter) and the primes on the summation signs mean that the boundary terms $j = \pm m/2$ or $k = \pm n/2$ are halved. The Fourier coefficients are defined by

$$\hat{u}_{jk} = \frac{1}{nm} \sum_{p=1}^{m} \sum_{q=1}^{n} \tilde{u}(\lambda_q, \theta_p) e^{-i(j\theta_p + k\lambda_q)}, \ -\frac{m}{2} \le j \le \frac{m}{2} - 1, \ -\frac{n}{2} \le k \le \frac{n}{2} - 1, \tag{5.9}$$

with $\hat{u}_{j,n/2} = \hat{u}_{j,-n/2}$ for all j and $\hat{u}_{m/2,k} = \hat{u}_{-m/2,k}$ for all k, and correspond to a 2D uniform grid with n points in longitude and m points in latitude,

$$\lambda_q = -\pi + (q-1)\frac{2\pi}{n}, \ 1 \le q \le n, \quad \theta_p = -\pi + (p-1)\frac{2\pi}{m}, \ 1 \le p \le m. \tag{5.10}$$

The nm Fourier coefficients $\hat{u}_{jk}$ can be computed by sampling $\tilde{u}$ on the grid and using the 2D FFT, costing $\mathcal{O}(nm \log nm)$ operations. In practice we take $m = n$ since it

[5] As in Chapter 4, we assume that the grid sizes n and m are large enough so the Fourier coefficients $\hat{u}(t)$ of $\tilde{u}(\lambda, \theta)$ and those of its trigonometric interpolant are indistinguishable.

leads to the same resolution in each direction around the equator where the spacing is the coarsest.

As mentioned in [163], every smooth function $u(\lambda, \theta)$ on the sphere is associated with a smooth bi-periodic function $\tilde{u}(\lambda, \theta)$ on $[-\pi, \pi]^2$ via (5.7), but the converse is not true since smooth bi-periodic functions might not be constant along the lines $\theta = 0$ and $\theta = \pm\pi$ corresponding to the poles. To be smooth on the sphere, functions of the form (5.7) have to satisfy the *pole conditions*, which asserts that $\tilde{u}(\lambda, \theta)$ is single-valued at the poles despite the fact that latitude circles degenerate into a single point there. For approximations of the form (5.8)–(5.9), this is given by

$$\sum_{j=-m/2}^{m/2}{}' \hat{u}_{jk} = \sum_{j=-m/2}^{m/2}{}' (-1)^j \hat{u}_{jk} = 0, \quad |k| \geq 1. \tag{5.11}$$

As we will see in the numerical experiments presented of Sections 5.2.4 and 5.2.5, when solving a PDE involving the Laplacian operator, if the right-hand side (for Poisson's equation) or the initial condition (for the heat equation) is a smooth function on the sphere, then the solutions obtained with our DFS method are also smooth functions on the sphere. Therefore, we do not have to impose the conditions (5.11). Similarly, we do not impose the "doubled-up" symmetry in (5.7), as it was preserved throughout our experiments (we leave as an open problem to prove this). For brevity we only illustrate the condition (5.11) in our experiments.

Let us finish this section with some comments about Fourier series for solving PDEs on the sphere. One way of using them is to use standard double Fourier series on a "doubled-up" version of u, i.e., the DFS method (5.7)–(5.9). This is what Merilees did and, combined with a Fourier spectral method in value space, he solved the shallow water equations [107]. Another way is to use half-range cosine or sine series [21, 37, 125, 147, 175]—after all, spherical harmonics are represented as proper combinations of half-ranged cosine or sine series. For example, Orszag [125] suggested the use of approximations of the form

$$u(\lambda, \theta) \approx \sum_{k=-n/2}^{n/2} \sum_{j=0}^{n/2} \hat{u}_{jk} \sin^s \theta \cos j\theta e^{ik\lambda}, \tag{5.12}$$

where $s = 0$ if k is even, $s = 1$ if k is odd. (Note that the Fourier series in (5.12) approximates u directly, as opposed to $\tilde{u}$.) When using half-range cosine or sine terms as basis functions, one must be careful that the pole conditions are satisfied. One can either select basis functions that satisfy the pole conditions or impose a constraint on the Fourier coefficients to enforce it. For solving PDEs involving the Laplacian

operator (in both value and coefficient spaces), Orszag imposed certain constraints on the coefficients $\hat{u}_{jk}$ in (5.12), analogous to those in (5.11),

$$\sum_{j=0}^{n/2} \hat{u}_{jk} = \sum_{j=0}^{n/2} (-1)^j \hat{u}_{jk} = 0, \quad |k| \geq 2. \tag{5.13}$$

Boyd [21] studied Orszag's method and showed that the constraints (5.13) are actually not necessary for solving *time-independent* PDEs in value space but mentioned that the "absence of pole constraints is still risky" when working with coefficients.

The spectral method we present in this chapter is based on Merilees' approach and is similar to the method of Townsend et al. [163], but the Fourier multiplication matrices we use are different. It is simpler to implement than the half-range cosine/sine methods since there are no constraints to impose, and gives comparable accuracy.

5.2.2 Fourier multiplication matrices in coefficient space

The Laplacian operator on the sphere is given by

$$\Delta u = \frac{1}{\sin\theta}\big(\sin\theta\, u_\theta\big)_\theta + \frac{1}{\sin^2\theta} u_{\lambda\lambda}, \tag{5.14}$$

which we write as

$$\Delta u = u_{\theta\theta} + \frac{\cos\theta\sin\theta}{\sin^2\theta} u_\theta + \frac{1}{\sin^2\theta} u_{\lambda\lambda}. \tag{5.15}$$

In this section, we show how to construct a matrix for multiplication by $\sin^2\theta$ that is nonsingular, using the matrices presented in Section 2.3.2. This is crucial because some of the time-stepping methods we shall describe in Section 5.3 need to compute the inverse of such a matrix.

To multiply by $\sin^2\theta$ we apply (2.38) with $a(\theta) = \sin^2(\theta)$ and note that multiplying by $\sin^2\theta$ will increase the length of the representation by a factor of four. The resulting matrix for multiplication by $\sin^2\theta$, which we denote by $\mathbf{T}_{\sin^2}$, is given by

$$\mathbf{T}_{\sin^2} = \mathbf{Q}\mathbf{M}_{\sin^2}(:, 3 : m+3)\mathbf{P}, \tag{5.16}$$

where $\mathbf{M}_{\sin^2}$ is the $(m+1+4) \times (m+1+4)$ matrix defined by

$$\mathbf{M}_{\sin^2} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{4} & & & & \\ 0 & \frac{1}{2} & 0 & -\frac{1}{4} & & & \\ -\frac{1}{4} & 0 & \ddots & \ddots & \ddots & & \\ & -\frac{1}{4} & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & -\frac{1}{4} \\ & & & \ddots & \ddots & \ddots & 0 \\ & & & & -\frac{1}{4} & 0 & \frac{1}{2} \end{pmatrix}, \tag{5.17}$$

$\mathbf{P}$ is given by (2.32), and $\mathbf{Q}$ is the following $m \times (m+1+4)$ matrix,

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 1 & & & & 1 & 0 & 0 \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{5.18}$$

This leads to

$$\mathbf{T}_{\sin^2} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{4} & & & & & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{4} & & & & & 0 \\ -\frac{1}{8} & 0 & \ddots & \ddots & \ddots & & & & \\ & -\frac{1}{4} & \ddots & \ddots & \ddots & \ddots & & & \\ & & -\frac{1}{4} & \ddots & \ddots & \ddots & -\frac{1}{4} & & \\ & & & \ddots & \ddots & \ddots & \ddots & -\frac{1}{4} & \\ -\frac{1}{8} & & & & -\frac{1}{4} & \ddots & \frac{1}{2} & 0 & \\ 0 & 0 & & & & -\frac{1}{4} & 0 & \frac{1}{2} & \end{pmatrix}. \tag{5.19}$$

Using the Gershgorin circle theorem [173] we see that the $m \times m$ matrix (5.19) is nonsingular since it is row diagonally dominant, with strict diagonal dominance in the second row, and irreducible.

Let us add some comments about (5.19). If we operated in value space, we would obtain a singular matrix, since the multiplication matrix in value space, $\mathbf{M}^v_{\sin^2}$, a diagonal matrix with entries $\{\sin^2\theta_p\}_{p=1}^m$, has two zeros corresponding to $\theta_p = -\pi$ and $\theta_p = 0$. (The standard remedy in that case is to shift the θ-grid so that it does not contain the poles [107].) From this matrix, we can obtain a multiplication in coefficient space by multiplying by the DFT matrix $\mathbf{F}$ and its inverse,

$$\mathbf{F}\mathbf{M}^v_{\sin^2}\mathbf{F}^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{4} & & & & & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{4} & & & & & -\frac{1}{4} \\ -\frac{1}{4} & 0 & \ddots & \ddots & \ddots & & & & \\ & -\frac{1}{4} & \ddots & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & \ddots & -\frac{1}{4} & & \\ & & & \ddots & \ddots & \ddots & \ddots & -\frac{1}{4} & \\ -\frac{1}{4} & & & & -\frac{1}{4} & \ddots & \frac{1}{2} & 0 & \\ 0 & -\frac{1}{4} & & & & -\frac{1}{4} & 0 & \frac{1}{2} & \end{pmatrix}. \tag{5.20}$$

This matrix is indeed singular since $\mathbf{F}$ defines a unitary transformation, and its null space contains the vectors $(1, 1, \ldots)^T$ and $(1, -1, 1, -1, \ldots)^T$, which correspond to the Fourier coefficients of the delta functions at $\theta = 0$ and $\theta = -\pi$.

In [163], the authors use the $m \times m$ version of (5.17) (which is also nonsingular) as opposed to (5.19); this leads to incorrect results for trigonometric polynomials of degree $m/2-2$. To illustrate this, let us consider $m = 6$ and multiply $\sin^2(\theta)$ by a trigonometric polynomial of degree $m/2-2 = 1$, e.g., $\cos(\theta)$. In the representation (2.30), the function $\cos(\theta) = 1/2(e^{-i\theta} + e^{i\theta})$ has coefficients

$$c = \Big(0, 0, \frac{1}{2}, 0, \frac{1}{2}, 0\Big)^T, \tag{5.21}$$

while the product $\cos(\theta)\sin(\theta) = -1/8(e^{-3i\theta} + e^{3i\theta}) + 1/8(e^{-i\theta} + e^{i\theta})$ has coefficients

$$d = \Big(-\frac{1}{4}, 0, \frac{1}{8}, 0, \frac{1}{8}, 0\Big)^T. \tag{5.22}$$

For $m = 6$, we have

$$\mathbf{T}_{\sin^2} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{4} & 0 & 0 \\ -\frac{1}{8} & 0 & \frac{1}{2} & 0 & -\frac{1}{4} & 0 \\ 0 & -\frac{1}{4} & 0 & \frac{1}{2} & 0 & -\frac{1}{4} \\ -\frac{1}{8} & 0 & -\frac{1}{4} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & -\frac{1}{4} & 0 & \frac{1}{2} \end{pmatrix}, \quad \mathbf{T}_{\sin^2} c = \Big(-\frac{1}{4}, 0, \frac{1}{8}, 0, \frac{1}{8}, 0\Big)^T = d, \tag{5.23}$$

while

$$\mathbf{M}_{\sin^2} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{4} & 0 & 0 \\ -\frac{1}{4} & 0 & \frac{1}{2} & 0 & -\frac{1}{4} & 0 \\ 0 & -\frac{1}{4} & 0 & \frac{1}{2} & 0 & -\frac{1}{4} \\ 0 & 0 & -\frac{1}{4} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & -\frac{1}{4} & 0 & \frac{1}{2} \end{pmatrix}, \quad \mathbf{M}_{\sin^2} c = \Big(-\frac{1}{8}, 0, \frac{1}{8}, 0, \frac{1}{8}, 0\Big)^T \neq d. \tag{5.24}$$

Similarly, the matrix for multiplication by $\cos\theta\sin\theta$ is the product of three matrices,

$$\mathbf{T}_{\cos\sin} = \mathbf{Q}\mathbf{M}_{\cos\sin}(:, 3 : m+3)\mathbf{P}, \tag{5.25}$$

with

$$\mathbf{M}_{\cos\sin} = \begin{pmatrix} 0 & 0 & \frac{i}{4} & & & & \\ 0 & 0 & 0 & \frac{i}{4} & & & \\ -\frac{i}{4} & 0 & \ddots & \ddots & \ddots & & \\ & -\frac{i}{4} & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & \frac{i}{4} \\ & & & \ddots & \ddots & \ddots & 0 \\ & & & & -\frac{i}{4} & 0 & 0 \end{pmatrix}, \tag{5.26}$$

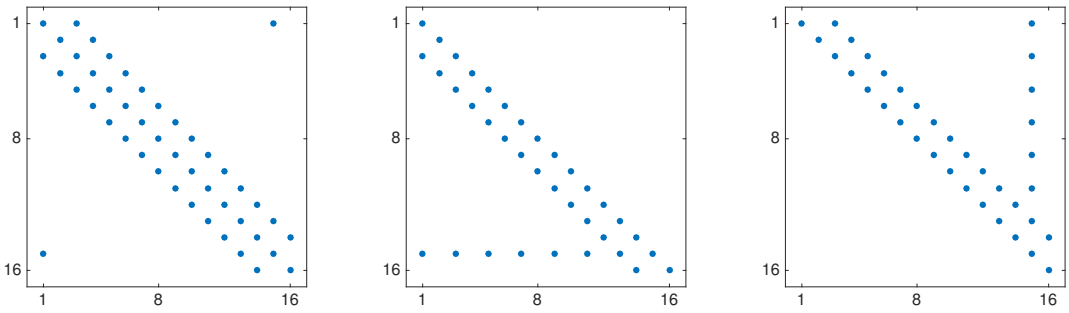

Figure 5.2: *Sparsity pattern of the matrix* $z\mathbf{T}_{\sin^2} + w\mathbf{T}_{\sin^2}\mathbf{L}_i$ *(left), and its* L *(middle) and* U *(right) factors for* $m = 16$*. Triangular systems involving* L *and* U *are solvable in* $\mathcal{O}(m)$ *operations.*

and is given by

$$\mathbf{T}_{\cos\sin} = \begin{pmatrix} 0 & 0 & \frac{i}{4} & & & & -\frac{i}{4} & 0 \\ 0 & 0 & 0 & \frac{i}{4} & & & & 0 \\ -\frac{i}{8} & 0 & \ddots & \ddots & \ddots & & & \\ & -\frac{i}{4} & \ddots & \ddots & \ddots & \ddots & & \\ & & -\frac{i}{4} & \ddots & \ddots & \ddots & \frac{i}{4} & \\ & & & \ddots & \ddots & \ddots & \ddots & \frac{i}{4} \\ \frac{i}{8} & & & & -\frac{i}{4} & \ddots & 0 & 0 \\ 0 & 0 & & & & -\frac{i}{4} & 0 & 0 \end{pmatrix}. \tag{5.27}$$

5.2.3 Laplacian matrix and linear systems

We want to discretize the Laplacian operator (5.15) with a matrix $\mathbf{L}$ using a Fourier spectral method in coefficient space on an $n \times m$ uniform longitude-latitude grid (5.10), and we look for a solution of the form (5.8)–(5.9). Using Kronecker products, we can write $\mathbf{L}$ as

$$\mathbf{L} = \mathbf{I}_n \otimes (\mathbf{D}_m^{(2)} + \mathbf{T}_{\sin^2}^{-1}\mathbf{T}_{\cos\sin}\mathbf{D}_m) + \mathbf{D}_n^{(2)} \otimes (\mathbf{T}_{\sin^2}^{-1}), \tag{5.28}$$

where $\mathbf{T}_{\sin^2}$ and $\mathbf{T}_{\cos\sin}$ have been defined in the previous section, $\mathbf{I}_n$ is the $n \times n$ identity matrix, and $\mathbf{D}_m$ and $\mathbf{D}_m^{(2)}$ are the differentiation matrices (2.35) and (2.36). Note that $\mathbf{L}$ is block diagonal with n dense blocks of size $m \times m$. Let us emphasise that the n blocks correspond to the n longitudinal wavenumbers $-n/2 \le k \le n/2-1$ and that the size m of each block corresponds to the m latitudinal wavenumbers $-m/2 \le j \le m/2-1$.

Some of the time-stepping schemes we shall describe in Section 5.3 involve solving linear systems of the form $(z\mathbf{I}_{nm}+w\mathbf{L})x = b$, where $\mathbf{I}_{nm}$ denotes the $nm \times nm$ identity matrix. Fortunately, the matrix structure allows for a linear-cost direct solver. The key observation is that $\mathbf{L}$ is block diagonal, and each $m \times m$ block $\mathbf{L}_i$ of $\mathbf{L}$,

$$\mathbf{L}_i = \mathbf{D}_m^{(2)} + \mathbf{T}_{\sin^2}^{-1}\mathbf{T}_{\cos\sin}\mathbf{D}_m + \mathbf{D}_n^{(2)}(i,i)\mathbf{T}_{\sin^2}^{-1}, \tag{5.29}$$

is dense but $(z\mathbf{I}_m + w\mathbf{L}_i)x = b$ can be solved in $\mathcal{O}(m)$ operations since it is equivalent to solving

$$(z\mathbf{T}_{\sin^2} + w\mathbf{T}_{\sin^2}\mathbf{L}_i)x = \mathbf{T}_{\sin^2}b, \tag{5.30}$$

and $(z\mathbf{T}_{\sin^2} + w\mathbf{T}_{\sin^2}\mathbf{L}_i)$ is pentadiagonal with two (near-)corner elements. Therefore, using a sparse direct solver based on the standard LU factorization without pivots [43], the L and U factors have the sparsity patterns indicated in Figure 5.2 (due to the diagonal dominance in most blocks, the LU factorization without pivoting completes without breaking down[6]). Since L and U have at most three nonzero elements per column and row respectively, each triangular linear system is solvable in $\mathcal{O}(m)$ operations; thus once an LU factorization is computed, the linear system (5.30) can be solved in $\mathcal{O}(m)$ operations. Therefore, linear systems of the form

$$(z\mathbf{I}_{nm} + w\mathbf{L})x = b \tag{5.31}$$

can be solved blockwise in $\mathcal{O}(nm)$ operations.[7] Because of the structure of the LU factors (every other term is zero, see Figure 5.2), the coefficients one gets when solving systems of the form (5.31) have the property that the even modes in λ correspond to even functions in θ and the odd modes to odd functions.

5.2.4 Poisson's equation

To test the accuracy of (5.28), we first solve a *time-independent* PDE, *Poisson's equation*, with a zero-mean condition for uniqueness,

$$\begin{aligned} &\Delta u = f(\lambda,\theta), \quad (\lambda,\theta) \in [-\pi,\pi] \times [0,\pi], \\ &\int_0^\pi \int_{-\pi}^\pi u(\lambda,\theta)\sin\theta d\lambda d\theta = 0, \end{aligned} \tag{5.32}$$

[6]For the IMEX schemes of Section 5.3.2, we can prove that all the blocks but one are diagonally dominant; for ETDRK4-CF (see Section 5.3.1), the diagonal dominance is violated much more frequently. Nonetheless, diagonal dominance is merely a sufficient condition for the LU factorization to not require pivoting, and in practice, all the methods result in linear systems for which the LU factorization causes no stability issues.

[7]In practice we do not solve linear systems of the form (5.31) blockwise, i.e., by solving n linear systems (5.30). (If we were solving these linear systems blockwise, we could use parallel computing.) Instead, a more efficient approach is to store the left-hand sides of (5.30) altogether as a sparse matrix and use MATLAB's sparse linear solver.

where f also has zero mean on $[-\pi,\pi]\times[0,\pi]$. Using the DFS method, we seek a solution $\tilde{u}$ of the "doubled-up" version of (5.32),

$$\begin{aligned}\Delta\tilde{u} &= \tilde{f}(\lambda,\theta), \quad (\lambda,\theta)\in[-\pi,\pi]^2,\\ \int_0^{\pi}\int_{-\pi}^{\pi}\tilde{u}(\lambda,\theta)\sin\theta d\lambda d\theta &= 0,\end{aligned} \tag{5.33}$$

of the form (5.8)–(5.9). The true solution u can be recovered by restricting $\tilde{u}$ to $(\lambda,\theta)\in[-\pi,\pi]\times[0,\pi]$. (Note that the zero-mean condition in (5.33) is on the original domain $[-\pi,\pi]\times[0,\pi]$ since $\tilde{u}$ must coincide with u on this domain.) Townsend et al. [163] showed that the zero-mean condition can be discretized as

$$\begin{aligned}\int_0^{\pi}\int_{-\pi}^{\pi}\tilde{u}(\lambda,\theta)\sin\theta d\lambda d\theta &\approx \sum_{j=-m/2}^{m/2}{}' \sum_{k=-n/2}^{n/2}{}' \hat{u}_{jk}\int_0^{\pi}\sin\theta e^{ij\theta}d\theta\int_{-\pi}^{\pi}e^{ik\lambda}d\lambda\\ &= 2\pi\sum_{j=-m/2}^{m/2}{}' \hat{u}_{j0}\frac{1+e^{ij\pi}}{1-j^2} = 0.\end{aligned} \tag{5.34}$$

Poisson's equation (5.33) is then discretized by

$$\mathbf{L}\hat{u} = \hat{f}, \tag{5.35}$$

where $\hat{u}$ and $\hat{f}$ are the vectors of nm Fourier coefficients (5.9) of $\tilde{u}$ and $\tilde{f}$, and $\mathbf{L}$ is the Laplacian matrix (5.28). We impose the zero-mean condition by replacing the $(m/2+1)$st row of the $(n/2+1)$st block of $\mathbf{L}$ by (5.34).[8] Note that, given the Fourier coefficients of $\tilde{u}$ and $\tilde{f}$, (5.35) can be solved in $\mathcal{O}(nm)$ operations since it is of the form (5.31) with $z=0$.

Since the Laplacian operator on the sphere has real eigenvalues $-l(l+1)$, $l\geq 0$, with eigenfunctions the spherical harmonics $Y_l^m(\lambda,\theta)$ [10], a simple test is to take the right-hand side f to be a spherical harmonic Y_l^m. In this case, the exact solution is $u=-1/[l(l+1)]Y_l^m$. A slightly more complicated test is given in [37] and this is what we shall present here. We solve Poisson's equation for a family of right-hand sides f_l defined by

$$f_l(\lambda,\theta) = l(l+1)\sin^l\theta\cos(l\lambda) + (l+1)(l+2)\cos\theta\sin^l\theta\cos(l\lambda), \quad l\geq 1. \tag{5.36}$$

The exact solution $u_l^{ex}(\lambda,\theta)$ is given by

$$u_l^{ex}(\lambda,\theta) = -\sin^l\theta\cos(l\lambda) - \cos\theta\sin^l\theta\cos(l\lambda), \quad l\geq 1. \tag{5.37}$$

[8]The $(n/2+1)$st block of $\mathbf{L}$ corresponds to longitudinal wavenumber $k=0$ while the $(m/2+1)$st row of this block corresponds to latitudinal wavenumber $j=0$.

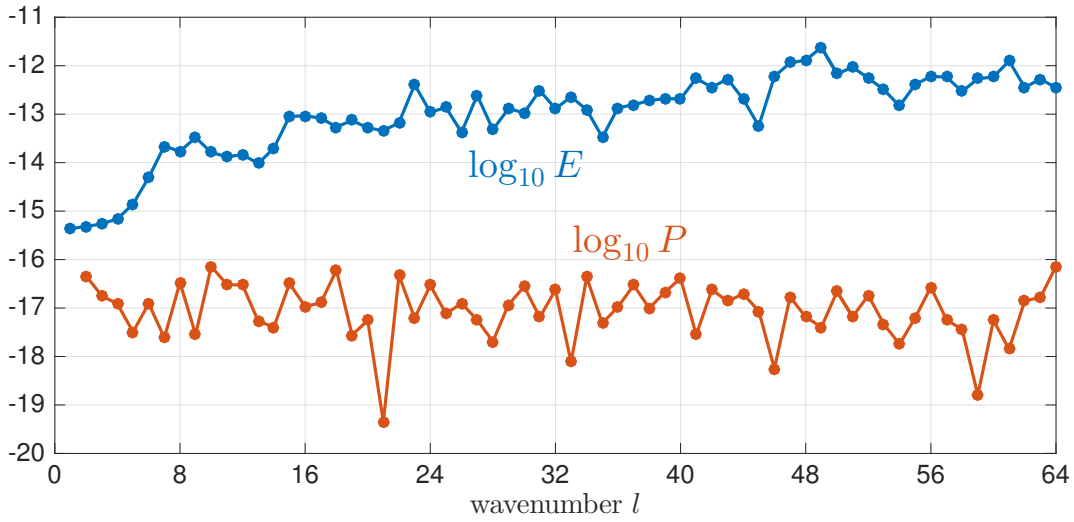

Figure 5.3: *Variation of the relative error* $\log_{10} E$ *and of the error in the pole conditions* $\log_{10} P$ *with wavenumber* l *for* $m = n = 128$. *The accuracy is excellent for every wavenumber* $1 \leq l \leq 64$.

We compute the solutions u_l for $m = n = 128$ grid points in each direction for $1 \leq l \leq 64$ and, following [37, Fig. 1], we plot the logarithm (base 10) of the relative L^2-error E,

$$E = \frac{||u_l(\lambda,\theta) - u_l^{ex}(\lambda,\theta)||_2}{||u_l^{ex}(\lambda,\theta)||_2}, \tag{5.38}$$

with (continuous) L^2-norm on the sphere $||\cdot||_2$, against l. (The L^2-norm of a `spherefun` can be computed in Chebfun with the `norm` command.) We also plot the logarithm of the error P in satisfying the pole conditions (5.11), i.e.,

$$P = \max\left(\max_{k\neq 0}\left|\sum_{j=-m/2}^{m/2}{}' \hat{u}_{jk}^l\right|, \max_{k\neq 0}\left|\sum_{j=-m/2}^{m/2}{}' (-1)^j \hat{u}_{jk}^l\right|\right), \tag{5.39}$$

where the $\hat{u}_{jk}^l$ are the Fourier coefficients of $\tilde{u}_l$; see Figure 5.3. The accuracy is excellent; the results are similar to those shown in [37, Fig. 1] and to results we have obtained with the Poisson solver of [163] (although the discretization matrices are different as noted in Section 5.2.2, the effect on the solution is negligible when m and n are taken large enough).

5.2.5 Heat equation

As we mentioned in the introduction, Orszag [125] and Boyd [21] showed that when solving a *time-dependent* PDE involving the Laplacian with half-range cosine/sine

series à la (5.12), it is crucial to impose the pole conditions (5.13); otherwise, they observed that "the numerical solution still converged as the time step was shortened—to a wildly wrong answer" [21]. Therefore, let us now test the accuracy of our method with a time-dependent PDE, and illustrate that the pole conditions (5.11) is also automatically satisfied in this case.

We consider the *heat equation* with a thermal diffusivity and an initial condition that lead to a particularly simple exact solution,

$$u_t = \frac{1}{l(l+1)}\Delta u, \quad u(t=0,\lambda,\theta) = Y_l^m(\lambda,\theta), \quad (\lambda,\theta) \in [-\pi,\pi]\times[0,\pi]. \tag{5.40}$$

The exact solution is $u^{ex}(t,\lambda,\theta) = e^{-t}Y_l^m(\lambda,\theta)$. Using the DFS method, we seek a solution $\tilde{u}$ of the "doubled-up" version of (5.40),

$$\tilde{u}_t = \frac{1}{l(l+1)}\Delta\tilde{u}, \quad \tilde{u}(t=0,\lambda,\theta) = Y_l^m(\lambda,\theta), \quad (\lambda,\theta) \in [-\pi,\pi]^2, \tag{5.41}$$

of the form

$$\tilde{u}(t,\lambda,\theta) \approx \sum_{j=-m/2}^{m/2}{}' \sum_{k=-n/2}^{n/2}{}' \hat{u}_{jk}(t)e^{ij\theta}e^{ik\lambda}. \tag{5.42}$$

We discretize the Laplacian operator with (5.28) and use the fourth-order backward differentiation formula to march in time. We take $l = 64$, $\tilde{u}(t=0,\lambda,\theta) = Y_{64}^{64}(\lambda,\theta)$, $m = n = 128$ grid points and solve (5.41) up to $t = 1$ for various time-steps h. We plot the relative L^2-error E at $t = 1$,

$$E = \frac{||u(t=1,\lambda,\theta) - u^{ex}(t=1,\lambda,\theta)||_2}{||u^{ex}(t=1,\lambda,\theta)||_2}, \tag{5.43}$$

and the error in the pole conditions (as defined in (5.39)) against h; the error E scales as $\mathcal{O}(h^4)$, see Figure 5.4. (With $m = n = 128$ grid points, the error due to the spatial discretization is small compared to the error due to the time discretization, so we are really measuring the latter.)

5.2.6 Bound for the eigenvalues of the Laplacian matrix

As mentioned in Section 5.2.4, the eigenvalues of the Laplacian operator are $-l(l+1)$, for integers $l \geq 0$. What about the eigenvalues of the Laplacian matrix (5.28)? We show in Appendix D that they are all real and nonpositive, and have observed numerically that some of them are spectrally accurate approximations to the eigenvalues $-l(l+1)$, but some others, the so-called *outliers* [164, Chap. 10], are of order $\mathcal{O}(n^2m^2)$ as $n = m \to \infty$. We shall prove the latter fact below. These large eigenvalues are

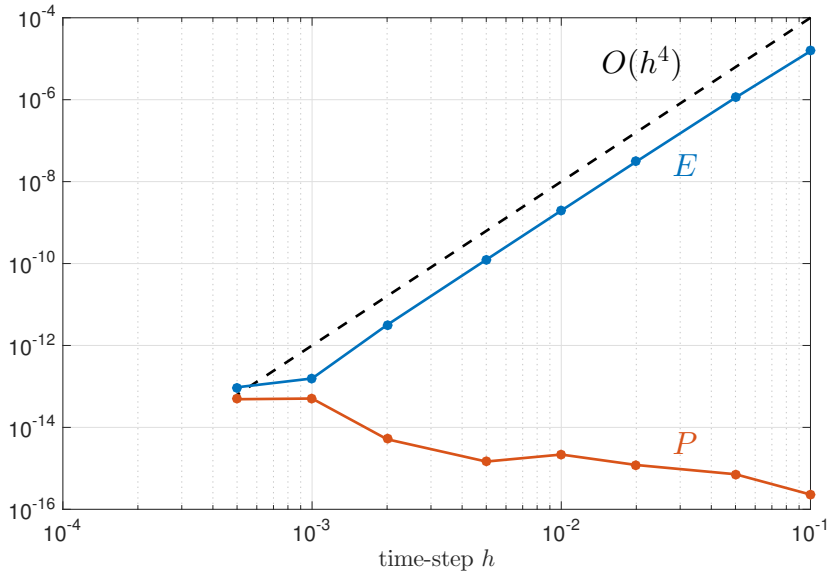

Figure 5.4: *Variation of the relative error E at $t = 1$ and of the error in the pole conditions P with time-step h for $m = n = 128$. The error E scales as $\mathcal{O}(h^4)$. Note that the initial condition $Y_{64}^{64}(\lambda, \theta)$ corresponds to the highest wavenumbers that can be resolved on a 128×128 grid.*

meaningless physically but of crucial importance in practice since for time-stepping algorithms applied to (5.2) to be stable, we need the eigenvalues of (5.28), scaled by the time-step, to lie in their stability region.

