Multiscale flow in porous media:
A finite element method for the Stokes-Brinkman equation

Kamran Arora

June 23, 2024

Contents

1 Introduction

In this project, we are concerned with modelling resin flow within composite materials. A composite material is simply a material formed as a combination of other materials, which have different properties. Two of the most common examples of composite materials are reinforced concrete and carbon fibres. These materials have the benefit of being strong and often light. The typical manufacturing process uses liquid composite moulding (LCM) technology, where a resin is infused to harden the composite material into its final state. Knowledge of permeability variations can help optimise the LCM stage of the manufacturing process [3], but these cannot be experimentally measured. Instead, numerical simulations are used to predict this quantity. The goal is to develop a fast and accurate numerical method for prediction that can be used to optimise and streamline LCM.

This project stems from the work done in [7], in which the author develops a numerical method based on the Fast Fourier Transform (FFT) to predict the permeability of composite materials. We aim to build upon this in three ways:

1. Encapsulate the variation in local fibre orientation via a distribution, and use this to generate random realisations of a composite material.
2. Implement a solver based on Finite Element Methods (FEM), and compare its performance to the solver proposed in [7].
3. Construct a neural network to predict macroscopic permeability, given image data of the composite material.

The focus of this report is on the development of a numerical solver based on FEM. In particular, in Section 2 we introduce the problem from a technical perspective, and review the FFT method proposed by [7]. In Section 3, we introduce the relevant mathematical techniques used to design our solver. We then perform and analyse a selection of numerical experiments in Section 4, and conclude with a discussion of future directions in Section 5.

2 Problem formulation and motivation

In this section we introduce the problem, and the partial differential equation (PDE) to be solved following the presentation of [7].

The composites we concern ourselves with can be considered at three scales (see Figure 1):

- In the micro-scale, we have individual material fibres placed next to each-other.
- In the meso-scale, the collections of fibres form bundles, oriented in different directions.
- In the macro-scale, the bundles are weaved together to create a dry composite material, into which resin can be injected.

Consider the dual-scale porous medium shown in Figure 2. We consider fluid flow through bundles at the meso-scale, and between bundles at the macro-scale. L_1 is the characteristic length scale of the clear fluid region, and L_2 of the porous solid region.

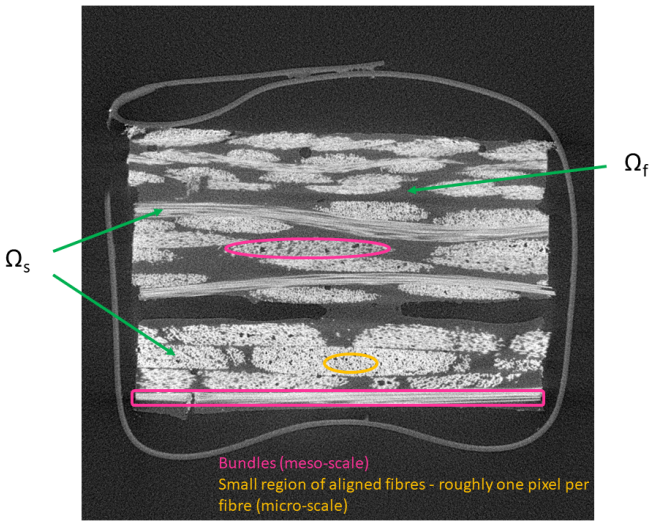


Figure 1: 2D slice of an X-ray CT scan of a composite material. Bundles are outlined in pink, fibres are outlined in orange, and regions of fluid flow are labelled in green.

