A sketched finite element method for elliptic models

Robert Lung[a,1,*], Yue Wu[a,b,c,2], Dimitris Kamilis[a,2], Nick Polydorides[a,c,2]

[a] *School of Engineering, University of Edinburgh, Edinburgh, EH9 3FB, UK.*
[b] *Mathematical Institute, University of Oxford, Oxford, OX2 6GG, UK.*
[c] *The Alan Turing Institute, London, UK.*

Abstract

We consider a sketched implementation of the finite element method for elliptic partial differential equations on high-dimensional models. Motivated by applications in real-time simulation and prediction we propose an algorithm that involves projecting the finite element solution onto a low-dimensional subspace and sketching the reduced equations using randomised sampling. We show that a sampling distribution based on the leverage scores of a tall matrix associated with the discrete Laplacian operator, can achieve nearly optimal performance and a significant speedup. We derive an expression of the complexity of the algorithm in terms of the number of samples that are necessary to meet an error tolerance specification with high probability, and an upper bound for the distance between the sketched and the high-dimensional solutions. Our analysis shows that the projection not only reduces the dimension of the problem but also regularises the reduced system against sketching error. Our numerical simulations suggest speed improvements of two orders of magnitude in exchange for a small loss in the accuracy of the prediction.

Keywords: Randomised linear algebra, Galerkin finite element method, statistical leverage scores, real-time simulation.
2000 MSC: 65F05, 65M60, 68W20

*Corresponding author
Email addresses: robert.lung@ed.ac.uk (Robert Lung), Yue.Wu@ed.ac.uk,Yue.Wu@maths.ox.ac.uk (Yue Wu), d.kamilis@ed.ac.uk (Dimitris Kamilis), n.polydorides@ed.ac.uk (Nick Polydorides)

[1]RL acknowledges the support of the James Clerk Maxwell Foundation.
[2]NP, YW and DK are grateful to EPSRC for funding this work through the grant EP/R041431/1, titled 'Randomness: a resource for real-time analytics'.

1. Introduction

Motivated by applications in digital manufacturing twins and real-time simulation in robotics, we consider the implementation of the Finite Element Method (FEM) in high-dimensional discrete models associated with elliptic partial differential equations (PDE). In particular, we focus on the many-query context, where a stream of approximate solutions are sought for various PDE parameter fields [1], aiming to expedite computations in situations where speedy model prediction is critical. Realising real-time simulation with high-dimensional models is instrumental to enable digital economy functions and has been driving developments in model reduction over the last decade [2], including the popular and, in many cases, effective Reduced-Basis method, which approximates the PDE solution manifold via a low-dimensional reduced basis, built from solution snapshots using either a POD or greedy construction [3, 4, 5]. Reducing the computational complexity of models is also central to the practical performance of statistical inference and uncertainty quantification algorithms, where a multitude of model evaluations are necessary to achieve convergence [6]. When real-time prediction is coupled with noisy sensor data, as in the digital twins paradigm, a fast, somewhat inaccurate model prediction typically suffices [7].

Our approach is thus tailored to applications where some of the accuracy of the solution can be traded off with speed. In these circumstances the framework of randomised linear algebra presents a competitive alternative [8]. In the seminal work [9], Drineas and Mahoney propose an algorithm for computing the solution of the Laplacian of a graph, making the case for sampling the rows of the matrices involved based on their statistical leverage scores. Despite aimed explicitly for the symmetric diagonally dominant systems arising in these problems, their approach provides inspiration for the numerical solution of symmetric, positive definite and possibly ill-conditioned systems originating from the discretisation of elliptic PDEs on unstructured meshes. Apart from the algebraic resemblance to the Galerkin FEM systems, the authors introduced sampling based on leverage scores of matrices through the concept of 'effective resistance' of a graph derived by mimicking Ohmic relations in resistor networks. As it turns out the complexity of computing the leverage scores is similar to that of solving the high-dimensional problem deterministically, however efficient methods to approximate them have

since been suggested [10]. More recently, Avron and Toledo have proposed an extension of [9] for preconditioning the FEM equations by introducing the 'effective stiffness' of an element in a finite element mesh [11]. Specifically, for sparse symmetric positive definite (SSPD) stiffness matrices, they derive an expression for the effective stiffness of an element and show its equivalence to the statistical leverage scores. Sampling $O(n \log n)$ elements leads to a sparser preconditioner.

In situations where a single, high-dimensional linear system is sought, randomised algorithms suited to SSPD systems are readily available. The methods of Gower and Richtarik for example randomises the row-action iterative methods by taking a sequence of random projections onto convex sets [12]. This algorithm is equivalent to a stochastic gradient descent method with provable convergence, while their alternative approach in [13] iteratively sketches the inverse of the matrix. In [14], Bertsekas and Yu present a Monte Carlo method for simulating approximate solutions to linear fixed-point equations, arising in evaluating the cost of stationary policies in Markovian decisions. Their algorithm is based on approximate dynamic programming and has subsequently led to [15], that extends some of the proposed importance sampling ideas in the context of linear ill-posed inverse problems.

Real-time FEM computing at the many query paradigm, is hindered by two fundamental challenges: the fast assembly of the stiffness matrix for each parameter field, unless the domain consists of a small number of regions with homogeneous isotropic materials, and the efficient solution of the resulting system to the required accuracy. To mitigate these, is to compromise slightly on the accuracy in order to capitalise on speed. To achieve this we first transform the linear SSPD system into an overdetermined least squares problem, and then project its solution this onto a low-dimensional subspace. This mounts to inverting a low-dimensional, dense matrix whose entries are perturbed by random errors. Our emphasis and contributions are in developing the projected sketching algorithm, and in optimising the sampling process so that it is both efficient in the multi-query context and effective in suppressing the variance of the solution. We also analyse the complexity of our algorithm and derive, probabilistic error bounds for quality of the approximation.

Our paper is organised as follows: In section 2 we provide a concise introduction to the Galerkin formulation for elliptic boundary value problems, and subsequently derive the projected least squares formulation of the problem. We then describe the sampling distribution used in the sketching and provide the conditions under which the reduced sketched system has a unique solu-

tion. Section 4 contains a description of our algorithm, and our main result that describes the complexity of our algorithm in achieving an error tolerance in high probability. We then provide an error analysis addressing the various types of errors imparted on the solution through the various stages of the methodology, before concluding with some numerical experiments based on the steady-state diffusion equation.

1.1. Notation

Let $[m]$ denote the set of integers between 1 and m inclusive. For a matrix $X \in \mathbb{R}^{m\times n}$, $X_{(\ell)}$ and $X^{(\ell)}$ denote its ℓ-th row and column respectively, and X_{ij} its (i,j)-th entry. $X^\dagger$ is the pseudo-inverse of X and $\kappa(X)$ its condition number. If $m \geq n$ we define the singular value decomposition $X = U_X \Sigma_X V_X^T$ where $U_X \in \mathbb{R}^{m\times n}$, $\Sigma_X \in \mathbb{R}^{n\times n}$ and $V_X \in \mathbb{R}^{n\times n}$. Notice that the form of the SVD used in this work is the more economical reduced/thin variant where the matrix U_X is not square and due to $n \leq m$ the matrix Σ_X is invertible whenever X has full column rank. Unless stated otherwise, singular values and eigenvalues are ordered in non-increasing order. Analogously, for a symmetric and positive definite matrix $A \in \mathbb{R}^{m\times m}$, $\lambda_{\max(A)}$ is the largest eigenvalue, and $\lambda_{\min(A)}$ the smallest. By $\mathrm{nnz}(A)$ we denote the number of non-zero elements in A. Further we write $\|\cdot\|$ for the Euclidean norm for a vector or the spectral norm of a matrix and $\|\cdot\|_F$ the Frobenius norm of a matrix. For matrices X and Y with the same number of rows $(X|Y)$ is the augmented matrix formed by column concatenation. The identity matrix is expressed as I or I_n to specify its dimension n when important to the context. We write $y \otimes 1_n$ for the Kronecker product of vector y with the ones vector in n dimensions.

2. Galerkin finite element method preliminaries

Consider the elliptic partial differential equation

$$-\nabla \cdot p\nabla u = f \quad \text{in} \quad \Omega, \tag{1}$$

on a bounded, simply connected domain $\Omega \subset \mathbb{R}^d$, $d \in \{2,3\}$ with Dirichlet conditions

$$u = g^{(D)} \quad \text{on} \quad \partial\Omega, \tag{2}$$

on a Lipschitz smooth boundary $\partial\Omega$ for a sufficiently smooth function $g^{(D)}$. Further let p a bounded positive parameter function in the Banach space

$L^\infty(\Omega)$ such that

$$0 < p_{\min} \leq p \leq p_{\max} < \infty \quad \text{on} \quad \Omega \cup \partial\Omega, \tag{3}$$

If we consider $\mathcal{T}_\Omega \doteq \{\Omega_1, \ldots, \Omega_k\}$ a mesh comprising k elements, having n interior and n_∂ boundary vertices (nodes) and

$$\mathcal{S}^1_\Omega \doteq \text{span}\{\phi_1, \ldots, \phi_n, \ldots, \phi_{n+n_\partial}\}$$

to comprise linear interpolation shape functions with local support over the elements in $\mathcal{T}_\Omega$ then the weak form of (1), see [16] chapter 6, can be discretised to yield the Galerkin system of equations for the vector $\{u_1, \ldots, u_n\}$ and each $i = 1, \ldots, n$

$$\begin{aligned}\sum_{j=1}^{n}\Big(\sum_{\Omega_\ell \in \mathcal{T}_\Omega} \int_{\Omega_\ell} \mathrm{d}x\, \nabla\phi_i \cdot p_\ell \nabla\phi_j\Big) u_j = &\sum_{\Omega_\ell \in \mathcal{T}_\Omega} \int_{\Omega_\ell} \mathrm{d}x\, f_\ell \phi_i \\ &- \sum_{j=n+1}^{n+n_\partial}\Big(\sum_{\Omega_l \in \mathcal{T}_\Omega} \int_{\Omega_\ell} \mathrm{d}x \nabla\phi_i \cdot p_\ell \nabla\phi_j\Big) u_j.\end{aligned} \tag{4}$$

At the same time (2) gives the boundary conditions of the form

$$u_j = g_j, \qquad j = n+1, \ldots, n_\partial \tag{5}$$

with g_j given by evaluating the boundary function $g^{(D)}$ at the j-th node. The coefficients in the above equation are defined as the element-average coefficients

$$p_\ell = \frac{1}{|\Omega_\ell|}\int_{\Omega_\ell} \mathrm{d}x\, p, \quad \text{and} \quad f_\ell = \frac{1}{|\Omega_\ell|}\int_{\Omega_\ell} \mathrm{d}x\, f, \quad \ell = 1, \ldots, k \tag{6}$$

which correspond to the piecewise constant approximations of the parameter and forcing term. The linear equations in (4) and (5) are expressed in a matrix form as

$$Au = b, \tag{7}$$

where $A \in \mathbb{R}^{n\times n}$ is the symmetric, sparse and positive-definite stiffness matrix, whose dependence on the parameters p is implicit and suppressed for clarity. The FEM construction guarantees the consistency of the system (7), thus $b \in \mathbb{R}^n$ is always in the column space of A and consequently it admits a unique solution $u_{\text{opt}} = A^{-1}b$. As we focus to the efficient approximation of u_{opt} in the many query context we content with two challenges: the efficient assembly of the stiffness matrix, and the speedy solution of the resulted FEM system.