We examine now the largest (in magnitude) eigenvalues of (5.28). It suffices to examine each block

$$\mathbf{L}_i = \big(\mathbf{D}_m^{(2)} + \mathbf{T}_{\sin^2}^{-1}\mathbf{T}_{\cos\sin}\mathbf{D}_m + \mathbf{D}_n^{(2)}(i,i)\mathbf{T}_{\sin^2}^{-1}\big), \tag{5.44}$$

whose largest eigenvalue can be bounded as

$$|\lambda_{\max}(\mathbf{L}_i)| \le \|\mathbf{L}_i\| \le \|\mathbf{D}_m^{(2)}\| + \|\mathbf{T}_{\sin^2}^{-1}\|\|\mathbf{T}_{\cos\sin}\mathbf{D}_m + \mathbf{D}_n^{(2)}(i,i)\mathbf{I}_m\|. \tag{5.45}$$

We trivially have $\|\mathbf{D}_m^2\| = \mathcal{O}(m^2)$ and

$$\|\mathbf{T}_{\cos\sin}\mathbf{D}_m + \mathbf{D}_n^{(2)}(i,i)\mathbf{I}_m\| \le \|\mathbf{T}_{\cos\sin}\mathbf{D}_m\| + |\mathbf{D}_n^{(2)}(i,i)| = \mathcal{O}(m + i^2). \tag{5.46}$$

It remains to bound $\|\mathbf{T}_{\sin^2}^{-1}\|$; we claim that this is $\mathcal{O}(m^2)$. To verify this, recalling (5.16), we first examine $\mathbf{M}_{\sin^2}$. Its eigenvalues are explicitly known to be

$$\sigma_j = \frac{1}{2}\big(\cos(\pi j/(m/2+1)) + 1\big), \quad 1 \le j \le m/2, \tag{5.47}$$

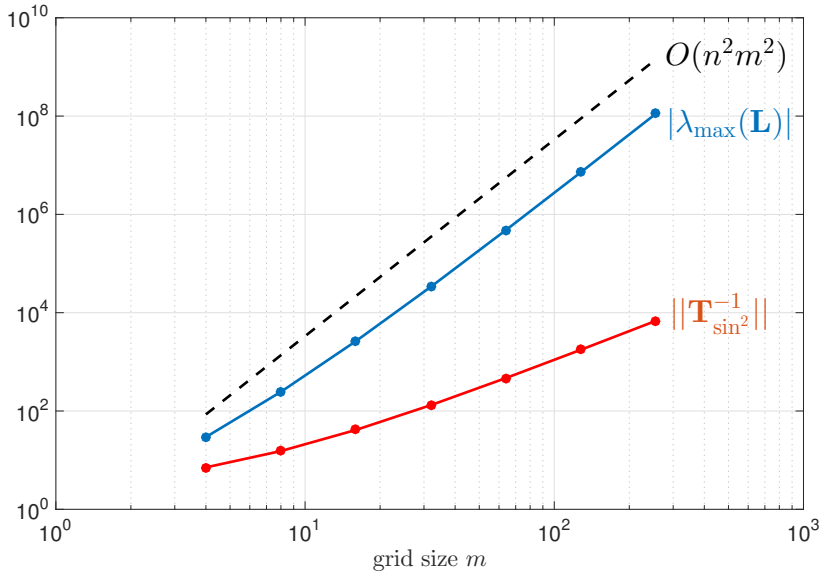

Figure 5.5: *Variation of* $|\lambda_{\max}(\mathbf{L})|$ *with* $m = n$*; the bound* (5.50) *is accurately reflected.*

with each eigenvalue of multiplicity two. Since $\mathbf{M}_{\sin^2}$ is symmetric positive definite, these are also its singular values. Now, using $\sigma_{\min}(\mathbf{ABC}) \geq \sigma_{\min}(\mathbf{A})\sigma_{\min}(\mathbf{B})\sigma_{\min}(\mathbf{C})$, $\sigma_{\min}(\mathbf{P}) = \frac{1}{\sqrt{2}}$ and $\sigma_{\min}(\mathbf{Q}) = 1$, we have

$$\sigma_{\min}(\mathbf{T}_{\sin^2}) \geq \frac{1}{\sqrt{2}}\sigma_{\min}(\mathbf{M}_{\sin^2}(:, 3 : m+3)). \tag{5.48}$$

The bottom $m \times m$ part of $\mathbf{M}_{\sin^2}(:, 3 : m+3)$ has singular values σ_j. Since the singular values have the monotonicity property $\sigma_i\left(\left[\begin{smallmatrix}\mathbf{A}\\ \mathbf{B}\end{smallmatrix}\right]\right) \geq \sigma_i(\mathbf{B})$ if $\mathbf{B}$ is square (or tall-skinny), we have

$$\sigma_{\min}(\mathbf{M}_{\sin^2}(:, 3 : m+3)) \geq \sigma_{\min}(\mathbf{M}_{\sin^2}(3 : m+3, 3 : m+3)) = \min_j \sigma_j, \tag{5.49}$$

with $\min_j \sigma_j = \mathcal{O}(1/m^2)$. Therefore we obtain $\|\mathbf{T}^{-1}_{\sin^2}\| = 1/\sigma_{\min}(\mathbf{T}_{\sin^2}) = \mathcal{O}(m^2)$, as claimed. We conclude that $\|\mathbf{L}_i\| = \mathcal{O}(i^2m^2)$; since this holds for every i, it gives

$$|\lambda_{\max}(\mathbf{L})| = \mathcal{O}(n^2m^2 + m^3) = \mathcal{O}(n^2m^2), \tag{5.50}$$

when $m = n$. We illustrate (5.50) in Figure 5.5, which suggests that it is sharp.

The fact that the largest eigenvalue has order n^2m^2 makes time-dependent PDEs on the sphere particularly stiff. This is a consequence of the second order of the Laplacian operator and the clustering of the points near the poles in (5.47). It would imply severe restrictions on the time-steps for generic explicit algorithms—this is why we use exponential integrators and IMEX schemes, which we describe next. (In

the literature, the severe time-stepping restrictions due to uniform longitude-latitude grids is sometimes called the *pole problem* [21, 125]). Note that the time-stepping restrictions resulting from the clustering near the poles can be addressed by truncating high-frequency terms in the space discretization (see, e.g., [61, Sec. 2.1.6] and [64]). However, this approach does not overcome stiffness resulting from the second-order operator.

5.3 Fourth-order time-stepping on the sphere

Using the DFS method, we seek a solution $\tilde{u}$ of the "doubled-up" version of (5.6),

$$\tilde{u}_t = \alpha\Delta\tilde{u} + \mathcal{N}(\tilde{u}), \quad \tilde{u}(t=0,\lambda,\theta) = \tilde{u}_0(\lambda,\theta), \quad (\lambda,\theta) \in [-\pi,\pi]^2, \tag{5.51}$$

of the form (5.42). Discretizing the Laplacian operator with the Laplacian matrix (5.28), we obtain the system of nm ODEs (5.2) where $\mathbf{L}$ is (5.28) multiplied by α. Time is discretized with time-step h and the problem is to find the Fourier coefficients $\hat{u}^{n+1}$ of $\tilde{u}$ at $t_{n+1} = (n+1)h$ from the coefficients $\hat{u}^n$ at $t_n = nh$ and also coefficients at previous time-steps (for multistep schemes). Note that, in practice, nonlinear evaluations $\mathbf{N}(\hat{u}^n)$ are carried out in value space.

We present in this section four time-stepping algorithms for solving (5.2), and show how it is possible to achieve $\mathcal{O}(nm \log nm)$ complexity per time-step in most cases. Two of them are based on the ETRDK4 exponential integrator (4.19) with different strategies for computing the matrix exponential and related functions, while the two others are IMEX schemes. As before, we have observed numerically that these schemes combined with our spatial discretization preserve both the "doubled-up" symmetry (5.7) and the pole conditions (5.11), i.e., if one starts with a smooth "doubled-up" initial condition, then the solution at time t is also a smooth "doubled-up" function.

5.3.1 Exponential integrators

Based on the experience of the last chapter, we take as representative exponential integrator the ETDRK4 scheme (4.19). When all the eigenvalues of $\mathbf{L}$ are real (i.e., $\alpha \in \mathbb{R}$, diffusive PDE), matrix-vector products $\varphi_l(h\mathbf{L})v$ can be evaluated using rational approximations computed by the Carathéodory–Fejér (CF) method, as described in [143]. If $\mathbf{L}$ has some imaginary eigenvalues—the extreme case being when all the eigenvalues are imaginary (i.e., $\alpha \in i\mathbb{R}$, dispersive PDE)—methods based on rational

approximations necessarily become expensive, and the φ-functions have to be precomputed before the time-stepping starts, e.g., using the eigenvalue decomposition of $\mathbf{L}$ and contour integrals.[9] We denote by ETDRK4-CF the method with CF approximations and by ETDRK4-EIG the method with eigenvalue decomposition. We shall give details about the precomputation of the coefficients for both ETDRK4-CF and ETDRK4-EIG below.

Note that Du and Zhu computed in [54] the stability region of (4.19) and showed that it includes parts of both the negative real axis and the imaginary axis.

ETDRK4-CF When all the eigenvalues of $\mathbf{L}$ are real, one can compute matrix-vector products $\varphi_l(h\mathbf{L})v$ in $\mathcal{O}(mn)$ operations using near-best rational approximations to the φ-functions on the negative real axis. For an efficient implementation, we use the algorithm that uses common poles for approximating the exponential and other φ-functions [143], as we summarize below. Using the CF method for the negative real line [165], we obtain a rational approximant

$$e^z \approx r_\infty + \sum_{j=1}^{p} \frac{c_j}{z - z_j}, \tag{5.52}$$

which has error decaying like $\approx 9.28903^{-p}$ with a type (p, p) function. (We use the MATLAB `cf` code of Trefethen, Weideman and Schmelzer [167] in our experiments.) To obtain an approximant to $\varphi_l(z)$ using the approximant (5.52) to $e^z = \varphi_0(z)$, we use the fact [143, Prop. 4.1] that defining

$$B_z = \begin{pmatrix} z & 1 \\ 0 & 0 \end{pmatrix} \tag{5.53}$$

we have

$$\varphi_l(B_z) = \begin{pmatrix} \varphi_l(z) & \varphi_{l+1}(z) \\ 0 & \varphi_l(0) \end{pmatrix}, \quad l \geq 0. \tag{5.54}$$

Together with the identity

$$(B_z - z_j I)^{-1} = \begin{pmatrix} (z - z_j)^{-1} & (z - z_j)^{-1} z_j^{-1} \\ 0 & -z_j^{-1} \end{pmatrix}, \tag{5.55}$$

we obtain the approximation

$$\varphi_l(z) \approx \sum_{j=1}^{p} \frac{c_j z_j^{-l}}{z - z_j}, \quad l \geq 0. \tag{5.56}$$

[9] A comparison of methods for computing the φ-functions can be found in [9].

As suggested in [143, Prop. 4.1], we further incorporate a shift 1 in (5.52), which with $p = 12$ gives accuracy $\approx 10^{-10}$ on the negative real axis for all φ_l, $0 \le l \le 3$. (The accuracy is $\approx 10^{-8}$ with $p = 10$.) Given (5.52) and (5.56), evaluating $\varphi_l(h\mathbf{L})b$ for a vector b as in (4.19) can be approximated as

$$\varphi_l(h\mathbf{L})b \approx \sum_{j=1}^{p} c_j z_j^{-l}(h\mathbf{L} - z_j\mathbf{I})^{-1}b, \quad 0 \le l \le 3, \tag{5.57}$$

which reduces to $p = 12$ shifted linear systems of the form (5.31), which we do with linear cost, as described in Section 5.2.3. In practice, we compute and store the LU factorizations of the matrices that appear in (5.57) before the time-stepping starts. Note that computing different products $\varphi_l(hL)v$ at once with the same v requires no further linear systems.

Let us emphasize three aspects of this approach. First, it is not necessary to explicitly compute and store the φ-functions; instead, their action on vectors is directly computed via (5.57). Second, the most expensive operation is the 2D FFT, which costs $\mathcal{O}(nm \log nm)$ operations; see Table 5.1. Third, this method is not applicable when $\mathbf{L}$ has some imaginary eigenvalues; this is because low-degree rational functions are unable to approximate the exponential on the imaginary axis, which is oscillatory. In this case one has to compute and store the φ-functions, and the complexity increases to $\mathcal{O}(nm^2)$ per time-step, as we describe next.

ETDRK4-EIG To compute the φ-functions, one can use a method based on eigenvalues and eigenvectors. The idea is to diagonalize $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ and then apply the φ-functions to the eigenvalues,

$$\varphi_l(h\mathbf{L}) = \mathbf{V}\varphi_l(h\mathbf{\Lambda})\mathbf{V}^{-1}, \quad 0 \le l \le 3, \tag{5.58}$$

with

$$\varphi_l(h\mathbf{\Lambda}) = \begin{pmatrix} \varphi_l(h\lambda_1) & & & \\ & \varphi_l(h\lambda_2) & & \\ & & \ddots & \\ & & & \varphi_l(h\lambda_{nm}) \end{pmatrix}. \tag{5.59}$$

For a general $nm \times nm$ matrix $\mathbf{L}$, this would require $\mathcal{O}(n^3m^3)$ operations, but for the Laplacian matrix (5.28), this can be done blockwise in $\mathcal{O}(nm^3)$ operations. Note that this corresponds to Method 14 of [110]. Theoretically, this approach only works when $\mathcal{L}$ is nondefective, that is, when it has a complete set of linearly independent eigenfunctions—this is a well known result for the Laplacian operator on the

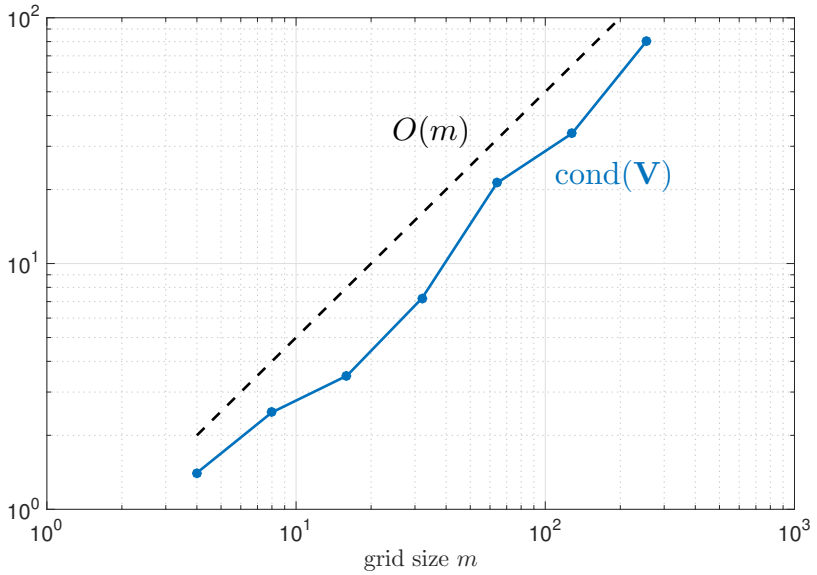

Figure 5.6: *Variation of* cond($\mathbf{V}$) *with* $m = n$*. It is of order* m *and reasonably small; therefore, an approach based on eigenvalues and eigenvectors for the computation of the* φ*-functions is effective.*

sphere [10]. In practice, difficulties occur when the discretizaion $\mathbf{L}$ is "nearly" defective, i.e., when the condition number $\text{cond}(\mathbf{V}) = ||\mathbf{V}||\,||\mathbf{V}^{-1}||$ is large. Fortunately, we have observed numerically that the condition number is small and of order m as $m = n$ increases; see Figure 5.6.

Once we have computed the eigenvalue decomposition, we evaluate the φ-functions at each scaled eigenvalue $h\lambda$ using Cauchy's integral formula, as in Section 4.2.2.

The precomputation step costs $\mathcal{O}(nm^3)$ operations, while the cost per time-step is $\mathcal{O}(nm^2)$ since one has to compute block diagonal matrix-vector products $\varphi_l(h\mathbf{L})v$; see Table 5.1.

5.3.2 Implicit-explicit schemes

We present in this section the two IMEX schemes we consider in this chapter. The first one, IMEX-BDF4 [8], is a multistep scheme which is stable only for diffusive PDEs. The second one, LIRK4 [27], is a one-step scheme, stable for both diffusive and dispersive PDEs.

IMEX-BDF4 Following Kassam and Trefethen [84], we consider a scheme known either as SBDF4 (in [8]), AB4BD4 (in [42]) or IMEX-BDF4 (in [81]), which combines a fourth-order Adams–Bashforth formula and a fourth-order backward differentiation

Table 5.1: *Computational costs* (*per time-step*) *of the time-stepping algorithms with* $p = 12$ *in the CF approximation. The IMEX-BDF*4 *scheme is particularly cheap while for ETDRK*4*-CF one needs to solve an extremely large number of linear systems. ETDRK*4*-EIG is the only scheme that has a* $\mathcal{O}(nm^2)$ *cost per time-step and a* $\mathcal{O}(nm^3)$ *precomputation step. Precomputations for the other schemes* (LU *factorizations*) *cost* $\mathcal{O}(nm)$ *operations.*

	ETDRK4		**IMEX**	
	CF	**EIG**	**BDF4**	**LIRK4**
# $\mathcal{O}(nm \log nm)$ **FFTs**	8	8	2	12
# $\mathcal{O}(nm)$ **linear solves**	$9p = 108$	0	1	5
# $\mathcal{O}(nm^2)$ **matrix-vector products**	0	9	0	0
diffusive PDEs	✓	✓	✓	✓
dispersive PDEs	×	✓	×	✓

scheme in a nontrivial way. The method is given by:

$$\begin{aligned}(25\mathbf{I}_{nm} - 12h\mathbf{L})\hat{u}^{n+1} = 48\hat{u}^n - 36\hat{u}^{n-1} + 16\hat{u}^{n-2} - 3\hat{u}^{n-3} + 48h\mathbf{N}(\hat{u}^n) \\ - 72h\mathbf{N}(\hat{u}^{n-1}) + 48h\mathbf{N}(\hat{u}^{n-2}) - 12h\mathbf{N}(\hat{u}^{n-3}).\end{aligned} \tag{5.60}$$

At each time-step, one has to solve a linear system to get the Fourier coefficients $\hat{u}^{n+1}$, which we do with linear cost by multiplying each block of (5.60) by $\mathbf{T}_{\sin^2}$, as explained in Section 5.2.3. Therefore, the dominant cost in (5.60) is the $\mathcal{O}(nm \log nm)$ 2D FFT for the nonlinear evaluations; see Table 5.1.

Let us add three comments about (5.60). First, the LU factorization of the left-hand side is computed and stored before the time-stepping starts. Second, this is a multistep formula so it has to be started with a one-step scheme—in the numerical comparisons of Section 5.4, we initialize it with three steps of ETDRK4-CF. Third, it is unstable for dispersive PDEs since the stability region of the fourth-order backward differentiation formula does not contain the portion of the imaginary axis near the origin. In these cases, one can use IMEX Runge–Kutta schemes [27, 86, 127] or extrapolation-based IMEX schemes [32, 40]. We have decided to focus on the former.

LIRK4 IMEX Runge–Kutta schemes combine explicit Runge–Kutta formulas to adavance the nonlinear part and implicit Runge–Kutta to advance the linear part [27, 86, 127]. In this chapter, we use the fourth-order LIRK4 scheme of Calvo, de Frutos

and Novo [27]. This combines an implicit L-stable five-stage fourth-order Runge–Kutta method, whose Butcher tableau is given in [69, Table 6.5], with a six-stage fourth-order explicit Runge–Kutta method, whose coefficients were derived in [27]. The formula for this scheme is:

$$\begin{aligned}
(\mathbf{I}_{nm} - \tfrac{1}{4}h\mathbf{L})\hat{a}^n &= \hat{u}^n + \tfrac{1}{4}h\mathbf{N}(\hat{u}^n),\\
(\mathbf{I}_{nm} - \tfrac{1}{4}h\mathbf{L})\hat{b}^n &= \hat{u}^n + \tfrac{1}{2}h\mathbf{L}\hat{a}^n - \tfrac{1}{4}h\mathbf{N}(\hat{u}^n) + h\mathbf{N}(\hat{a}^n),\\
(\mathbf{I}_{nm} - \tfrac{1}{4}h\mathbf{L})\hat{c}^n &= \hat{u}^n + \tfrac{17}{50}h\mathbf{L}\hat{a}^n - \tfrac{1}{25}h\mathbf{L}\hat{b}^n - \tfrac{13}{100}h\mathbf{N}(\hat{u}^n) + \tfrac{43}{75}h\mathbf{N}(\hat{a}^n)\\
&\qquad + \tfrac{8}{75}h\mathbf{N}(\hat{b}^n),\\
(\mathbf{I}_{nm} - \tfrac{1}{4}h\mathbf{L})\hat{d}^n &= \hat{u}^n + \tfrac{371}{1360}h\mathbf{L}\hat{a}^n - \tfrac{137}{2720}h\mathbf{L}\hat{b}^n + \tfrac{15}{544}h\mathbf{L}\hat{c}^n - \tfrac{6}{85}h\mathbf{N}(\hat{u}^n),\\
&\qquad + \tfrac{42}{85}h\mathbf{N}(\hat{a}^n) + \tfrac{179}{1360}h\mathbf{N}(\hat{b}^n) - \tfrac{15}{272}h\mathbf{N}(\hat{c}^n),\\
(\mathbf{I}_{nm} - \tfrac{1}{4}h\mathbf{L})\hat{e}^n &= \hat{u}^n + \tfrac{25}{24}h\mathbf{L}\hat{a}^n - \tfrac{49}{48}h\mathbf{L}\hat{b}^n + \tfrac{125}{16}h\mathbf{L}\hat{c}^n - \tfrac{85}{12}h\mathbf{L}\hat{d}^n + \tfrac{79}{24}h\mathbf{N}(\hat{a}^n)\\
&\qquad - \tfrac{5}{8}h\mathbf{N}(\hat{b}^n) + \tfrac{25}{2}h\mathbf{N}(\hat{c}^n) - \tfrac{85}{6}h\mathbf{N}(\hat{d}^n),\\
\hat{u}^{n+1} &= \hat{u}^n + \tfrac{25}{24}h\mathbf{L}\hat{a}^n - \tfrac{49}{48}h\mathbf{L}\hat{b}^n + \tfrac{125}{16}h\mathbf{L}\hat{c}^n - \tfrac{85}{12}h\mathbf{L}\hat{d}^n + \tfrac{1}{4}h\mathbf{L}\hat{e}^n + \tfrac{25}{24}h\mathbf{N}(\hat{a}^n)\\
&\qquad - \tfrac{49}{48}h\mathbf{N}(\hat{b}^n) + \tfrac{125}{16}h\mathbf{N}(\hat{c}^n) - \tfrac{85}{12}h\mathbf{N}(\hat{d}^n) + \tfrac{1}{4}h\mathbf{N}(\hat{e}^n).
\end{aligned} \tag{5.61}$$

The LU factorization of the left-hand sides of (5.61) is computed and stored before the time-stepping starts. Note that the most expensive operations in the computation of the internal stages $\hat{a}^n$, $\hat{b}^n$, $\hat{c}^n$, $\hat{d}^n$ and $\hat{e}^n$ are the nonlinear evaluations ($\mathcal{O}(nm \log nm)$ work); see Table 5.1. The other operations, i.e., linear solves and matrix-vector products in the right-hand side (each block being multiplied by $\mathbf{T}_{\sin^2}$), can be carried out in linear time. The computation of $\hat{u}^{n+1}$ from $\hat{u}^n$ requires matrix-vector products of the form $\mathbf{L}v$, which reduce to n dense matrix-vector products $\mathbf{L}_i v$; each can be done in $\mathcal{O}(m)$ operations since $\mathbf{L}_i v = \mathbf{T}_{\sin^2}^{-1}(\mathbf{T}_{\sin^2}\mathbf{L}_i)v$. (The LU factorization of $\mathbf{T}_{\sin^2}$ is also computed and stored before the time-stepping starts.)

5.4 Numerical comparisons

5.4.1 Methodology

To compare time-stepping schemes, we follow the methodology of Chapter 4. We solve a given PDE up to $t = T$ for various time-steps h and a fixed number of grid points.

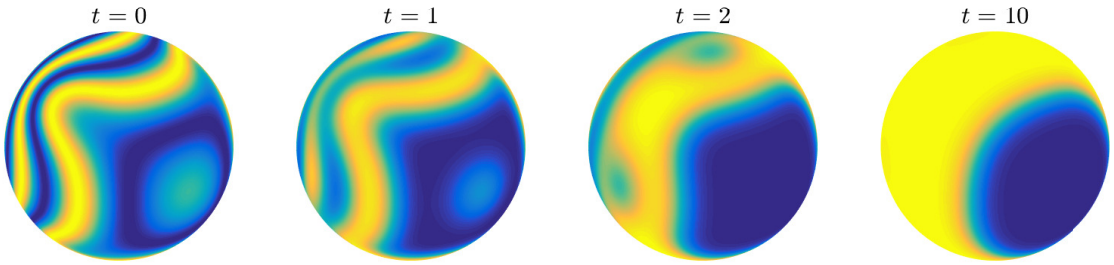

Figure 5.7: *Initial condition* (5.64) *and solution at times* $t = 1, 2, 10$ *of the Allen–Cahn equation* (5.63).

We estimate the "exact" solution $u^{ex}(t = T, \lambda, \theta)$ by using a very small time-step (half the smallest time-step h) and ETDRK4-EIG (the most accurate time-stepping scheme). We then measure the relative L^2-error E at $t = T$ between the computed solution $u(t = T, \lambda, \theta)$ and $u^{ex}(t = T, \lambda, \theta)$,

$$E = \frac{\|u(t = T, \lambda, \theta) - u^{ex}(t = T, \lambda, \theta)\|_2}{\|u^{ex}(t = T, \lambda, \theta)\|_2}. \tag{5.62}$$

For both u and u^{ex} we use $m = n = 256$ grid points. (With these grid sizes, the error due to the spatial discretization is small compared to the error due to the time discretization.) We use $p = 12$ in the CF approximation for ETDRK4-CF and $M = 32$ points to compute the contour integrals for ETDRK4-EIG. We plot (5.62) against relative time-steps h/T and computer times on a pair of graphs.[10]

5.4.2 Results for the diffusive case

The Allen–Cahn equation (4.25) was studied in the ball and on the sphere in, e.g., [55]. It is given by

$$u_t = \epsilon \Delta u + u - u^3, \quad \epsilon \ll 1. \tag{5.63}$$

We take $\epsilon = 10^{-2}$ and

$$u(t = 0, x, y, z) = \cos(\cosh(5xz) - 10y), \tag{5.64}$$

and solve up to $t = 10$. The initial condition and the solution at times $t = 1, 2, 10$ are shown in Figure 5.7. The initial condition quickly converges to a metastable $u \approx \pm 1$ solution (at around $t = 10$) and eventually to the stable constant solution $u = 1$ (at around $t = 60$).

[10]The precomputation of the coefficients of the exponential integrators, the LU factorizations for the IMEX schemes and the starting phase of IMEX-BDF4 are not included in the computing time. Timings were done on a 2.8 GHz Intel i7 machine with 16 GB of RAM.

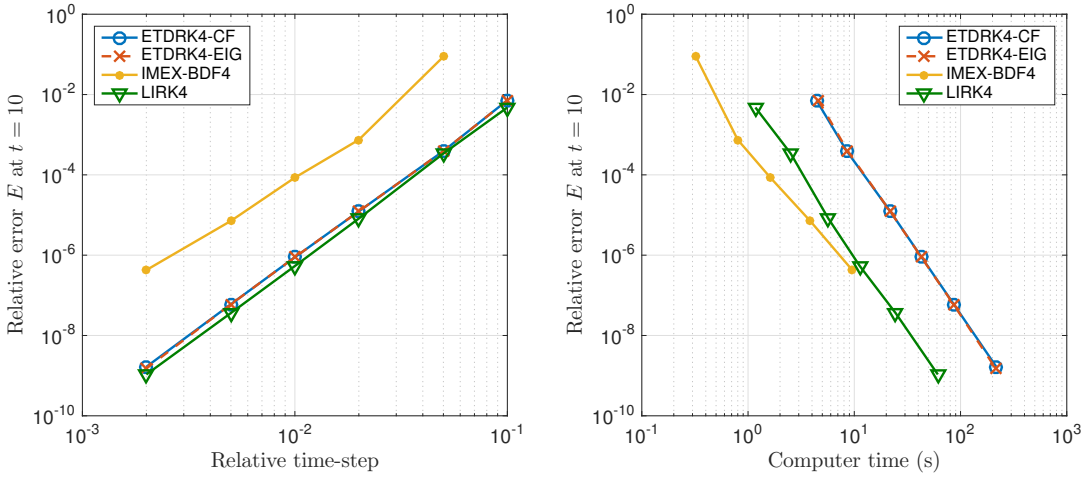

Figure 5.8: *Relative error E at $t = 10$ versus relative time-step and computer time for the Allen–Cahn equation* (5.63).

The results are shown in Figure 5.8. All the schemes are stable for the time-steps we have considered, except IMEX-BDF4 for the largest time-step. The ETDRK4 schemes and LIRK4 have similar accuracy, while IMEX-BDF4 is significantly less accurate. However, IMEX-BDF4 is the most efficient scheme. This can be explained by looking at Table 5.1: IMEX-BDF4 requires very few operations per time-step. Note that in this experiment ETDRK4-CF ($\mathcal{O}(nm \log nm)$ work per time-step) is not more efficient than ETDRK4-EIG ($\mathcal{O}(nm^2)$). Again, the reason can be found in Table 5.1: ETDRK4-CF requires the solution of 108 linear systems per time-step. (For $m = n$ sufficiently large, ETDRK4-CF will be indeed more efficient than ETDRK4-EIG.)

5.4.3 Results for the dispersive case

We consider the NLS equation (4.40) on the sphere, given by

$$u_t = i\Delta u + i|u|^2 u. \tag{5.65}$$

A recent theoretical study of its solutions can be found in [157]. We take

$$u(t = 0, \lambda, \theta) = A\left(\frac{2B^2}{2 - \sqrt{2}\sqrt{2 - B^2}\cos(AB\theta)} - 1\right) + Y_3^3(\lambda, \theta), \tag{5.66}$$

with $A = B = 1$, and solve up to $t = 1$. The initial condition and the real part of the solution at times $t = 0.3, 0.6, 1$ are shown in Figure 5.9. The initial condition is the superposition of two nonlinear waves in which energy concentrates in a localized and oscillatory fashion, a *breather* and a spherical harmonic.

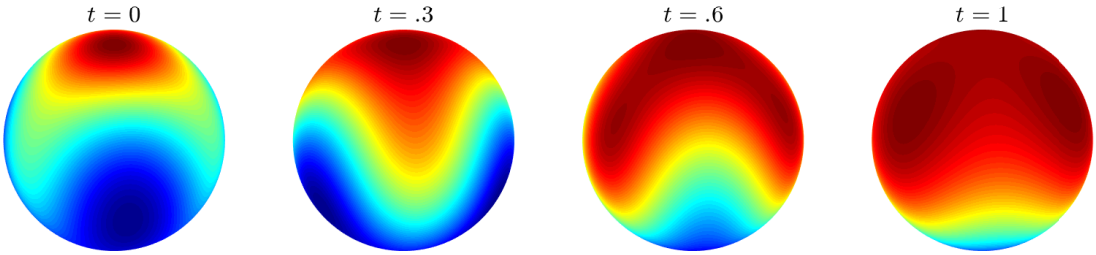

Figure 5.9: *Initial condition* (5.66) *and real part of the solution at times* $t = 0.3, 0.6, 1$ *of the NLS equation* (5.65).

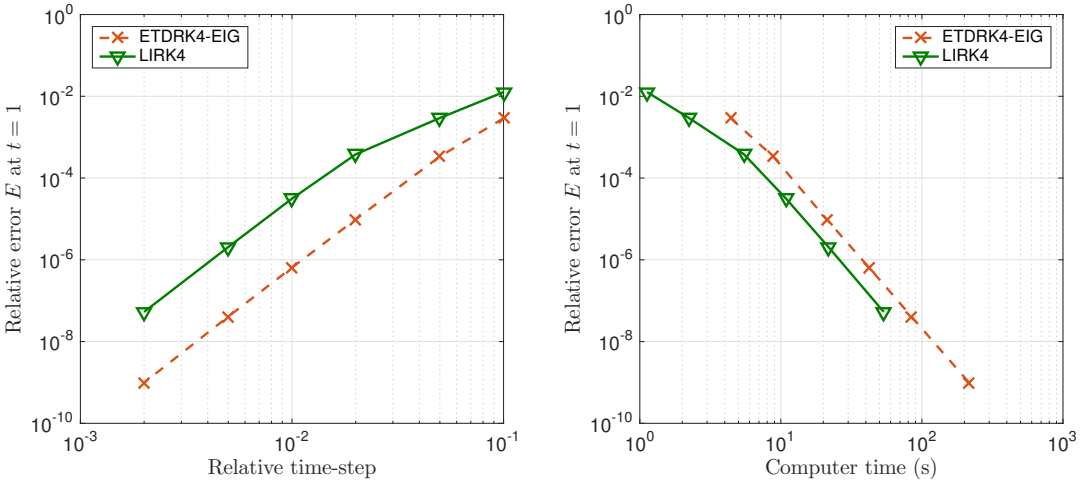

Figure 5.10: *Relative error* E *at* $t = 1$ *versus relative time-step and computer time for the NLS equation* (5.65).

The results are shown in Figure 5.10. Both schemes are stable for the time-steps we have considered. LIRK4 is less accurate than ETDRK4-EIG in this case, but since it has a much lower cost per time-step, it is more efficient. Let us emphasize that timings do not include the precomputation step, which takes significantly longer for ETDRK4-EIG ($\mathcal{O}(nm^3)$ versus $\mathcal{O}(nm)$). Also, since LIRK4 and ETDRK4-EIG have different scaling in $m = n$, the results ultimately depend on the size of the discretization. For example, for $m = n = 128$ instead of 256, ETDRK4-EIG is slightly more efficient.