Assume that $L_2/L_1 \ll 1$ so that the length scales are separable. We consider the flow of an incompressible, Newtonian fluid. In the clear fluid region we use the Stokes equations with dynamic viscosity μ. In the solid porous region we use a Darcy-Brinkman equation [6] with effective permeability μ_e and local permeability tensor $\mathbf{k}_s$. The flow across the unified domain $\Omega = \Omega_f \cup \Omega_s$ can thus be described by a Stokes-Brinkman PDE

$$\begin{cases} \varphi(x)\Delta u(x) - \beta(x)u(x) - \nabla p(x) = f(x) \\ \nabla \cdot u(x) = 0 \end{cases} \quad x \in \Omega, \tag{1}$$

where u is the local velocity, p is the local pressure, and

$$\varphi(x) = \begin{cases} \mu, & x \in \Omega_f, \\ \mu_e, & x \in \Omega_s, \end{cases} \quad \text{and} \quad \beta(x) = \begin{cases} 0, & x \in \Omega_f, \\ \mu \mathbf{k}_s^{-1}, & x \in \Omega_s. \end{cases} \tag{2}$$

To ensure well-posedness of the PDE, appropriate boundary conditions must be enforced. One possible choice is periodic boundary conditions for both velocity and pressure. Following [7], this is the choice we make.

Recall from Section 1, that the problem is to quickly and accurately compute the permeability of a composite material. More precisely, the quantity of interest is the macroscopic permeability $\mathbf{K}$. We can compute this using Darcy's law at macroscale:

$$U = -\frac{1}{\mu}\mathbf{K} \cdot G, \tag{3}$$

where $U = \langle u \rangle_\Omega$ is the macroscopic velocity, $G = \langle \nabla p \rangle_\Omega$ is the macroscopic pressure gradient, and $\langle \cdot \rangle_\Omega$ denotes an average over the domain Ω. Therefore, by forcing the system with a macroscopic pressure gradient, we can compute the macroscopic permeability. More precisely, we set $f(x)$ in (1) to be G, solve (1) for u, and then use (3) to compute $\mathbf{K}$.

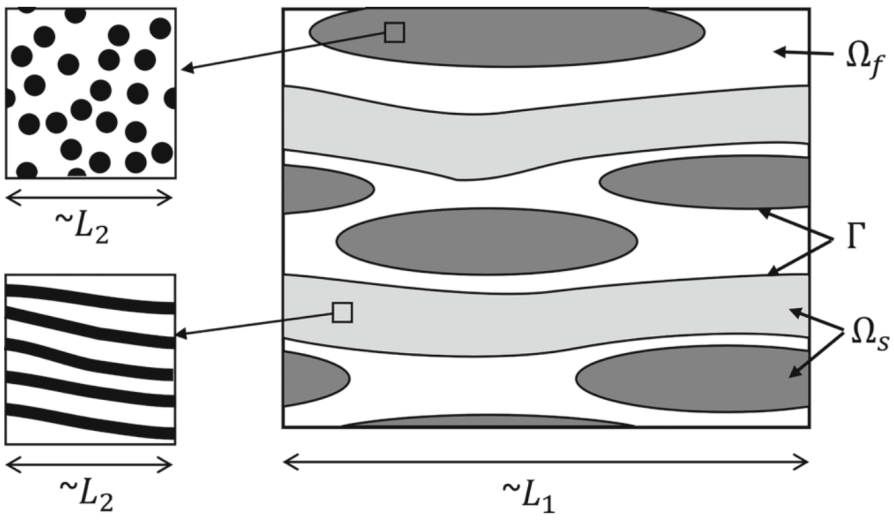


Figure 2: Dual scale porous medium with clear fluid region Ω_f and porous solid region Ω_s with respective length scales L_1 and L_2. Γ is the boundary between the two regions. This figure is taken from [7].