2.1. The stiffness matrix

Let $\mathcal{I}_\ell$ be the index set of the $d+1$ vertices of the ℓth element, and consider $D_\ell \in \mathbb{R}^{d\times n}$ to be the sparse matrix holding the gradients of the linear shape functions ϕ_i where $i \in \mathcal{I}_\ell$. In this $D_\ell^{(i)}$ is then a constant gradients vector associated with the ith node of Ω_ℓ, and let $z_\ell = |\Omega_\ell| p_\ell$ the element of a vector $z \in \mathbb{R}^k$ such that $Z^2 = \mathrm{diag}(z \otimes 1_d)$ and $D \in \mathbb{R}^{kd\times n}$ a row concatenation of D_ℓ matrices for all elements. If we define as $Y_\ell = \sqrt{z_\ell} D_\ell$ and $Y \in \mathbb{R}^{kd\times n}$ the concatenation of the Y_ℓ matrices as

$$Y = ZD \tag{8}$$

then the stiffness matrix takes the form of a high-dimensional sum or product of sparse matrices

$$A = \sum_{\ell=1}^{k} Y_\ell^T Y_\ell = Y^T Y. \tag{9}$$

The above construction typically leads to a stiffness matrix that is well-conditioned for inversion with the exception of acute element skewness [17] and parameter vectors with wild variation [18], which cause the condition number $\kappa(A)$ to increase dramatically. Explicit bounds on the largest and smallest eigenvalues of A, and respectively the singular values of Y, are given in [19].

3. A regularised sketched formulation

Broadly speaking, the randomised sketching technique [8] provides a rigorous framework for speeding up numerical linear algebra operations, such as regression or low-rank approximation problems, at the cost of introducing a provably controllable error. This is achieved by compressing high dimensional vectors or matrices to a much smaller size by multiplying them by a random sketching matrix S. The matrix S should ideally be such that for a high dimensional quantity Y, i.e. a large matrix or vector,

- $\hat{Y} = S^T Y$ can be computed (substantially) faster than the solution to the original un-sketched problem and
- $\hat{Y}$ is a good enough approximation of Y in a way specific to the problem at hand.

The first criterion is very simple and ensures that computing the sketch isn't prohibitively expensive. The second criterion should be understood in terms of the solution x of the original problem which often involves some form of optimisation. More specifically, an ε-accurate sketch $\hat{Y}$ of Y for finding an approximate solution $\hat{x}$ ensures that $\|\hat{x} - x\| \leq \varepsilon\|x\|$. In other words, the sketched matrix is a good approximation of its high-dimensional counterpart if it can be used to solve the problem of interest subject to a small relative error. The acceleration from sketched methods consequently scales with the amount of compression that can be applied to Y while keeping the error acceptable.

In the case of linear regression problems, good computational gains can usually be expected when the matrices that should be sketched have significantly more rows than columns and the resulting systems are highly over-determined. Intuitively this observation can be explained by noticing that there is a certain amount of redundancy in an over-determined system and thus there is some hope that it can be compressed and solved more efficiently. In order to understand how this technique can be applied in our context we start by observing that the sought solution $u_{\text{opt}} = A^{-1}b$ can be alternatively obtained by solving the over-determined least squares problem

$$u_{\text{opt}} = u_{\text{LS}} = \arg\min_{u \in \mathbb{R}^n} \|Yu - (Y^T)^\dagger b\|^2, \tag{10}$$

since

$$u_{\text{LS}} = (Y^T Y)^{-1} Y^T (Y^T)^\dagger b = A^{-1} Y^T (Y^T)^\dagger b = A^{-1} b = u_{\text{opt}}.$$

The fact that the above problem is over-determined implies, at least to some extent, robustness against noise, such as random perturbations on the elements of the matrix Y and vector b. A similar error is induced by randomised sketching where we replace (10) with

$$\hat{u}_{\text{LS}} = \arg\min_{u \in \mathbb{R}^n} \|\hat{Y}u - (\hat{Y}^T)^\dagger b\|^2, \tag{11}$$

and look for a random approximation $\hat{Y}$ of Y in the sense that $\hat{u}_{\text{LS}} \approx u_{\text{LS}}$. We note that $\hat{Y}$ and Y don't have to be similar as such, e.g. have the same dimensions, as long as the problems are well defined and the optimisers remain similar. Following [9] and [20] we seek to approximate Y with some sketch $\hat{Y}$ by sampling and scaling rows according to probabilities that will be specified later. The number of rows in $\hat{Y}$ in that case equals the number

of drawn samples. Clearly $\hat{Y}$ must have at least n rows as otherwise the problem (11) will be under-determined and, due to the non-uniqueness of the solution, the error could become arbitrarily large. On the other hand, if around $n\log(n)$ rows are sampled from a suitable distribution, then Drineas and Mahoney [9] show that the resulting sketch is a good approximation with high probability. However, if substantially less than $n\log(n)$ samples are drawn then the sketching induced error outweighs its computational benefits. In order to understand how this issue can be addressed we note that, if $\hat{Y}$ has full column-rank and thus the optimiser of (11) is unique, the solution of the sketched problem can be obtained by solving the linear system

$$\hat{Y}^T\hat{Y}u = b,$$

which is equivalent to solving

$$Y^TYu = b + (Y^TY(\hat{Y}^T\hat{Y})^{-1} - I)b = \hat{b}. \tag{12}$$

From (12) it becomes clear that the sketching induced error can be regarded as an error on the right-hand side of the linear system (7) or the least squares problem (10). We can easily obtain a bound for the relative error given by

$$\frac{\|\hat{b} - b\|}{\|b\|} \leq \|Y^TY(\hat{Y}^T\hat{Y})^{-1} - I\|$$

A standard way of dealing with noise as in (12) is regularisation [21]. Suppose that there exists a low-dimensional subspace

$$\mathcal{S}_\rho \doteq \{\Psi r \,|\, r \in \mathbb{R}^\rho\}, \tag{13}$$

spanned by a basis of $\rho \ll n$ orthonormal functions arranged in the columns of matrix Ψ, and assume that is sufficient to approximate u_{opt} within some acceptable level of accuracy, in the sense of incurring a small subspace error $\|(I - \Pi)u_{\text{opt}}\|$. The orthogonal projection operator $\Pi \doteq \Psi\Psi^T$ maps vectors from $\mathbb{R}^n$ onto the subspace $\mathcal{S}_\rho$. Of course, such a subspace can't accommodate all but rather only sufficiently regular $u \in \mathbb{R}^n$. For that reason $\mathcal{S}_\rho$ has to be constructed using prior information (e.g. degree of smoothness) about the solution. Orthogonality of Ψ ensures for any $u_{\text{opt}} = \Pi u_{\text{opt}} + (I - \Pi)u_{\text{opt}}$ the existence of a unique, optimal low-dimensional vector r_{opt} satisfying

$$\Psi r_{\text{opt}} = \Pi u_{\text{opt}}. \tag{14}$$

In these conditions we can pose a projected-regularised least-squares problem replacing (10) by

$$\Pi u_{\text{opt}} \approx u_{\text{reg}} = \arg\min_{u \in \mathcal{S}_\rho} \|Yu - (Y^T)^\dagger b\|^2, \tag{15}$$

in order to improve the robustness of the solution against sketching-induced errors. The problem in (15) still involves high-dimensional quantities such as Y and b, but the solution is unique as soon as $\mathcal{S}_\rho$ and the null-space of Y have $\{0\}$ intersection. We start by introducing the low dimensional problem[3]

$$r_{\text{reg}} = \arg\min_{r \in \mathbb{R}^\rho} \|Y\Psi r - (Y^T)^\dagger b\|^2. \tag{16}$$

A solution r_{reg} of (16) yields a solution $u_{\text{reg}} = \Psi r_{\text{reg}}$ of (15) because the columns of Ψ form an orthonormal basis (ONB) of its column-space $\mathcal{S}_\rho$ by construction. In addition, we have the following.

Lemma 3.1. *If Y has full column rank and the columns of Ψ form an ONB of $\mathcal{S}_\rho$ so that $\Pi = \Psi\Psi^T$ is the projection onto $\mathcal{S}_\rho$, then*

$$\arg\min_{u \in \mathcal{S}_\rho} \|Yu - (Y^T)^\dagger b\|^2 = \arg\min_{u \in \mathcal{S}_\rho} \|Y\Pi u - (\Psi^T Y^T)^\dagger \Psi^T b\|^2. \tag{17}$$

In particular, both problems have a unique solution.

Proof. Both problems have unique solutions because $\mathcal{S}_\rho$ is convex and Y has (by assumption) full column rank. Therefore it suffices to show that there exists an element $u_{\text{reg}} \in \mathcal{S}_\rho$ that solves both problems. The solution r_{reg} of (16) can be found explicitly by solving the linear system

$$\Psi^T Y^T Y \Psi r = \Psi^T Y^T (Y^T)^\dagger b \iff r_{\text{reg}} = (\Psi^T Y^T Y \Psi)^{-1} \Psi^T b.$$

[3] We emphasise the contrast between the projected equations in (16) and the projected variable least squares problem

$$r' = \arg\min_{r \in \mathbb{R}^\rho} \left\| A\Psi r - b \right\|^2,$$

whose solution is

$$r' = (\Psi^T A^2 \Psi)^{-1} \Psi^T A b = \Psi^T u + (\Psi^T A^2 \Psi)^{-1} \Psi^T A^2 (I - \Pi) u,$$

and incurs a subspace regression error term that is quadratic in A. Moreover, note that the right hand side vector in the normal equations $\Psi^T A^T A \Psi r' = \Psi^T A^T b$ has dependence on the parameter through A.