5.5 Discussion

We have presented algorithms for solving stiff PDEs on the sphere with spectral accuracy in space and fourth-order in time. For the spatial discretization, we have used

Table 5.2: *Computation costs per time-step for a grid with N points and most efficient time-stepping scheme in each case. Diagonal problems were considered in* [84]. *In this chapter, we investigated non-diagonal problems that allow for fast numerical linear algebra (i.e., linear systems can be solved in linear time). In the latter case, IMEX schemes outperform exponential integrators. It is not clear which scheme would perform best in the dense numerical linear algebra case.*

	diffusive PDEs	**dispersive PDEs**
diagonal problems	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log N)$
	ETDRK4	ETDRK4
non-diagonal problems	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log N)$
fast sparse direct solver	IMEX-BDF4	LIRK4
non-diagonal problems	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
dense solver	TBD	TBD

a novel variant of the DFS method in coefficient space. The main advantages of our method are that it is not necessary to numerically impose the pole conditions (5.11), while operating in coefficient space avoids the coordinate singularity without using shifted grids and leads to matrices $\mathbf{L}$ that can be computed and inverted efficiently. We have tested our method with the Poisson and heat equations and obtained excellent results.

For solving nonlinear time-dependent PDEs, we have used IMEX schemes and exponential integrators to circumvent the time-stepping restrictions due to the large eigenvalues of the Laplacian matrix. For diagonal problems, exponential integrators are particularly efficient since the computation and the action of the matrix exponential can trivially be computed in linear time. For problems that allow for fast numerical linear algebra (fast sparse direct solver), as in this chapter, we have given numerical evidence that IMEX schemes outperform exponential integrators. The IMEX-BDF4 time-stepping scheme is remarkably efficient for diffusive PDEs but since it is unstable for dispersive PDEs and needs to be started with another algorithm, it might be preferable to use LIRK4 for both diffusive and dispersive PDEs. For problems that generate dense matrices $\mathbf{L}$, it is not clear which one would perform best. On a grid with N points, both exponential integrators (dense matrix-vector products) and IMEX schemes (triangular systems from precomputed LU factorizations) would have

a $\mathcal{O}(N^2)$ cost per time-step. Note that dense matrices correspond to, e.g., the DFS method applied to (5.1) with highly oscillatory variable coefficients or a Chebyshev discretization in value space of PDEs of the form (5.1) in 1D/2D/3D. However, as recently shown by Aurentz in [11], it seems that all spectral differentiation matrices (even the dense ones!) might allow for fast numerical linear algebra, which would render IMEX schemes more efficient for value-based Chebyshev discretizations—this is a story to be continued. We summarize these observations in Table 5.2.

We have not considered the "sliders" of Fornberg and Driscoll [51, 62]. While this can be an efficient alternative to IMEX schemes and exponential integrators for diagonal problems [84], it is not clear how it can be extended to non-diagonal ones.

Chapter 6

Conclusions

In the preceding chapters, we studied algorithms for solving differential equations with periodicity, and presented new contributions to the fields of numerical analysis and scientific computing. In Chapter 2, we proposed a novel approach for computing multiplication matrices. In Chapter 3, we developed an algorithm to compute choreographies to high accuracy, and found new choreographies on the sphere and on the Poincaré disk. In Chapter 4, we compared 30 exponential integrators applied to 11 PDEs in 1D, 2D and 3D periodic domains. We showed numerical evidence that it is hard do to much better that one of the simplest of these formulas, the ETDRK4 scheme of Cox and Matthews. Finally, in Chapter 5, we studied algorithms for solving stiff PDEs on the sphere with spectral accuracy in space and fourth-order accuracy in time. These were based on a new variant of the DFS method that uses the multiplication matrices of Chapter 2, and standard implicit-explicit time-stepping schemes. We compared implicit-explicit schemes with exponential integrators, and found that implicit-explicit schemes performed better.

There are many ways in which this work could be profitably continued. A possible extension of Chapter 3 would be to classify choreographies by looking at their singularities in the complex plane. Choreographies are smooth solutions on a real time interval, but invariably have singularities in the complex plane. Approximations with rational functions allow one to capture these singularities, and looking at their positions in the complex plane might tell us more about the structure of these very special solutions.

We believe that the techniques described in Chapter 3 can be applied not only to particle dynamics but also to other types of dynamics. Therefore, another extension of this chapter would be the study of choreographies of the n-vortex problem [116], which describes the motion of n vortices, complex potentials associated with the two-dimensional, irrotational and incompressible Euler equations.

The method described in Chapter 5 for solving stiff PDEs on the sphere could be extended to hyperbolic problems, e.g., the barotropic vorticity equation or the shallow water equations. Such problems involve nonlinear differential operators $\mathcal{N}$ with large eigenvalues. While stiffness in the linear part (as in the Allen–Cahn and NLS equations) can be treated by using IMEX schemes or exponential integrators, it is not obvious how to deal with a stiff nonlinear operator. For the barotropic vorticity equation,

$$u_t = \mathcal{N}(u) = -\frac{(\Delta^{-1}u)_\theta}{\sin\theta}u_\lambda + \frac{(\Delta^{-1}u)_\lambda}{\sin\theta}(u_\theta - 2\Omega\sin\theta), \tag{6.1}$$

a possible approach would be to use the EPIRK schemes mentioned in the introduction of Chapter 4 [159], e.g., the EPIRK2 scheme given by

$$\hat{u}^{n+1} = \hat{u}^n + \mathbf{J}^{-1}(e^{h\mathbf{J}} - \mathbf{I})\mathbf{N}(\hat{u}^n), \quad \mathbf{J} = \frac{d\mathbf{N}}{d\hat{u}}(\hat{u}),$$

with Arnoldi iteration for the matrix-vector products involving the Jacobian matrix $\mathbf{J}$ of the discretization $\mathbf{N}$ of $\mathcal{N}$—the cost per time-step would also be $\mathcal{O}(N \log N)$ operations. Another approach would be to use the preconditioned implicit exponential integrators of Luan, Tokman and Rainwater [102], which we mentioned in the conclusion of Chapter 4.

Appendix A

Closed-form expressions for the gradient and the Hessian matrix to compute choreographies

A.1 Gradient and Hessian matrix for planar choreographies

Let $p_n(t)$ be the trigonometric interpolant (2.14) of a choreography $q(t)$ at $N = 2n+1$ equispaced points on $[0, 2\pi]$, and let $\tilde{q}_k = \tilde{u}_k + i\tilde{v}_k$ denote its Fourier coefficients. We can decompose the action (3.8) into the sum $A = A_K + A_U$ with $q(t)$ and $q'(t)$ approximated by $p_n(t)$ and $p'_n(t)$, and

$$A_K = \frac{n}{2}\int_0^{2\pi} \left|p'_n(t) + i\omega p_n(t)\right|^2 dt,\ A_U = \frac{n}{2}\sum_{j=1}^{n-1}\int_0^{2\pi} \left|p_n(t) - p_n\Big(t + \frac{2\pi j}{n}\Big)\right|^{-1} dt. \quad \text{(A.1)}$$

The two terms A_K and A_U depend on the $2N$ real variables $\{\tilde{u}_k, \tilde{v}_k\}$, $|k| \leq (N-1)/2$, via p_n and p'_n.

Let ∇ denote the gradient of the action A with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s, i.e., $\nabla = (\nabla_{\tilde{u}}, \nabla_{\tilde{v}})^T$ with $\nabla_{\tilde{u}} = (\partial/\partial\tilde{u}_k)^T$ and $\nabla_{\tilde{v}} = (\partial/\partial\tilde{v}_k)^T$. We wish to derive a closed-form expression for $\nabla A = \nabla A_K + \nabla A_U$. (The closed-form expression for ∇A_K was derived in (3.10).) Let us write A_U as

$$A_U(\tilde{u}_k, \tilde{v}_k) = \frac{n}{2}\sum_{j=1}^{n-1}\int_0^{2\pi} \frac{dt}{\sqrt{f_j(\tilde{u}_k, \tilde{v}_k, t)}}, \quad \text{(A.2)}$$

with

$$\sqrt{f_j(\tilde{u}_k, \tilde{v}_k, t)} = \left|p_n(t) - p_n\Big(t + \frac{2\pi j}{n}\Big)\right|. \quad \text{(A.3)}$$

Expanding $p_n(t)$ and $p_n(t+2\pi j/n)$ and regrouping real and imaginary parts lead to

$$f_j(\tilde{u}_k,\tilde{v}_k,t)=\left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}a_{k,j}(t)\tilde{u}_k+b_{k,j}(t)\tilde{v}_k\right)^2+\left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}a_{k,j}(t)\tilde{v}_k-b_{k,j}(t)\tilde{u}_k\right)^2 \quad \text{(A.4)}$$

with

$$\begin{aligned}a_{k,j}(t)&=\big[1-\cos(2\pi jk/n)\big]\cos(kt)+\sin(2\pi jk/n)\sin(kt),\\ b_{k,j}(t)&=\big[-1+\cos(2\pi jk/n)\big]\sin(kt)+\sin(2\pi jk/n)\cos(kt).\end{aligned} \quad \text{(A.5)}$$

The partial derivatives of A_U with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s can then be computed with the chain rule,

$$\nabla A_U=\frac{n}{2}\sum_{j=1}^{n-1}\int_0^{2\pi}\nabla\left(\frac{1}{\sqrt{f_j}}\right)dt=-\frac{n}{4}\sum_{j=1}^{n-1}\int_0^{2\pi}\frac{\nabla f_j}{f_j^{3/2}}dt, \quad \text{(A.6)}$$

with

$$\frac{\partial f_j}{\partial\tilde{u}_k}=2\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}}\left(\Big[a_{l,j}(t)a_{k,j}(t)+b_{l,j}(t)b_{k,j}(t)\Big]\tilde{u}_l+\Big[b_{l,j}(t)a_{k,j}(t)-a_{l,j}(t)b_{k,j}(t)\Big]\tilde{v}_l\right), \quad \text{(A.7)}$$

and

$$\frac{\partial f_j}{\partial\tilde{v}_k}=2\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}}\left(\Big[a_{l,j}(t)b_{k,j}(t)-b_{l,j}(t)a_{k,j}(t)\Big]\tilde{u}_l+\Big[b_{l,j}(t)b_{k,j}(t)+a_{l,j}(t)a_{k,j}(t)\Big]\tilde{v}_l\right). \quad \text{(A.8)}$$

Let us now derive the formula for the exact Hessian matrix H,

$$H=\begin{bmatrix}\dfrac{\partial^2 A}{\partial\tilde{u}_l\partial\tilde{u}_k} & \dfrac{\partial^2 A}{\partial\tilde{u}_l\partial\tilde{v}_k}\\[2ex] \dfrac{\partial^2 A}{\partial\tilde{v}_l\partial\tilde{u}_k} & \dfrac{\partial^2 A}{\partial\tilde{v}_l\partial\tilde{v}_k}\end{bmatrix},\quad A=A_K+A_U. \quad \text{(A.9)}$$

H is a $2N\times 2N$ matrix, and each block is $N\times N$. Note that the $\partial^2A/\partial\tilde{v}_l\partial\tilde{u}_k$ block is the transpose of the $\partial^2A/\partial\tilde{u}_l\partial\tilde{v}_k$ block, so we are going to derive formulas for the $\partial^2A/\partial\tilde{u}_l\partial\tilde{u}_k$, $\partial^2A/\partial\tilde{u}_l\partial\tilde{v}_k$, and $\partial^2A/\partial\tilde{v}_l\partial\tilde{v}_k$ derivatives only. The closed-form expressions for the second derivatives of A_K with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s were derived in (3.11). The second derivatives of A_U can be obtained by differentiating (A.6) one more time, e.g.,

$$\frac{\partial^2 A_U}{\partial\tilde{u}_l\partial\tilde{u}_k}=-\frac{n}{4}\sum_{j=1}^{n-1}\int_0^{2\pi}\frac{\frac{\partial^2 f_j}{\partial\tilde{u}_l\partial\tilde{u}_k}f_j-\frac{3}{2}\frac{\partial f_j}{\partial\tilde{u}_l}\frac{\partial f_j}{\partial\tilde{u}_k}}{f_j^{5/2}}dt, \quad \text{(A.10)}$$

with

$$\frac{\partial^2 f_j}{\partial \tilde{u}_l \partial \tilde{u}_k} = 2(a_{l,j}a_{k,j} + b_{l,j}b_{k,j}). \tag{A.11}$$

There are similar formulas for the other derivatives, and these use

$$\frac{\partial^2 f_j}{\partial \tilde{v}_l \partial \tilde{v}_k} = \frac{\partial^2 f_j}{\partial \tilde{u}_l \partial \tilde{u}_k}, \quad \frac{\partial^2 f_j}{\partial \tilde{u}_l \partial \tilde{v}_k} = 2(a_{l,j}b_{k,j} - b_{l,j}a_{k,j}). \tag{A.12}$$

Note that all the second derivatives involving the real and imaginary parts $\tilde{u}_0$ and $\tilde{v}_0$ of the constant term $\tilde{q}_0$ are zeros, i.e.,

$$\frac{\partial^2 A_K}{\partial \tilde{u}_l \partial \tilde{u}_0} = \frac{\partial^2 A_K}{\partial \tilde{v}_l \partial v_0} = \frac{\partial^2 A_K}{\partial \tilde{u}_l \partial v_0} = 0, \tag{A.13}$$

and

$$\frac{\partial^2 A_U}{\partial \tilde{u}_l \partial \tilde{u}_0} = \frac{\partial^2 A_U}{\partial \tilde{v}_l \partial v_0} = \frac{\partial^2 A_U}{\partial \tilde{u}_l \partial v_0} = 0. \tag{A.14}$$

To prove (A.13), take $k = 0$ in (3.11), and to prove (A.14), note that $a_{0,j} = b_{0,j} = 0$. As a consequence, when using Newton's method with the exact Hessian (A.9), one needs to eliminate the lines and columns corresponding to those derivatives to obtain a nonsingular matrix.

A.2 Gradient and Hessian matrix for spherical choreographies

Again, let $p_n(t)$ be the trigonometric interpolant (2.14) of a spherical choreography $q(t)$ at $N = 2n + 1$ equispaced points on $[0, 2\pi]$, and let $\tilde{q}_k = \tilde{u}_k + i\tilde{v}_k$ denote its Fourier coefficients. We can decompose the action (3.26) into the sum of two terms A_K and A_U. The first term comes from the kinetic energy,

$$A_K = \frac{n}{2} \int_0^{2\pi} \left(\frac{2R^2 |p_n'(t) + i\omega p_n(t)|}{R^2 + |p_n(t)|^2} \right)^2 dt, \tag{A.15}$$

while the second term comes from the potential energy,

$$A_U = \frac{n}{2R} \sum_{j=1}^{n-1} \int_0^{2\pi} A_U^j(t) dt, \quad A_U^j(t) = \frac{2R^2 - D_j(t)^2}{D_j(t)\sqrt{4R^2 - D_j(t)^2}}, \tag{A.16}$$

with $D_j(t) = d\big(p_n(t), p_n\big(t + \frac{2\pi j}{n}\big)\big)$.

Again, let $\nabla = (\nabla_u, \nabla_v)^T$ denote the gradient of the action A with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s. Let us first derive the closed-form expression for ∇A_K. A straightforward calculation leads to

$$\nabla A_K = \frac{n}{2} \int_0^{2\pi} \nabla \left[\left(\frac{2R^2 |p_n'(t) + i\omega p_n(t)|}{R^2 + |p_n(t)|^2} \right)^2 \right] dt = 2nR^4 \int_0^{2\pi} \frac{h\nabla g - g\nabla h}{h^2} dt, \tag{A.17}$$

with

$$g(\tilde{u}_k, \tilde{v}_k, t) = |p_n'(t) + i\omega p_n(t)|^2, \quad h(\tilde{u}_k, \tilde{v}_k, t) = \big(R^2 + |p_n(t)|^2\big)^2. \tag{A.18}$$

The functions g and h are given by

$$g = \Bigg[\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} (k+\omega)\Big(\tilde{u}_k \sin(kt) + \tilde{v}_k \cos(kt)\Big)\Bigg]^2 + \Bigg[\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} (k+\omega)\Big(\tilde{u}_k \cos(kt) - \tilde{v}_k \sin(kt)\Big)\Bigg]^2, \tag{A.19}$$

and

$$h = \Bigg[R^2 + \Bigg(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \tilde{u}_k \sin(kt) + \tilde{v}_k \cos(kt)\Bigg)^2 + \Bigg(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \tilde{u}_k \cos(kt) - \tilde{v}_k \sin(kt)\Bigg)^2\Bigg]^2. \tag{A.20}$$

Their partial derivatives are given by the formulas

$$\begin{aligned}
\frac{\partial g}{\partial \tilde{u}_k} &= 2(k+\omega) \sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} (l+\omega)\Big(\tilde{u}_l \cos\big((k-l)t\big) + \tilde{v}_l \sin\big((k-l)t\big)\Big), \\
\frac{\partial g}{\partial \tilde{v}_k} &= 2(k+\omega) \sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} (l+\omega)\Big(\tilde{u}_l \sin\big((l-k)t\big) + \tilde{v}_l \cos\big((l-k)t\big)\Big),
\end{aligned} \tag{A.21}$$

and

$$\begin{aligned}
\frac{\partial h}{\partial \tilde{u}_k} &= 4\sqrt{h} \sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} \tilde{u}_l \cos\big((k-l)t\big) + \tilde{v}_l \sin\big((k-l)t\big), \\
\frac{\partial h}{\partial \tilde{v}_k} &= 4\sqrt{h} \sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} \tilde{u}_l \sin\big((l-k)t\big) + \tilde{v}_l \cos\big((l-k)t\big).
\end{aligned} \tag{A.22}$$

Let us now derive the closed-form expression for ∇A_U, with

$$\nabla A_U = \frac{n}{2R} \sum_{j=1}^{n-1} \int_0^{2\pi} \nabla A_U^j(t)dt, \quad \nabla A_U^j(t) = \frac{-8R^4 \nabla D_j(t)}{D_j(t)^2\big[4R^2 - D_j(t)^2\big]^{3/2}}. \tag{A.23}$$

Let us write

$$D_j(t) = \frac{2R^2\big|p_n(t) - p_n(t+\frac{2\pi j}{n})\big|}{\sqrt{\Big(R^2 + \big|p_n(t)\big|^2\Big)\Big(R^2 + \big|p_n(t+\frac{2\pi j}{n})\big|^2\Big)}} = 2R^2 \frac{\sqrt{f_j}}{r_0 r_j}, \tag{A.24}$$

with $\sqrt{f_j(\tilde{u}_k, \tilde{v}_k, t)} = |p_n(t) - p_n(t + 2\pi j/n)|$ as defined in (A.4)–(A.5), and r_j defined by

$$r_j(\tilde{u}_k, \tilde{v}_k, t) = \sqrt{R^2 + \Big|p_n\Big(t + \frac{2\pi j}{n}\Big)\Big|^2}. \tag{A.25}$$

We can rewrite r_j as

$$r_j = \sqrt{R^2 + \left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{k,j}(t)\tilde{u}_k + d_{k,j}(t)\tilde{v}_k\right)^2 + \left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{k,j}(t)\tilde{v}_k - d_{k,j}(t)\tilde{u}_k\right)^2}, \quad \text{(A.26)}$$

with

$$\begin{aligned} c_{k,j}(t) &= \cos(2\pi kj/n)\cos(kt) - \sin(2\pi kj/n)\sin(kt), \\ d_{k,j}(t) &= -\cos(2\pi kj/n)\sin(kt) - \sin(2\pi kj/n)\cos(kt). \end{aligned} \quad \text{(A.27)}$$

This leads to

$$\nabla D_j(t) = 2R^2 \frac{r_0 r_j \nabla f_j/2 - f_j \nabla(r_0 r_j)}{r_0^2 r_j^2 \sqrt{f_j}}. \quad \text{(A.28)}$$

The derivatives of f_j with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s are given by (A.7)–(A.8), while the derivatives of r_j are given by

$$\begin{aligned} \frac{\partial r_j}{\partial \tilde{u}_k} &= \frac{1}{r_j}\left[c_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{u}_l + d_{l,j}(t)\tilde{v}_l\right) - d_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{v}_l - d_{l,j}(t)\tilde{u}_l\right)\right], \\ \frac{\partial r_j}{\partial \tilde{v}_k} &= \frac{1}{r_j}\left[d_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{u}_l + d_{l,j}(t)\tilde{v}_l\right) + c_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{v}_l - d_{l,j}(t)\tilde{u}_l\right)\right]. \end{aligned} \quad \text{(A.29)}$$

Let us now derive the formula for the exact Hessian H on the sphere. Differentiating (A.17) gives the second derivatives of A_K with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s, e.g.,

$$\frac{\partial^2 A_K}{\partial \tilde{u}_l \partial \tilde{u}_k} = 2nR^4 \int_0^{2\pi} \frac{\frac{\partial p_k}{\partial \tilde{u}_l} h - 2\frac{\partial h}{\partial \tilde{u}_l} p_k}{h^3} dt, \quad p_k = h\frac{\partial g}{\partial \tilde{u}_k} - g\frac{\partial h}{\partial \tilde{u}_k}, \quad \text{(A.30)}$$

with

$$\frac{\partial p_k}{\partial \tilde{u}_l} = \frac{\partial h}{\partial \tilde{u}_l}\frac{\partial g}{\partial \tilde{u}_k} + h\frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{u}_k} - \frac{\partial g}{\partial \tilde{u}_l}\frac{\partial h}{\partial \tilde{u}_k} - g\frac{\partial^2 h}{\partial \tilde{u}_l \partial \tilde{u}_k}, \quad \text{(A.31)}$$

and

$$\frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{u}_k} = 2(k+\omega)(l+\omega)\cos\big((k-l)t\big), \ \frac{\partial^2 h}{\partial \tilde{u}_l \partial \tilde{u}_k} = \frac{1}{2h}\frac{\partial h}{\partial \tilde{u}_l}\frac{\partial h}{\partial \tilde{u}_k} + 4\sqrt{h}\cos\big((k-l)t\big). \quad \text{(A.32)}$$

There are similar formulas for the other derivatives, and these use

$$\frac{\partial^2 g}{\partial \tilde{v}_l \partial \tilde{v}_k} = \frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{u}_k}, \quad \frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{v}_k} = 2(k+\omega)(l+\omega)\sin\big((l-k)t\big), \quad \text{(A.33)}$$

and

$$\begin{aligned}\frac{\partial^2 h}{\partial\tilde{v}_l\partial\tilde{v}_k} &= \frac{1}{2h}\frac{\partial h}{\partial\tilde{v}_l}\frac{\partial h}{\partial\tilde{v}_k} + 4\sqrt{h}\cos\big((k-l)t\big),\\ \frac{\partial^2 h}{\partial\tilde{u}_l\partial\tilde{v}_k} &= \frac{1}{2h}\frac{\partial h}{\partial\tilde{u}_l}\frac{\partial h}{\partial\tilde{v}_k} + 4\sqrt{h}\sin\big((l-k)t\big).\end{aligned} \tag{A.34}$$

The second derivatives of A_U can be obtained by differentiating (A.23), e.g.,

$$\frac{\partial^2 A_U^j}{\partial\tilde{u}_l\partial\tilde{u}_k} = \frac{16R^4\frac{\partial D_j}{\partial\tilde{u}_l}\frac{\partial D_j}{\partial\tilde{u}_k}}{D_j^3\big[4R^2-D_j^2\big]^{3/2}} - \frac{24R^4\frac{\partial D_j}{\partial\tilde{u}_l}\frac{\partial D_j}{\partial\tilde{u}_k}}{D_j\big[4R^2-D_j^2\big]^{5/2}} - \frac{8R^4\frac{\partial^2 D_j}{\partial\tilde{u}_l\partial\tilde{u}_k}}{D_j^2\big[4R^2-D_j^2\big]^{3/2}}, \tag{A.35}$$

with

$$\frac{\partial^2 D_j}{\partial\tilde{u}_l\partial\tilde{u}_k} = 2R^2\frac{r_0r_jf_j\frac{\partial q_k}{\partial\tilde{u}_l} - q_k\big(2r_0f_j\frac{\partial r_j}{\partial\tilde{u}_l} + 2r_jf_j\frac{\partial r_0}{\partial\tilde{u}_l} + \frac{1}{2}r_0r_j\frac{\partial f_j}{\partial\tilde{u}_l}\big)}{r_0^3r_j^3f_j^{3/2}}, \tag{A.36}$$

and

$$q_k = \frac{1}{2}r_0r_j\frac{\partial f_j}{\partial\tilde{u}_k} - f_jr_j\frac{\partial r_0}{\partial\tilde{u}_k} - f_jr_0\frac{\partial r_j}{\partial\tilde{u}_k}. \tag{A.37}$$

The second derivatives (A.36) involve the derivatives of (A.37) with respect to the $\tilde{u}_l$'s—the chain rule leads to nine terms, and only two of them are new, $\partial^2 r_0/\partial\tilde{u}_l\partial\tilde{u}_k$ and $\partial^2 r_j/\partial\tilde{u}_l\partial\tilde{u}_k$. These are given by

$$\frac{\partial^2 r_j}{\partial\tilde{u}_l\partial\tilde{u}_k} = \frac{c_{k,j}c_{l,j} + d_{k,j}d_{l,j} - \frac{\partial r_j}{\partial\tilde{u}_l}\frac{\partial r_j}{\partial\tilde{u}_k}}{r_j}. \tag{A.38}$$

Similar formulas can be derived for the other derivatives, and these use

$$\frac{\partial^2 r_j}{\partial\tilde{v}_l\partial\tilde{v}_k} = \frac{c_{k,j}c_{l,j} + d_{k,j}d_{l,j} - \frac{\partial r_j}{\partial\tilde{v}_l}\frac{\partial r_j}{\partial\tilde{v}_k}}{r_j}, \quad \frac{\partial^2 r_j}{\partial\tilde{u}_l\partial\tilde{v}_k} = \frac{c_{l,j}d_{k,j} - c_{k,j}d_{l,j} - \frac{\partial r_j}{\partial\tilde{u}_l}\frac{\partial r_j}{\partial\tilde{u}_k}}{r_j}. \tag{A.39}$$

Note that, as in the plane, all the second derivatives involving the real and imaginary parts $\tilde{u}_0$ and $\tilde{v}_0$ of the constant terms are zeros. As a consequence, when using Newton's method with the exact Hessian, one needs to eliminate the lines and columns corresponding to those derivatives to obtain a nonsingular matrix.

A.3 Gradient and Hessian matrix for hyperbolic choreographies

The expressions for the gradient and the Hessian matrix for hyperbolic choreographies are very similar to those obtained for spherical choreographies. We decompose the

action (3.50) into the sum of two terms A_K and A_U. The first term comes from the kinetic energy,

$$A_K = \frac{n}{2}\int_0^{2\pi}\left(\frac{2R^2|p_n'(t)+i\omega p_n(t)|}{R^2-|p_n(t)|^2}\right)^2 dt, \tag{A.40}$$

while the second term comes from the potential energy,

$$A_U = \frac{n}{2R}\sum_{j=1}^{n-1}\int_0^{2\pi} A_U^j(t)dt, \quad A_U^j(t) = \frac{2R^2+D_j(t)^2}{D_j(t)\sqrt{4R^2+D_j(t)^2}}, \tag{A.41}$$

with $D_j(t) = d\big(p_n(t), p_n\big(t+\frac{2\pi j}{n}\big)\big)$.

Again, let $\nabla = (\nabla_u, \nabla_v)^T$ denote the gradient of the action A with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s. Let us first derive the closed-form expression for ∇A_K. A straightforward calculation leads to

$$\nabla A_K = \frac{n}{2}\int_0^{2\pi}\nabla\left[\left(\frac{2R^2|p_n'(t)+i\omega p_n(t)|}{R^2-|p_n(t)|^2}\right)^2\right]dt = 2nR^4\int_0^{2\pi}\frac{h\nabla g - g\nabla h}{h^2}dt, \tag{A.42}$$

with

$$g(\tilde{u}_k,\tilde{v}_k,t) = |p_n'(t)+i\omega p_n(t)|^2, \quad h(\tilde{u}_k,\tilde{v}_k,t) = \big(R^2-|p_n(t)|^2\big)^2. \tag{A.43}$$

The functions g and h are given by

$$g = \left[\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}(k+\omega)\Big(\tilde{u}_k\sin(kt)+\tilde{v}_k\cos(kt)\Big)\right]^2 + \left[\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}(k+\omega)\Big(\tilde{u}_k\cos(kt)-\tilde{v}_k\sin(kt)\Big)\right]^2, \tag{A.44}$$

and

$$h = \left[R^2 - \left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}\tilde{u}_k\sin(kt)+\tilde{v}_k\cos(kt)\right)^2 - \left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}\tilde{u}_k\cos(kt)-\tilde{v}_k\sin(kt)\right)^2\right]^2. \tag{A.45}$$

Their partial derivatives are given by the formulas

$$\begin{aligned}
\frac{\partial g}{\partial\tilde{u}_k} &= 2(k+\omega)\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}}(l+\omega)\Big(\tilde{u}_l\cos\big((k-l)t\big)+\tilde{v}_l\sin\big((k-l)t\big)\Big),\\
\frac{\partial g}{\partial\tilde{v}_k} &= 2(k+\omega)\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}}(l+\omega)\Big(\tilde{u}_l\sin\big((l-k)t\big)+\tilde{v}_l\cos\big((l-k)t\big)\Big),
\end{aligned} \tag{A.46}$$

and

$$\begin{aligned}
\frac{\partial h}{\partial\tilde{u}_k} &= -4\sqrt{h}\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}}\tilde{u}_l\cos\big((k-l)t\big)+\tilde{v}_l\sin\big((k-l)t\big),\\
\frac{\partial h}{\partial\tilde{v}_k} &= -4\sqrt{h}\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}}\tilde{u}_l\sin\big((l-k)t\big)+\tilde{v}_l\cos\big((l-k)t\big).
\end{aligned} \tag{A.47}$$

Let us now derive the closed-form expression for ∇A_U,

$$\nabla A_U = \frac{n}{2R}\sum_{j=1}^{n-1}\int_0^{2\pi} \nabla A_U^j(t)dt, \quad \nabla A_U^j(t) = \frac{-8R^4\nabla D_j(t)}{D_j(t)^2\big[4R^2 + D_j(t)^2\big]^{3/2}}. \tag{A.48}$$

Let us write

$$D_j(t) = \frac{2R^2\big|p_n(t) - p_n(t+\frac{2\pi j}{n})\big|}{\sqrt{\Big(R^2 + \big|p_n(t)\big|^2\Big)\Big(R^2 + \big|p_n(t+\frac{2\pi j}{n})\big|^2\Big)}} = 2R^2\frac{\sqrt{f_j}}{r_0 r_j}, \tag{A.49}$$

with $\sqrt{f_j(\tilde{u}_k, \tilde{v}_k, t)} = |p_n(t) - p_n(t+2\pi j/n)|$ as defined in (A.4)–(A.5), and r_j defined by

$$r_j(\tilde{u}_k, \tilde{v}_k, t) = \sqrt{R^2 - \Big|p_n\Big(t + \frac{2\pi j}{n}\Big)\Big|^2}. \tag{A.50}$$

Let us write r_j as

$$r_j = \sqrt{R^2 - \left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{k,j}(t)\tilde{u}_k + d_{k,j}(t)\tilde{v}_k\right)^2 - \left(\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{k,j}(t)\tilde{v}_k - d_{k,j}(t)\tilde{u}_k\right)^2}, \tag{A.51}$$

with

$$\begin{aligned} c_{k,j}(t) &= \cos(2\pi kj/n)\cos(kt) - \sin(2\pi kj/n)\sin(kt), \\ d_{k,j}(t) &= -\cos(2\pi kj/n)\sin(kt) - \sin(2\pi kj/n)\cos(kt). \end{aligned} \tag{A.52}$$

This leads to

$$\nabla D_j(t) = 2R^2\frac{r_0 r_j \nabla f_j/2 - f_j\nabla(r_0 r_j)}{r_0^2 r_j^2\sqrt{f_j}}. \tag{A.53}$$

The derivatives of f_j with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s are given by (A.7)–(A.8), while the derivatives of r_j are given by

$$\begin{aligned} \frac{\partial r_j}{\partial \tilde{u}_k} &= -\frac{1}{r_j}\left[c_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{u}_l + d_{l,j}(t)\tilde{v}_l\right) - d_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{v}_l - d_{l,j}(t)\tilde{u}_l\right)\right], \\ \frac{\partial r_j}{\partial \tilde{v}_k} &= -\frac{1}{r_j}\left[d_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{u}_l + d_{l,j}(t)\tilde{v}_l\right) + c_{k,j}(t)\left(\sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} c_{l,j}(t)\tilde{v}_l - d_{l,j}(t)\tilde{u}_l\right)\right]. \end{aligned} \tag{A.54}$$