In [7], the author proposes a FFT based solver for (1). Consider the Stokes-Brinkman PDE forced by a macroscopic pressure gradient G. Introduce a polarisation vector

$$\tau(x) = (\varphi(x) - \varphi_0)\Delta u(x) - (\beta(x) - \beta_0 I) \cdot u(x), \tag{4}$$

where φ_0 and β_0 are reference material coefficients. Rewriting (1) in terms of τ and G, and passing into Fourier space gives

$$\begin{cases} -\left(\varphi_0 \|\xi\|^2 + \beta_0\right) \hat{u} - i\xi\hat{p}' + \hat{\tau} = \hat{G} \\ -\|\xi\|^2 \hat{p}' = i\xi \cdot \hat{\tau} \end{cases} \quad \forall \xi \in \Xi, \tag{5}$$

where the hat denotes a variable in Fourier space, and Ξ is the set of frequency vectors. This problem has a closed form solution

$$\hat{u} = \begin{cases} \frac{1}{\beta_0}\left(\hat{\tau}(0) - G\right), & \xi = 0 \\ \hat{\mathbb{G}} \cdot \hat{\tau}, & \xi \neq 0, \end{cases} \quad \text{where} \quad \hat{\mathbb{G}} = \frac{I - \frac{\xi \otimes \xi}{\|\xi\|^2}}{\varphi_0 \|\xi\|^2 + \beta_0}. \tag{6}$$

For a given u, one can compute τ. By utilising the FFT, one can then use $\hat{\tau}$ to compute $\hat{u}$. The updated velocity field can then be found with the inverse FFT. Given a stopping criterion (see Section 4.1) this defines an iterative algorithm for solving (1). Full details can be found in Algorithm 3 of [7].

Moreover, the Anderson acceleration technique [18] is introduced in [7] to improve the convergence behaviour of the proposed algorithm. The author validates the algorithm against analytical results for the local velocity field and macroscopic permeability, and benchmarks its performance. Despite this, it is still natural to ask if an alternative solver to the FFT method is better suited to the problem.

In this project we wish to develop and implement a finite element method. In particular, this method avoids the approximation of the problem with reference coefficients, and solves the real problem directly. However, it does need to be suitably preconditioned. For this we use multigrid (see Section 3.3.3). Benefits of FEM include the ability to handle complex domain geometries, and more than just periodic boundary conditions. Moreover, the parallel scalability of FFT solvers is limited by global communcations but FEM solvers do not suffer with this issue so scale well in parallel. Since the solver will ultimately be used in a large-scale, industrial context, this is a highly desirable characteristic. A priori, it is not clear which solver is expected to have greater performance, therefore this is the key question that we aim to answer in this project.

3 Mathematical preliminaries

In this section we provide the necessary technical details that underpin our implementation of a finite element method for (1).

3.1 Weak formulation

Solving a PDE using a finite element method requires it to be written in weak form. Multiplying (1) by test functions $(v, q) \in H_0^1(\Omega; \mathbb{R}^3) \times L_0^2(\Omega; \mathbb{R})$, and integrating by parts gives the following weak form:

$$\begin{cases} \displaystyle\int_\Omega -\nabla u : \nabla(\varphi v) - \beta u \cdot v \,\mathrm{d}x + \int_\Omega p \nabla \cdot v \,\mathrm{d}x = \int_\Omega f \cdot v \,\mathrm{d}x, & \forall v \in H_0^1(\Omega; \mathbb{R}^3), \\ \displaystyle\int_\Omega q \nabla \cdot u \,\mathrm{d}x = 0, & \forall q \in L_0^2(\Omega; \mathbb{R}), \end{cases} \tag{7}$$

where $L_0^2(\Omega; \mathbb{R})$ is the space of $L^2(\Omega; \mathbb{R})$ functions that integrate to zero. Define the bilinear forms

$$a_\beta(u, v) \coloneqq \int_\Omega -\nabla u : \nabla(\varphi v) - \beta u \cdot v \,\mathrm{d}x, \quad \text{and} \quad b(u, q) \coloneqq \int_\Omega q \nabla \cdot u \,\mathrm{d}x, \tag{8}$$

and the continuous linear functional

$$L(v) \coloneqq \int_\Omega f \cdot v \,\mathrm{d}x. \tag{9}$$

The resulting abstract form of equation (7) is:

Find $(u, p) \in H_0^1(\Omega; \mathbb{R}^3) \times L_0^2(\Omega; \mathbb{R})$ such that

$$\begin{cases} a_\beta(u, v) + b(v, p) = L(v), \\ b(u, q) = 0, \end{cases} \tag{10}$$

for all $(v, q) \in H_0^1(\Omega; \mathbb{R}^3) \times L_0^2(\Omega; \mathbb{R})$. For our choice of φ and β, one can use Brezzi's theorem to prove that there exists a unique solution to (10) (see Appendix A).