We have used that Y has full column rank so that $Y^T(Y^T)^\dagger = I$ and $\Psi^T Y^T Y \Psi$ is invertible. Similarly we may consider

$$\arg\min_{r\in\mathbb{R}^\rho} \|Y\Pi\Psi r - (\Psi^T Y^T)^\dagger \Psi^T b\|^2,$$

which produces solutions r_Ψ such that Ψr_Ψ is a solution of the right-hand side of (17). Since $\Pi\Psi = \Psi$ and $Y\Psi$ has full column rank we can write r_Ψ as

$$\Psi^T Y^T Y \Psi r_\Psi = \Psi^T Y^T (\Psi^T Y^T)^\dagger \Psi^T b \iff r_\Psi = (\Psi^T Y^T Y \Psi)^{-1} \Psi^T b.$$

We conclude that $\Psi(\Psi^T Y^T Y \Psi)^{-1}\Psi^T b$ is a solution to both sides of (17) which completes the proof. □

The right hand side of (17) has a very natural interpretation and is obtained by embedding the rows of Y, the vector b and the variable u in $\mathcal{S}_\rho$ using its low dimensional representation from the basis induced by the columns of Ψ. In view of Lemma 3.1 we may regularise the problem from (11) and obtain an embedded sketched counterpart to (15) as

$$\hat{u}_{\text{reg}} = \arg\min_{u\in\mathcal{S}_\rho} \|\hat{Y}\Pi u - (\Psi^T \hat{Y}^T)^\dagger \Psi^T b\|^2. \tag{18}$$

We argue that (18) is much more robust to the noise imparted by the approximation $\hat{Y}$ and produces solutions with controlled errors even if substantially less than n suitably drawn samples are used for the approximation. In order to see why, notice that the problem (18) can be expressed in terms of the low-dimensional vector of coefficients

$$\hat{r}_{\text{reg}} = \arg\min_{r\in\mathbb{R}^\rho} \|\hat{Y}\Psi r - (\Psi^T \hat{Y}^T)^\dagger \Psi^T b\|^2. \tag{19}$$

so that $\Psi\hat{r}_{\text{reg}} = \hat{u}_{\text{reg}}$. Recalling that $A = Y^T Y$, it is convenient to introduce

$$X = Y\Psi \quad \text{and} \quad G = X^T X = \Psi^T A \Psi, \tag{20}$$

together with their sketched approximations

$$\hat{X} = \hat{Y}\Psi \quad \text{and} \quad \hat{G} = \hat{X}^T \hat{X}. \tag{21}$$

Lemma 3.2. *If $\hat{X} = \hat{Y}\Psi$ has full column rank then the solution of the least-squares problem (19) is given by $\hat{r}_{\text{reg}} = \hat{G}^{-1}\Psi^T b$ and we have*

$$\hat{u}_{\text{reg}} = \Psi\hat{r}_{\text{reg}} = u_{\text{reg}} + \Psi(\hat{G}^{-1}G - I)\Psi^T u_{\text{reg}}. \tag{22}$$

where u_{reg} and $\hat{u}_{\text{reg}}$ are the solutions of (15) and (18) respectively.

Proof. If $\hat{Y}\Psi$ has linearly independent columns then $\Psi^T\hat{Y}^T(\Psi^T\hat{Y}^T)^\dagger = I$ and the solution $\hat{r}_{\text{reg}}$ of (19) solves

$$\hat{G}r = \Psi^T b.$$

Again $\hat{G}$ is invertible because $\hat{Y}\Psi$ has linearly independent columns and the first claim follows. The matrix A is positive definite which implies that G is positive definite and $u_{\text{reg}} = \Psi G^{-1}\Psi^T b$. The matrix Ψ has orthonormal columns which implies $\Psi^T b = G\Psi^T u_{\text{reg}}$. Since $\hat{u}_{\text{reg}} = \Psi\hat{r}_{\text{reg}}$ we can use the formula we have just shown and obtain

$$\begin{aligned}
\hat{u}_{\text{reg}} &= \Psi\hat{r}_{\text{reg}} \\
&= \Psi\hat{G}^{-1}\Psi^T b \\
&= \Psi\hat{G}^{-1}G\Psi^T u_{\text{reg}} \\
&= \Psi\hat{G}^{-1}(\hat{G} + (G - \hat{G}))\Psi^T u_{\text{reg}} \\
&= \Pi u_{\text{reg}} + \Psi(\hat{G}^{-1}G - I)\Psi^T u_{\text{reg}} \\
&= u_{\text{reg}} + \Psi(\hat{G}^{-1}G - I)\Psi^T u_{\text{reg}}
\end{aligned}$$

where the last identity is due to $u_{\text{reg}} \in \mathcal{S}_\rho$. □

In order to understand the effect of row sampling and why it can be a good approximation, recall that k is the number of elements and d the dimension, we can start by writing

$$G = \sum_{j=1}^{kd} X_{(j)}^T X_{(j)} = X^T X \quad \text{and} \quad A = \sum_{j=1}^{kd} Y_{(j)}^T Y_{(j)} = Y^T Y \tag{23}$$

as a sum of outer products of rows. Introduce for some sample size $c \in \mathbb{N}$ the iid random indices $\mathbf{i}_1, \ldots, \mathbf{i}_c$ taking values in $[kd]$ with distribution

$$\mathbb{P}(\mathbf{i}_j = i) = q_i \tag{24}$$

for each $j \in [c]$ and $i \in [kd]$. Instead of (23) we may consider the sketch

$$\hat{G} = \frac{1}{c}\sum_{j=1}^{c} \frac{1}{q_{\mathbf{i}_j}} X_{(\mathbf{i}_j)}^T X_{(\mathbf{i}_j)}. \tag{25}$$

If we define the random matrix $R \in \mathbb{R}^{kd \times c}$ and the random diagonal matrix $W \in \mathbb{R}^{c \times c}$ via

$$R_{ij} = \begin{cases} 1 & \text{if} \quad \mathbf{i}_j = i \\ 0 & \text{if} \quad \mathbf{i}_j \neq i \end{cases}, \qquad W_{jj} = \frac{1}{\sqrt{c q_{\mathbf{i}_j}}}, \tag{26}$$

then can put $S = RW$ and construct the sketch $\hat{G}$ as

$$\hat{G} = X^T S S^T X = X^T R W^2 R^T X. \tag{27}$$

Lastly, we can write $\hat{Y} = S^T Y$ as well as $\hat{X} = \hat{Y}\Psi = S^T Y \Psi$ for the sketches of Y and X. A simple computation together with an application of the strong law of large numbers shows the following.

Proposition 3.3 (Lemma 3 and 4 in [22])**.** *Assume that the sampling probabilities satisfy the consistency condition*

$$X_{(j)} \neq 0 \implies q_j > 0 \qquad \forall j = 1, \ldots, kd. \tag{28}$$

In this case we have for the matrix $\hat{G}$ as defined in (25) that $\mathbb{E}[\hat{G}] = G$ and $\mathbb{E}[\|\hat{G} - G\|_F^2] = \mathcal{O}(c^{-1})$. As a consequence, $\hat{G} \to G$ almost surely for $c \to \infty$.

Proposition 3.3 summarises the asymptotic properties of the used sketch. The condition (28) is very mild and holds for a wide range of distributions such as sampling from scaled row norms or uniform sampling. The convergence rate of c^{-1} cannot be improved although the constant depends on the chosen probabilities q_j. In other words, as long as we sample all non-zero rows with positive probability we will obtain a sketch that has good asymptotic properties when considered as an approximation for G. However, in order to find good sampling probabilities q_j we have to consider the non-asymptotic behaviour of the sketch. In fact, the main purpose of the regularisation/dimensionality reduction was to avoid situations where sampling a large number of rows is necessary. If $\rho \ll n$, then the regularised problem (16) has substantially fewer degrees of freedom than the high dimensional formulation in (10). Consequently, the dependence of G on the rows of X is a lot smoother than the dependence of A on $Y_{(j)}$. In other words, approximating X by row sampling has a much smaller effect on the regularised solution u_{reg} than an approximation of Y with the same sample size c would have on the solution u of the full system (7). For example, a much smaller

number of rows needs to be sampled to obtain the correct null-space which results in a full-rank approximation of G. Note that, conditional on $\hat{G}$ being invertible, $u_{\text{reg}} \in \mathcal{S}_\rho$ in combination with Lemma 3.2 implies

$$\frac{\|u_{\text{reg}} - \hat{u}_{\text{reg}}\|}{\|u_{\text{reg}}\|} \leq \|\hat{G}^{-1}G - I\|, \tag{29}$$

so the randomisation error of the regularised problem is entirely controlled by low dimensional structures. This property is the key to a small sketching error and thus to an overall accurate approximation when only few samples are drawn. Using the notation from before and letting $X = U_X \Sigma_X V_X^T$ be the singular value decomposition of X, we can write the bound from (29) as

$$\|\hat{G}^{-1}G - I\| = \|\Sigma_X^{-1}(U_X^T SS^T U_X)^{-1}\Sigma_X - I\|.$$

From the above formulation it becomes apparent that the error will be small if the sketch is constructed such that $(U_X SS^T U_X)^{-1} \approx I$ in spectral norm. We argue that this is essentially equivalent to $U_X SS^T U_X \approx I$. Indeed, we have the following.

Lemma 3.4. *If* $\|U_X^T SS^T U_X - I\| < \varepsilon < 1$ *then*

$$1 - \varepsilon \leq \frac{\|U_X^T SS^T U_X - I\|}{\|(U_X^T SS^T U_X)^{-1} - I\|} \leq 1 + \varepsilon.$$

Proof. Under the condition of the lemma we know that $U_X SS^T U_X$ is invertible and that

$$\|U_X^T SS^T U_X\| \leq \|I\| + \|U_X^T SS^T U_X - I\| < 1 + \varepsilon$$

which implies the upper bound by considering the estimate

$$\begin{aligned}\|U_X^T SS^T U_X - I\| &\leq \|U_X^T SS^T U_X\|\|(U_X^T SS^T U_X)^{-1} - I\| \\ &\leq (1+\varepsilon)\|(U_X^T SS^T U_X)^{-1} - I\|.\end{aligned}$$

Denote by $\lambda_i(U_X SS^T U_X)$ the i-th eigenvalue of $U_X SS^T U_X$. Then we may write

$$\begin{aligned}\|(U_X SS^T U_X)^{-1} - I\| &= \max_{i=1}^{\rho} |1 - \lambda_i^{-1}(U_X^T SS^T U_X)| \\ &= \max_{i=1}^{\rho} \frac{|1 - \lambda_i(U_X^T SS^T U_X)|}{\lambda_i(U_X^T SS^T U_X)} \\ &\leq \frac{\|1 - U_X^T SS^T U_X\|}{\lambda_{\min}(U_X^T SS^T U_X)}\end{aligned}$$

where $\lambda_{\min}(U_X SS^T U_X)$ is the smallest eigenvalue. By assumption of the lemma

$$|1 - \lambda_{\min}(U_X^T SS^T U_X)| \leq \varepsilon \implies \lambda_{\min}(U_X^T SS^T U_X) \geq 1 - \varepsilon$$

which implies the claim after dividing by $\|1 - U_X^T SS^T U_X\|$ and taking the inverse. □

An approximation of $U_X^T SS^T U_X$ can be obtained by sampling with probabilities that are proportional to the statistical leverage scores

$$\ell_i(X) = \ell_i(U_X) = \|(U_X)_{(i)}\|^2, \tag{30}$$

i.e. the row norms of the left singular vectors of X [10]. At first sight it seems that taking sampling probabilities proportional to the leverage scores in (30) in order to obtain a sketch of (16) is very similar to using the leverage scores of Y to obtain (11) from (10) as was proposed by Drineas and Mahoney in [9] for a similar problem. A key difference is that X is tall and dense while Y is sparse and thus G is quite different to the initial stiffness matrix A. Consequently, an interpretation of the leverage scores from (30) in terms of effective stiffness [11] is, to the best of our knowledge, not possible. The following Lemma will be useful for our further developments.

Lemma 3.5 ([23] section 6.4)**.** *Assume that S is constructed as before with sampling probabilities q_i satisfying*

$$q_i \geq \beta \frac{\ell_i(X)}{\rho} \quad i = 1, \ldots, kd \tag{31}$$

for some $\beta \in (0, 1]$. Then we have $\forall \varepsilon > 0$

$$\mathbb{P}\left(\|U_X^T SS^T U_X - I\| \geq \varepsilon\right) \leq 2\rho \exp\left(-\frac{3c\beta\varepsilon^2}{12\rho + 4\rho\varepsilon}\right) \tag{32}$$

An important corollary of the above lemma is that a sketch which is constructed by sampling from leverage score probabilities will virtually always be invertible and therefore the sketched problem (19) has a unique solution. The following result states that this property is preserved even when the rows are re-weighted, an operation which changes the leverage scores.