Let us now derive the formula for the exact Hessian H. Differentiating (A.42) gives the second derivatives of A_K with respect to the $\tilde{u}_k$'s and $\tilde{v}_k$'s, e.g.,

$$\frac{\partial^2 A_K}{\partial\tilde{u}_l\partial\tilde{u}_k} = 2nR^4\int_0^{2\pi}\frac{\frac{\partial p_k}{\partial\tilde{u}_l}h - 2\frac{\partial h}{\partial\tilde{u}_l}p_k}{h^3}dt, \quad p_k = h\frac{\partial g}{\partial\tilde{u}_k} - g\frac{\partial h}{\partial\tilde{u}_k}, \tag{A.55}$$

with

$$\frac{\partial p_k}{\partial \tilde{u}_l} = \frac{\partial h}{\partial \tilde{u}_l}\frac{\partial g}{\partial \tilde{u}_k} + h\frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{u}_k} - \frac{\partial g}{\partial \tilde{u}_l}\frac{\partial h}{\partial \tilde{u}_k} - g\frac{\partial^2 h}{\partial \tilde{u}_l \partial \tilde{u}_k}, \tag{A.56}$$

and

$$\frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{u}_k} = 2(k+\omega)(l+\omega)\cos\big((k-l)t\big),\ \frac{\partial^2 h}{\partial \tilde{u}_l \partial \tilde{u}_k} = \frac{1}{2h}\frac{\partial h}{\partial \tilde{u}_l}\frac{\partial h}{\partial \tilde{u}_k} - 4\sqrt{h}\cos\big((k-l)t\big). \tag{A.57}$$

There are similar formulas for the other derivatives, and these use

$$\frac{\partial^2 g}{\partial \tilde{v}_l \partial \tilde{v}_k} = \frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{u}_k}, \quad \frac{\partial^2 g}{\partial \tilde{u}_l \partial \tilde{v}_k} = 2(k+\omega)(l+\omega)\sin\big((l-k)t\big), \tag{A.58}$$

and

$$\begin{aligned} \frac{\partial^2 h}{\partial \tilde{v}_l \partial \tilde{v}_k} &= \frac{1}{2h}\frac{\partial h}{\partial \tilde{v}_l}\frac{\partial h}{\partial \tilde{v}_k} - 4\sqrt{h}\cos\big((k-l)t\big), \\ \frac{\partial^2 h}{\partial \tilde{u}_l \partial \tilde{v}_k} &= \frac{1}{2h}\frac{\partial h}{\partial \tilde{u}_l}\frac{\partial h}{\partial \tilde{v}_k} - 4\sqrt{h}\sin\big((l-k)t\big). \end{aligned} \tag{A.59}$$

The second derivatives of A_U can be obtained by differentiating (A.48), e.g.,

$$\frac{\partial^2 A_U^j}{\partial \tilde{u}_l \partial \tilde{u}_k} = \frac{16R^4 \frac{\partial D_j}{\partial \tilde{u}_l}\frac{\partial D_j}{\partial \tilde{u}_k}}{D_j^3\big[4R^2 + D_j^2\big]^{3/2}} + \frac{24R^4 \frac{\partial D_j}{\partial \tilde{u}_l}\frac{\partial D_j}{\partial \tilde{u}_k}}{D_j\big[4R^2 + D_j^2\big]^{5/2}} - \frac{8R^4 \frac{\partial^2 D_j}{\partial \tilde{u}_l \partial \tilde{u}_k}}{D_j^2\big[4R^2 + D_j^2\big]^{3/2}}, \tag{A.60}$$

with

$$\frac{\partial^2 D_j}{\partial \tilde{u}_l \partial \tilde{u}_k} = 2R^2 \frac{r_0 r_j f_j \frac{\partial q_k}{\partial \tilde{u}_l} - q_k\big(2r_0 f_j \frac{\partial r_j}{\partial \tilde{u}_l} + 2r_j f_j \frac{\partial r_0}{\partial \tilde{u}_l} + \frac{1}{2} r_0 r_j \frac{\partial f_j}{\partial \tilde{u}_l}\big)}{r_0^3 r_j^3 f_j^{3/2}}, \tag{A.61}$$

and

$$q_k = \frac{1}{2} r_0 r_j \frac{\partial f_j}{\partial \tilde{u}_k} - f_j r_j \frac{\partial r_0}{\partial \tilde{u}_k} - f_j r_0 \frac{\partial r_j}{\partial \tilde{u}_k}. \tag{A.62}$$

The second derivatives (A.61) involve the derivatives of (A.62) with respect to the $\tilde{u}_l$'s—the chain rule leads to nine terms, and only two of them are new, $\partial^2 r_0/\partial \tilde{u}_l \partial \tilde{u}_k$ and $\partial^2 r_j/\partial \tilde{u}_l \partial \tilde{u}_k$. These are given by

$$\frac{\partial^2 r_j}{\partial \tilde{u}_l \partial \tilde{u}_k} = \frac{-c_{k,j}c_{l,j} - d_{k,j}d_{l,j} - \frac{\partial r_j}{\partial \tilde{u}_l}\frac{\partial r_j}{\partial \tilde{u}_k}}{r_j}. \tag{A.63}$$

Similar formulas can be derived for the other derivatives, and these use

$$\frac{\partial^2 r_j}{\partial \tilde{v}_l \partial \tilde{v}_k} = \frac{-c_{k,j}c_{l,j} - d_{k,j}d_{l,j} - \frac{\partial r_j}{\partial \tilde{v}_l}\frac{\partial r_j}{\partial \tilde{v}_k}}{r_j}, \tag{A.64}$$

and

$$\frac{\partial^2 r_j}{\partial \tilde{u}_l \partial \tilde{v}_k} = \frac{-c_{l,j}d_{k,j} + c_{k,j}d_{l,j} - \frac{\partial r_j}{\partial \tilde{u}_l}\frac{\partial r_j}{\partial \tilde{u}_k}}{r_j}. \tag{A.65}$$

Note that, as in the plane and on the sphere, all the second derivatives involving the real and imaginary parts $\tilde{u}_0$ and $\tilde{v}_0$ of the constant terms are zeros. One can use the same trick to obtain a nonsingular matrix.

Appendix B

Butcher tableaux of the integrators

We present here a complete list of the Butcher tableaux for the exponential integrators featured in Chapter 4. The format for the tableaux is given in Table 4.1. Within each tableau any unspecified coefficient is defined directly below. As we mentioned in Chapter 4, since almost all of the exponential integrators satisfy the summation property (4.9), it is notationally convenient to incorporate this condition by filling the first entry in the rows of A and B with a dot on the understanding that these method coefficients are prescribed through this.

B.1 ETD Adams–Bashforth integrators

Table B.1: *ABNørsett*4.

	$B_1 = \cdot$	$V_1 = -3\varphi_2 - 5\varphi_3 - 3\varphi_4 \quad V_2 = \frac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4 \quad V_3 = -\frac{1}{3}\varphi_2 - \varphi_3 - \varphi_4$

Table B.2: *ABNørsett*5.

	$B_1 = \cdot$	$V_1 = -(4\varphi_2 + \frac{26}{3}\varphi_3 + 9\varphi_4 + 4\varphi_5)$	$V_2 = 3\varphi_2 + \frac{19}{2}\varphi_3 + 12\varphi_4 + 6\varphi_5$	V_3	V_4

with

$$V_3 = -(\tfrac{4}{3}\varphi_2 + \tfrac{14}{3}\varphi_3 + 7\varphi_4 + 4\varphi_5), \quad V_4 = \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5. \tag{B.1}$$

Table B.3: *ABNørsett*6.

	$B_1 = \cdot$	$V_1 = -(5\varphi_2 + \frac{77}{6}\varphi_3 + \frac{71}{4}\varphi_4 + 14\varphi_5 + 5\varphi_6)$	V_2	V_3	V_4 V_5

with

$$
\begin{aligned}
V_2 &= 5\varphi_2 + \tfrac{107}{6}\varphi_3 + \tfrac{59}{2}\varphi_4 + 26\varphi_5 + 10\varphi_6,\\
V_3 &= -(\tfrac{10}{3}\varphi_2 + 13\varphi_3 + \tfrac{49}{2}\varphi_4 + 24\varphi_5 + 10\varphi_6),\\
V_4 &= \tfrac{5}{4}\varphi_2 + \tfrac{61}{12}\varphi_3 + \tfrac{41}{4}\varphi_4 + 11\varphi_5 + 5\varphi_6,\\
V_5 &= -(\tfrac{1}{5}\varphi_2 + \tfrac{5}{6}\varphi_3 + \tfrac{7}{4}\varphi_4 + 2\varphi_5 + \varphi_6).
\end{aligned}
\tag{B.2}
$$

B.2 ETD Runge–Kutta integrators

Table B.4: *ETDRK*4.

$\frac{1}{2}$	$A_{2,1} = \cdot$			
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \psi_{1,2}$		
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = 2\psi_{1,2}$	
	$B_1 = \cdot$	$B_2 = 2\varphi_2 - 4\varphi_3$	$B_3 = 2\varphi_2 - 4\varphi_3$	$B_4 = -\varphi_2 + 4\varphi_3$

Table B.5: *Friedli.*

$\frac{1}{2}$	$A_{2,1} = \cdot$			
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = 2\psi_{2,2}$		
1	$A_{4,1} = \cdot$	$A_{4,2} = -\frac{26}{25}\varphi_1 + \frac{2}{25}\varphi_2$	$A_{4,3} = \frac{26}{25}\varphi_1 + \frac{48}{25}\varphi_2$	
	$B_1 = \cdot$	$B_2 = 0$	$B_3 = 4\varphi_2 - 8\varphi_3$	$B_4 = -\varphi_2 + 4\varphi_3$

Table B.6: *Krogstad.*

$\frac{1}{2}$	$A_{2,1} = \cdot$			
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = 4\psi_{2,2}$		
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = 2\varphi_2$	
	$B_1 = \cdot$	$B_2 = 2\varphi_2 - 4\varphi_3$	$B_3 = 2\varphi_2 - 4\varphi_3$	$B_4 = -\varphi_2 + 4\varphi_3$

Table B.7: *Minchev.*

$\frac{1}{2}$	$A_{2,1} = \cdot$				
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{4}{25}\psi_{1,2} + \frac{24}{25}\psi_{2,2}$			
1	$A_{4,1} = \cdot$	$A_{4,2} = \frac{21}{5}\varphi_2 - \frac{108}{5}\varphi_3$	$A_{4,3} = \frac{1}{20}\varphi_1 - \frac{33}{10}\varphi_2 + \frac{123}{5}\varphi_3$		
	$B_1 = \cdot$	$B_2 = -\frac{1}{10}\varphi_1 + \frac{1}{5}\varphi_2 - 4\varphi_3 + 12\varphi_4$	$B_3 = \frac{1}{30}\varphi_1 + \frac{23}{5}\varphi_2 - 8\varphi_3 - 4\varphi_4$	B_4	

with

$$B_4 = \frac{1}{30}\varphi_1 - \frac{7}{5}\varphi_2 + 6\varphi_3 - 4\varphi_4. \tag{B.3}$$

Table B.8: *Strehmel–Weiner.*

$\frac{1}{2}$	$A_{2,1} = \cdot$			
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = 2\psi_{2,2}$		
1	$A_{4,1} = \cdot$	$A_{4,2} = -2\varphi_2$	$A_{4,3} = 4\varphi_2$	
	$B_1 = \cdot$	$B_2 = 0$	$B_3 = 4\varphi_2 - 8\varphi_3$	$B_4 = -\varphi_2 + 4\varphi_3$

Table B.9: *Hochbruck–Ostermann.*

$\frac{1}{2}$	$A_{2,1} = \cdot$				
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = 4\psi_{2,2}$			
1	$A_{4,1} = \cdot$	$A_{4,2} = \varphi_2$	$A_{4,3} = \varphi_2$		
$\frac{1}{2}$	$A_{5,1} = \cdot$	$A_{5,2} = \frac{1}{4}\varphi_2 - \varphi_3 + 2\psi_{2,2} - 4\psi_{3,2}$	$A_{5,3} = A_{5,2}$	$A_{5,4} = \psi_{2,2} - A_{5,2}$	
	$B_1 = \cdot$	$B_2 = 0$	$B_3 = 0$	$B_4 = -\varphi_2 + 4\varphi_3$	B_5

with

$$B_5 = 4\varphi_2 - 8\varphi_3. \tag{B.4}$$

Table B.10: *EXPRK5S8*.

$\frac{1}{2}$	$A_{2,1}=\cdot$							
$\frac{1}{2}$	$A_{3,1}=\cdot$	$A_{3,2}=2\psi_{2,2}$						
$\frac{1}{4}$	$A_{4,1}=\cdot$	$A_{4,2}=0$	$A_{4,3}=2\psi_{2,4}$					
$\frac{1}{2}$	$A_{5,1}=\cdot$	$A_{5,2}=0$	$A_{5,3}=-2\psi_{2,2}+16\psi_{3,2}$	$A_{5,4}$				
$\frac{1}{5}$	$A_{6,1}=\cdot$	$A_{6,2}=0$	$A_{6,3}=0$	$A_{6,4}$	$A_{6,5}$			
$\frac{2}{3}$	$A_{7,1}=\cdot$	$A_{7,2}=0$	$A_{7,3}=0$	$A_{7,4}$	$A_{7,5}$	$A_{7,6}$		
1	$A_{8,1}=\cdot$	$A_{8,2}=0$	$A_{8,3}=0$	$A_{8,4}=0$	$A_{8,5}$	$A_{8,6}$	$A_{8,7}$	
	$B_1=\cdot$	$B_2=0$	$B_3=0$	$B_4=0$	$B_5=0$	B_6	B_7	B_8

with

$$
\begin{aligned}
&A_{5,4}=8\psi_{2,2}-32\psi_{3,2}, && A_{6,4}=8\psi_{2,6}-32\psi_{3,6}\\
&A_{6,5}=-2\psi_{2,6}+16\psi_{3,6}, && A_{7,4}=-\tfrac{125}{162}A_{6,4}\\
&A_{7,5}=\tfrac{125}{1944}A_{6,4}-\tfrac{4}{3}\psi_{2,7}+\tfrac{40}{3}\psi_{3,7}, && A_{7,6}=\tfrac{3125}{3888}A_{6,4}+\tfrac{25}{3}\psi_{2,7}-\tfrac{100}{3}\psi_{3,7},\\
&A_{8,5}=-\tfrac{16}{3}\varphi_2+\tfrac{208}{3}\varphi_3-40\Phi, && A_{8,6}=\tfrac{250}{21}\varphi_2-\tfrac{250}{3}\varphi_3+\tfrac{250}{7}\Phi,\\
&A_{8,7}=\tfrac{27}{14}\varphi_2-27\varphi_3+\tfrac{135}{7}\Phi, && B_6=\tfrac{125}{14}\varphi_2-\tfrac{625}{14}\varphi_3+\tfrac{1125}{14}\varphi_4,\\
&B_7=-\tfrac{27}{14}\varphi_2+\tfrac{162}{7}\varphi_3-\tfrac{405}{7}\varphi_4, && B_8=\tfrac{1}{2}\varphi_2-\tfrac{13}{2}\varphi_3+\tfrac{45}{2}\varphi_4,
\end{aligned}
\tag{B.5}
$$

and

$$
\Phi=\tfrac{5}{32}A_{6,4}-\tfrac{25}{28}\psi_{2,6}+\tfrac{81}{175}\psi_{2,7}-\tfrac{162}{25}\psi_{3,7}+\tfrac{150}{7}\psi_{4,6}+\tfrac{972}{35}\psi_{4,7}+6\varphi_4. \tag{B.6}
$$

B.3 Lawson integrators

Table B.11: *ABLawson4*.

$B_1=\frac{55}{24}\varphi_0$	$V_1=-\frac{59}{24}\varphi_0(2h\mathbf{L})$	$V_2=\frac{37}{24}\varphi_0(3h\mathbf{L})$	$V_3=-\frac{3}{8}\varphi_0(4h\mathbf{L})$	

Table B.12: *Lawson* 4.

$\frac{1}{2}$	$A_{2,1} = \frac{1}{2}\psi_{0,2}$			
$\frac{1}{2}$	$A_{3,1} = 0$	$A_{3,2} = \frac{1}{2}$		
1	$A_{4,1} = 0$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$	
	$B_1 = \frac{1}{6}\varphi_0$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \frac{1}{6}$

B.4 Generalised Lawson integrators

Table B.13: *GenLawson* 41.

$\frac{1}{2}$	$A_{2,1} = \cdot$			
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$		
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$	
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \frac{1}{6}$

Table B.14: *GenLawson* 42.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1} = -\psi_{2,2}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1} = -\psi_{2,2} + \frac{1}{4}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1} = -\varphi_2 + \frac{1}{2}\psi_{0,2}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \frac{1}{6}$	$V_1 = -\varphi_2 + \frac{1}{3}\psi_{0,2} + \frac{1}{6}$

Table B.15: *GenLawson* 43.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1} = -2\psi_{2,2} - 2\psi_{3,2}$	$U_{2,2}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1} = -2\psi_{2,2} - 2\psi_{3,2} + \frac{5}{8}$	$U_{3,2}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1} = -2\varphi_2 - 2\varphi_3 + \frac{5}{4}\psi_{0,2}$	$U_{4,2}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \frac{1}{6}$	$V_1 = -2\varphi_2 - 2\varphi_3 + \frac{5}{6}\psi_{0,2} + \frac{1}{2}$	V_2

with

$$\begin{aligned} U_{2,2} &= \tfrac{1}{2}\psi_{2,2} + \psi_{3,2}, & U_{3,2} &= \tfrac{1}{2}\psi_{2,2} + \psi_{3,2} - \tfrac{3}{16}, \\ U_{4,2} &= \tfrac{1}{2}\varphi_2 + \varphi_3 - \tfrac{3}{8}\psi_{0,2}, & V_2 &= \tfrac{1}{2}\varphi_2 + \varphi_3 - \tfrac{1}{4}\psi_{0,2} - \tfrac{1}{6}. \end{aligned} \tag{B.7}$$

Table B.16: *GenLawson* 44.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1}$	$U_{2,2}$	$U_{2,3}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1}$	$U_{3,2}$	$U_{3,3}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1}$	$U_{4,2}$	$U_{4,3}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \frac{1}{6}$	V_1	V_2	V_3

with

$$\begin{aligned} U_{2,1} &= -3\psi_{2,2} - 5\psi_{3,2} - 3\psi_{4,2}, & U_{3,1} &= U_{2,1} + \tfrac{35}{32}, \\ U_{4,1} &= -3\varphi_2 - 5\varphi_3 - 3\varphi_4 + \tfrac{35}{16}\psi_{0,2}, & V_1 &= -3\varphi_2 - 5\varphi_3 - 3\varphi_4 + \tfrac{35}{24}\psi_{0,2} + 1, \\ U_{2,2} &= \tfrac{3}{2}\psi_{2,2} + 4\psi_{3,2} + 3\psi_{4,2}, & U_{3,2} &= U_{2,2} - \tfrac{21}{32}, \\ U_{4,2} &= \tfrac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4 - \tfrac{21}{16}\psi_{0,2}, & V_2 &= \tfrac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4 - \tfrac{7}{8}\psi_{0,2} - \tfrac{2}{3}, \\ U_{2,3} &= -\tfrac{1}{3}\psi_{2,2} - \psi_{3,2} - \psi_{4,2}, & U_{3,3} &= U_{2,3} + \tfrac{5}{32}, \\ U_{4,3} &= -\tfrac{1}{3}\varphi_2 - \varphi_3 - \varphi_4 + \tfrac{5}{16}\psi_{0,2}, & V_3 &= -\tfrac{1}{3}\varphi_2 - \varphi_3 - \varphi_4 + \tfrac{5}{24}\psi_{0,2} + \tfrac{1}{6}. \end{aligned} \tag{B.8}$$

Table B.17: *GenLawson* 45.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1}$	$U_{2,2}$	$U_{2,3}$	$U_{2,4}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1}$	$U_{3,2}$	$U_{3,3}$	$U_{3,4}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1}$	$U_{4,2}$	$U_{4,3}$	$U_{4,4}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \frac{1}{6}$	V_1	V_2	V_3	V_4

with

$$\begin{aligned} V_1 &= -4\varphi_2 - \tfrac{26}{3}\varphi_3 - 9\varphi_4 - 4\varphi_5 + \tfrac{35}{16}\psi_{0,2} + \tfrac{5}{3}, \\ V_2 &= 3\varphi_2 + \tfrac{19}{2}\varphi_3 + 12\varphi_4 + 6\varphi_5 - \tfrac{63}{32}\psi_{0,2} - \tfrac{5}{3}, \\ V_3 &= -\tfrac{4}{3}\varphi_2 - \tfrac{14}{3}\varphi_3 - 7\varphi_4 - 4\varphi_5 + \tfrac{15}{16}\psi_{0,2} + \tfrac{5}{6}, \\ V_4 &= \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5 - \tfrac{35}{192}\psi_{0,2} - \tfrac{1}{6}, \end{aligned} \tag{B.9}$$

and

$$
\begin{aligned}
U_{2,1} &= -4\psi_{2,2} - \tfrac{26}{3}\psi_{3,2} - 9\psi_{4,2} - 4\psi_{5,2}, & U_{3,1} &= U_{2,1} + \tfrac{105}{64},\\
U_{2,2} &= 3\psi_{2,2} + \tfrac{19}{2}\psi_{3,2} + 12\psi_{4,2} + 6\psi_{5,2}, & U_{3,2} &= U_{2,2} - \tfrac{189}{128},\\
U_{2,3} &= -\tfrac{4}{3}\psi_{2,2} - \tfrac{14}{3}\psi_{3,2} - 7\psi_{4,2} - 4\psi_{5,2}, & U_{3,3} &= U_{2,3} + \tfrac{45}{64},\\
U_{2,4} &= \tfrac{1}{4}\psi_{2,2} + \tfrac{88}{96}\psi_{3,2} + \tfrac{3}{2}\psi_{4,2} + \psi_{5,2}, & U_{3,4} &= U_{2,4} - \tfrac{35}{256},\\
U_{4,1} &= -4\varphi_2 - \tfrac{26}{3}\varphi_3 - 9\varphi_4 - 4\varphi_5 + \tfrac{105}{32}\psi_{0,2},\\
U_{4,2} &= 3\varphi_2 + \tfrac{19}{2}\varphi_3 + 12\varphi_4 + 6\varphi_5 - \tfrac{189}{64}\psi_{0,2},\\
U_{4,3} &= -\tfrac{4}{3}\varphi_2 - \tfrac{14}{3}\varphi_3 - 7\varphi_4 - 4\varphi_5 + \tfrac{45}{32}\psi_{0,2},\\
U_{4,4} &= \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5 - \tfrac{35}{128}\psi_{0,2}.
\end{aligned}
\tag{B.10}
$$

B.5 Modified generalised Lawson integrators

Table B.18: *ModGenLawson* 41.

$\frac{1}{2}$	$A_{2,1} = \cdot$			
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$		
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$	
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \varphi_2 - \frac{1}{3}\psi_{0,2}$

Table B.19: *ModGenLawson* 42.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1} = -\psi_{2,2}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1} = -\psi_{2,2} + \frac{1}{4}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1} = -\varphi_2 + \frac{1}{2}\psi_{0,2}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	$B_4 = \frac{1}{2}\varphi_2 + \varphi_3 - \frac{1}{4}\psi_{0,2}$	$V_1 = -\frac{1}{2}\varphi_2 + \varphi_3 + \frac{1}{12}\psi_{0,2}$

Table B.20: *ModGenLawson* 43.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1} = -2\psi_{2,2} - 2\psi_{3,2}$	$U_{2,2}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1} = -2\psi_{2,2} - 2\psi_{3,2} + \frac{5}{8}$	$U_{3,2}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1} = -2\varphi_2 - 2\varphi_3 + \frac{5}{4}\psi_{0,2}$	$U_{4,2}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	B_4	$V_1 = -\varphi_2 + \varphi_3 + 3\varphi_4 + \frac{5}{24}\psi_{0,2}$	V_2

with

$$
\begin{aligned}
&U_{2,2} = \tfrac{1}{2}\psi_{2,2} + \psi_{3,2}, && U_{3,2} = \tfrac{1}{2}\psi_{2,2} + \psi_{3,2} - \tfrac{3}{16},\\
&U_{4,2} = \tfrac{1}{2}\varphi_2 + \varphi_3 - \tfrac{3}{8}\psi_{0,2}, && V_2 = \tfrac{1}{6}\varphi_2 - \varphi_4 - \tfrac{1}{24}\psi_{0,2},\\
&B_4 = \tfrac{1}{3}\varphi_2 + \varphi_3 + \varphi_4 - \tfrac{5}{24}\psi_{0,2}.
\end{aligned}
\tag{B.11}
$$

Table B.21: *ModGenLawson* 44.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1}$	$U_{2,2}$	$U_{2,3}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1}$	$U_{3,2}$	$U_{3,3}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1}$	$U_{4,2}$	$U_{4,3}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	B_4	V_1	V_2	V_3

with

$$
\begin{aligned}
&U_{2,1} = -3\psi_{2,2} - 5\psi_{3,2} - 3\psi_{4,2}, && U_{3,1} = U_{2,1} + \tfrac{35}{32},\\
&U_{2,2} = \tfrac{3}{2}\psi_{2,2} + 4\psi_{3,2} + 3\psi_{4,2}, && U_{3,2} = U_{2,2} - \tfrac{21}{32},\\
&U_{2,3} = -\tfrac{1}{3}\psi_{2,2} - \psi_{3,2} - \psi_{4,2}, && U_{3,3} = U_{2,3} + \tfrac{5}{32},\\
&U_{4,1} = -3\varphi_2 - 5\varphi_3 - 3\varphi_4 + \tfrac{35}{16}\psi_{0,2}, && U_{4,2} = \tfrac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4 - \tfrac{21}{16}\psi_{0,2},\\
&U_{4,3} = -\tfrac{1}{3}\varphi_2 - \varphi_3 - \varphi_4 + \tfrac{5}{16}\psi_{0,2},\\
&V_1 = -\tfrac{3}{2}\varphi_2 + \tfrac{1}{2}\varphi_3 + 6\varphi_4 + 6\varphi_5 + \tfrac{35}{96}\psi_{0,2},\\
&V_2 = \tfrac{1}{2}\varphi_2 + \tfrac{1}{3}\varphi_3 - 3\varphi_4 - 4\varphi_5 - \tfrac{7}{48}\psi_{0,2},\\
&V_3 = -\tfrac{1}{12}\varphi_2 - \tfrac{1}{12}\varphi_3 + \tfrac{1}{2}\varphi_4 + \varphi_5 + \tfrac{5}{192}\psi_{0,2},\\
&B_4 = \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5 - \tfrac{35}{192}\psi_{0,2}.
\end{aligned}
\tag{B.12}
$$

Table B.22: *ModGenLawson* 45.

$\frac{1}{2}$	$A_{2,1} = \cdot$				$U_{2,1}$	$U_{2,2}$	$U_{2,3}$	$U_{2,4}$
$\frac{1}{2}$	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{2}$			$U_{3,1}$	$U_{3,2}$	$U_{3,3}$	$U_{3,4}$
1	$A_{4,1} = \cdot$	$A_{4,2} = 0$	$A_{4,3} = \psi_{0,2}$		$U_{4,1}$	$U_{4,2}$	$U_{4,3}$	$U_{4,4}$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\psi_{0,2}$	$B_3 = \frac{1}{3}\psi_{0,2}$	B_4	V_1	V_2	V_3	V_4

with

$$
\begin{aligned}
B_4 &= \tfrac{12}{59}\varphi_2 + \tfrac{50}{59}\varphi_3 + \tfrac{105}{59}\varphi_4 + \tfrac{120}{59}\varphi_5 - \tfrac{60}{59}\varphi_6 - \tfrac{157}{944}\psi_{0,2},\\
V_1 &= -\tfrac{116}{59}\varphi_2 - \tfrac{34}{177}\varphi_3 + \tfrac{519}{59}\varphi_4 + \tfrac{964}{59}\varphi_5 - \tfrac{600}{59}\varphi_6 + \tfrac{495}{944}\psi_{0,2},\\
V_2 &= \tfrac{57}{59}\varphi_2 + \tfrac{121}{118}\varphi_3 - \tfrac{342}{59}\varphi_4 - \tfrac{846}{59}\varphi_5 + \tfrac{600}{59}\varphi_6 - \tfrac{577}{1888}\psi_{0,2},\\
V_3 &= -\tfrac{56}{177}\varphi_2 - \tfrac{76}{177}\varphi_3 + \tfrac{112}{59}\varphi_4 + \tfrac{364}{59}\varphi_5 - \tfrac{300}{59}\varphi_6 + \tfrac{25}{236}\psi_{0,2},\\
V_4 &= \tfrac{11}{236}\varphi_2 + \tfrac{49}{708}\varphi_3 - \tfrac{33}{118}\varphi_4 - \tfrac{61}{59}\varphi_5 + \tfrac{60}{59}\varphi_6 - \tfrac{181}{11328}\psi_{0,2},
\end{aligned}
\tag{B.13}
$$

and

$$
\begin{aligned}
U_{2,1} &= -4\psi_{2,2} - \tfrac{26}{3}\psi_{3,2} - 9\psi_{4,2} - 4\psi_{5,2}, & U_{3,1} &= U_{2,1} + \tfrac{105}{64},\\
U_{2,2} &= 3\psi_{2,2} + \tfrac{19}{2}\psi_{3,2} + 12\psi_{4,2} + 6\psi_{5,2}, & U_{3,2} &= U_{2,2} - \tfrac{189}{128},\\
U_{2,3} &= -\tfrac{4}{3}\psi_{2,2} - \tfrac{14}{3}\psi_{3,2} - 7\psi_{4,2} - 4\psi_{5,2}, & U_{3,3} &= U_{2,3} + \tfrac{45}{64},\\
U_{2,4} &= \tfrac{1}{4}\psi_{2,2} + \tfrac{88}{96}\psi_{3,2} + \tfrac{3}{2}\psi_{4,2} + \psi_{5,2}, & U_{3,4} &= U_{2,4} - \tfrac{35}{256},\\
U_{4,1} &= -4\varphi_2 - \tfrac{26}{3}\varphi_3 - 9\varphi_4 - 4\varphi_5 + \tfrac{105}{32}\psi_{0,2},\\
U_{4,2} &= 3\varphi_2 + \tfrac{19}{2}\varphi_3 + 12\varphi_4 + 6\varphi_5 - \tfrac{189}{64}\psi_{0,2},\\
U_{4,3} &= -\tfrac{4}{3}\varphi_2 - \tfrac{14}{3}\varphi_3 - 7\varphi_4 - 4\varphi_5 + \tfrac{45}{32}\psi_{0,2},\\
U_{4,4} &= \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5 - \tfrac{35}{128}\psi_{0,2}.
\end{aligned}
\tag{B.14}
$$

B.6 Exponential predictor-corrector integrators

Table B.23: $PEC423$.

1	$A_{2,1} = \cdot$		$U_{2,1} = -2\varphi_2 - 2\varphi_3$	$U_{2,2} = \frac{1}{2}\varphi_2 + \varphi_3$
	$B_1 = \cdot$	$B_2 = \frac{1}{3}\varphi_2 + \varphi_3 + \varphi_4$	$V_1 = -\varphi_2 + \varphi_3 + 3\varphi_4$	$V_2 = \frac{1}{6}\varphi_2 - \varphi_4$

Table B.24: $PEC433$.

1	$A_{2_1} = \cdot$			$U_{2,1} = -2\varphi_2 - 2\varphi_3$	$U_{2,2}$
1	$A_{3,1} = \cdot$	$A_{3,2} = \frac{1}{3}\varphi_2 + \varphi_3 + \varphi_4$		$U_{3,1} = -\varphi_2 + \varphi_3 + 3\varphi_4$	$U_{3,2}$
	$B_1 = \cdot$	$B_2 = 0$	$B_3 = \frac{1}{3}\varphi_2 + \varphi_3 + \varphi_4$	$V_1 = -\varphi_2 + \varphi_3 + 3\varphi_4$	V_2

with

$$U_{2,2} = \tfrac{1}{2}\varphi_2 + \varphi_3, \quad U_{3,2} = \tfrac{1}{6}\varphi_2 - \varphi_4, \quad V_2 = \tfrac{1}{6}\varphi_2 - \varphi_4. \tag{B.15}$$

Table B.25: $PEC524$.