In Appendix B we show that (12) arises from the following constrained minimisation problem

$$\text{Minimise} \quad J(u) = \frac{1}{2}a_\beta(u,u) - L(u) \quad \text{subject to} \quad \nabla \cdot u = 0, \tag{11}$$

where pressure is the Lagrange multiplier for the divergence-free condition. The unique solution to (12) is a saddle point of the Lagrangian associated with (11), therefore the Stokes-Brinkman equations are called a saddle point system.

In Section 3.2, we define a linear system (12) that is the discrete analogue of the saddle point system given in (10). These systems are a general class of problems that arise throughout science and engineering. In particular, there exists susbtantial literature devoted to the efficient numerical solution of such systems [2, 1], including papers specific to Stokes-type problems [8, 4, 12]. In Section 3.3, we use this information to choose an appropriate numerical method for solving the linear system.

3.2 Finite element approximation

To obtain a finite element discretisation of (10) we consider finite-dimensional closed subspaces V_h and Q_h of V and Q respectively, and equip them with basis functions. For this problem, the finite element discretisation must satisfy an inf-sup condition (Appendix C) in order to be stable. We choose the Taylor-Hood finite element which was shown to give a stable discretisation in [17]. This choice corresponds to piecewise quadratic basis functions for velocity, and piecewise linear basis functions for pressure. We denote these bases by $\{\phi_i\}_{i=0}^{N-1}$ and $\{\psi_j\}_{j=0}^{M-1}$ respectively. This is not the only stable choice, but we have picked it due to it being the simplest such choice to implement.

Inserting the basis functions into the forms defined in (8) and (9) yields a finite-dimensional system of linear equations

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}, \tag{12}$$

where

$$A_{ij} = a_\beta(\phi_j, \phi_i) = \int_\Omega -\nabla\phi_j : \nabla(\varphi\phi_i) - \beta\phi_j \cdot \phi_i \,\mathrm{d}x, \tag{13}$$

$$B_{ij} = b(\phi_j, \psi_i) = \int_\Omega \psi_i \nabla \cdot \phi_j \,\mathrm{d}x, \tag{14}$$

$$F_i = L(\phi_i) = \int_\Omega f \cdot \phi_i \,\mathrm{d}x. \tag{15}$$

3.3 Iterative methods

Recall that we are solving the system of linear equations given in (12). A first choice for solving such a system would be a direct solver utilising, for example, a Cholesky or LU decomposition.

Consider solving (1) on a 2D mesh of size $N \times N$. The Taylor-Hood finite element corresponds to 1 pressure degree of freedom (DoF) per cell, and 4 velocity DoFs per cell. Thus, for a

problem size of $N \times N$, we have $5N^2$ DoFs. As we increase N, using a direct solver quickly becomes computationally expensive and numerically unstable.

We turn to iterative methods. The basic idea is as follows. Consider a general linear system $Ax = b$, and choose an initial guess for the solution x^0. The iterate x^{k+1} can then be computed from x^k according to some chosen update rule. The goal is to define a scheme such that x^k is close to the true solution after a small number of iterations.