Proposition 3.6. *Let $\Gamma \in \mathbb{R}^{kd\times kd}$ be a diagonal matrix with positive entries, i.e. $\Gamma_{ii} > 0$ for each $i = 1, \dots, kd$. Assume that the sketching matrix S is constructed with sampling probabilities $q_i = \rho^{-1}\ell_i(X)$. For the scaled sketch $\hat{H} = X^T\Gamma SS^T\Gamma X$ we have*

$$\mathbb{P}(\hat{H} \text{ is invertible}) = \mathbb{P}(\hat{G} \text{ is invertible}) \geq 1 - 2\rho \exp\left(-\frac{3c}{16\rho}\right) \tag{33}$$

Proof. It is sufficient to show that

$$\hat{H} \text{ is invertible} \iff \hat{G} \text{ is invertible} \iff U_X^T SS^T U_X \text{ is invertible}$$

because the probability bound follows immediately from

$$\mathbb{P}(U_X^T SS^T U_X \text{ is invertible}) \geq 1 - \mathbb{P}\left(\|U_X^T SS^T U_X - I\| \geq 1\right)$$

after applying (32) from Lemma 3.5. The above matrices are always positive semi-definite and therefore invertibility is equivalent to positive definiteness. For any diagonal matrix Γ it holds that $S^T\Gamma = \hat{\Gamma}S^T$ where $\hat{\Gamma}$ is a random diagonal matrix with entries $\hat{\Gamma}_{jj} = \Gamma_{\mathbf{i}_j\mathbf{i}_j}$. Thus for any $x \in \mathbb{R}^\rho$ we have

$$x^T\hat{H}x = (\Sigma_X V_X^T x)^T U_X^T S\hat{\Gamma}^2 S^T U_X(\Sigma_X V_X^T x).$$

Since X has full column rank we know that $\Sigma_X V_X^T$ corresponds to a change of basis and $\Sigma_X V_X^T x \neq 0$ whenever $x \neq 0$. It follows that $\hat{H}$ is positive definite if and only if $U_X^T S\hat{\Gamma}^2 S^T U_X$ is positive definite. As $\hat{\Gamma}$ is a diagonal such that $\hat{\Gamma}_{jj} > 0$ with probability 1, the latter is equivalent to $U_X^T SS^T U_X$ being positive definite. The case of $\hat{G}$ is covered by $\Gamma = I$. □

Proposition 3.6 states that re-scaling of rows doesn't affect the quality of the sketching matrix regarding its invertibility and after sampling $\rho\log(\rho)$ rows the probability of the sketch being singular decays exponentially fast with each additional draw. In practice this makes knowledge of $\ell_i(X)$ valuable because we only need to sample $\rho\log(\rho) + M$ rows for some moderately large M and obtain a sketch that is virtually never singular. On the other hand, we need at least ρ samples so that there is any hope in obtaining a non-singular matrix. The remarkable thing about Proposition 3.6 is that the failure probability is *independent* of both, the inner dimension kd of the product $X^T X$ as well as the scaling matrix Γ and equivalent to the bound which could be obtained by sampling from $\ell_i(\Gamma X)$. This suggests that a

sketch which is constructed by drawing samples from $\ell_i(X)$ is not too different compared to sampling from $\ell_i(\Gamma X)$. This intuition is supported by the following result which describes the change in the leverage scores after re-weighting a single row.

Proposition 3.7 ([24] Lemma 5)**.** *Let $\Gamma^{\langle i\rangle} \in \mathbb{R}^{kd\times kd}$ be a diagonal matrix with $\Gamma^{\langle i\rangle}_{ii} = \sqrt{\gamma} \in (0,1)$ and $\Gamma^{\langle i\rangle}_{jj} = 1$ for each $j \neq i$. Then*

$$\ell_i(\Gamma^{\langle i\rangle}X) = \frac{\gamma\ell_i(X)}{1-(1-\gamma)\ell_i(X)} \leq \ell_i(X) \tag{34}$$

and for $i \neq j$

$$\ell_j(\Gamma^{\langle i\rangle}X) = \ell_j(X) + \frac{(1-\gamma)\ell_{ij}^2(X)}{1-(1-\gamma)\ell_i(X)} \geq \ell_j(X) \tag{35}$$

where $\ell_{ij}(X) = (U_X U_X^T)_{ij}$ are the cross leverage scores.

Since U_X has orthogonal columns, we have $\|v\| = \|U_X v\|$ for any $v \in \mathbb{R}^\rho$ and thus the cross leverage scores from the above Lemma satisfy

$$\ell_i(X) = \sum_{j=1}^{kd} \ell_{ij}^2(X). \tag{36}$$

For a general diagonal matrix Γ as in Proposition 3.6 we may without loss of generality assume that each entry lies in $(0,1]$ since we can divide the elements by their maximum. The re-weighting can thus be considered as a superposition of single row operations

$$\Gamma = \prod_{i=1}^{kd} \Gamma^{\langle i\rangle} \tag{37}$$

where the $\Gamma^{\langle i\rangle}$ are as in Proposition 3.7. Since the $\Gamma^{\langle i\rangle}$ commute we can apply them in any order without changing the outcome. Considering Lemma 3.5, if we could ensure that $\ell_i(X)$ isn't substantially smaller than $\ell_i(\Gamma X)$ then sampling from $q_i = \rho^{-1}\ell_i(X)$ will produce good sketches for ΓX.

Large leverage scores $\ell_i(X) \approx 1$. Equation (34) shows that the relative change of the i-th leverage score after a re-weighting of the i-th row shrinks when $\ell_i(X) \to 1$. In the extreme case when $\ell_i(X) = 1$ the re-weighting has no effect. In addition to this stability property it trivially holds that $\ell_i(X) \leq 1$ which suggests that large leverage scores are fairly stable when rows are re-weighted.

Small leverage scores $\ell_i(X) \ll 1$. From Equation (35) we know that the increase of $\ell_j(X)$ after re-weighting of row i is proportional to $\ell_{ij}(X)$. If the entries of the scaling matrix Γ don't vary too much, then (36) suggests that we can expect the total increase, i.e. after applying $\Gamma^{\langle j \rangle}$ for each $j \neq i$ to be roughly of order $\ell_i(X) - \ell_i^2(X) \approx \ell_i(X)$. On the other hand, small $\ell_i(X)$ are fairly sensitive to re-weighting of row i since $\ell_i(\Gamma^{\langle i \rangle} X) \approx (\Gamma_{ii}^{\langle i \rangle})^2 \ell_i(X)$ in that case. Thus we can expect that the re-weighting of row i will counterbalance the effects from re-weighting the other rows. In addition, we know that

$$\sum_{i=1}^{kd} \ell_i(X) = \sum_{i=1}^{kd} \ell_i(\Gamma X).$$

Since large leverage scores will likely be quite stable and $\ell_i(\Gamma X) \geq 0$ we would expect that not too many small leverage scores will become large.

So far we have discussed the projection of the high-dimensional system without providing explicit details on how the basis Ψ is selected. A desired property is to sustain a small projection error for all admissible parameter choices under the constraint $\rho \ll n$. Suitable options include subsets of the right singular vectors of A or orthogonalised Krylov-subspace bases [25], however these have to be computed for each individual parameter vector which can be detrimental to the speed of the solver. Alternatively, we opt for a generic basis exploiting the smoothness of u on domains with smooth Lipschitz boundaries. A simple choice is to select the basis among the eigenvectors of the discrete Laplacian operator

$$\Delta \doteq D^T Z_\Delta^2 D, \tag{38}$$

for $Z_\Delta^2 = \operatorname{diag}\big([|\Omega_1|, \ldots, |\Omega_k|] \otimes 1_d\big)$. From $U_\Delta^T \Delta U_\Delta = \Sigma_\Delta$ and splitting the eigenvectors as

$$U_\Delta = \big(U_\Delta{}^{(1:n-\rho-1)} | \Psi\big),$$

such that the columns of Ψ correspond to the last ρ columns of U_Δ, and respectively to the ρ smallest eigenvalues $\{\lambda_{n-\rho-1}(\Delta), \ldots, \lambda_n(\Delta)\}$. In effect, with Δ constrained by the Dirichlet boundary conditions, the norm $\|\Delta \Psi^{(i)}\|$ provides a measure of the smoothness of $\Psi^{(i)}$ in the interior of Ω. It is not difficult to see that this basis satisfies

$$\|\Delta \Psi^{(i)}\| \geq \|\Delta \Psi^{(j)}\| \quad \text{for} \quad \rho \geq i > j \geq 1.$$

We remark that the computation of the basis is computationally very expensive for large n, as the eigen-decomposition of Δ is necessary, however this is only computed once, prior to the beginning of the simulation (offline stage) in an offline stage. After the matrix Ψ has been obtained we can compute the leverage scores $\ell_i(Z_\Delta D\Psi)$. The Laplacian Δ differs from a general stiffness matrix A only by different diagonal weights, i.e. Z_Δ^2 is replaced by the diagonal matrix $Z^2 = Z_\Delta^2 \mathrm{diag}\big[(p_1, \dots, p_k) \otimes 1_d\big]$ where the p_i contain information about the parameter from (1). Propositions 3.6 and 3.7 along with the developments thereafter suggest that the Laplacian leverage scores $\ell_i(Z_\Delta D\Psi)$ can nonetheless be used to construct sketches $\hat{G} = X^T S S^T X$ of the projected matrix $G = X^T X = \Psi^T Y^T Y \Psi$ because the difference in the stiffness matrices is just a diagonal weighting.

4. Complexity and error analysis

Motivated by the developments from the previous sections we propose the following algorithm for computing solutions to a sequence of N problem of the form (1). We assume that each problem is specified by its parameter vector $z^{(t)} \in \mathbb{R}^{kd}$ for $t = 1, \dots, N$ (see section 2.1).