1	$A_{2,1} = \cdot$		$U_{2,1} = -3\varphi_2 - 5\varphi_3 - 3\varphi_4$	$U_{2,2} = \frac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4$	$U_{2,3}$
	$B_1 = \cdot$	B_2	$V_1 = -\frac{3}{2}\varphi_2 + \frac{1}{2}\varphi_3 + 6\varphi_4 + 6\varphi_5$	$V_2 = \frac{1}{2}\varphi_2 + \frac{1}{3}\varphi_3 - 3\varphi_4 - 4\varphi_5$	V_3

with

$$\begin{aligned} &B_2 = \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5, \qquad U_{2,3} = -\tfrac{1}{3}\varphi_2 - \varphi_3 - \varphi_4, \\ &V_3 = -\tfrac{1}{12}\varphi_2 - \tfrac{1}{12}\varphi_3 + \tfrac{1}{2}\varphi_4 + \varphi_5. \end{aligned} \tag{B.16}$$

Table B.26: $PEC534$.

1	$A_{2,1} = \cdot$			$U_{2,1} = -3\varphi_2 - 5\varphi_3 - 3\varphi_4$	$U_{2,2}$	$U_{2,3}$
1	$A_{3,1} = \cdot$	$A_{3,2}$		$U_{3,1} = -\frac{3}{2}\varphi_2 + \frac{1}{2}\varphi_3 + 6\varphi_4 + 6\varphi_5$	$U_{3,2}$	$U_{3,3}$
	$B_1 = \cdot$	$B_2 = 0$	$B_3 = A_{3,2}$	$V_1 = -\frac{3}{2}\varphi_2 + \frac{1}{2}\varphi_3 + 6\varphi_4 + 6\varphi_5$	V_2	$V_3 = U_{3,3}$

with

$$\begin{aligned} A_{3,2} &= \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5, & U_{2,2} &= \tfrac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4, \\ U_{2,3} &= -\tfrac{1}{3}\varphi_2 - \varphi_3 - \varphi_4, & U_{3,2} &= \tfrac{1}{2}\varphi_2 + \tfrac{1}{3}\varphi_3 - 3\varphi_4 - 4\varphi_5, \\ U_{3,3} &= -\tfrac{1}{12}\varphi_2 - \tfrac{1}{12}\varphi_3 + \tfrac{1}{2}\varphi_4 + \varphi_5, & V_2 &= \tfrac{1}{2}\varphi_2 + \tfrac{1}{3}\varphi_3 - 3\varphi_4 - 4\varphi_5. \end{aligned} \tag{B.17}$$

Table B.27: $PEC625$.

1	$A_{2,1} = \cdot$		$U_{2,1} = -4\varphi_2 - \frac{26}{3}\varphi_3 - 9\varphi_4 - 4\varphi_5$	$U_{2,2}$	$U_{2,3}$	$U_{2,4}$
	$B_1 = \cdot$	B_2	$V_1 = -2\varphi_2 - \frac{1}{3}\varphi_3 + \frac{17}{2}\varphi_4 + 16\varphi_5 + 10\varphi_6$	V_2	V_3	V_4

with

$$\begin{aligned} B_2 &= \tfrac{1}{5}\varphi_2 + \tfrac{5}{6}\varphi_3 + \tfrac{7}{4}\varphi_4 + 2\varphi_5 + \varphi_6, & U_{2,2} &= 3\varphi_2 + \tfrac{19}{2}\varphi_3 + 12\varphi_4 + 6\varphi_5, \\ U_{2,3} &= -\tfrac{4}{3}\varphi_2 - \tfrac{14}{3}\varphi_3 - 7\varphi_4 - 4\varphi_5, & U_{2,4} &= \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5, \\ V_2 &= \varphi_2 + \tfrac{7}{6}\varphi_3 - \tfrac{11}{2}\varphi_4 - 14\varphi_5 - 10\varphi_6, & V_3 &= -\tfrac{1}{3}\varphi_2 - \tfrac{1}{2}\varphi_3 + \tfrac{7}{4}\varphi_4 + 6\varphi_5 + 5\varphi_6, \\ V_4 &= \tfrac{1}{20}\varphi_2 + \tfrac{1}{12}\varphi_3 - \tfrac{1}{4}\varphi_4 - \varphi_5 - \varphi_6. \end{aligned} \tag{B.18}$$

Table B.28: $PEC635$.

1	$A_{2,1} = \cdot$		$U_{2,1} = -4\varphi_2 - \frac{26}{3}\varphi_3 - 9\varphi_4 - 4\varphi_5$	$U_{2,2}$	$U_{2,3}$	$U_{2,4}$
1	$A_{3,1} = \cdot$	$A_{3,2}$	$U_{3,1} = -2\varphi_2 - \frac{1}{3}\varphi_3 + \frac{17}{2}\varphi_4 + 16\varphi_5 + 10\varphi_6$	$U_{3,2}$	$U_{3,3}$	$U_{3,4}$
	$B_1 = \cdot$	$B_2 = 0$ B_3	$V_1 = -2\varphi_2 - \frac{1}{3}\varphi_3 + \frac{17}{2}\varphi_4 + 16\varphi_5 + 10\varphi_6$	V_2	V_3	V_4

with

$$\begin{aligned} A_{3,2} &= \tfrac{1}{5}\varphi_2 + \tfrac{5}{6}\varphi_3 + \tfrac{7}{4}\varphi_4 + 2\varphi_5 + \varphi_6, & B_3 &= A_{3,2}, \\ U_{2,2} &= 3\varphi_2 + \tfrac{19}{2}\varphi_3 + 12\varphi_4 + 6\varphi_5, & V_2 &= U_{3,2}, \\ U_{2,3} &= -\tfrac{4}{3}\varphi_2 - \tfrac{14}{3}\varphi_3 - 7\varphi_4 - 4\varphi_5, & V_3 &= U_{3,3}, \\ U_{2,4} &= \tfrac{1}{4}\varphi_2 + \tfrac{11}{12}\varphi_3 + \tfrac{3}{2}\varphi_4 + \varphi_5, & V_4 &= U_{3,4}, \end{aligned} \tag{B.19}$$

and

$$\begin{aligned} U_{3,2} &= \varphi_2 + \tfrac{7}{6}\varphi_3 - \tfrac{11}{2}\varphi_4 - 14\varphi_5 - 10\varphi_6, \\ U_{3,3} &= -\tfrac{1}{3}\varphi_2 - \tfrac{1}{2}\varphi_3 + \tfrac{7}{4}\varphi_4 + 6\varphi_5 + 5\varphi_6, \\ U_{3,4} &= \tfrac{1}{20}\varphi_2 + \tfrac{1}{12}\varphi_3 - \tfrac{1}{4}\varphi_4 - \varphi_5 - \varphi_6. \end{aligned} \tag{B.20}$$

Table B.29: $PEC726$.

1	$A_{2,1} = \cdot$	$U_{2,1} = -5\varphi_2 - \frac{77}{6}\varphi_3 - \frac{71}{4}\varphi_4 - 14\varphi_5 - 5\varphi_6$	$U_{2,2}$	$U_{2,3}$	$U_{2,4}$	$U_{2,5}$
	$B_1 = \cdot$ B_2	$V_1 = -\frac{5}{2}\varphi_2 - \frac{17}{12}\varphi_3 + \frac{83}{8}\varphi_4 + \frac{57}{2}\varphi_5 + \frac{65}{2}\varphi_6 + 15\varphi_7$	V_2	V_3	V_4	V_5

with

$$
\begin{aligned}
B_2 &= \tfrac{1}{6}\varphi_2 + \tfrac{137}{180}\varphi_3 + \tfrac{15}{8}\varphi_4 + \tfrac{17}{6}\varphi_5 + \tfrac{5}{2}\varphi_6 + \varphi_7,\\
U_{2,2} &= 5\varphi_2 + \tfrac{107}{6}\varphi_3 + \tfrac{59}{2}\varphi_4 + 26\varphi_5 + 10\varphi_6,\\
U_{2,3} &= -\tfrac{10}{3}\varphi_2 - 13\varphi_3 - \tfrac{49}{2}\varphi_4 - 24\varphi_5 - 10\varphi_6,\\
U_{2,4} &= \tfrac{5}{4}\varphi_2 + \tfrac{61}{12}\varphi_3 + \tfrac{41}{4}\varphi_4 + 11\varphi_5 + 5\varphi_6,\\
U_{2,5} &= -\tfrac{1}{5}\varphi_2 - \tfrac{5}{6}\varphi_3 - \tfrac{7}{4}\varphi_4 - 2\varphi_5 - \varphi_6,\\
V_2 &= \tfrac{5}{3}\varphi_2 + \tfrac{47}{18}\varphi_3 - 8\varphi_4 - \tfrac{92}{3}\varphi_5 - 40\varphi_6 - 20\varphi_7,\\
V_3 &= -\tfrac{5}{6}\varphi_2 - \tfrac{19}{12}\varphi_3 + \tfrac{29}{8}\varphi_4 + \tfrac{37}{2}\varphi_5 + \tfrac{55}{2}\varphi_6 + 15\varphi_7,\\
V_4 &= \tfrac{1}{4}\varphi_2 + \tfrac{31}{60}\varphi_3 - \varphi_4 - 6\varphi_5 - 10\varphi_6 - 6\varphi_7,\\
V_5 &= -\tfrac{1}{30}\varphi_2 - \tfrac{13}{180}\varphi_3 + \tfrac{1}{8}\varphi_4 + \tfrac{5}{6}\varphi_5 + \tfrac{3}{2}\varphi_6 + \varphi_7.
\end{aligned}
\tag{B.21}
$$

Table B.30: $PEC736$.

1	$A_{2,1} = \cdot$		$U_{2,1}$	$U_{2,2}$	$U_{2,3}$	$U_{2,4}$	$U_{2,5}$
1	$A_{3,1} = \cdot$	$A_{3,2}$	$U_{3,1}$	$U_{3,2}$	$U_{3,3}$	$U_{3,4}$	$U_{3,5}$
	$B_1 = \cdot$	$B_2 = 0$ $B_3 = A_{3,2}$	V_1	$V_2 = U_{3,2}$	$V_3 = U_{3,3}$	$V_4 = U_{3,4}$	$V_5 = U_{3,5}$

with

$$
\begin{aligned}
U_{2,1} &= -5\varphi_2 - \tfrac{77}{6}\varphi_3 - \tfrac{71}{4}\varphi_4 - 14\varphi_5 - 5\varphi_6,\\
U_{2,2} &= 5\varphi_2 + \tfrac{107}{6}\varphi_3 + \tfrac{59}{2}\varphi_4 + 26\varphi_5 + 10\varphi_6,\\
U_{2,3} &= -\tfrac{10}{3}\varphi_2 - 13\varphi_3 - \tfrac{49}{2}\varphi_4 - 24\varphi_5 - 10\varphi_6,\\
U_{2,4} &= \tfrac{5}{4}\varphi_2 + \tfrac{61}{12}\varphi_3 + \tfrac{41}{4}\varphi_4 + 11\varphi_5 + 5\varphi_6,\\
U_{2,5} &= -\tfrac{1}{5}\varphi_2 - \tfrac{5}{6}\varphi_3 - \tfrac{7}{4}\varphi_4 - 2\varphi_5 - \varphi_6,
\end{aligned}
\tag{B.22}
$$

and

$$
\begin{aligned}
U_{3,1} &= -\tfrac{5}{2}\varphi_2 - \tfrac{17}{12}\varphi_3 + \tfrac{83}{8}\varphi_4 + \tfrac{57}{2}\varphi_5 + \tfrac{65}{2}\varphi_6 + 15\varphi_7,\\
U_{3,2} &= \tfrac{5}{3}\varphi_2 + \tfrac{47}{18}\varphi_3 - 8\varphi_4 - \tfrac{92}{3}\varphi_5 - 40\varphi_6 - 20\varphi_7,\\
U_{3,3} &= -\tfrac{5}{6}\varphi_2 - \tfrac{19}{12}\varphi_3 + \tfrac{29}{8}\varphi_4 + \tfrac{37}{2}\varphi_5 + \tfrac{55}{2}\varphi_6 + 15\varphi_7,\\
U_{3,4} &= \tfrac{1}{4}\varphi_2 + \tfrac{31}{60}\varphi_3 - \varphi_4 - 6\varphi_5 - 10\varphi_6 - 6\varphi_7,\\
U_{3,5} &= -\tfrac{1}{30}\varphi_2 - \tfrac{13}{180}\varphi_3 + \tfrac{1}{8}\varphi_4 + \tfrac{5}{6}\varphi_5 + \tfrac{3}{2}\varphi_6 + \varphi_7,\\
A_{3,2} &= \tfrac{1}{6}\varphi_2 + \tfrac{137}{180}\varphi_3 + \tfrac{15}{8}\varphi_4 + \tfrac{17}{6}\phi_5 + \tfrac{5}{2}\varphi_6 + \varphi_7,\\
V_1 &= -\tfrac{5}{2}\varphi_2 - \tfrac{17}{12}\varphi_3 + \tfrac{83}{8}\varphi_4 + \tfrac{57}{2}\varphi_5 + \tfrac{65}{2}\varphi_6 + 15\varphi_7.
\end{aligned}
\tag{B.23}
$$

Appendix C

Comparison plots of the integrators

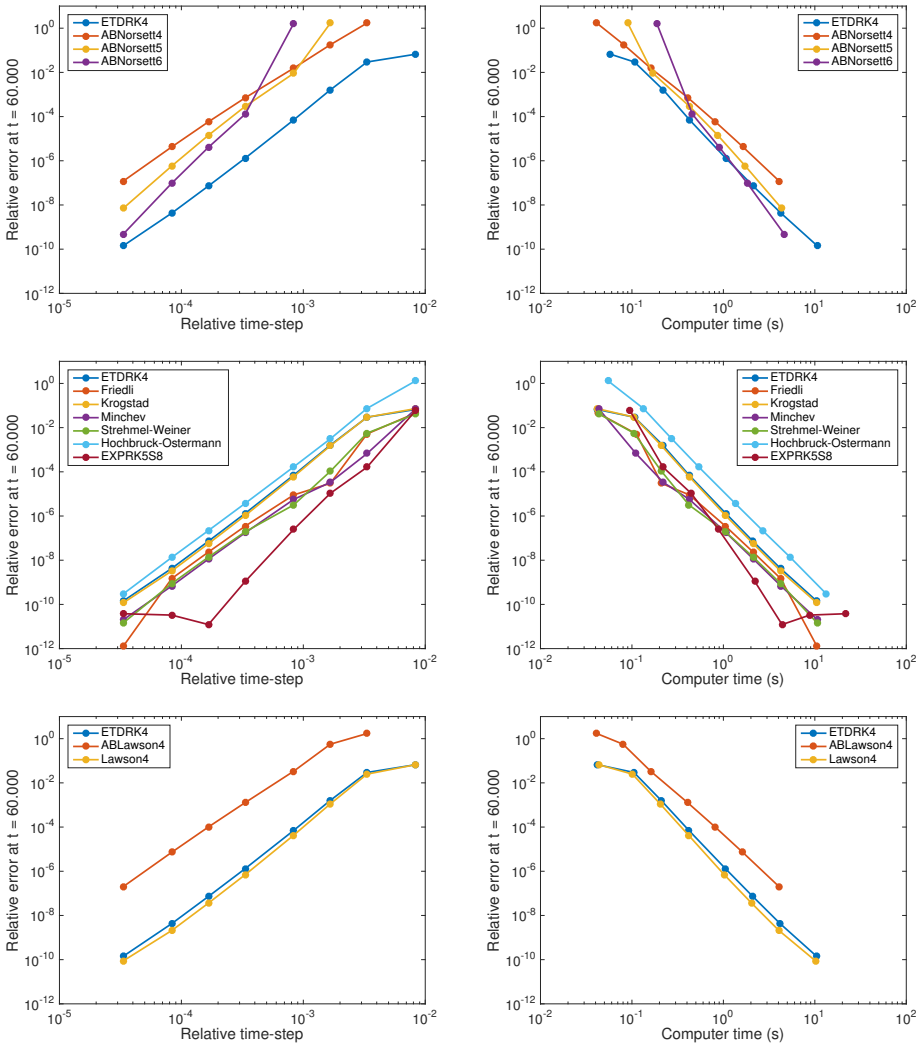

Figure C.1: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the* 1*D Allen–Cahn equation.*

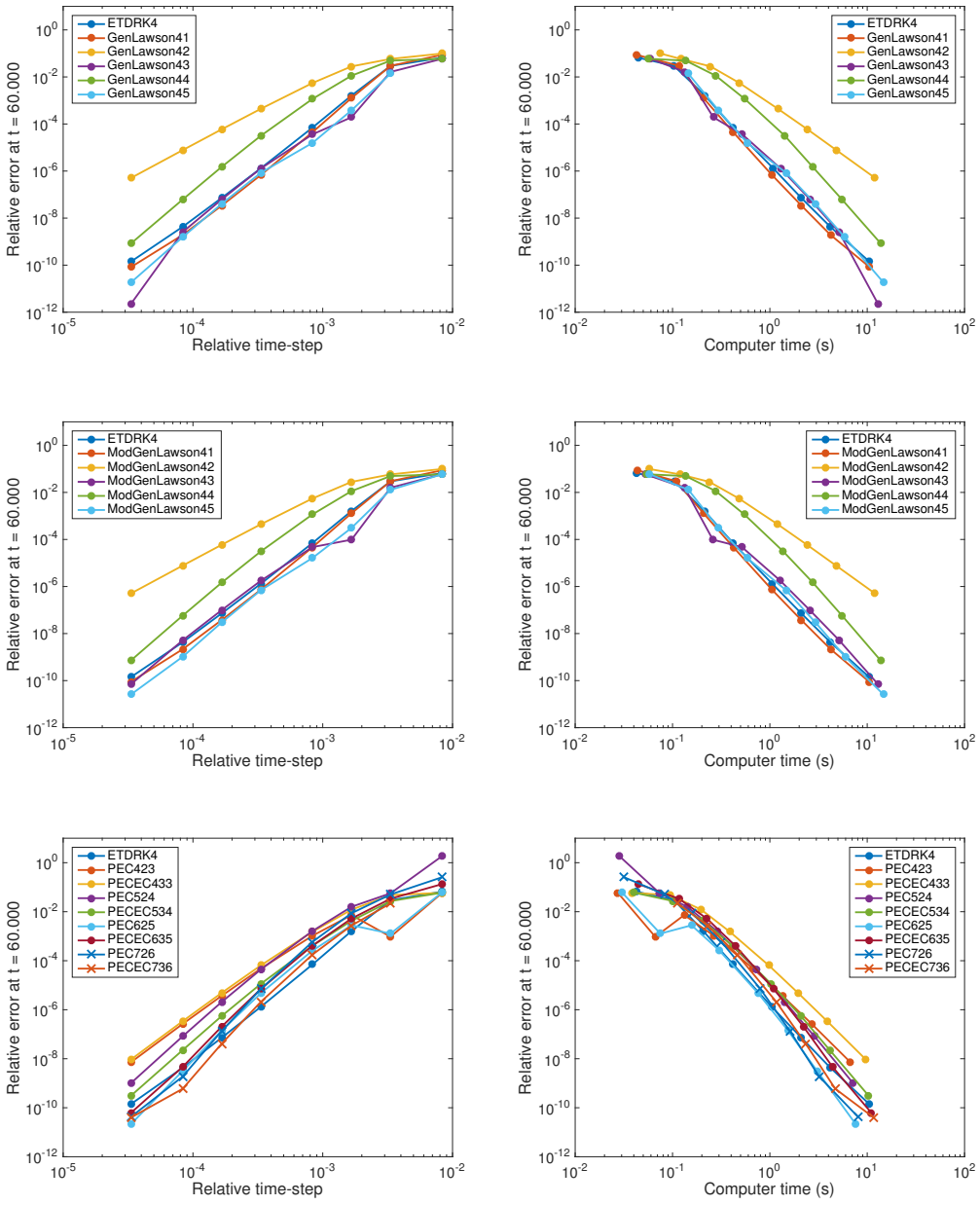

Figure C.2: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the* 1*D Allen–Cahn equation.*

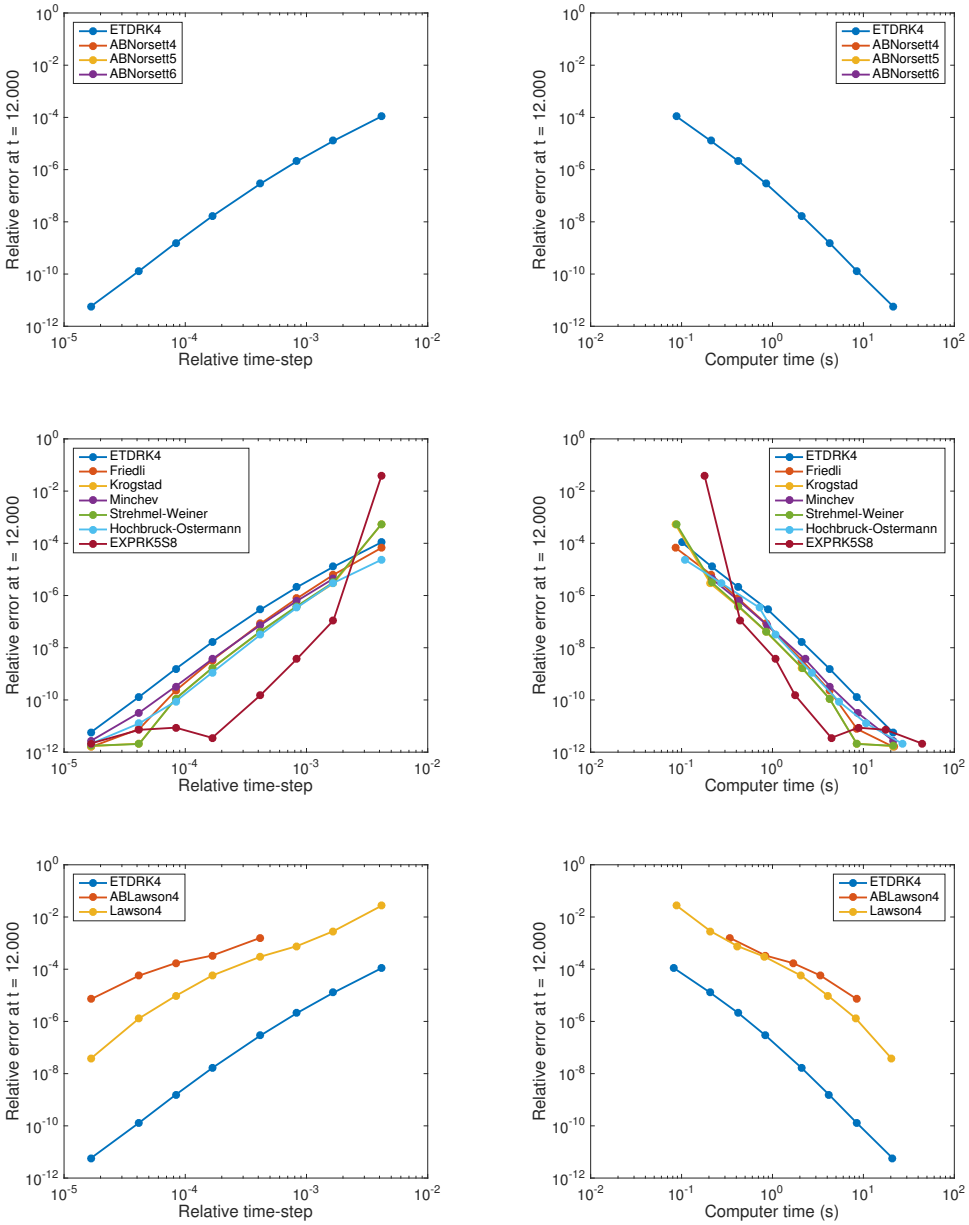

Figure C.3: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the* 1*D Cahn–Hilliard equation.*

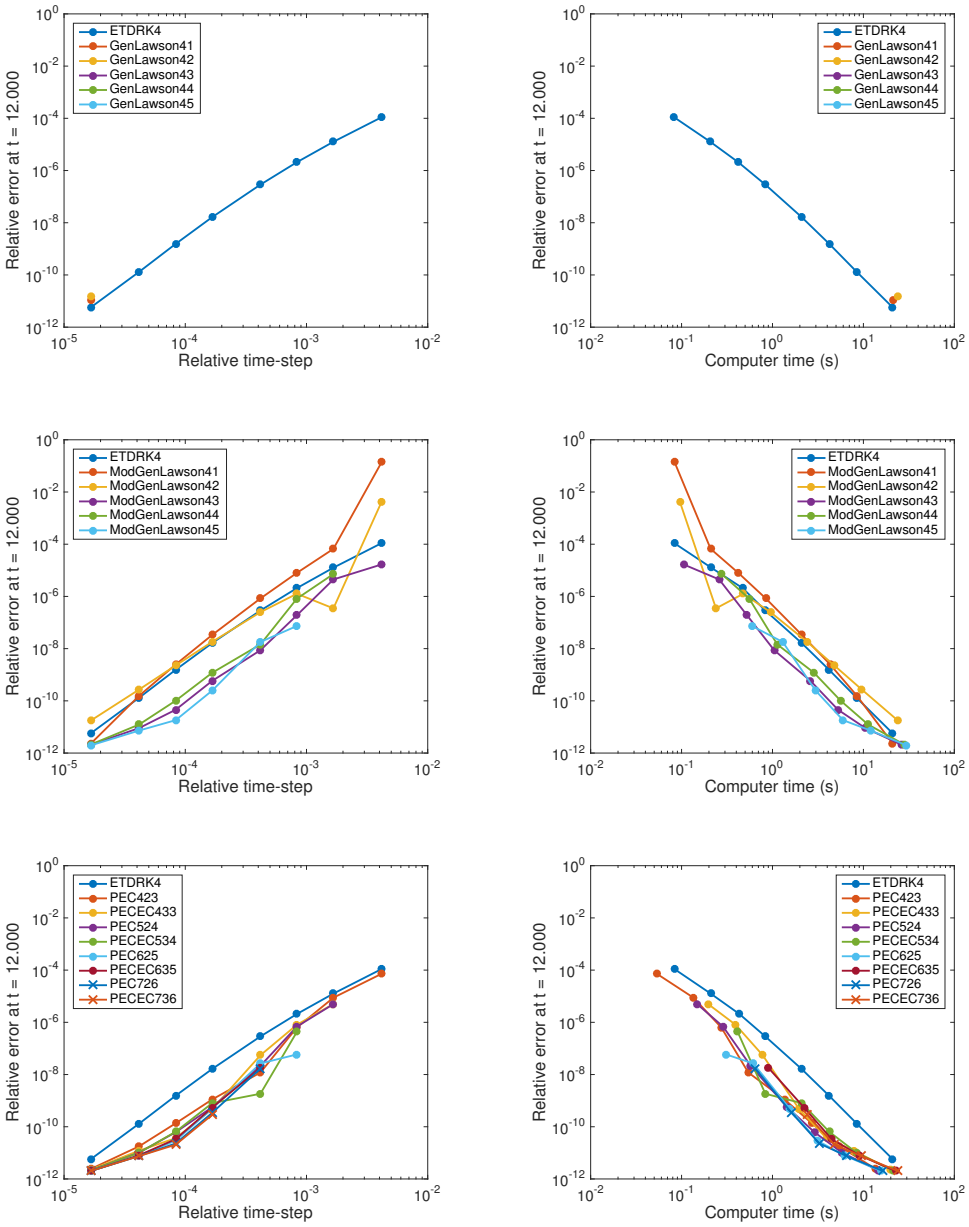

Figure C.4: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the* 1*D Cahn–Hilliard equation.*

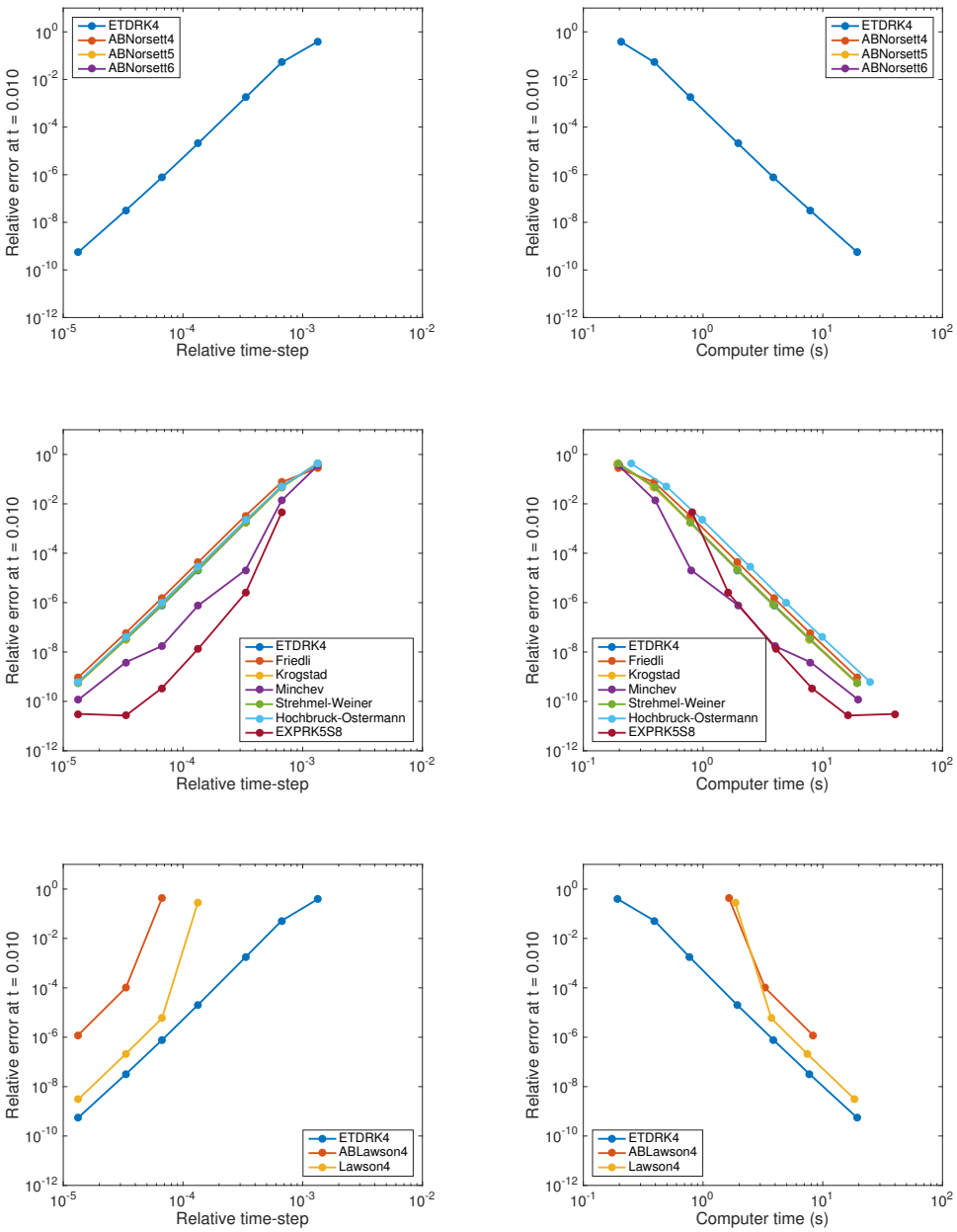

Figure C.5: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the* 1*D KdV equation.*