In [12], the authors study iterative solvers for a generalised Stokes problem

$$-\nu\Delta u + \xi u + \nabla p = f, \quad \nu > 0, \xi \geq 0. \tag{16}$$

Observe that (up to a change of sign) this is our Stokes-Brinkman PDE with the functions $\varphi(x)$ and $\beta(x)$ replaced by ν and ξ. Two smoothers, Braess-Sarazin and Vanka, are introduced, and convergence results are proved for when these are coupled with multigrid methods. The authors also numerically verify their theoretical result. Moreover, the results in [12] correspond to the Taylor-Hood finite element. Motivated by this, these are the two methods we choose to implement for solving (12).

3.3.1 The Braess-Sarazin smoother

The Braess-Sarazin (BS) smoother was first introduced in [4]. We give a brief overview.

Computing the block factorisation of the matrix in (12), and inverting gives

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix}^{-1} = \begin{pmatrix} I & -A^{-1}B^{\mathrm{T}} \\ 0 & I \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -BA^{-1} & I \end{pmatrix}, \tag{17}$$

where $S \coloneqq -BA^{-1}B^{\mathrm{T}}$ is the Schur complement. The solution of (12) thus relies on efficient computations of A^{-1} and S^{-1}. Unfortunately, A^{-1} is dense therefore S is also dense, and therefore not easily determined.

The BS smoother replaces A with αD_A where $D_A = \mathrm{diag}(A)$ and $\alpha > 0$ is chosen to be greater than or equal to the largest eigenvalue of A. This gives the following iteration

$$\begin{pmatrix} u_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} u_k \\ p_k \end{pmatrix} - \begin{pmatrix} \alpha D_A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix}^{-1} \left[\begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} u_k \\ p_k \end{pmatrix} - \begin{pmatrix} f \\ 0 \end{pmatrix} \right]. \tag{18}$$

Each step of the iteration involves solving

$$\begin{pmatrix} \alpha D_A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} \tilde{u} \\ \tilde{p} \end{pmatrix} = \begin{pmatrix} r_k \\ Bu_k \end{pmatrix}, \tag{19}$$

where $\tilde{u} = u_k - u_{k+1}$, $\tilde{p} = p_k - p_{k+1}$ and $r_k = Au_k + B^{\mathrm{T}}p_k - f$. Crucially, it follows that $Bu_{k+1} = 0$ so that the BS iteration preserves the divergence-free condition. Equation (19) can be reduced to solving

$$BD_A^{-1}B^{\mathrm{T}}\tilde{p} = BD_A^{-1}r_k - \alpha Bu_k, \tag{20}$$

which is a Poisson-like equation for $\tilde{p}$. This can be computed using an iterative solver, then $\tilde{u}$ can be recovered via

$$\alpha D_A \tilde{u} = r_k - B^{\mathrm{T}}\tilde{p}. \tag{21}$$

Crucially, D_A is diagonal meaning $\tilde{u}$ can be computed very cheaply.

3.3.2 The Vanka smoother

In [12], the authors introduce a multiplicative Vanka smoother. We instead choose to use an additive Vanka smoother. The latter has both proven convergence results [15], and is simpler to implement. We give a brief overview following [15].

Consider the saddle point problem given in (12). Suppose we have n_p degrees of freedom for the pressure variable. Split the domain into subdomains $\{\mathcal{S}_i\}_{i=1}^{n_p}$ such that $\mathcal{S}_i$ consists of the ith pressure DoF and the connected velocity DoFs. Let R_i denote the restriction operator to $\mathcal{S}_i$, and its transpose R_i^{T}, the associated prolongation. Writing

$$\mathcal{A} = \begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix},$$

define the local block $\mathcal{A}_i = R_i \mathcal{A} R_i^{\mathrm{T}}$. For a given weighting matrix ω_i, the additive Vanka smoothing operator is then given by

$$S_{AV} \coloneqq \sum_{i=1}^{n_p} R_i^{\mathrm{T}} \omega_i \mathcal{A}_i^{-1} R_i. \tag{22}$$

Intuitively, one step of the additive Vanka smoother involves a loop over the pressure DoFs. For each DoF a small, local system is solved, usually with a direct solver. For a multiplicative Vanka smoother, the DoF updates would take place in a Gauss-Seidel manner, however for the additive version these are computed simultaneously.