The complexity and approximation error of Algorithm 1 are obviously linked. The more samples we draw the better we expect our solutions to be. Although the size of the reduced system matrix G (and therefore its sketched counterpart $\hat{G}$ as well) is independent of c, the computational burden for building $\hat{G}$ is higher when drawing more samples. More precisely, we need:

- $\mathcal{O}(c)$ operations in order to find $\mathbf{i}_1, \dots \mathbf{i}_c \overset{\text{iid}}{\sim} q$. This is possible because q is fixed and we can perform the necessary pre-processing offline [26].
- $\mathcal{O}(c)$ operations for computing the sampled indices $\{\mathbf{j}_1, \dots, \mathbf{j}_{c'}\}$ and their frequencies m_j as this requires a single loop through the set $\{\mathbf{i}_1, \dots, \mathbf{i}_c\}$ of initial samples.
- $\mathcal{O}(c')$ operation for assembling the diagonal matrices M and $\hat{Z}$.
- $\mathcal{O}(c'\rho)$ operations for computing $M\hat{Z}D_{(J)}\Psi$. This can be achieved since computing $M\hat{Z}D_{(J)}$ requires $\mathrm{nnz}(D_{(J)}) = \mathcal{O}(c')$ multiplications and $\rho \cdot \mathrm{nnz}(M\hat{Z}D_{(J)}) = \rho \cdot \mathrm{nnz}(D_{(J)}) = \mathcal{O}(\rho c')$ multiplications are enough for computing $[M\hat{Z}D_{(J)}]\Psi$ due to sparsity of D.

input : Matrices $D \in \mathbb{R}^{kd\times n}$, $\Psi \in \mathbb{R}^{n\times\rho}$, data vector $\Psi^T b \in \mathbb{R}^\rho$, and sampling probabilities $q_i = \rho^{-1}\ell_i(Z_\Delta D\Psi)$ (offline)
output: Parameter dependent solutions $\hat{r}^{(t)} \in \mathbb{R}^\rho$ where $t = 1,\dots,N$
Online Simulation;
for $t \leftarrow 1$ **to** N **do**
  input : Parameter vector $z^{(t)} \in \mathbb{R}^k$, sample size c
  draw row indices $\mathbf{i}_1,\dots\mathbf{i}_c \overset{\text{iid}}{\sim} q$ from $[kd]$;
  get the sampled indices $J = \bigcup_{j=1}^{c}\{\mathbf{i}_j\}$;
  set $c' = |J|$ and write $J = \{\mathbf{j}_1,\dots,\mathbf{j}_{c'}\}$;
  compute the frequencies $m_j = \sum_{k=1}^{c}\delta(\mathbf{i}_k = \mathbf{j}_j)$ for $j = 1,\dots,c'$;
  find $M^2_{jj} = c^{-1}m_j q^{-1}_{\mathbf{j}_j}$ for $j = 1,\dots,c'$ and the diagonal matrix M;
  find $\hat{Z}^2_{jj} = z^{(t)}_{\mathbf{j}_j}$ for $j = 1,\dots,c'$ and the diagonal matrix $\hat{Z}^2$;
  assemble the $c'\times\rho$ matrix $\hat{X} = M\hat{Z}D_{(J)}\Psi$;
  compute reduced system $\hat{G} = \hat{X}^T\hat{X}$;
  compute and store $\hat{r}^{(t)} \leftarrow \text{solve}(\hat{G}, \Psi^T b)$;
end

Algorithm 1: Algorithm for simulating the low-dimensional projected solution of the FEM equations for different choices of parameter vectors p. Note that as we are sampling with replacement, $c' \leq c$. In the above $\delta(\cdot)$ denotes the indicator function where $\delta(E) = 1$ if the event E has occurred and it is zero otherwise otherwise. $D_{(J)}$ is the sub-matrix of D whose rows are the (ordered) elements of J

- $\mathcal{O}(c'\rho^2)$ operations in order to build $\hat{G}$ which corresponds to the cost of multiplication for dense matrices.
- $\mathcal{O}(\rho^3)$ operations for solving $\hat{G}r = \Psi^T b$ with a direct method.

The sketch $\hat{G}$ will be singular if we draw $c' < \rho$ distinct samples which means that building the sketch $\hat{G}$ dominates the complexity of Algorithm 1. In particular, the worst case complexity doesn't exceed $\mathcal{O}(c\rho^2)$ since we require $c \geq c' \geq \rho$. If the sampling probabilities are a good approximation in the sense that β in Lemma 3.5 can be chosen close to 1, then we need $c = \mathcal{O}(\varepsilon^{-2}\rho\log(\rho))$ samples in order to have a provably controlled error. The worst case, i.e. the the largest increase of $\ell_i(X)$, will be observed if $z^{(t)}_j \ll z^{(t)}_i$ for $j \neq i$. A parameter p corresponding to such a situation essentially renders

the implementation of the classical Galerkin FEM problematic, as $\kappa(A)$ scales to $p_{\max}/p_{\min}$, see Theorem 5.2 in [19] The following theorem summarises the findings of this section.

Theorem 4.1. *Let $\varepsilon \in (0,1)$ and $\beta \in (0,1]$ is such that the sampling probabilities q_i from Algorithm 1 satisfy (31), i.e.*

$$q_i \geq \beta \frac{\ell_i(ZD\Psi)}{\rho} \quad i = 1, \dots, kd$$

where $Z^2 = \operatorname{diag}(z^{(t)})$. Let $G = X^T X = \Psi^T D^T Z^2 D\Psi$ be the reduced system matrix corresponding to parameter $z^{(t)}$ and $\kappa(G)$ its condition number. For the choice $c = 15\rho\log(15\rho)\beta^{-1}\varepsilon^{-2}$ Algorithm 1 requires $\mathcal{O}(\rho^3\log(\rho)\beta^{-1}\varepsilon^{-2})$ operations and outputs, with probability exceeding 0.999, a vector $\hat{r}^{(t)}$ that satisfies

$$\frac{\|\hat{r}^{(t)} - G^{-1}\Psi^T b\|}{\|G^{-1}\Psi^T b\|} \leq \sqrt{\kappa(G)}\frac{\varepsilon}{1-\varepsilon}. \tag{39}$$

Proof. As stated before, the complexity of Algorithm 1 is $\mathcal{O}(c\rho^2)$ which immediately implies that it requires $\mathcal{O}(\rho^3\log(\rho)\beta^{-1}\varepsilon^{-2})$ operations for a single query. It remains to prove the error bound. In view of (29) and the developments thereafter it follows, conditional on $\hat{G}$ being invertible, that

$$\begin{aligned}\frac{\|\hat{r}^{(t)} - G^{-1}\Psi^T b\|}{\|G^{-1}\Psi^T b\|} &\leq \|\Sigma_X^{-1}(U_X^T S S^T U_X)^{-1}\Sigma_X - I\| \\ &\leq \kappa(X)\|(U_X^T S S^T U_X)^{-1} - I\| \\ &\leq \kappa(X)\frac{1}{1-\varepsilon}\|U_X^T S S^T U_X - I\|.\end{aligned}$$

Since $\kappa^2(X) = \kappa(G)$ we only need to show that

$$\mathbb{P}(\|U_X^T S S^T U_X - I\| \geq \varepsilon) \leq 0.001$$

because $\hat{G}$ is necessarily invertible on that event which implies validity of the estimates from before. But plugging the value for c into (32) we obtain for any $\rho \geq 1$

$$\mathbb{P}(\|U_X^T S S^T U_X - I\| \geq \varepsilon) \leq \frac{2}{15}\exp\left(-\frac{29}{16}\log(15\rho)\right) < 0.001.$$

□

Algorithm 1 is most attractive when we can tolerate an error somewhere between 1% to 10% in which case we can obtain the solution to a single query in about $\mathcal{O}(\beta^{-1}\rho^3 \log(\rho))$ time. In practice the value for β is unobtainable since it requires knowledge of the true leverage scores but considering Lemma 3.7 and the arguments thereafter, we expect that for a moderately large β^{-1} the required bound will hold for all but a few small leverage scores. The statement in Lemma 3.5 is rather pessimistic when there are few misaligned leverage scores since it requires a uniform bound. For practical purposes we expect that β^{-1} can be substituted with a small constant and we take $\varepsilon = 0.1$ which will ensure reglarity of the sketch. Up until now we have only considered the randomisation error of the sketched solution, i.e. we have analysed $\|\hat{u}_{\text{reg}} - u_{\text{reg}}\|$. However, the the total error of $\hat{u}_{\text{reg}}$ compared to the high dimensional solution u of (7) has two components. If we decompose the process into two steps

$$\min_{u\in\mathbb{R}^n} \|Yu - (Y^T)^\dagger b\|^2 \xrightarrow[\|u_{\text{opt}}-u_{\text{reg}}\|]{\text{Projection}} \min_{u\in\mathcal{S}_\rho} \|Yu - (Y^T)^\dagger b\|^2 \tag{40}$$

$$\min_{u\in\mathcal{S}_\rho} \|Yu - (Y^T)^\dagger b\|^2 \xrightarrow[\|\hat{u}_{\text{reg}}-u_{\text{reg}}\|]{\text{Sketching}} \min_{u\in\mathcal{S}_\rho} \|\hat{Y}\Pi u - (\Psi^T\hat{Y}^T)^\dagger \Psi^T b\|^2, \tag{41}$$

it becomes apparent that even with a perfect sketch, i.e. if we solved the noiseless projected problem (15) and (41) is negligible, we could still not achieve an error smaller than $\|u_{\text{opt}} - \Pi u_{\text{opt}}\|$. The next result tells us that the error from (40) is close to the optimal one.

Theorem 4.2. *Let u_{opt} be the solution of (7) and u_{reg} be the optimum of (15). If $\kappa(A)$ is the condition number of the stiffness matrix A and $\Pi = \Psi\Psi^T$ the projection ont $\mathcal{S}_\rho$, then*

$$\|u_{\text{opt}} - u_{\text{reg}}\| \leq \left(1 + \sqrt{\kappa(A)}\right) \|u_{\text{opt}} - \Pi u_{\text{opt}}\|.$$

Proof. Recall that $A = Y^TY$ and $G = X^TX = \Psi^TY^TY\Psi$. From the developments in Lemma 3.2 we know that $u_{\text{reg}} = \Psi G^{-1}\Psi^T b$. We may write as before $X = U_X\Sigma_X V_X^T$ so that $G^{-1} = V_X\Sigma_X^{-2}V_X^T$ and

$$\begin{aligned}
\|u_{\text{opt}} - u_{\text{reg}}\| &= \|u_{\text{opt}} - \Psi G^{-1}\Psi^T b\| \\
&= \|u_{\text{opt}} - \Psi G^{-1}\Psi^T A u_{\text{opt}}\| \\
&= \|u_{\text{opt}} - \Psi G^{-1}\Psi^T A[\Pi + (I-\Pi)]u_{\text{opt}}\| \\
&\leq \|u_{\text{opt}} - \Psi G^{-1}\Psi^T A\Psi\Psi^T u_{\text{opt}}\| + \|\Psi G^{-1}\Psi^T A(I-\Pi)u_{\text{opt}}\|
\end{aligned}$$

where the last line follows from the triangle inequality and the fact that $\Pi = \Psi\Psi^T$. Since $\Psi^T A\Psi = \Psi^T Y^T Y\Psi = X^T X = G$ the expression in the first term of the above equation simplifies to

$$u_{\text{opt}} - \Psi G^{-1}(\Psi^T A\Psi)\Psi^T u_{\text{opt}} = u_{\text{opt}} - \Pi u_{\text{opt}}.$$

In order to simply the second term we can start by writing

$$\Psi^T A = \Psi^T Y^T Y = X^T Y = (U_X \Sigma_X V_X^T)^T Y = V_X \Sigma_X U_X^T Y$$

which implies that

$$G^{-1}\Psi^T A = V_X \Sigma_X^{-2} V_X^T V_X \Sigma_X U_X^T Y = V_X \Sigma_X^{-1} U_X^T Y.$$

From those observation it follows that

$$\begin{aligned}\|u_{\text{opt}} - u_{\text{reg}}\| &\leq \|u_{\text{opt}} - \Pi u_{\text{opt}}\| + \|\Psi V_X \Sigma_X^{-1} U_X^T Y(I - \Pi)u_{\text{opt}}\| \\ &\leq \|u_{\text{opt}} - \Pi u_{\text{opt}}\| \left(1 + \|\Psi V_X \Sigma_X^{-1} U_X^T Y\|\right).\end{aligned}$$