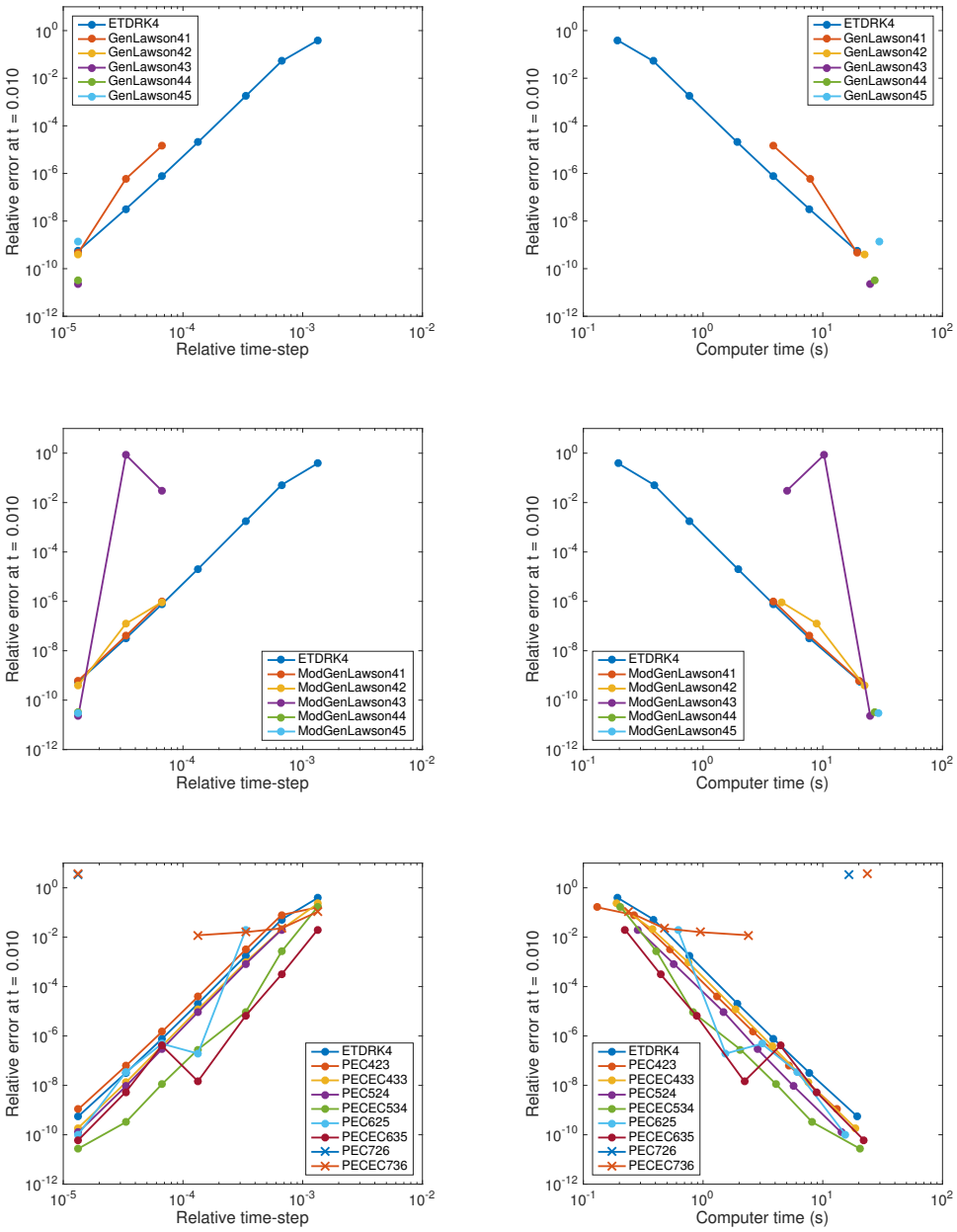

Figure C.6: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the* 1*D KdV equation.*

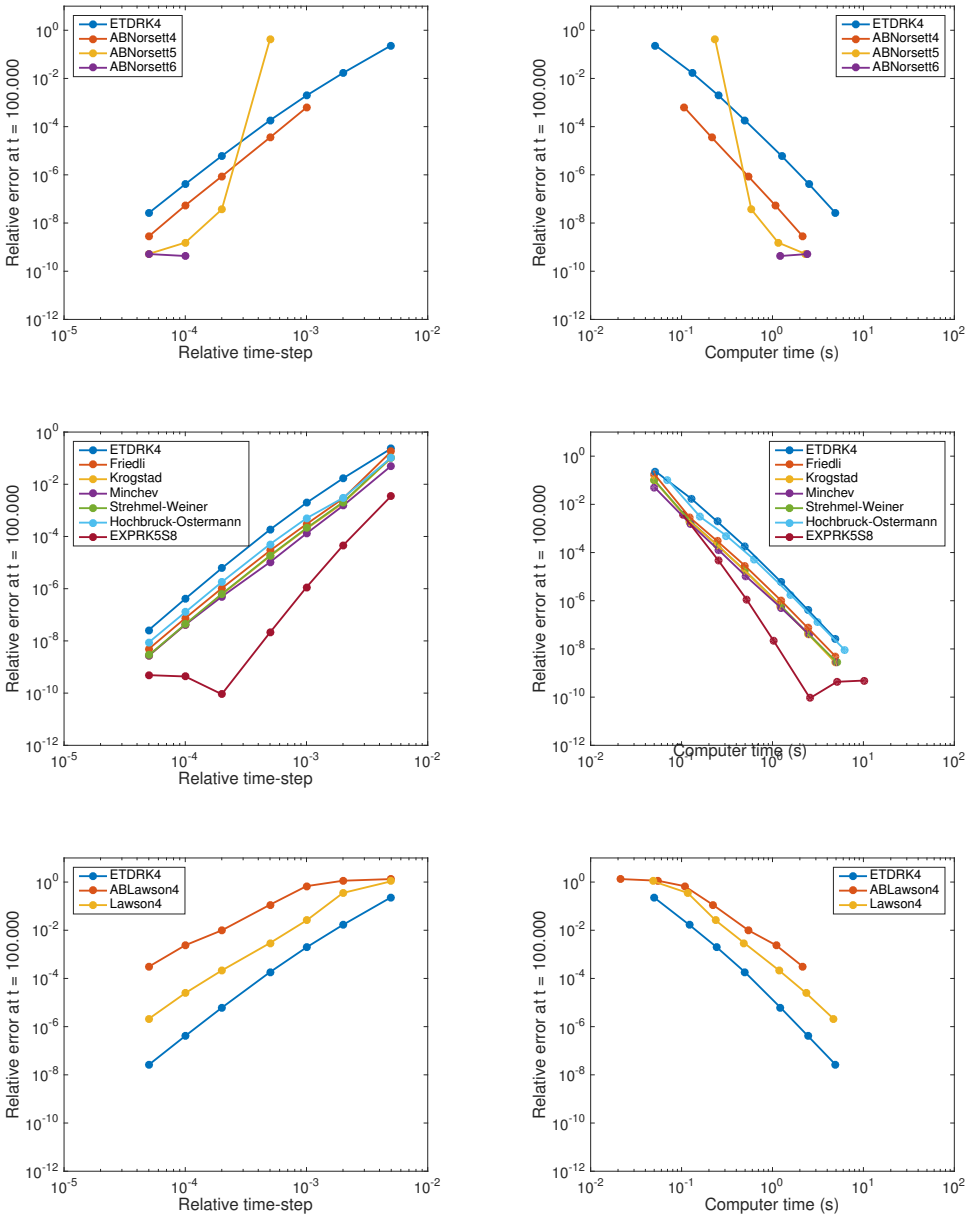

Figure C.7: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the* 1*D KS equation.*

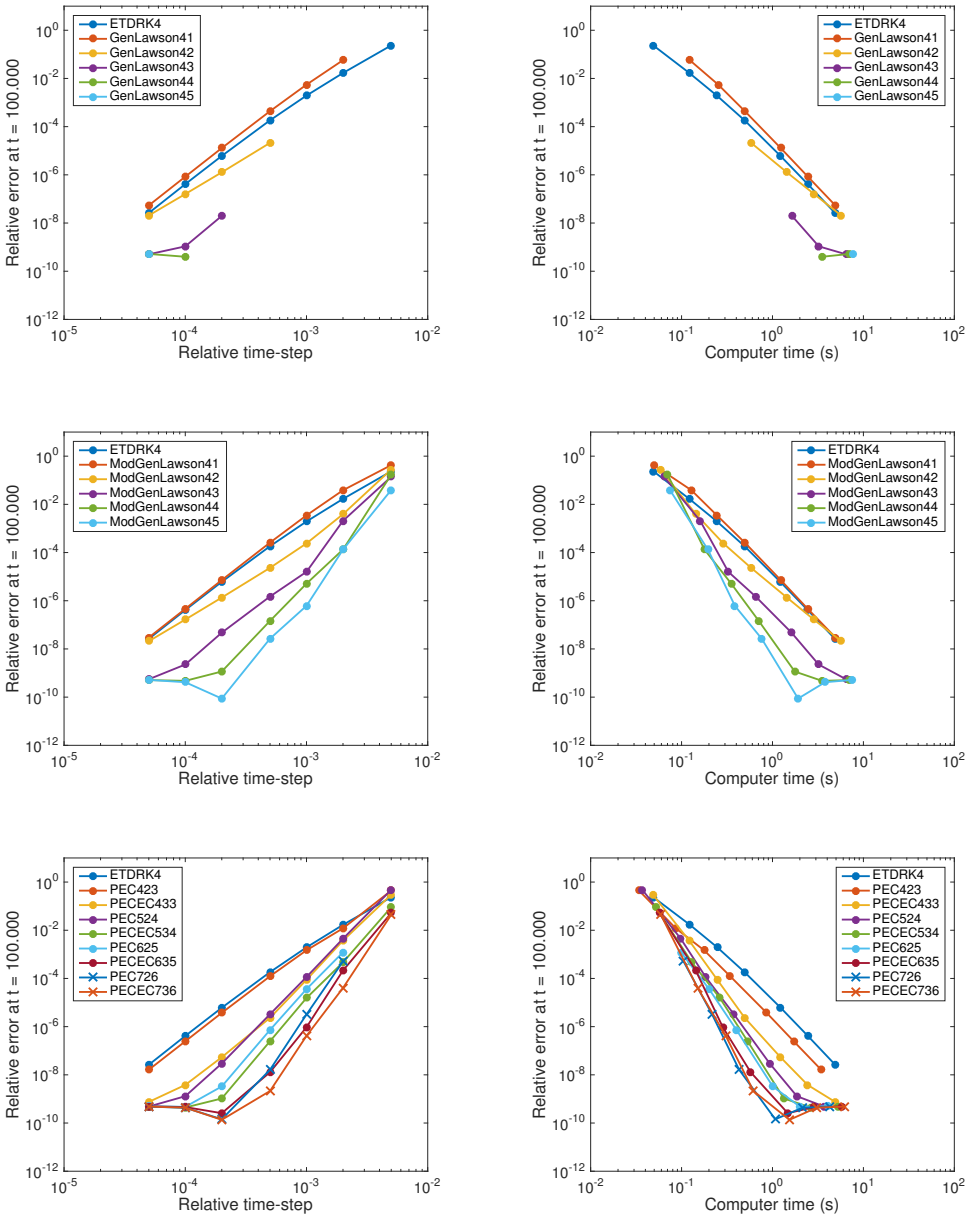

Figure C.8: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the* 1*D KS equation.*

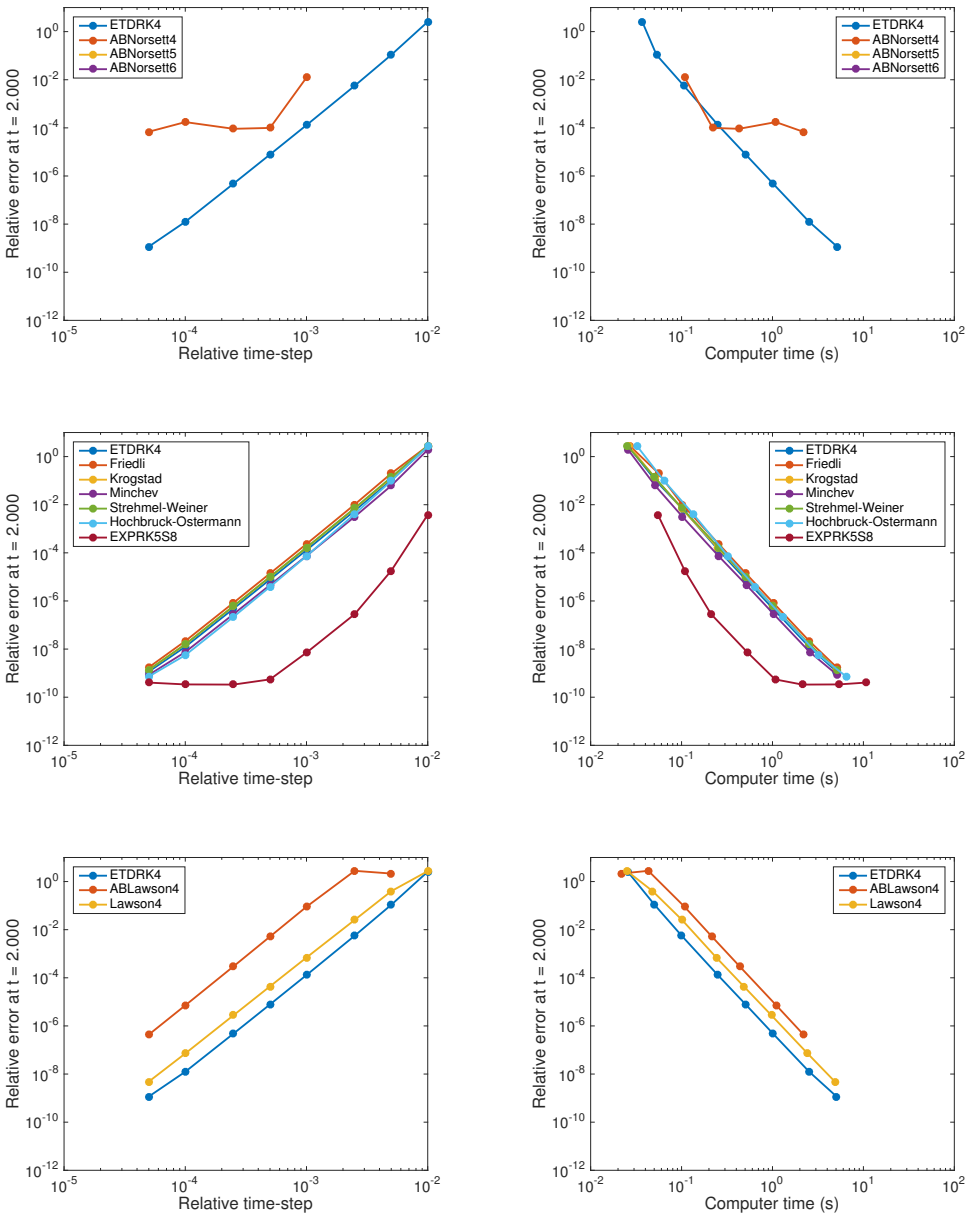

Figure C.9: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the* 1*D NLS equation.*

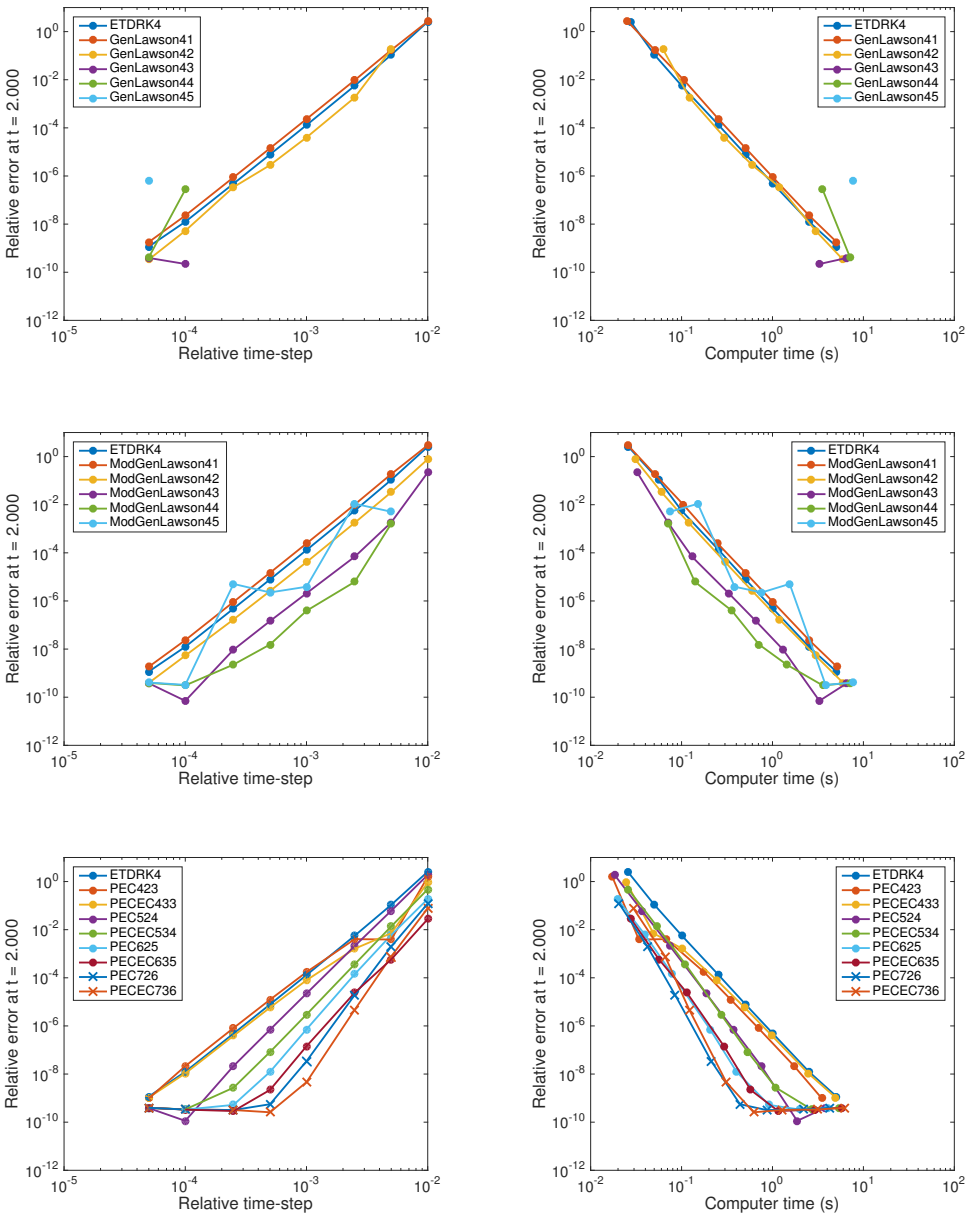

Figure C.10: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the* 1*D NLS equation.*

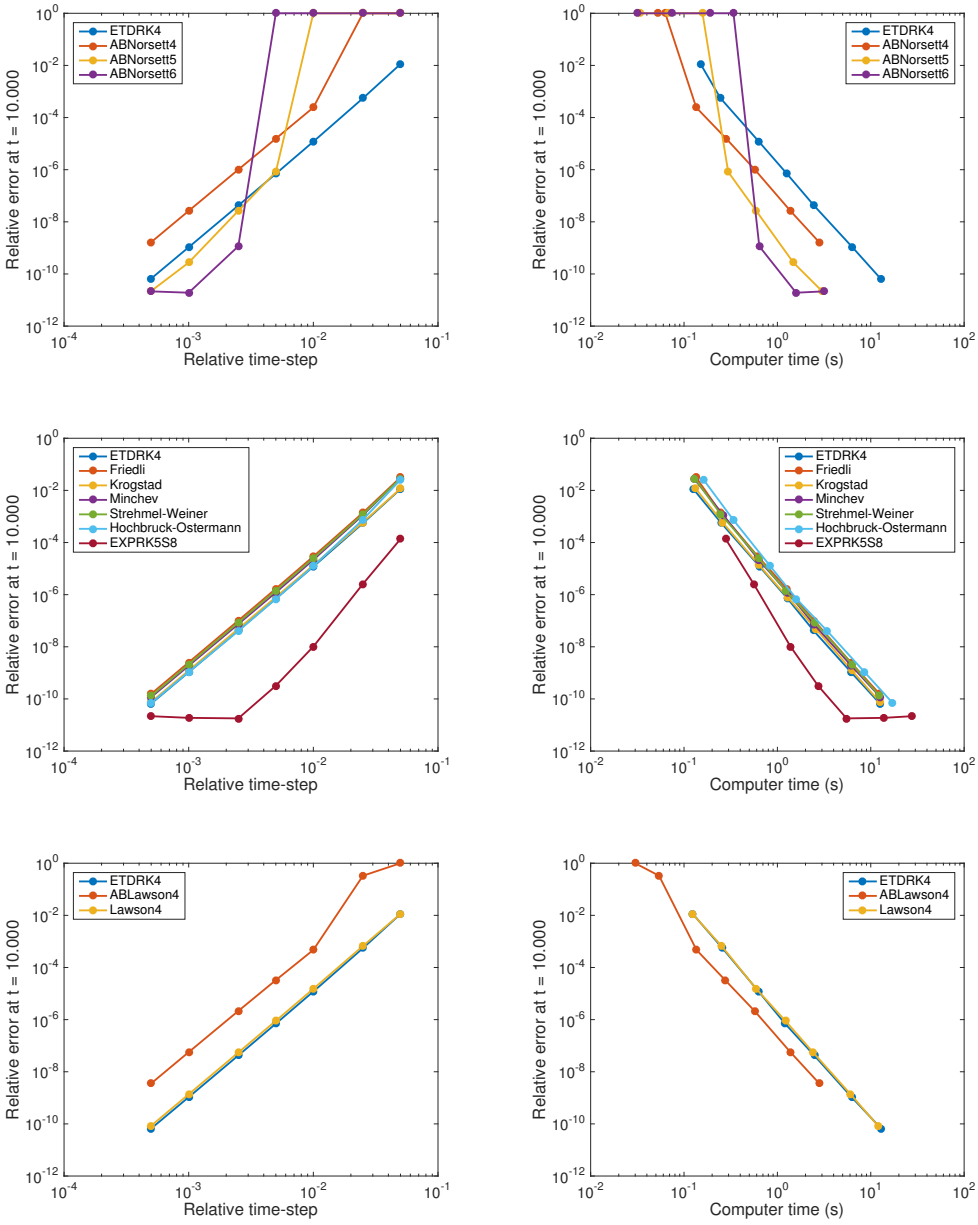

Figure C.11: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the 2D Ginzburg–Landau equation.*

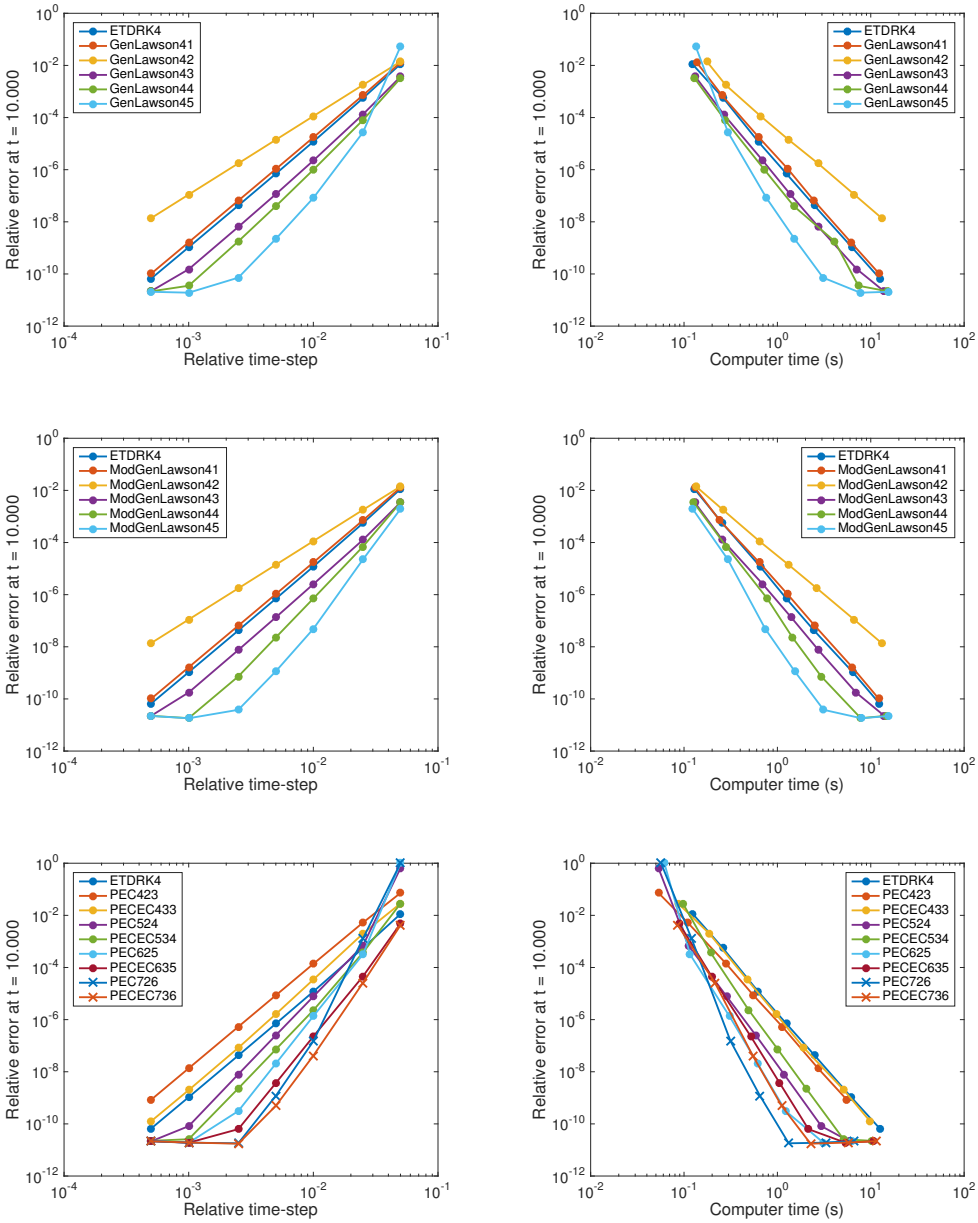

Figure C.12: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the 2D Ginzburg–Landau equation.*

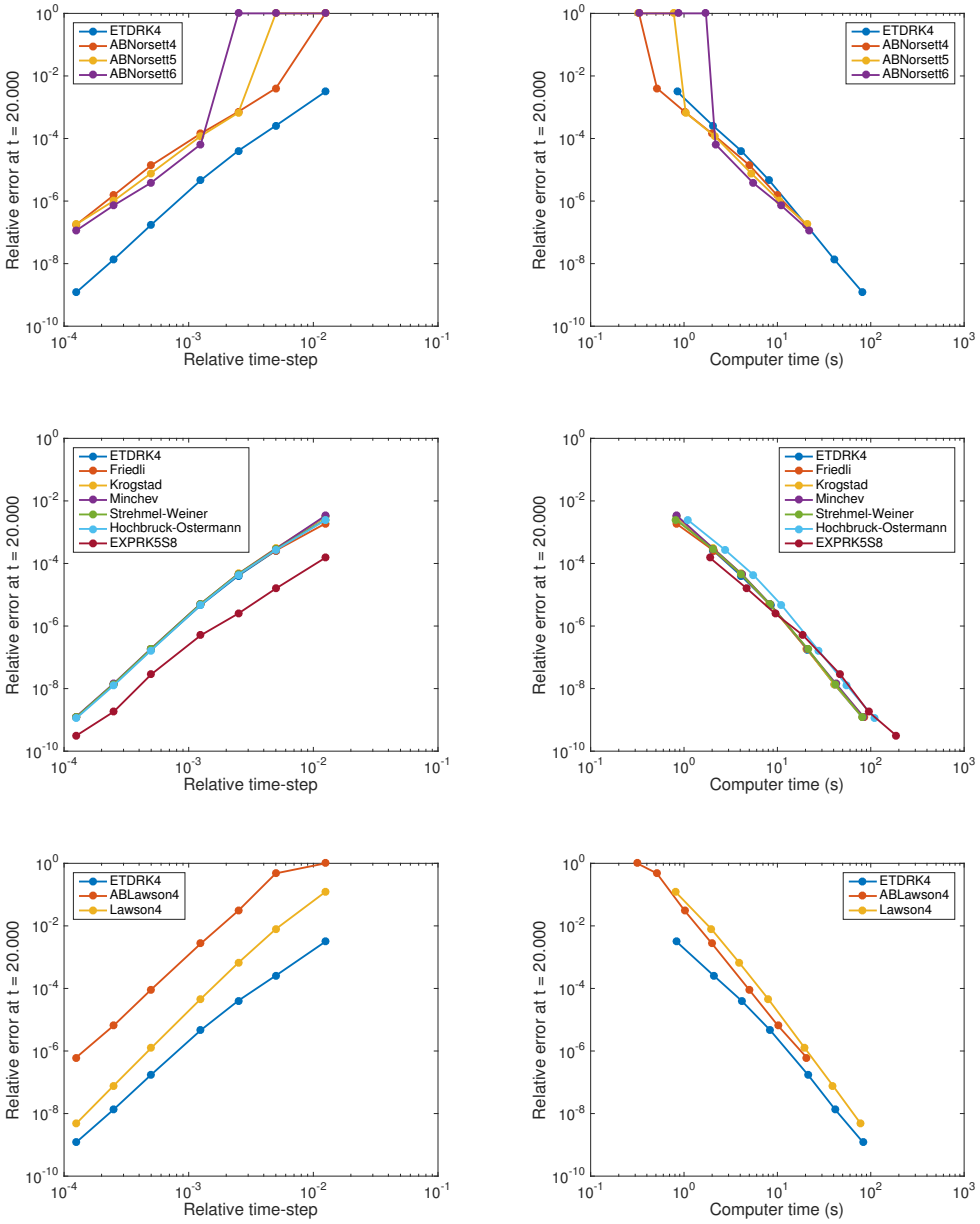

Figure C.13: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the 2D Schnakenberg equations.*

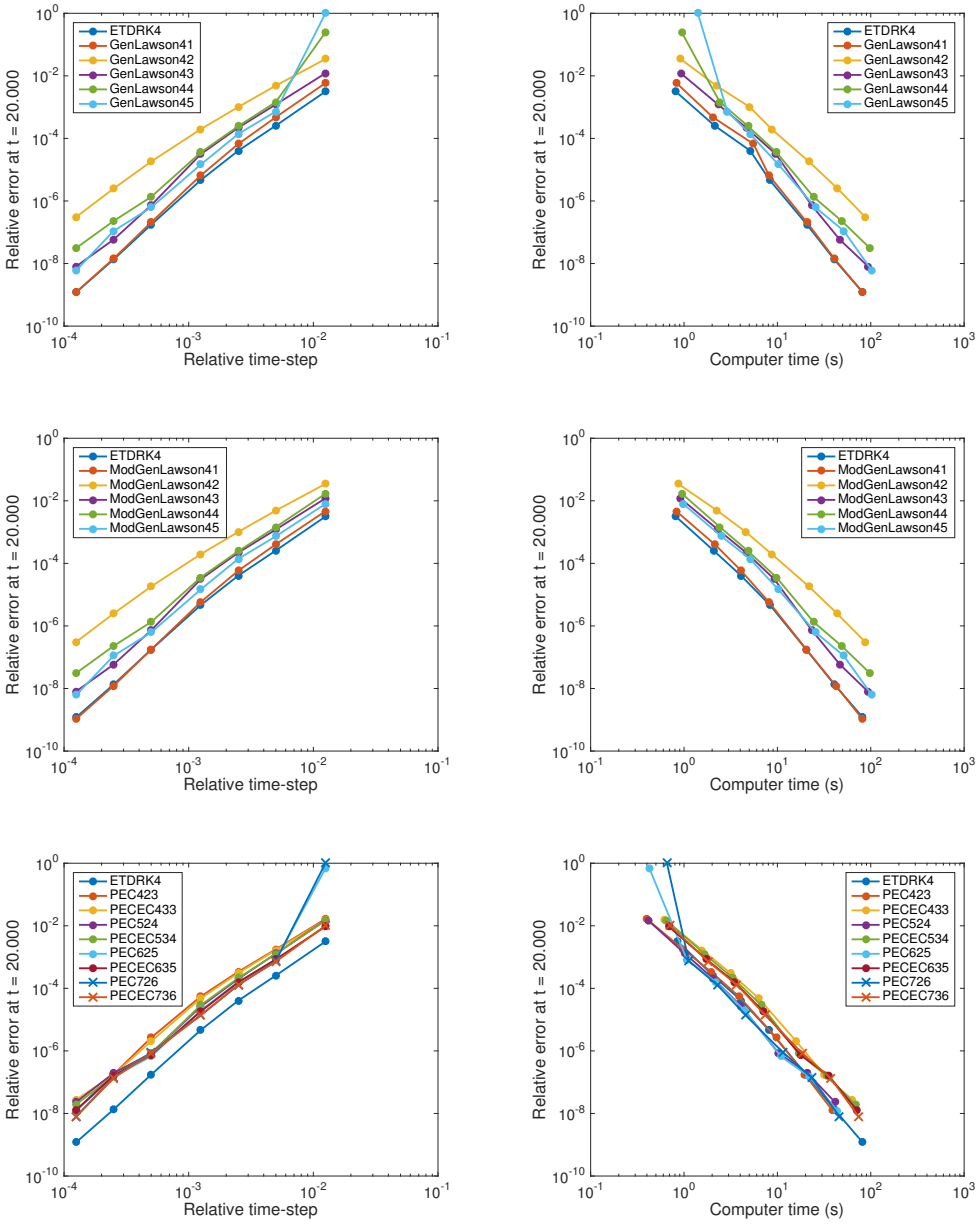

Figure C.14: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the 2D Schnakenberg equations.*