3.3.3 Multigrid

The two smoothers introduced thus far will have very slow convergence if used as stand-alone iterations because they only reduce the low-frequency components of the error. Multigrid methods address this problem by reducing the error on all length scales. Moreover, multigrid methods have been shown to be one of the most efficient techniques for solving saddle point problems [12], and there exists theoretical convergence guarantees when BS or Vanka are used as smoothers. Therefore, it is clear that such methods should be included in our implementation. We give a brief overview of the key ideas underpinning multigrid following the presentation in [13].

Consider a nested sequence of mixed finite element spaces

$$(V_{\ell-1}, Q_{\ell-1}) \subset (V_\ell, Q_\ell), \quad \ell = 1, \ldots, L$$

with a fixed finest level $\ell = L \geq 2$. Associated to these spaces are restriction operators R_ℓ to move from level ℓ to $\ell - 1$, and prolongation operators P_ℓ to move from level $\ell - 1$ to ℓ. Taking a finite element discretisation of a suitable, weakly formulated PDE on each level leads to a linear system on each level, $A_\ell x_\ell = b_\ell$, with true solution x_ℓ^*. Also define $p_\ell = P_\ell^{-1} P_{\ell-1}$, and $r_\ell = p_\ell^{\mathrm{T}}$.

For brevity, we discuss a two grid setup. The basic idea is to perform a small number of iterations of a smoother on the fine grid. This smooths the error. The error on the coarse

grid is cheap to compute, and prolongated onto the fine grid as an approximation to the fine error. This is then used as a correction term. It is common to apply postsmoothing steps after adding the correction because the prolongation introduces additional errors. We summarise this in Algorithm 1.

Input: x_ℓ, b_ℓ
if $\ell = 0$ **then**
  $x_0 = A_0^{-1} b_0$;
else
  $x_\ell = \text{SMOOTHER}(x_\ell, b_\ell)$; /* smoother e.g. BS or Vanka */
  $d_{\ell-1} = r_\ell(b_\ell - A_\ell x_\ell)$; /* restriction of defect */
  $e_{\ell-1} = A_{\ell-1}^{-1} d_{\ell-1}$; /* solve coarse grid problem */
  $x_\ell = x_\ell + p_\ell e_{\ell-1}$; /* add correction */
  $x_\ell = \text{SMOOTHER}(x_\ell, b_\ell)$; /* postsmoothing */
end
Output: x_ℓ

Algorithm 1: Two grid method. Reproduced from [13].

In general, multigrid methods use a hierarchy of coarse grids where the two-grid method is applied recursively. More details can be found in [13].

For our implementation we use the aformentioned iterative solvers as preconditioners for a Krylov-subspace iteration. More precisely, we consider the Krylov-subspace method GRMES [14]. We then either use BS or Vanka as a standalone preconditioner, or use multigrid as a preconditioner with BS or Vanka as the smoother. In [18], it was shown that Anderson acceleration is equivalent to GMRES for linear problems such as the one under consideration. Thus the solver in [7] can be viewed as GMRES preconditioned with FFT, and, as described above, the FEM solver we develop is a preconditioned GMRES method. This justifies the choice to consider a FEM solver, namely, the work in this project is equivalent to exploring if there exists a better preconditioner for solving (1) using a preconditioned GMRES iteration.

4 Numerical experiments

4.1 Implementation

We implement our solvers using the Firedrake python library [10]. All the results in this paper were generated using the code available at the following link https://github.com/kamran-arora/Stokes-Brinkman-IRP. We also made use of the code used to perform the experiments in [7]. This code is available at https://github.com/yang-chen-2022/fftpermeability-StokesBrinkman.