If we write $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ for the smallest and largest eigenvalues of A, then it must hold that

$$\lambda_{\min}(A) \leq \lambda_{\min}(G) \leq \lambda_{\max}(G) \leq \lambda_{\max}(A)$$

because Ψ has orthogonal columns. Indeed, if $\mathbb{S}^{n-1} \doteq \{w \in \mathbb{R}^n : \|w\| = 1\}$ is the n-dimensional unit sphere, then

$$\min_{w\in\mathbb{S}^{n-1}} w^T Aw \leq \min_{w\in\mathcal{S}_\rho\cap\mathbb{S}^{n-1}} w^T Aw \leq \max_{w\in\mathcal{S}_\rho\cap\mathbb{S}^{n-1}} w^T Aw \leq \max_{w\in\mathbb{S}^{n-1}} w^T Aw$$

is obviously true. Since the columns of Ψ form an ONB of $\mathcal{S}_\rho$ we have

$$\begin{aligned}\min_{w\in\mathcal{S}_\rho\cap\mathbb{S}^{n-1}} w^T Aw &= \min_{w\in\mathbb{S}^{\rho-1}} w^T\Psi^T A\Psi w = \min_{w\in\mathbb{S}^{\rho-1}} w^T Gw = \lambda_{\min}(G) \\ \max_{w\in\mathcal{S}_\rho\cap\mathbb{S}^{n-1}} w^T Aw &= \max_{w\in\mathbb{S}^{\rho-1}} w^T\Psi^T A\Psi w = \max_{w\in\mathbb{S}^{\rho-1}} w^T Gw = \lambda_{\max}(G).\end{aligned}$$

Thus, $\|\Sigma_X^{-1}\|^2 = \lambda_{\min}^{-1}(G) \leq \lambda_{\min}^{-1}(A)$. Clearly we also have $\|Y\|^2 = \lambda_{\max}(A)$. Due to orthogonality we know that $\|\Psi\| = \|V_X\| = \|U_X\| = 1$. Combining those estimates we obtain

$$\|\Psi V_X \Sigma_X^{-1} U_X^T Y\| \leq \sqrt{\frac{\lambda_{\max}(A)}{\lambda_{\min}(G)}} \leq \sqrt{\kappa(A)},$$

which yields the desired bound. □

If the subspace $\mathcal{S}_\rho$ is such that the relative projection error is small, then the norm of u_{reg} will be similar to the norm of u_{opt}. More precisely,

$$\frac{\|u_{\text{reg}} - u_{\text{opt}}\|}{\|u_{\text{opt}}\|} \leq \delta \implies \frac{\|u_{\text{reg}}\|}{\|u_{\text{opt}}\|} \in [1-\delta, 1+\delta]$$

so that Theorem 4.1 applies to $\|u_{\text{reg}} - \hat{u}_{\text{reg}}\|/\|u_{\text{opt}}\|$ with a small δ-dependent constant. By combining the previous two theorems we obtain the following.

Corollary 4.3. *Let $\varepsilon_{\text{R}} \in (0,1)$ and assume that the assumptions of Theorem 4.1 are satisfied for $\varepsilon = \varepsilon_{\text{R}}$. If u_{opt} is the solution of (7) and the subspace $\mathcal{S}_\rho$ is such that*

$$\|u_{\text{opt}} - \Pi u_{\text{opt}}\| \leq \|u_{\text{opt}}\|\varepsilon_{\text{P}}$$

for some $\varepsilon_{\text{P}} \in (0,1)$. Then the total error of the solutions $\hat{u}_{\text{reg}} = \Psi\hat{r}$ produced by Algorithm 1 satisfy the bound

$$\frac{\|u_{\text{opt}} - \hat{u}_{\text{reg}}\|}{\|u_{\text{opt}}\|} \leq \left(1 + \varepsilon_{\text{P}}\sqrt{\kappa(A)}\right)\sqrt{\kappa(G)}\frac{\varepsilon_{\text{R}}}{1-\varepsilon_{\text{R}}} + \left(1+\sqrt{\kappa(A)}\right)\varepsilon_{\text{P}}. \quad (42)$$

Proof. We can start with the estimate

$$\frac{\|u_{\text{opt}} - \hat{u}_{\text{reg}}\|}{\|u_{\text{opt}}\|} \leq \frac{\|u_{\text{opt}} - u_{\text{reg}}\|}{\|u_{\text{opt}}\|} + \frac{\|u_{\text{reg}} - \hat{u}_{\text{reg}}\|}{\|u_{\text{opt}}\|}.$$

Using the estimate from Theorem 4.2 we get

$$\frac{\|u_{\text{opt}} - u_{\text{reg}}\|}{\|u_{\text{opt}}\|} \leq \left(1+\sqrt{\kappa(A)}\right)\frac{\|u_{\text{opt}} - \Pi u_{\text{opt}}\|}{\|u_{\text{opt}}\|} \leq \left(1+\sqrt{\kappa(A)}\right)\varepsilon_{\text{P}}.$$

It remains to bound the other term. Since Ψ has orthogonal columns we obtain from Theorem 4.1

$$\frac{\|u_{\text{reg}} - \hat{u}_{\text{reg}}\|}{\|u_{\text{reg}}\|} \leq \sqrt{\kappa(G)}\frac{\varepsilon_{\text{R}}}{1-\varepsilon_{\text{R}}} \implies \frac{\|u_{\text{reg}} - \hat{u}_{\text{reg}}\|}{\|u_{\text{opt}}\|} \leq \frac{\|u_{\text{reg}}\|}{\|u_{\text{opt}}\|}\sqrt{\kappa(G)}\frac{\varepsilon_{\text{R}}}{1-\varepsilon_{\text{R}}}.$$

Since we have shown in the proof of Theorem 4.2 that

$$u_{\text{reg}} = \Pi u_{\text{opt}} + \Psi G^{-1}\Psi^T A(I-\Pi)u_{\text{opt}}$$

we can estimate

$$\|u_{\text{reg}}\| \leq \|\Pi u_{\text{opt}}\| + \|\Psi G^{-1}\Psi^T A(I-\Pi)u_{\text{opt}}\| \leq \|u_{\text{opt}}\| + \sqrt{\kappa(A)}\|(I-\Pi)u_{\text{opt}}\|.$$

As before, we have used the fact that

$$\Psi G^{-1}\Psi^T A = \Psi V_X \Sigma_X^{-1} U_X^T Y \implies \|\Psi G^{-1}\Psi^T A\| \leq \sqrt{\kappa(A)}.$$

From $\|u_{\text{opt}} - \Pi u_{\text{opt}}\| \leq \varepsilon_{\text{P}} \|u_{\text{opt}}\|$ it follows that

$$\frac{\|u_{\text{reg}}\|}{\|u_{\text{opt}}\|} \leq 1 + \varepsilon_{\text{P}} \sqrt{\kappa(A)},$$

which completes the proof. □

If we assume that $\varepsilon_{\text{P}} \sqrt{\kappa(G)} \approx 1$, then the error estimate from Corollary 4.3 states, with small leading constants, that

$$\frac{\|u_{\text{opt}} - \hat{u}_{\text{reg}}\|}{\|u_{\text{opt}}\|} \leq \mathcal{O}\left((\varepsilon_{\text{R}} + \varepsilon_{\text{P}}) \sqrt{\kappa(A)} \right).$$

It therefore makes sense to have a sketching error ε_{R} that is of the same order as the projection error ε_{P}. In practice we found that projection errors of roughly 1% to 10% can be expected so that the sketching induced error isn't very harmful if we choose the sample size as in Theorem 4.1 with $\varepsilon_{\text{R}} = 0.1$. As illustrated in (40) and (41), the accuracy in our approach is limited by both, the subspace projection and sketching error. The proposed method is therefore most useful when a moderate error of 1%-10% is acceptable in each query. In situations where the solutions of the FEM system are used for further computations that require substantially more accurate solutions it would be necessary to select a large number of basis elements and thus a large number of samples as well. This makes the queries computationally more expensive and the approach much less appealing.

5. Numerical results

To test the performance of Algorithm 1 we consider the finite element formulation of the elliptic equation (1) with homogeneous Dirichlet boundary conditions $u = 0$ on $\partial\Omega$ and a forcing term derived from a piecewise constant approximation of the function

$$f(x) = \begin{cases} 5 & \text{if } \sqrt{(x_1 + \frac{1}{2})^2 + x_2^2 + x_3^2} \leq 0.3, \\ 0 & \text{otherwise,} \end{cases}.$$

We discretise the model on a spherical domain Ω ($d = 3$) of unit radius comprising $k = 684560$ unstructured linear tetrahedral elements. This leads to a total 116805 nodes of which $n = 101509$ are situated in the interior of the domain. In these circumstances X is a tall matrix with 2053680 rows, the stiffness matrix A has dimensions 101509×101509 and the sample space is [2053680]. To test how out sketching algorithm performs in increasing problem dimensions, we run some tests on a finer discretisation of the domain with $k = 1688869$ elements and 315744 with $n = 257374$ are in the interior, yielding sample space of dimension 5066607. Given that the corresponding results are very similar and result in the same conclusions we haven't included those in full detail.

We seek to assess the practical performance of our algorithm in terms of its speed and accuracy in computing the sketched solution under various choices sampling budgets and low-dimensional subspaces, for the proposed sampling distribution. To achieve this we perform three benchmark tests involving realisations of (i) a uniformly distributed random parameter field, (ii) a smoothly varying lognormal random field, and (iii) a random field with jump discontinuities. For each of these we run a sequence of $N = 100$ simulations, i.e. p queries, and record timings and error measures on average. For each realisation we compute also the conventional FEM solution to provide a reference for comparison. The high-dimensional u_{opt} is computed using Matlab's built-in `A\b` command [27]. Given that this is not very efficient and thus not the best performance benchmark we have additionally provided times corresponding to the computation of an approximate, i.e. stopped at 10% error tolerance, solution u_{PCG} using a preconditioned conjugate gradient (PCG) method. Our code was implemented in Matlab R2018b and executed on a workstation equipped with two 14-core Intel Xeon dual processors, running Linux NixOS with 384GB RAM.

In the offline phase of Algorithm 1 we form a low-dimensional ONB for the projection by computing the last eigenfunctions of the sparse Laplacian matrix discretised on Ω. For this time consuming and memory demanding operation we have resorted to the `svds` and `qr` commands which avoid computing the complete spectrum or they produce a sparse ONB respectively. The computation of the sampling distribution based on the leverage scores of $X_\Delta = Z_\Delta D\Psi$ was also performed once during the offline phase and took about 4 hours, using the `svd(,'econ')` command. The distribution q was sampled with replacement during the online phase of the algorithm using uniformly random numbers in combination with `histc` (which performs a

binary search on the cumulative probabilities), which indicatively, for the chosen q, outputs a million samples in about 0.3 s. Notice that although this sampling implementation is not independent of the dimension kd, there exist alternative schemes that can handle arbitrarily large distributions with constant complexity [26].