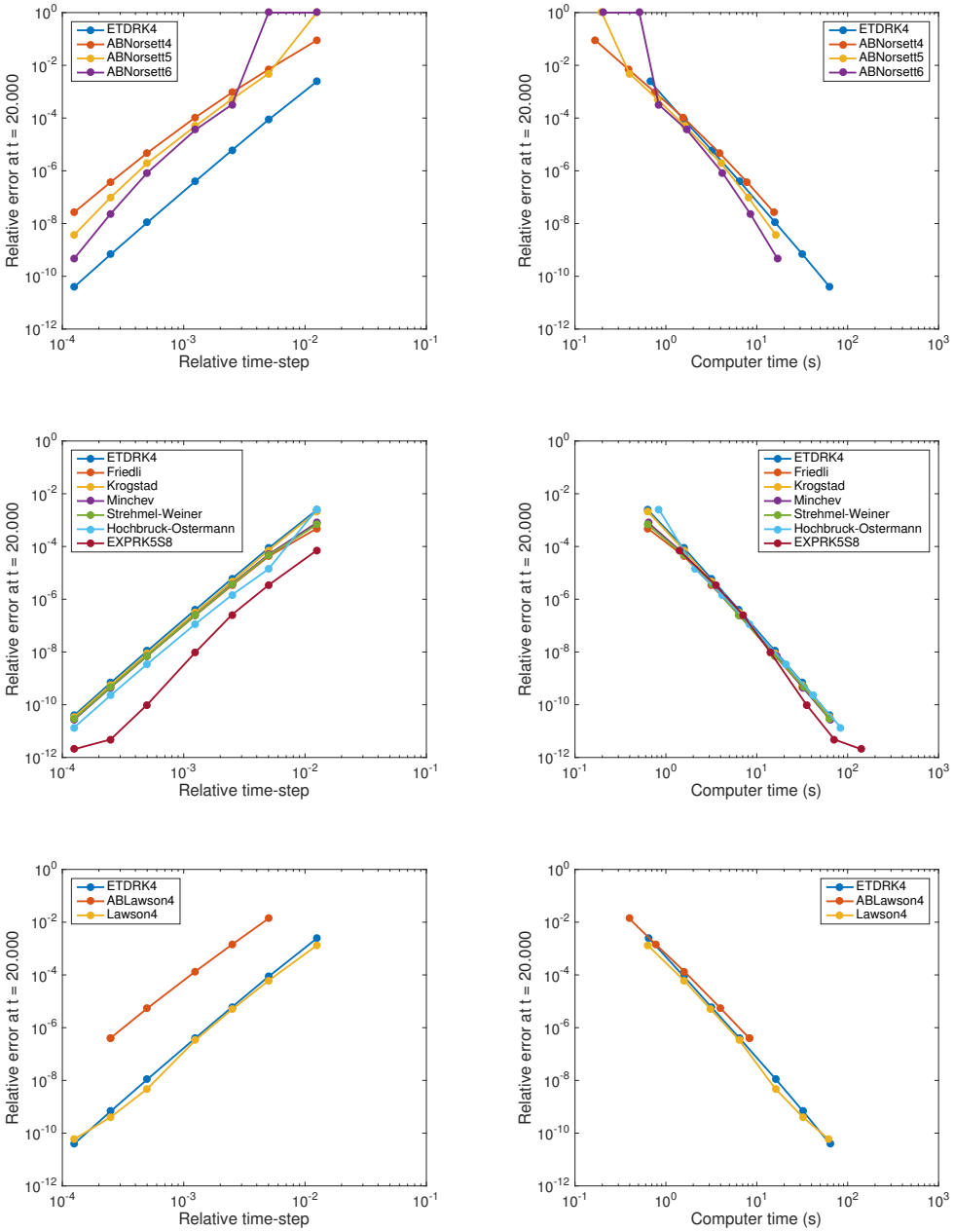

Figure C.15: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the 2D Swift–Hohenberg equation.*

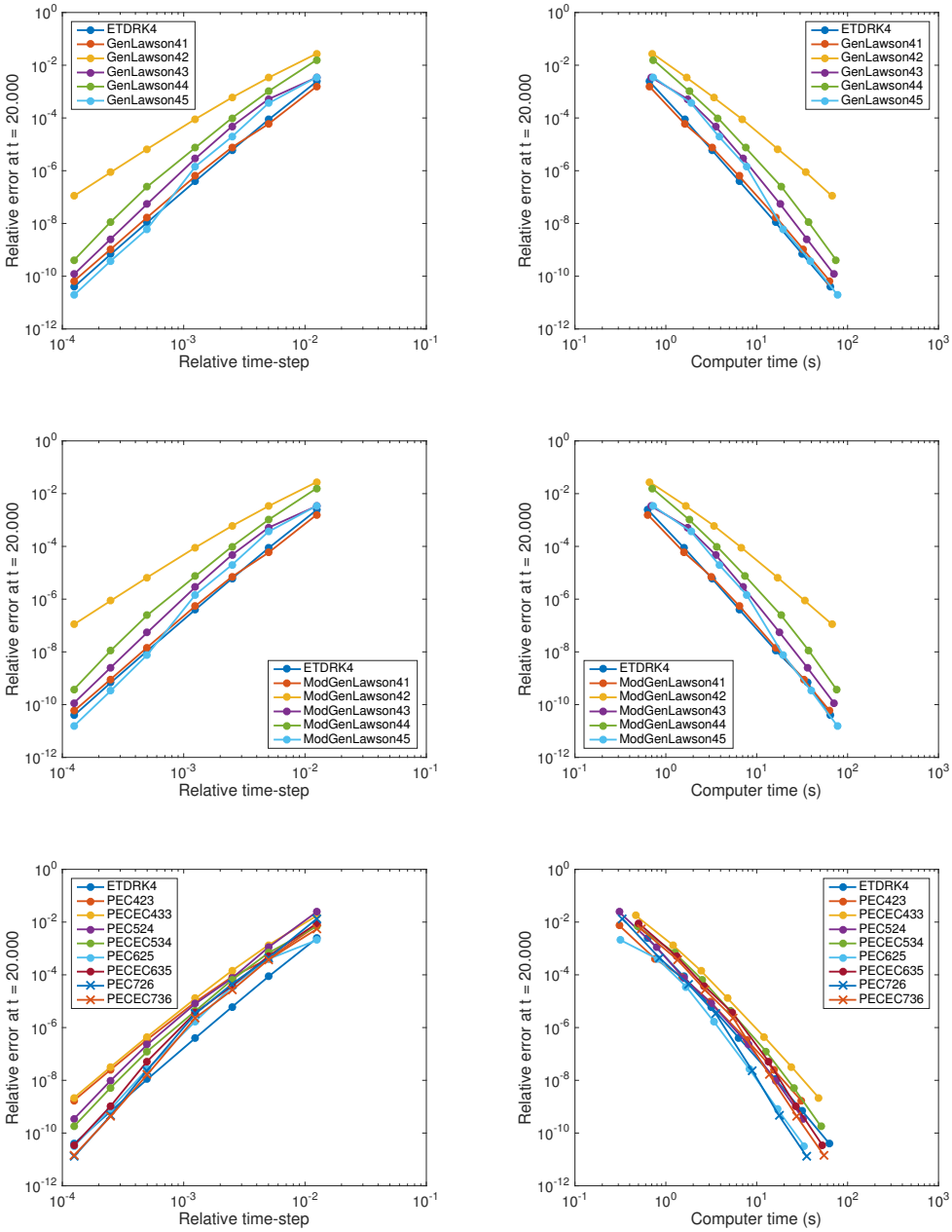

Figure C.16: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the 2D Swift–Hohenberg equation.*

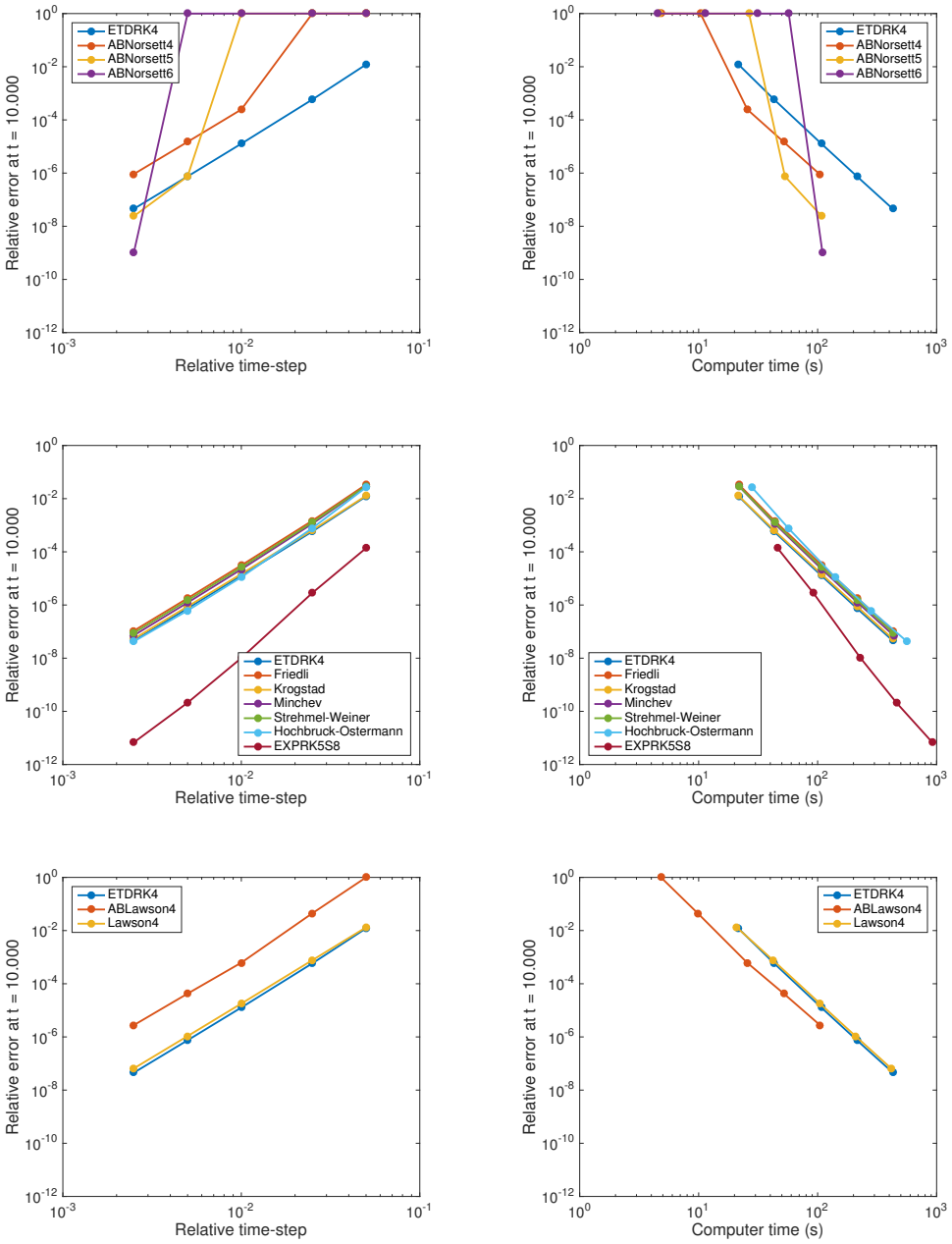

Figure C.17: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the 3D Ginzburg–Landau equation.*

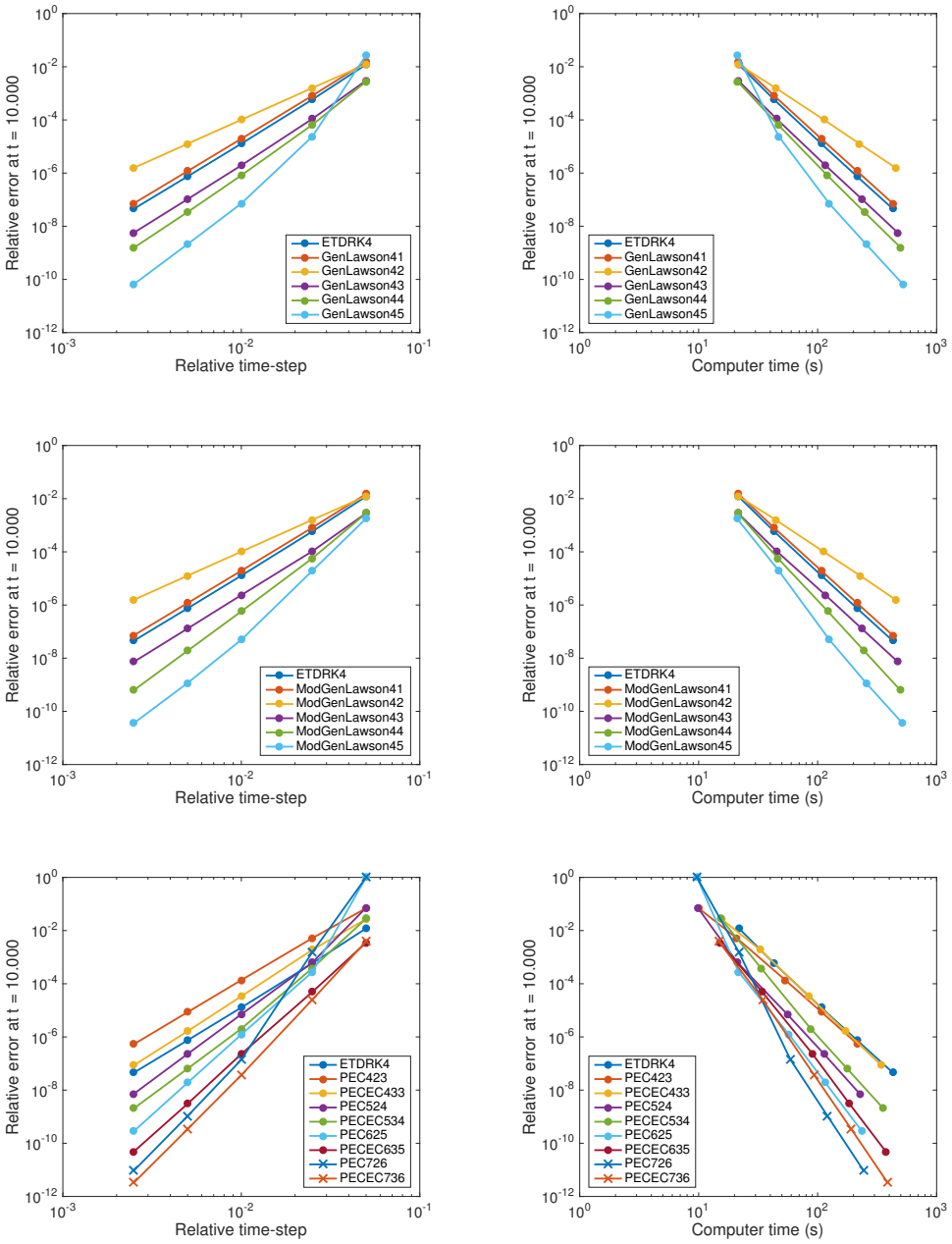

Figure C.18: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the 3D Ginzburg–Landau equation.*

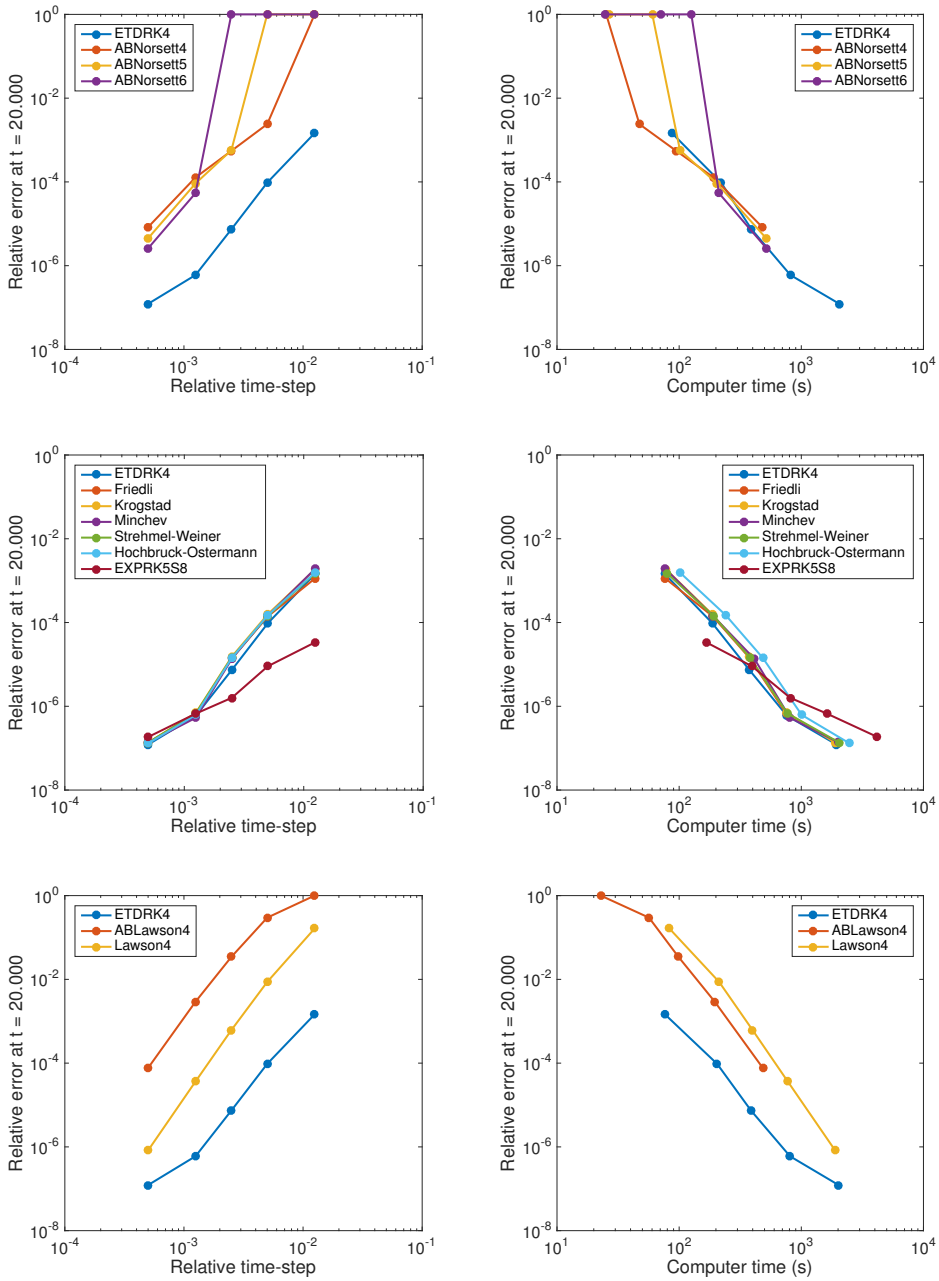

Figure C.19: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the 3D Schnakenberg equations.*

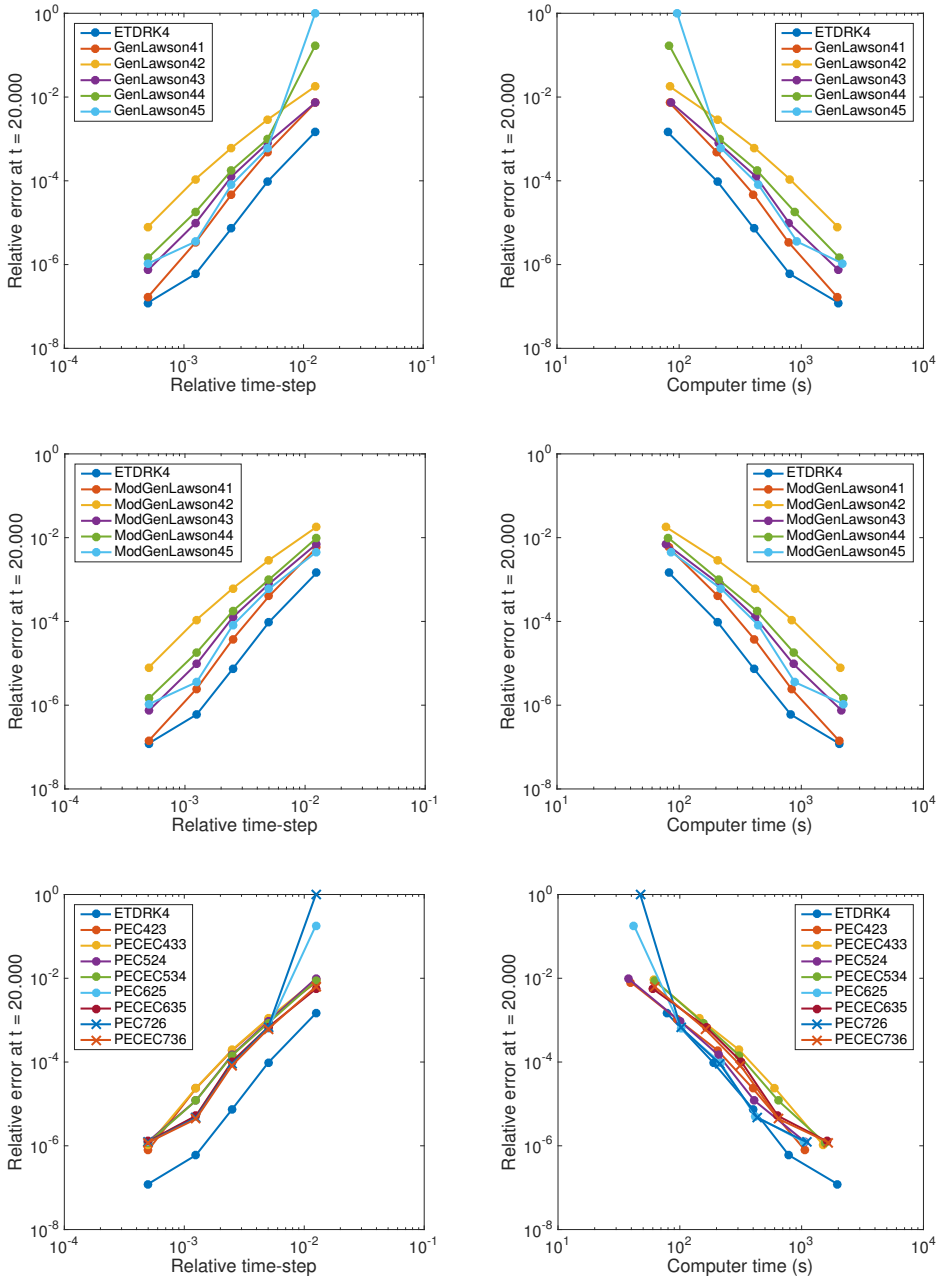

Figure C.20: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the 3D Schnakenberg equations.*

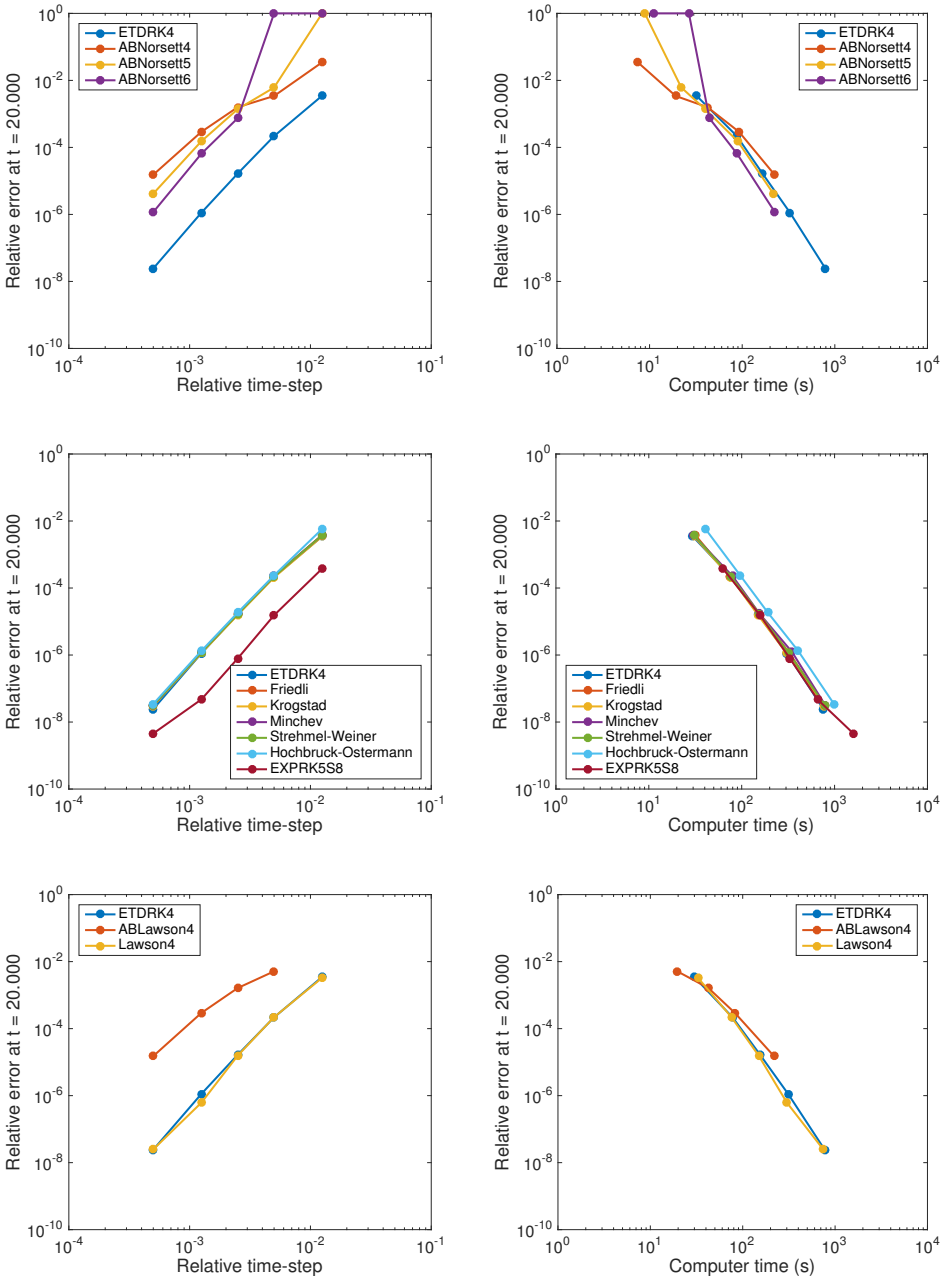

Figure C.21: *Accuracy versus time-step and computer time for the ETD Adams–Bashforth, ETD Runge–Kutta and Lawson schemes for the 3D Swift–Hohenberg equation.*

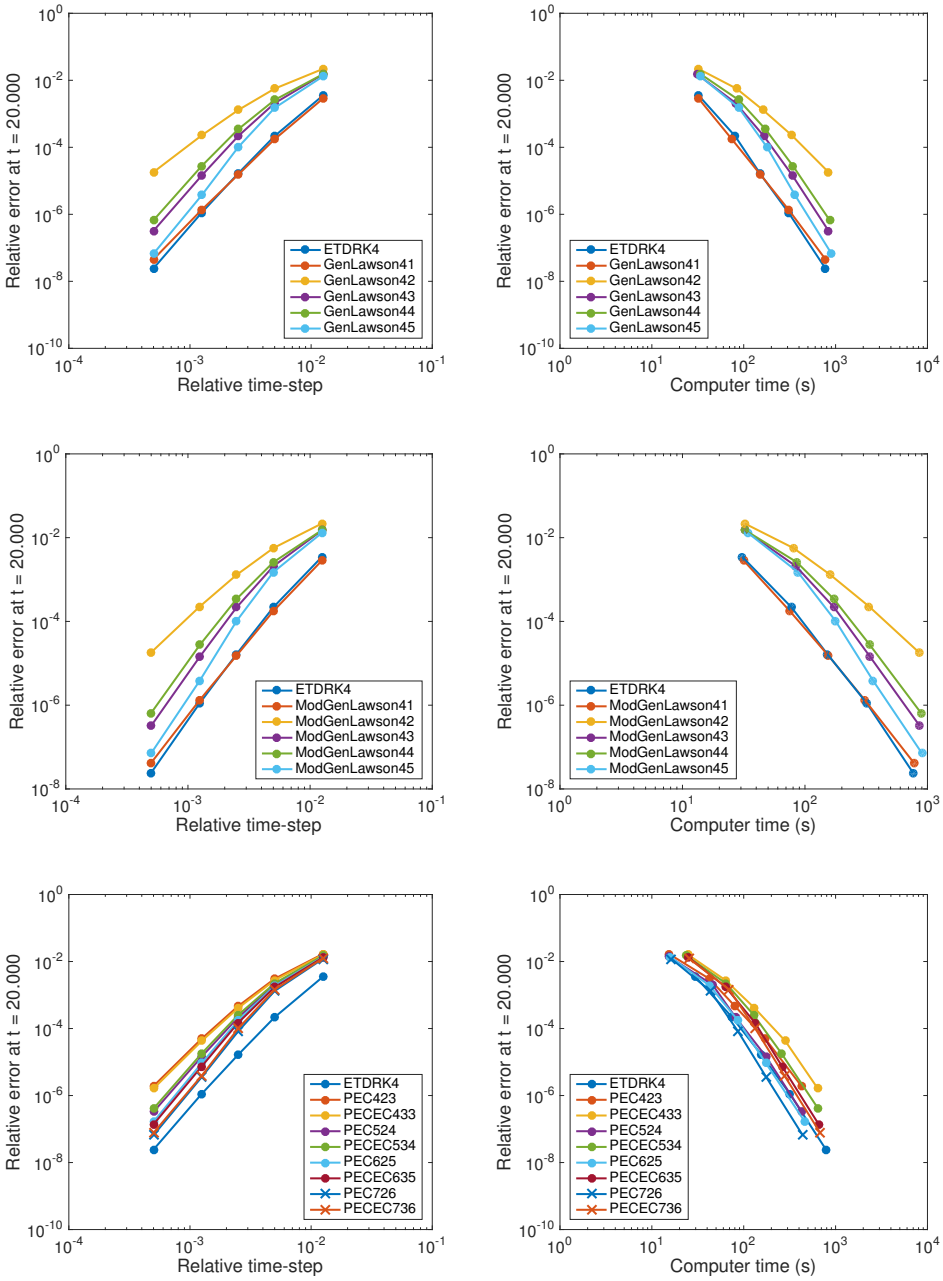

Figure C.22: *Accuracy versus time-step and computer time for the generalised Lawson, modified generalised Lawson and predictor-corrector schemes for the 3D Swift–Hohenberg equation.*

Appendix D

Eigenvalues of the Laplacian matrix on the sphere

The Laplacian matrix (5.28) is a discretized Laplacian, so one might expect that the eigenvalues are all real and nonpositive. For example, Gottlieb and Lustman [67] give a nontrivial proof that the discretization of the second derivative operator in the Chebyshev collocation method has real and negative eigenvalues. Here we show that essentially the same holds on the sphere for (5.28). (A slight difference is that we have one zero eigenvalue since the Laplacian of a constant is zero.)

Theorem D.1. *The eigenvalues of* $\mathbf{L}$ *in* (5.28) *are all real and nonpositive.*

Proof. It suffices to examine each block $\mathbf{L}_i = \mathbf{D}_m^{(2)} + \mathbf{T}_{\sin^2}^{-1}\mathbf{T}_{\cos\sin}\mathbf{D}_m + \mathbf{D}_n^{(2)}(i,i)\mathbf{T}_{\sin^2}^{-1}$. We first note that the eigenvalues of $\mathbf{L}_i$ are equal to those of the matrix pencil

$$\mathbf{A}_i - \lambda\mathbf{B}_i := \mathbf{T}_{\sin^2}\mathbf{D}_m^{(2)} + \mathbf{T}_{\cos\sin}\mathbf{D}_m + \mathbf{D}_n^{(2)}(i,i)\mathbf{I}_m - \lambda\mathbf{T}_{\sin^2}, \tag{D.1}$$

which corresponds to the generalized eigenvalue problem $\mathbf{A}_i x = \lambda\mathbf{B}_i x$. We shall prove that this pencil has negative real eigenvalues, regardless of i; we drop the subscript i for simplicity. Our proof proceeds as follows:

1. The eigenvalues of $\mathbf{A} - \lambda\mathbf{B}$ are the values of λ_0 for which the matrix $\mathbf{A} - \lambda_0\mathbf{B}$ has a zero eigenvalue.

2. For any fixed $\lambda_0 \in (-\infty, 0]$, all the eigenvalues of the matrix $\mathbf{A} - \lambda_0\mathbf{B}$ are real. Therefore we can define m real continuous functions $f_j(\lambda_0) := \lambda_j(\mathbf{A} - \lambda_0\mathbf{B})$ for $j = 1, \ldots, m$.

3. For every j, we have $f_j(0) \le 0$, and $f_j(\lambda_0) \ge 0$ for negative λ_0 with sufficiently large $|\lambda_0|$.

4. By the intermediate value theorem to each $f_j(\lambda_0)$, there is at least one root $f_j(\lambda_0) = 0$ in $\lambda_0 \in (-\infty, 0]$ for each j. It follows that $\mathbf{A} - \lambda\mathbf{B}$ has m real eigenvalues (counting multiplicities), hence so does $\mathbf{L}_i$.

The only nontrivial parts are the second step and the claim $f_j(0) \leq 0$.

We first prove the second step, i.e., that the matrix $\mathbf{C}(\lambda_0) := \mathbf{A} - \lambda_0\mathbf{B}$ has only real eigenvalues. To do this we apply the similarity transformation with the matrix $\mathbf{P} = \mathbf{I}_m(:, [1{:}2{:}m, 2{:}2{:}m])$, which gives $\mathbf{P}^T\mathbf{C}(\lambda_0)\mathbf{P} = \operatorname{diag}(\mathbf{C}_1(\lambda_0), \mathbf{C}_2(\lambda_0))$, a block-diagonal matrix with two $\frac{m}{2} \times \frac{m}{2} := \ell \times \ell$ blocks. Here $\mathbf{C}_1(\lambda_0)$ is tridiagonal with extra elements in the upper-right and lower-left corners (as in (D.3) below), and $\mathbf{C}_2(\lambda_0)$ is tridiagonal.[1]

For $\mathbf{C}_2(\lambda_0)$, we can verify that the products of the neighboring off-diagonals are positive,

$$\mathbf{C}_2(\lambda_0)_{j,j+1}\mathbf{C}_2(\lambda_0)_{j+1,j} > 0, \tag{D.2}$$

for all j (when $\lambda_0 = 0$ the product can be 0; we exclude this case for the moment and assume $\lambda_0 < 0$). Hence, $\mathbf{C}_2(\lambda_0)$ is diagonally similar to a symmetric matrix, thus its eigenvalues are all real.

It remains to prove that the eigenvalues of $\mathbf{C}_1(\lambda_0)$ are all real. Note that $\mathbf{C}_1(\lambda_0)$ is of the form

$$\mathbf{C}_1(\lambda_0) = \begin{pmatrix} \alpha & \beta' & & & & & & \beta' \\ \beta & \alpha_1 & \beta_1 & & & & & \\ & \beta_{\ell-2} & \alpha_2 & \beta_2 & & & & \\ & & \beta_{\ell-3} & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \beta_{\ell-1} & \\ & & & & & \beta_2 & \alpha_2 & \beta_{\ell-2} \\ \beta & & & & & & \beta_1 & \alpha_1 \end{pmatrix}. \tag{D.3}$$

Several properties of $\mathbf{C}_1(\lambda_0)$ are worth noting: (i) the $(\ell-1) \times (\ell-1)$ submatrix obtained by removing the first row and column is symmetric about the antidiagonal, both in the diagonal and off-diagonal elements (note the double appearance of β_i) and (ii) the products of the neighboring off-diagonal blocks are all positive $\beta_j\beta_{\ell-j-1} > 0$ for all j, and $\beta\beta' > 0$. In Lemma D.2 below we prove that any matrix (D.3) with such structure has only real eigenvalues, establishing that $f_j(\lambda_0)$ is real for any $\lambda_0 < 0$. The claim extends to $\lambda_0 = 0$ by continuity of the eigenvalues, completing the second step in the proof.

[1]It is possible to use this structure in the linear solvers. We did not do this in our experiments, as applying the permutation also requires $O(nm)$ operations.

It remains to show $f_j(0) \le 0$ for every j, that is, $\mathbf{A}$ has only nonpositive eigenvalues. Since $\mathbf{D}_n^{(2)}(i,i) \le 0$ for all i, it suffices to treat the case for which $\mathbf{D}_n^{(2)}(i,i) = 0$. We again examine $\mathbf{C}_1(0)$ and $\mathbf{C}_2(0)$ separately. For each of these, after deflating the zero eigenvalue (if present) we can apply a diagonal similarity transformation so that the Gershgorin disks, whose centers lie on the negative half line, do not contain the origin, implying that all the eigenvalues are nonpositive. This completes the proof of Theorem D.1. ∎

It remains to prove that the eigenvalues of matrices of the form in (D.3) are all real. A key fact is that a real tridiagonal matrix with the neighboring off-diagonals having the same sign is diagonally similar to a symmetric tridiagonal matrix, and an analogous result holds for arrowhead matrices.

Lemma D.2. *For any real matrix of the form* (D.3)*, with* $\beta_i \beta_{\ell-i-1} > 0$ *for all* i *and* $\beta\beta' > 0$*, all the eigenvalues are real.*

Proof. Denote the matrix by $\mathbf{C}$. We can apply a diagonal similarity transformation to the bottom-right $(\ell-1) \times (\ell-1)$ part $\mathbf{C}_2$, to obtain a symmetric matrix $\mathbf{D}^{-1}\mathbf{C}_2\mathbf{D}$. Since $\mathbf{C}_2$ is symmetric about the antidiagonal, so is the diagonal matrix $\mathbf{D}$; thus $\mathbf{D}^{-1}\mathbf{C}_2\mathbf{D}$ is both symmetric and symmetric about the antidiagonal, and so the transformation $\widehat{\mathbf{C}} = \left[\begin{smallmatrix} 1 & \\ & \mathbf{D}^{-1} \end{smallmatrix}\right] \mathbf{C} \left[\begin{smallmatrix} 1 & \\ & \mathbf{D} \end{smallmatrix}\right]$ preserves the property that the off-diagonal parts of the first row and column are parallel (when one is transposed). Now let $\mathbf{Q}$ be an orthogonal matrix of eigenvectors of $\mathbf{D}^{-1}\mathbf{C}_2\mathbf{D}$ such that $\mathbf{Q}^T(\mathbf{D}^{-1}\mathbf{C}_2\mathbf{D})\mathbf{Q}$ is diagonal, and consider the matrix $\widetilde{\mathbf{C}} = \left[\begin{smallmatrix} 1 & \\ & \mathbf{Q}^T \end{smallmatrix}\right] \widehat{\mathbf{C}} \left[\begin{smallmatrix} 1 & \\ & \mathbf{Q} \end{smallmatrix}\right]$. By the antidiagonal symmetry of $\mathbf{D}^{-1}\mathbf{C}_2\mathbf{D}$, each eigenvector (column of $\mathbf{Q}$) has the property that it is either in the form $[v_1, v_2, \ldots, -v_2, -v_1]^T$ or $[v_1, v_2, \ldots, v_2, v_1]^T$. Therefore, $\widetilde{\mathbf{C}}$ is an arrowhead matrix with the property that for every j, we have either $\widetilde{\mathbf{C}}_{j,1} = \widetilde{\mathbf{C}}_{1,j} = 0$, or $\widetilde{\mathbf{C}}_{j,1}\widetilde{\mathbf{C}}_{1,j} > 0$. It follows that there exists a diagonal similarity transformation that brings $\widetilde{\mathbf{C}}$ to symmetric form, hence has real eigenvalues. ∎

References

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York, 1965.

[2] J. C. Adams and P. N. Swarztrauber, *SPHEREPACK* 3.0*: a model development facility*, Mon. Wea. Rev., 127 (1999), pp. 1872–1878.

[3] A. H. Al-Mohy and N. J. Higham, *A new scaling and squaring method for the matrix exponential*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 970–989.

[4] ——, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM J. Sci. Comput., 33 (2011), pp. 488–511.

[5] S. M. Allen and J. W. Cahn, *A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening*, Acta Metall., 27 (1979), pp. 1085–1095.

[6] E. Anderson et al., *LAPACK Users' Guide*, SIAM, Philadelphia, 1999.

[7] A. Arakawa, *Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow*, J. Comput. Phys., 1 (1966), pp. 119–143.

[8] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 797–823.

[9] H. A. Ashi, L. J. Cummings, and P. C. Matthews, *Comparison of methods for evaluating functions of a matrix exponential*, Appl. Numer. Math., 59 (2009), pp. 468–486.

[10] K. Atkinson and W. Han, *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction*, Springer, Berlin, 2012.

[11] J. Aurentz, *Fast algorithms for spectral differentiation matrices*, Electron. Trans. Numer. Anal., 44 (2015), pp. 281–288.

[12] A. P. Austin and K. Xu, *On the numerical stability of the second barycentric formula for trigonometric interpolation in shifted equispaced points*, IMA J. Numer. Anal., to appear (2017).

[13] H. Berland, B. Skaflestad, and W. M. Wright, *EXPINT—A MATLAB package for exponential integrators*, ACM Trans. Math. Softw. (TOMS), 33 (2007), pp. 4:1–4:17.

[14] G. Beyklin, J. M. Keiser, and L. Vozovoi, *A new class of time discretization schemes for the solution of nonlinear PDEs*, J. Comput. Phys., 147 (1998), pp. 362–387.

[15] S. Bhattacharya, *Galerkin model for Turing patterns on a sphere*, Phys. Rev. E, 72 (2005), p. 036208.

[16] A. Birkisson and T. A. Driscoll, *Automatic Fréchet differentiation for the numerical solution of boundary-value problems*, ACM Trans. Math. Softw. (TOMS), 38 (2012), pp. 26:1–26:29.

[17] N. J. Bootland, *Exponential integrators for stiff PDEs*, Master's thesis, University of Oxford, 2014.

[18] A. V. Borisov and I. S. Mamaev, *The restricted two-body problem in constant curvature spaces*, Celest. Mech. Dyn. Astron., 96 (2006), pp. 1–17.

[19] ——, *Relations between integrable systems in plane and curved spaces*, Celest. Mech. Dyn. Astron., 99 (2007), pp. 253–260.

[20] A. V. Borisov, I. S. Mamaev, and A. A. Kilin, *Two-body problem on a sphere: reduction, stochasticity, periodic orbits*, Regul. Chaotic Dyn., 9 (2004), pp. 265–279.

[21] J. P. Boyd, *The choice of spectral functions on a sphere for boundary and eigenvalue problems: A comparison of Chebyshev, Fourier and associated Legendre expansions*, Mon. Wea. Rev., 106 (1978), pp. 1184–1191.

[22] ——, *Chebyshev and Fourier Spectral Methods*, Dover Publications, Mineola, 2001.

[23] ——, *Computing the zeros, maxima and inflection points of Chebyshev, Legendre and Fourier series: Solving transcendental equations by spectral interpolation and polynomial rootfinding*, J. Eng. Math., 56 (2006), pp. 203–219.

[24] H. Brézis, *Functional Analysis, Sobolev Spaces and Partial Differential Equations*, Springer, New York, 2010.

[25] T. Buvoli, *A class of exponential integrators based on spectral deferred correction*, submitted (2014).

[26] J. W. Cahn and J. E. Hilliard, *Free energy of a nonuniform system. I. Interfacial free energy*, The Journal of Chemical Physics, 28 (1958), pp. 258–267.

[27] M. P. Calvo, J. de Frutos, and J. Novo, *Linearly implicit Runge–Kutta methods for advection-reaction-diffusion equations*, Appl. Numer. Math., 37 (2001), pp. 535–549.

[28] M. P. Calvo and C. Palencia, *A class of multistep exponential integrators for semilinear problems*, Numer. Math., 102 (2006), pp. 367–381.

[29] B. Cano and A. Gonzáles-Pachón, *Exponential time integration of solitary waves of cubic Schrödinger equation*, Appl. Numer. Math., 91 (2015), pp. 26–45.

[30] ——, *Projected explicit Lawson methods for the integration of Schrödinger equation*, Numer. Methods Partial Differential Eq., 31 (2015), pp. 78–104.

[31] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*, Springer, Berlin, 2007.

[32] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang, *Extrapolation-based implicit-explicit general linear methods*, Numer. Algor., 65 (2014), pp. 377–399.

[33] J. F. Cariñena and M. F. Rañada, *Central potential on spaces of constant curvature: the Kepler problem on the two-dimensional sphere S^2 and the hyperbolic plane H^2*, J. Math. Phys., 46 (2005), p. 052702.

[34] J. Certaine, *The solution of ordinary differential equations with large time constants*, in Mathematical methods for digital computers, A. Ralston and H. S. Wilf, eds., Wiley, New York, 1960, pp. 128–132.

[35] K.-C. Chen, *Existence and minimizing properties of retrograde orbits to the three-body problem with various choices of masses*, Ann. of Math., 167 (2008), pp. 325–348.

[36] A. Chenciner and R. Montgomery, *A remarkable periodic solution of the three-body problem in the case of equal masses*, Ann. of Math., 152 (2000), pp. 881–901.

[37] H.-B. Cheong, *Double Fourier series on a sphere: Applications to elliptic and vorticity equations*, J. Comput. Phys., 157 (2000), pp. 327–349.

[38] V. A. Chernoivan and I. S. Mamaev, *The restricted two-body problem and the Kepler problem in the constant curvature spaces*, Regul. Chaotic Dyn., 4 (1999), pp. 112–124.

[39] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*, Krieger Publishing Company, Malabar, 1955.

[40] E. M. Constantinescu and A. Sandu, *Extrapolated implicit-explicit time stepping*, SIAM J. Sci. Comput., 31 (2010), pp. 4452–4477.

[41] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp., 19 (1965), pp. 297–301.

[42] S. M. Cox and P. C. Matthews, *Exponential time differencing for stiff systems*, J. Comput. Phys., 176 (2002), pp. 430–455.

[43] T. A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2006.

[44] F. Diacu, *Relative equilibria of the curved N-body problem*, Springer, New York, 2012.

[45] F. Diacu and S. Kordlou, *Rotopulsators of the curved N-body problem*, J. Differential Equations, 255 (2013), pp. 2709–2750.

[46] F. Diacu, R. Martínez, E. Pérez-Chavala, and C. Simó, *On the stability of tetrahedral relative equilibria in the positively curved 4-body problem*, Phys. D, 256-257 (2013), pp. 21–35.

[47] F. Diacu and E. Pérez-Chavala, *Homographic solutions of the curved 3-body problem*, J. Differential Equations, 250 (2011), pp. 340–366.

[48] F. Diacu, E. Pérez-Chavala, and J. G. Reyes Victoria, *An intrinsic approach in the curved n-body problem. The negative curvature case*, J. Differential Equations, 252 (2012), pp. 4529–4562.

[49] F. Diacu, E. Pérez-Chavala, and M. Santoprete, *The n-body problem in spaces of constant curvature. Part I: Relative equilibria*, J. Nonlinear Sci., 22 (2012), pp. 247–266.

[50] ——, *The n-body problem in spaces of constant curvature. Part II: Singularities*, J. Nonlinear Sci., 22 (2012), pp. 267–275.

[51] T. A. Driscoll, *A composite Runge–Kutta method for the spectral solution of semilinear PDEs*, J. Comput. Phys., 182 (2002), pp. 357–367.

[52] T. A. Driscoll, F. Bornemann, and L. N. Trefethen, *The chebop system for automatic solution of differential equations*, BIT, 48 (2008), pp. 701–723.

[53] T. A. Driscoll, N. Hale, and L. N. Trefethen, eds., *Chebfun Guide*, Pafnuty Publications, Oxford, 2014; see also `www.chebfun.org`.

[54] Q. Du and W. Zhu, *Analysis and applications of the exponential time differencing schmes and their contour integral modifications*, BIT, 45 (2005), pp. 307–328.

[55] Y. Du, *The heterogeneous Allen–Cahn equation in a ball: Solutions with layers and spikes*, J. Differential Equations, 244 (2008), pp. 117–169.

[56] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT, 40 (2000), pp. 241–266.

[57] M. S. P. Eastham, *The spectral theory of periodic differential equations*, Scottish Academic Press, Edinburgh, 1973.

[58] B. L. Ehle and J. D. Lawson, *Generalized Runge–Kutta processes for stiff initial-value problems*, J. Inst. Maths. Appl., 16 (1975), pp. 11–21.

[59] G. Floquet, *Sur les équations différentielles linéaires à coefficients périodiques*, Annales scientifiques de l'E.N.S., 12 (1883), pp. 47–88.

[60] B. Fornberg, *On a Fourier method for the Iintegration of Hyperbolic Equations*, SIAM J. Numer. Anal., 12 (1975), pp. 509–528.

[61] ——, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, 1998.

[62] B. Fornberg and T. A. Driscoll, *A fast spectral algorithm for nonlinear wave equations with linear dispersion*, J. Comput. Phys., 155 (1999), pp. 456–467.

[63] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2015.

[64] B. Fornberg and D. Merrill, *Comparison of finite difference- and pseudospectral methods for convective flow over a sphere*, Geophys. Res. Lett., 24 (1997), pp. 3245–3248.

[65] A. Friedli, *Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme*, in Numerical Treatment of Differential Equations, R. Burlirsch, R. Grigorieff, and J. Schröder, eds., Springer, Berlin, 1978, pp. 35–50.

[66] F. Garcia, L. Bonaventura, M. Net, and J. Sánchez, *Exponential versus IMEX high-order time integrators for thermal convection in rotating spherical shells*, J. Comput. Phys., 264 (2014), pp. 41–54.

[67] D. Gottlieb and L. Lustman, *The spectrum of the Chebyshev collocation operator for the heat equation*, SIAM J. Numer. Anal., 20 (1983), pp. 909–921.

[68] E. Hairer, G. Bader, and C. Lubich, *On the stability of semi-implicit methods for ordinary differential equations*, BIT, 22 (1982), pp. 211–232.

[69] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer, New York, 1991.

[70] P. Henrici, *Barycentric formulas for interpolating trigonometric polynomials and their conjugates*, Numer. Math., 33 (1979), pp. 225–234.

[71] P. Henrici, *Applied and Computational Complex Analysis*, Wiley, New York, 1986.

[72] J. Hersch, *Contribution à la méthode des équations aux différences*, Z. Angew. Math. und Phys., 9 (1958), pp. 129–180.

[73] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, Cambridge, 2007.

[74] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1179–1193.

[75] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM J. Sci. Comput., 19 (1998), pp. 1552–1574.

[76] M. Hochbruck and A. Ostermann, *Explicit exponential Runge–Kutta methods for semilinear parabolic problems*, SIAM J. Numer. Anal., 43 (2005), pp. 1069–1090.

[77] ——, *Exponential integrators*, Acta Numer., 19 (2010), pp. 209–286.

[78] ——, *Exponential multistep methods of Adams-type*, BIT, 51 (2011), pp. 889–908.

[79] M. Hochbruck, A. Ostermann, and J. Schweitzer, *Exponential Rosenbrock-type methods*, SIAM J. Numer. Anal., 47 (2009), pp. 786–803.

[80] W. G. Horner, *A new method of solving numerical equations of all orders by continuous approximation*, Phil. Trans. Roy. Soc. Lon., 109 (1819), pp. 308–335.

[81] W. Hundsdorfer and S. J. Ruuth, *IMEX extensions of linear multistep methods with general monotonicity and boundedness properties*, J. Comput. Phys., 225 (2007), pp. 2016–2042.

[82] M. Javed and L. N. Trefethen, *The Remez algorithm for trigonometric approximation of periodic functions*, submitted (2015).

[83] A.-K. Kassam, *Solving reaction-diffusion equations 10 times faster*, Tech. Rep. 1192, Numerical Analysis Group, University of Oxford, 2003.

[84] A.-K. Kassam and L. N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233.

[85] Y. Katznelson, *An Introduction to Harmonic Analysis*, Cambridge University Press, Cambridge, 2004.

[86] C. A. Kennedy and M. H. Carpenter, *Additive Runge–Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.

[87] A. A. Kilin, *Libration points in spaces S^2 and L^2*, Regul. Chaotic Dyn., 4 (1999), pp. 91–103.

[88] C. Klein, *Fourth order time-stepping for low dispersion Korteweg–de Vries and nonlinear Schrödinger equations*, Electron. Trans. Numer. Anal., 29 (2008), pp. 116–135.

[89] C. Klein and K. Roidot, *Fourth order time-stepping for Kadomtsev–Petviashvili and Davey–Stewartson equations*, SIAM J. Sci. Comput., 33 (2011), pp. 3333–3356.

[90] D. Korteweg and G. de Vries, *On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves*, Philos. Mag. (Ser. 5), 39 (1895), pp. 422–443.

[91] H.-O. Kreiss and J. Oliger, *Comparison of accurate methods for the integration of hyperbolic equations*, Tellus, 24 (1972), pp. 199–215.

[92] ——, *Stability of the Fourier Method*, SIAM J. Numer. Anal., 16 (1979), pp. 421–433.

[93] S. Krogstad, *Generalized integrating factor methods for stiff PDEs*, J. Comput. Phys., 203 (2005), pp. 72–88.

[94] Y. Kuramoto and T. Tsuzuki, *Persistent propagation of concentration waves in dissipative media far from thermal equilibrium*, Progress of Theoretical Physics, 55 (1976), pp. 356–369.

[95] J.-L. Lagrange, *Essai sur le problème des trois corps*, in Prix de l'Académie Royale des Sciences, vol. IX, Paris, 1772, pp. 229–332.

[96] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Stand., 45 (1950), pp. 255–282.

[97] J. Lawson, *Generalized Runge–Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal., 4 (1967), pp. 372–380.

[98] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide*, SIAM, Philadelphia, 1998.

[99] J. Loffeld and M. Tokman, *Comparative performance of exponential, implicit, and explicit integrators for stiff systems of ODEs*, J. Comput. Appl. Math., 241 (2013), pp. 45–67.

[100] V. T. Luan and A. Ostermann, *Explicit exponential Runge–Kutta methods of high order for parabolic problems*, J. Comput. Appl. Math., 256 (2014), pp. 168–179.

[101] ——, *Exponential Rosenbrock methods of order five—construction, analysis and numerical comparisons*, J. Comput. Appl. Math., 255 (2014), pp. 417–431.

[102] V. T. Luan, M. Tokman, and G. Rainwater, *Preconditioned implicit-exponential integrators (IMEXP) for stiff PDEs*, J. Comput. Phys., 335 (2017), pp. 846–864.

[103] S. H. Lui, *Legendre spectral collocation in space and time for PDEs*, Numer. Math., 136 (2017), pp. 75–99.

[104] R. H. Martin, *Nonlinear Operators and Differential Equations in Banach Spaces*, Wiley, New York, 1976.

[105] E. Mathieu, *Mémoire sur le mouvement vibratoire d'une membrane de forme ellipitque*, Journal de mathématiques pures et appliquées, 13 (1868), pp. 137–203.

[106] P. L. Maupertuis, *Accord de différentes loix de la nature qui avoient jusqu'ici paru incompatibles*, in Mémoires de l'Académie Royale des Sciences de l'année 1744, Paris, 1748, pp. 417–426.

[107] P. E. Merilees, *The pseudospectral approximation applied to the shallow water equations on a sphere*, Atmosphere, 11 (1973), pp. 13–20.

[108] B. V. Minchev, *Exponential integrators for semilinear problems*, PhD thesis, University of Bergen, 2004.

[109] B. V. Minchev and W. M. Wright, *A review of exponential integrators for first order semi-linear problems*, Tech. Rep. 2/2005, Norwegian University of Science and Technology, 2005.

[110] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev., 45 (2003), pp. 3–49.

[111] H. Montanelli, *Computing hyperbolic choreographies*, Regul. Chaotic Dyn., 21 (2016), pp. 523–531.

[112] H. Montanelli and N. Bootland, *Solving periodic semilinear stiff PDEs in 1D, 2D and 3D with exponential integrators*, submitted (2016).

[113] H. Montanelli and N. I. Gushterov, *Computing planar and spherical choreographies*, SIAM J. Appl. Dyn. Syst., 15 (2016), pp. 235–256.

[114] H. Montanelli and Y. Nakatsukasa, *Fourth-order time-stepping for stiff PDEs on the sphere*, submitted (2017).

[115] C. Moore, *Braids in classical dynamics*, Phys. Rev. Lett., 70 (1993), pp. 3675–3679.

[116] P. K. Newton, *The N-vortex Problem*, Springer, New York, 2001.

[117] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2006.

[118] S. P. Nørsett, *An A-stable modification of the Adams–Bashforth methods*, in Conference on the numerical solution of differential equations (Dundee, 1969), Springer, Berlin, 1969, pp. 214–219.

[119] S. Olver and A. Townsend, *A fast and well-conditioned spectral method*, SIAM Rev., 55 (2013), pp. 462–489.

[120] S. A. Orszag, *Numerical methods for the simulation of turbulence*, Phys. Fluids, 12 (1969), pp. 250–257.

[121] ——, *Transform method for the calculation of vector-coupled sums: appliaction to the spectral form of the vorticity equation*, J. Atmos. Sci., 27 (1970), pp. 890–895.

[122] ——, *Accurate solution of the Orr–Sommerfeld stability equation*, J. Fluid Mech., 50 (1971), pp. 689–703.

[123] ——, *Numerical simulation of incompressible flows within simple boundaries: Accuracy*, J. Fluid Mech., 49 (1971), pp. 75–112.

[124] ——, *Numerical simulation of incompressible flows within simple boundaries: Galerkin (spectral) representations*, Studies in Appl. Math., 50 (1971), pp. 293–327.

[125] ——, *Fourier series on spheres*, Mon. Wea. Rev., 102 (1974), pp. 56–75.

[126] A. Ostermann, M. Thalhammer, and W. M. Wright, *A class of explicit exponential general linear methods*, BIT, 46 (2006), pp. 409–431.

[127] L. Pareschi and G. Russo, *Implicit-explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation*, J. Sci. Comput., 25 (2005), pp. 129–155.

[128] E. Pérez-Chavela and J. G. Reyes Victoria, *An intrinsic approach in the curved n-body problem. The positive curvature case*, Trans. Amer. Math. Soc., 364 (2012), pp. 3805–3827.

[129] R. Peyret, *Spectral Methods for Incompressible Viscous Flow*, Springer, New York, 2002.

[130] N. A. Phillips, *The general circulation of the atmosphere: a numerical experiment*, Quart. J. Roy. Meteor. Soc., 82 (1956), pp. 123–164.

[131] ——, *An Example of Non-Linear Computational Instability*, in The Atmosphere and the Sea in Motion, The Rockefeller Institute Press, New York, 1959.

[132] L. Pismen and J. Rubinstein, *Dynamics of disclinations in liquid crystals*, Quart. Appl. Math., 50 (1992), pp. 535–545.

[133] H. Poincaré, *Les Méthodes Nouvelles de la Mécanique Céleste, Vol. I: Solutions périodiques, Non-existence des Intégrales Uniformes, Solutions Asymptotiques*, Gauthier-Villars et Fils, Paris, 1892.

[134] ——, *Sur les solutions périodiques et le principe de moindre action*, Comptes Rendus de l'Académie des Sciences, 123 (1896), pp. 915–918.

[135] Y. Pomeau, S. Zaleski, and P. Manneville, *Dislocation motion in cellular structures*, Phys. Rev. A, 27 (1983), pp. 2710–2726.

[136] D. A. Pope, *An exponential method of numerical integration of ordinary differential equations*, Communications of the ACM, 6 (1963), pp. 491–493.

[137] G. Rainwater and M. Tokman, *A new class of split exponential propagation iterative methods of Runge–Kutta type (sEPIRK) for semilinear systems of ODEs*, J. Comput. Phys., 269 (2014), pp. 40–60.

[138] ——, *A new approach to constructing efficient stiffly accurate EPIRK methods*, J. Comput. Phys., 323 (2016), pp. 283–309.

[139] M. E. Rognes, D. A. Ham, C. J. Cotter, and A. T. T. McRae, *Automating the solution of PDEs on the sphere and other manifolds in FEniCS* 1.2, Geosci. Model Dev., 6 (2013), pp. 2099–2119.

[140] V. Rokhlin and M. Tygert, *Fast algorithms for spherical harmonic expansions*, SIAM J. Sci. Comput., 27 (2006), pp. 1903–1928.

[141] H. H. Rosenbrock, *Some general implicit processes for the numerical solution of differential equations*, Computer, 5 (1963), pp. 329–330.

[142] J. Rubinstein and P. Sternberg, *On the slow motion of vortices in the Ginzburg–Landau heat flow*, SIAM J. Math. Anal., 26 (1995), pp. 1452–1466.

[143] T. Schmelzer and L. N. Trefethen, *Evaluating matrix functions for exponential integrators via Carathéodory–Fejér approximation and contour integrals*, Electron. Trans. Numer. Anal., 29 (2007), pp. 1–18.

[144] J. Schnakenberg, *Simple chemical reaction systems with limit cycle behaviour*, J. Theor. Biol., 81 (1979), pp. 389–400.

[145] D. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Math. Comp., 24 (1970), pp. 647–656.

[146] A. V. Shchepetilov, *Calculus and Mechanics on Two-Point Homogeneous Riemannian Spaces*, Springer, Berlin, 2006.

[147] J. Shen, *Efficient spectral-Galerkin methods IV. Spherical geometries*, SIAM J. Sci. Comput., 20 (1999), pp. 1438–1455.

[148] J. Shen, T. Tang, and L. L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, Springer, Berlin, 2011.

[149] C. Simó, *New families of solutions in N-body problems*, in Proceedings of the Third European Congress of Mathematics, Basel, 2001, Birkhäuser.

[150] G. I. Sivashinsky, *Nonlinear analysis of hydrodymanic instability in laminar flames—I. Derivation of basic equations*, Acta Astronaut., 4 (1977), pp. 1177–1206.

[151] R. M. Slevinsky, *Fast and backward stable transforms between spherical harmonic expansions and bivariate Fourier series*, submitted (2017).

[152] K. Stewartson and J. T. Stuart, *A non-linear instability theory for a wave system in plane Poiseuille flow*, J. Fluid Mech., 48 (1971), pp. 529–545.

[153] K. Strehmel and R. Weiner, *Behandlung steifer Anfangswertprobleme gewöhnlicher Differentialgleichungen mit adaptiven Runge–Kutta-Methoden*, Computing, 29 (1982), pp. 153–165.

[154] J. Swift and P. C. Hohenberg, *Hydrodynamic fluctuations at the convective instability*, Phys. Rev. A, 15 (1977), pp. 319–328.

[155] E. Tadmor, *The exponential accuracy of Fourier and Chebyshev differencing methods*, SIAM J. Numer. Anal., 23 (1986), pp. 1–10.

[156] ——, *Stability analysis of finite-difference, pseudospectral and Fourier–Galerkin approximations*, SIAM Rev., 29 (1987), pp. 525–555.

[157] H. Takaoka, *Local well-posedness of the nonlinear Schrödinger equations on the sphere for data in modulation spaces*, Commun. Part. Diff. Eq., 41 (2016), pp. 732–747.

[158] G. Teschl, *Ordinary Differential Equations and Dynamical Systems*, American Mathematical Society, Providence, 2012.

[159] M. Tokman, *Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods*, J. Comput. Phys., 213 (2006), pp. 748–776.

[160] ——, *A new class of exponential propagation iterative methods of Runge–Kutta type (EPIRK)*, J. Comput. Phys., 230 (2011), pp. 8762–8778.

[161] M. Tokman and J. Loffeld, *Efficient design of exponential-Krylov integrators for large scale computing*, Procedia Comp. Sci., 1 (2012), pp. 229–237.

[162] M. Tokman, J. Loffeld, and P. Tranquilli, *New adaptive exponential propagation iterative methods of Runge–Kutta type*, SIAM J. Sci. Comput., 34 (2012), pp. A2650–A2669.

[163] A. Townsend, H. Wilber, and G. B. Wright, *Computing with functions in spherical and polar geometries, I. The sphere*, SIAM J. Sci. Comput., 38 (2016), pp. C403–C425.

[164] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, Philadelphia, 2000.

[165] L. N. Trefethen and M. H. Gutknecht, *The Carathéodory–Fejér method for real rational approximation*, SIAM J. Numer. Anal., 20 (1983), pp. 420–436.

[166] L. N. Trefethen and J. A. C. Weideman, *The exponentially convergent trapezoidal rule*, SIAM Rev., 56 (2014), pp. 385–458.

[167] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer, *Talbot quadratures and rational approximations*, BIT, 46 (2006), pp. 653–670.

[168] P. H. Trinh and M. J. Ward, *The dynamics of localized spot patterns for reaction-diffusion systems on the sphere*, Nonlinearity, 29 (2016), pp. 766–806.

[169] A. M. Turing, *The chemical basis of morphogenesis*, Phil. Trans. Roy. Soc. Lon. (Ser. B), 237 (1952), pp. 37–72.

[170] M. Tygert, *Fast algorithms for spherical harmonic expansions, II*, J. Comput. Phys., 227 (2008), pp. 4260–4279.

[171] ——, *Fast algorithms for spherical harmonic expansions, III*, J. Comput. Phys., 229 (2010), pp. 6181–6192.

[172] P. J. van der Houwen, *Construction of Integration Formulas for Initial Value Problems*, North-Holland Publishing Co., Amsterdam, 1977.

[173] R. S. Varga, *Geršgorin and His Circles*, Springer, Berlin, 2004.

[174] G. B. Wright, M. Javed, H. Montanelli, and L. N. Trefethen, *Extension of Chebfun to periodic functions*, SIAM J. Sci. Comput., 37 (2015), pp. C554–C573.

[175] S. Y. K. Yee, *Solution of Poisson's equation on a sphere by truncated double Fourier series*, Mon. Wea. Rev., 109 (1980), pp. 501–505.

[176] A. Zygmund, *Trigonometric Series*, Cambridge University Press, Cambridge, 1959.