In the implementation of the algorithm we record the following quantities– diagnostics that provide evidence on the performance in the conditions of each benchmark: the ratio $c'/3k$ indicating how many of the rows of X are used in the sketch, the relative subspace projection error $\|\Pi u_{\text{opt}} - u_{\text{opt}}\|/\|u_{\text{opt}}\|$, the upper bound of the randomisation error $\|\hat{G}^{-1}G - I\|$, the relative regression error $\|\hat{u}_{\text{reg}} - u_{\text{reg}}\|/\|u_{\text{reg}}\|$, and the relative total error $\|\hat{u}_{\text{reg}} - u_{\text{opt}}\|/\|u_{\text{opt}}\|$. In the context of real-time model prediction in manufacturing processes an upper limit of 10% for the total error is deemed reasonable.

5.1. Uniformly random parameter field

In this first instance we simulate sketched solutions for 100 parameter vectors $p \in \mathbb{R}^k$ drawn at random from $\mathcal{U}\big([10^{-1}, 10^2]\big)$. Five sets of simulations were performed using ONBs incorporating the last $\rho = \{50, 100\}$ singular functions of the Laplacian. Our focus was on monitoring the trade-off between accuracy and time consumption when $c = \{5 \times 10^5, 10^6, 5 \times 10^6\}$ iid samples are drawn from p. The results are tabulated in table 1.

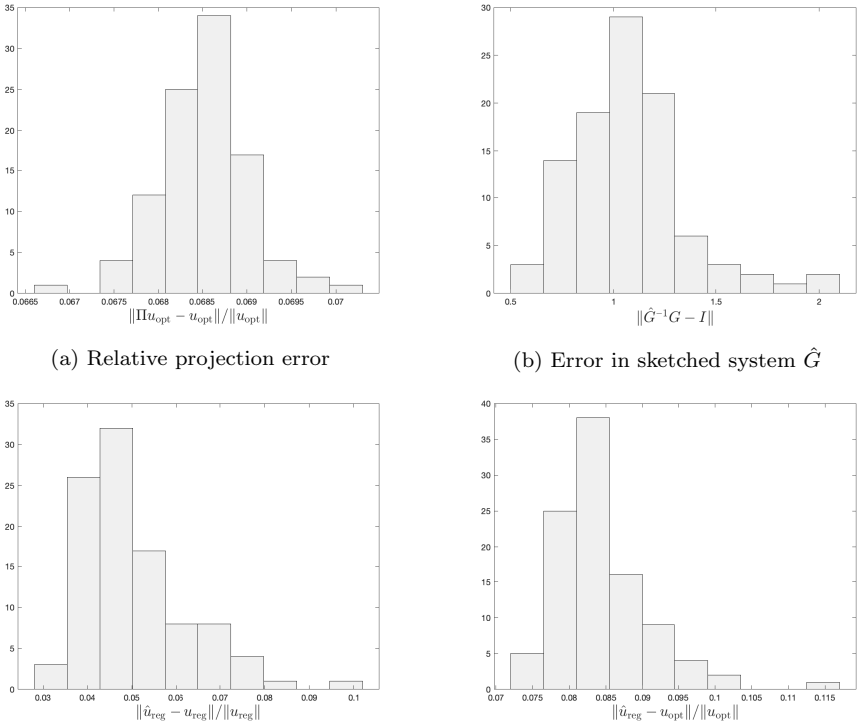
Although the values in p vary over four orders of magnitude, the parameter has a homogeneous expectation within the domain and thus overall the algorithm yields sketched solutions at 10% or less total error, with only 100 basis functions. The results show that the sampling is highly non-uniform since even in the case where a million idd samples were taken these involved only 41074, a mere 6%, of the rows of X. The sketching-induced error factor $\|\hat{G}^{-1}G - I\|$ appears to reduce almost linearly with the number of samples c. Comparing the relative subspace projection $\|\Pi u_{\text{opt}} - u_{\text{opt}}\|$ and total $\|\hat{u}_{\text{reg}} - u_{\text{opt}}\|$ errors note that for $\|\hat{G}^{-1}G - I\| \approx 1$ the later is kept marginally larger than the former, which verifies the regularising effect of the projection on the sketching-induced noise. It is also important to see that in switching from $\rho = 50$ to $\rho = 100$ the projection error is halved to 0.03, however the number of samples necessary to yield the same levels of the error increases by about 5 times. For relative error tolerances around the 10% mark, the times recorded for the smaller mesh are about 0.5 s, while by comparison the time for computing a solution u_{PCG} using a preconditioned conjugate gradient solver (up to the same 10% error tolerance) took 2.40 s (on average from 100

ρ	c [10^6]	time [s]	$c'/3k$	$\frac{\|\Pi u_{\text{opt}} - u_{\text{opt}}\|}{\|u_{\text{opt}}\|}$	$\|\hat{G}^{-1}G - I\|$	$\frac{\|\hat{u}_{\text{reg}} - u_{\text{reg}}\|}{\|u_{\text{reg}}\|}$	$\frac{\|\hat{u}_{\text{reg}} - u_{\text{opt}}\|}{\|u_{\text{opt}}\|}$
50	0.5	0.25	0.04	0.07	1.60	0.07	0.09
50	1	0.46	0.06	0.07	1.07	0.05	0.08
100	0.5	0.34	0.04	0.03	3.99	0.11	0.11
100	1	0.52	0.06	0.03	2.30	0.06	0.07
100	5	2.36	0.11	0.03	0.77	0.02	0.04

Table 1: Numerical results for the tests performed with $p \sim \mathcal{U}([10^{-1}, 10^2])$. The quantities above are averages over 100 runs with different p realisations. The results show the impact of c and ρ on the various error components and the computing times. Note that for a sufficiently large c the total error is only marginally larger than the projection error, which manifest the regularising effect of the projection on the sketching induced error.

runs) on the smaller mesh ($n = 101509$). Computing a PCG solution on the larger grid ($n = 257374$) took on average 3.46 s with relative improvements similar to those on the smaller mesh for a 10% error tolerance.

The trade-off between speed and accuracy can be seen by comparing the results in the first and last rows of the table 1 where the algorithm achieves a 4% total error, when the projection error is at 3%, after five million samples. On the other hand, solutions within a 10% error margin, when the projection error is at 7%, are obtained in less than 0.5 s, which is about 5 times faster than computing a comparable PCG solution. The histograms in figure 1 provide a further insight on how the various error components vary within the ensemble of the 100 problems. We point out that the numerical results are in good agreement with the assertion of Theorem 4.1. For the example shown in figure 1, i.e. when $\rho = 50$ and the error tolerance is $\varepsilon = 10\%$, our theorem predicts $c = 15\rho \log(15\rho)\beta^{-1}\varepsilon^{-2} \approx 5.0 \cdot 10^5 \beta^{-1}$ samples which is consistent to the observed $c = 1$ when $\beta^{-1} \approx 2$. In the histograms we see that the sketching error virtually never exceeds 10% and that $\|\hat{G}^{-1}G - I\|$ exhibits the same pattern as $\|u_{\text{opt}} - u_{\text{reg}}\|/\|u_{\text{opt}}\|$ which supports the claim that this quantity is driving the sketching error. Similar observations can be made for the other cases of table 1. Figure 1 also shows that, although their magnitude is comparable, the variability in the projection error is much smaller than that of the sketching error. This is not surprising as the sketching is an intrinsically random method while the differences in the projection are only due to perturbations in the parameter.



(a) Relative projection error

(b) Error in sketched system $\hat{G}$

(c) Relative sketching error

(d) Relative total error

Figure 1: Histograms showing the variation in the various error quantities relating to the performance of our algorithm, as recorded in the table 1 for 100 different realisations of the p vector from $\mathcal{U}([10^{-1}, 10^2])$ of the code with $\rho = 50$ and $c = 1$ million.

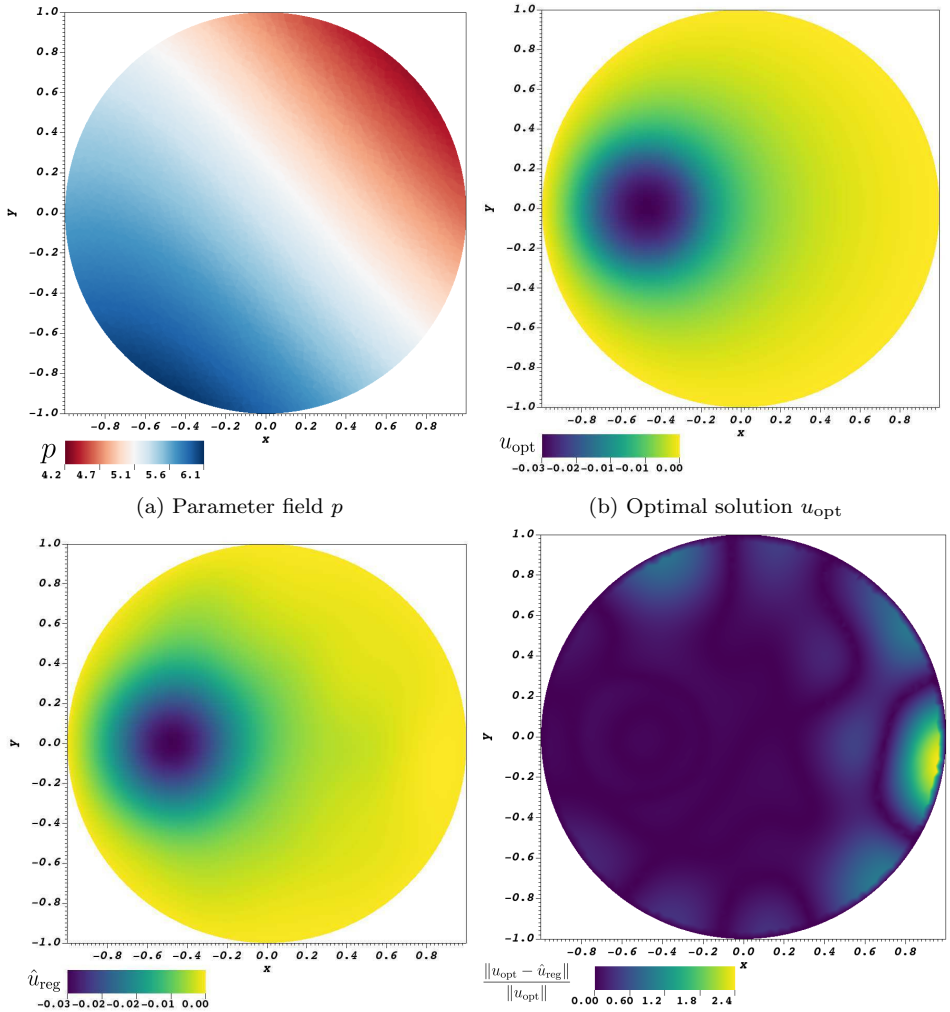
5.2. Smooth parameter field

In the second benchmark we turn our attention to parameter functions with smooth spatial variation like those encountered in the context of uncertainty quantification for PDEs [6]. As the anticipated FEM solution is smooth we maintain the bases used in 5.1. In this case, the parameter p is a lognormal random field given by $p \doteq \exp(b)$, where b is a zero-mean Gaussian random field with Whittle-Matérn covariance function with smoothness parameter $\nu > 0$ given by

$$C_b(x,y) = \frac{\mathrm{Var}[b]}{2^{\nu-1}\Gamma(\nu)} \left(\|x-y\|_M\right)^{\nu} K_\nu \left(\|x-y\|_M\right), \quad x, y \in \Omega, \tag{43}$$

where $\Gamma(\nu)$ is the Gamma function, $\|x\|_M^2 = x^T M^{-1} x$ is the weighted Euclidean norm with positive definite matrix M and K_ν is the order $\nu > 0$ modified Bessel function of the second kind. Here we use $\nu = 15/2$, $M^{1/2} = \mathrm{diag}(1/5, 1/5, 1/5)$ and $\mathrm{Var}[b] = 1$. We draw realisations of p by calculating once the Karhunen-Loève expansion of b and then drawing iid from $\mathcal{N}(0,1)$.

The results presented in table 2 show a similar performance to the uniformly random case in subsection 5.1. The suitability of the low-dimensional subspace is evidenced by the 7% relative projection error attained at $\rho = 50$. Sketched solutions within an error tolerance of 10% were computed in less than 1 s. The timings of the PCG solutions were similar to those corresponding to the uniformly random parameter fields from the previous section and took approximately 5 times longer to compute. Further, note that the total error is within a 2% margin from the projection error, which demonstrates the effectiveness of our sketching regularisation approach, apart from the test with $\rho = 100$ and $c = 1$ where $\|\hat{G}^{-1}G - I\|$ is considerably higher, implying that c was insufficiently small for that test. This observation is consistent with our error bound in (4.1). Comparing the results for $(\rho = 50, c = 5)$ and $(\rho = 100, c = 1)$ shows that in the former case, although using half the number of basis functions and five times more samples, due to the larger projection error, the total error is still 1% larger than that of the later. The images presented in figure 2 correspond to one of the simulations in this benchmark with $\rho = 100$ and $c = 1$ million, illustrating a cross section of the profile of p, the exact FEM solution, the sketched solution and the relative error between the two.



(a) Parameter field p

(b) Optimal solution u_{opt}

(c) Sketched solution $\hat{u}_{\text{reg}}$

(d) Error of the sketched solution

Figure 2: At the top left (a), a view of a lognormal field p sampled from the Whittle-Mattérn class, and to its right (b) the corresponding view of u_{opt}. Below to the left (c), the sketched projected solution $\hat{u}_{\text{reg}}$ and to its right (d) the profile of the relative error between u_{opt} and $\hat{u}_{\text{reg}}$. All illustrations correspond to cross-sections of 3-dimensional functions at $z = 0$.

ρ	c [10^6]	time [s]	$c'/3k$	$\frac{\|\Pi u_{\text{opt}} - u_{\text{opt}}\|}{\|u_{\text{opt}}\|}$	$\|\hat{G}^{-1}G - I\|$	$\frac{\|\hat{u}_{\text{reg}} - u_{\text{reg}}\|}{\|u_{\text{reg}}\|}$	$\frac{\|\hat{u}_{\text{reg}} - u_{\text{opt}}\|}{\|u_{\text{opt}}\|}$
25	0.5	0.23	0.04	0.15	0.73	0.05	0.16
50	1	0.45	0.06	0.07	0.95	0.04	0.08
50	5	1.99	0.12	0.07	0.35	0.02	0.07
100	1	0.56	0.06	0.03	1.97	0.05	0.06
100	5	2.16	0.12	0.03	0.65	0.04	0.04

Table 2: Numerical results for the tests with lognormal random field drawn from a Whittle-Matérn model with a smooth covariance. The algorithm yields solutions with less than 10% error with as few as 50 basis functions. Similar to the uniformly random case in table 1, the total errors are sustained close to the projection errors when $\|\hat{G}^{-1}G - I\| < 1$.

5.3. Non-smooth parameter field

A more challenging benchmark test is to consider the FEM solution for a parameter field with non-smooth variation. In this case it is natural to anticipate that any significant jump discontinuities in the profile of p will have an adverse effect on the condition number of the stiffness matrix [19]. For our simulations we choose a piecewise constant approximation of the positive function

$$p(x) \doteq 9.1 + \text{sgn}(x_1) + 3\text{sgn}(x_2) + 5\text{sgn}(x_3) + 0.1\mathcal{U}\big([0,1]\big)$$

which is discontinuous along the three axes. The sign function $\text{sgn} : \mathbb{R} \to \mathbb{R}$ is given by $\text{sgn}(x) = x/|x|$ when $x \neq 0$ and $\text{sgn}(0) = 0$. In constructing the projection subspace we found that the smooth basis utilised in the previous cases was not appropriate to this case and we thus resorted in a sparse ONB taking a subset of the columns of the sparse unitary matrix computed from the QR decomposition of the Laplacian.

The results in table 3 suggest that the chosen basis is not very appropriate since not only the number of basis functions is substantially larger, but also the reduction in the projection error for a 100% increase in ρ is quiet marginal. In turn, this increase in the dimension of $\hat{G}$ affects the level of sketching error, as even with $c = 5$ million samples $\|\hat{G}^{-1}G - I\| > 1$. Consequently, this has a profound effect on timings which are slightly worse than those corresponding to a PCG approach. For the tests for $(\rho = 2 \times 10^3, c = 10^6)$ and $(\rho = 2 \times 10^3, c = 5 \times 10^6)$ notice that increasing the samples by five times does not yield a significant improvement in the results, which is likely triggered by the large $\kappa(A) \approx 10^5$ in the error term of Theorem 4.2 which causes the $\|u_{\text{reg}} - u_{\text{opt}}\|$ to grow.

ρ	c $[10^6]$	time [s]	$c'/3k$	$\frac{\|\Pi u_{\text{opt}} - u_{\text{opt}}\|}{\|u_{\text{opt}}\|}$	$\|\hat{G}^{-1}G - I\|$	$\frac{\|\hat{u}_{\text{reg}} - u_{\text{reg}}\|}{\|u_{\text{reg}}\|}$	$\frac{\|\hat{u}_{\text{reg}} - u_{\text{opt}}\|}{\|u_{\text{opt}}\|}$
1000	1	2.67	0.06	0.07	4.61	0.01	0.26
1000	5	5.96	0.12	0.05	1.25	0.01	0.26
2000	1	4.87	0.06	0.02	77.36	0.02	0.08
2000	5	9.95	0.12	0.03	9.64	0.01	0.08

Table 3: Numerical results for the non-smooth parameter field. In this case the algorithm requires a far more extensive basis, and thus considerably more samples and computing time to yield solutions within the required 10% error margin.

6. Conclusions

We have considered expediting the solution of the finite element method equations arising from the discretisation of elliptic PDEs on high-dimensional models. Taking into consideration the multi-query context and the smooth profile of the FEM solution, we proposed a practical sketch-based algorithm that involves projection onto lower-dimensional subspace and sketching using a generic, sampling distribution derived from the leverage scores of a tall matrix associated with the Laplacian operator. We have elaborated on the impact of the projection in reducing the dimensionality as well as mitigating the effects of sketching noise. The performance of our method was evaluated in a series of benchmark tests of FEM simulations that demonstrated substantial speed improvements at the cost of a small compromise in accuracy when the stiffness matrix is well conditioned.

7. References

References

[1] H. Elman, D. Silvester, A. Wathen, Finite Elements and Fast Iterative Solvers, Oxford University Press, 2nd edition, 2014.

[2] D. Hartmann, M. Herz, U. Wever, Model Order Reduction a Key Technology for Digital Twins, Springer International Publishing, Cham, pp. 167–179.

[3] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, volume 92, Springer, 2016.

[4] P. Benner, A. Cohen, M. Ohlberger, K. Willcox, Model Reduction and Approximation: Theory and Algorithms, SIAM, 2017.

[5] J. S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer, 2015.

[6] G. J. Lord, C. E. Powell, T. Shardlow, An Introduction to Computational Stochastic PDEs, Cambridge University Press, 2014.

[7] D. Calvetti, M. Dunlop, E. Somersalo, A. M. Stuart, Iterative updating of model error for Bayesian inversion, Inverse Problems 34 (2018) 25008.

[8] D. P. Woodruff, Sketching as a tool for numerical linear algebra, Foundations and Trends® in Theoretical Computer Science 10 (2014) 1–157.

[9] P. Drineas, M. W. Mahoney, Effective resistances, statistical leverage, and applications to linear equation solving, ArXiv, 2010.

[10] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, D. P. Woodruff, Fast approximation of matrix coherence and statistical leverage, Journal of Machine Learning Research 13 (2012) 3441–3472.

[11] H. Avron, S. Toledo, Effective stiffness: Generalizing effective resistance sampling to finite element matrices, ArXiv, 2011.

[12] R. M. Gower, P. Richtárik, Randomized iterative methods for linear systems, SIAM Journal on Matrix Analysis and Applications 36 (2015) 1660–1690.

[13] R. M. Gower, P. Richtárik, Linearly convergent randomized iterative methods for computing the pseudoinverse, ArXiv, 2016.

[14] D. P. Bertsekas, H. Yu, Projected equation methods for approximate solution of large linear systems, Journal of Computational and Applied Mathematics 227 (2009) 27–50.

[15] N. Polydorides, M. Wang, D. P. Bertsekas, A quasi Monte Carlo method for large-scale inverse problems, in: H. Woźniakowski (Ed.), Springer Proceedings in Mathematics and Statistics, volume 23, Monte Carlo and Quasi-Monte Carlo Methods 2010. Springer Proceedings in Mathematics & Statistics, 23, Springer, 2012, pp. 623–637.

[16] L. C. Evans, Partial Differential Equations, American Mathematical Society, 2nd edition, 2010.

[17] R. Kannan, S. Hendry, N. J. Higham, F. Tisseur, Detecting the causes of ill-conditioning in structural finite element models, Computers & Structures 133 (2014) 79–89.

[18] S. A. Vavasis, Stable finite elements for problems with wild coefficients, SIAM Journal on Numerical Analysis 33 (1996) 890–916.

[19] L. Kamenski, W. Huang, H. Xu, Conditioning of finite element equations with arbitrary anisotropic meshes, Mathematics of Computation 83 (2014) 2187–2211.

[20] M. Pilanci, M. J. Wainwright, Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares, Journal of Machine Learning Research 17 (2014) 1–38.

[21] A. Neumaier, Solving ill-conditioned and singular linear systems: A tutorial on regularization, SIAM Review 40 (1998) 636–666.

[22] P. Drineas, R. Kannan, M. W. Mahoney, Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication, SIAM J. Comput. 36 (2006) 132–157.

[23] J. A. Tropp, An introduction to matrix concentration inequalities, Foundations and Trends in Machine Learning 8 (2015) 1–230.

[24] M. B. Cohen, Y. T. Lee, C. Musco, C. Musco, R. Peng, A. Sidford, Uniform sampling for matrix approximation, in: Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS '15, ACM, New York, NY, USA, 2015, pp. 181–190.

[25] N. Halko, P. G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Review 53 (2011) 217–288.

[26] K. Bringmann, K. Panagiotou, Efficient sampling methods for discrete distributions, Algorithmica 79 (2017) 484–508.

[27] MATLAB, version 9.5.0.944444 (R2018b), The MathWorks Inc., Natick, Massachusetts, 2019.