Motivated by Section 3.2, we use the Taylor-Hood finite element. For the BS smoother we choose $\alpha = 1.25$ because this is shown to be optimal for the problem in [12], however in future work it would be instructive to consider varying this value. Similarly, we choose $\mu = \mu_e$ for simplicity but in future work these values should be varied. All experiments

in this paper used periodic boundary conditions in velocity and pressure for the chosen mesh. Moreover, any experiments that use multigrid have the following setup: the mesh is of resolution $2^K \times 2^K$ where $K \geq 4$, and the number of levels (see Section 3.3.3) is $K - 3$. Parameter details for individual experiments are described in their corresponding sections of this report.

4.2 Stopping criteria

We now describe the different criteria used to determine convergence of the FFT and FEM solvers.

4.2.1 FFT

The FFT solver uses the difference in subsequent iterations of velocity to determine convergence [7]. In particular, given a tolerance $\varepsilon > 0$, the solver is said to have converged if

$$\frac{\sqrt{\left\langle \|u^{k+1} - u^k\|_2^2 \right\rangle_\Omega}}{\|U\|_2} < \varepsilon, \tag{23}$$

where u^k denotes the kth iterate of the local velocity field.

4.2.2 FEM

The FEM solver uses the residual of the linear system to determine convergence. In particular, rewrite (12) as $\mathcal{A}x = b$. For the kth iterate x^k, define the (unpreconditioned) residual

$$r^k \coloneqq b - \mathcal{A}x^k. \tag{24}$$

Convergence is determined by a relative tolerance, RTOL > 0. This means that the solver is said to converge if

$$\left\|r^k\right\|_2 < \text{RTOL} \times \left\|r^0\right\|_2. \tag{25}$$

For some experiments, we instead use the preconditioned residual defined by

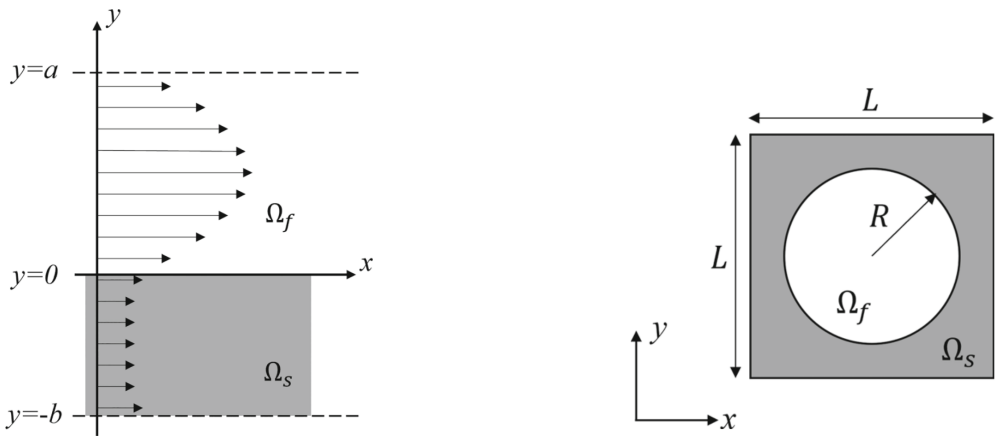
$$r_p^k \coloneqq P^{-1}\left(b - \mathcal{A}x^k\right), \tag{26}$$

where P^{-1} is an arbitrary preconditioner. The choice of residual for a given experiment is given in the associated section of this report. Determining which choice of residual should be used in practice is a topic for future work.

4.3 Domain geometries

In Sections 4.4 and 4.5, we use the following two domain geometries for testing. These are the geometries used in [7] for algorithm validation and benchmarking, therefore are suitable for initial experiments using our FEM solver.

As described in Section 2, $\beta(x)$ takes the value 0 when $x \in \Omega_f$, and the value μ/k_s when $x \in \Omega_s$.



(a) Parallel channel

(b) Square hole

Figure 3: Schematics of the toy problem domain geometries. These figures are taken from [7].

4.4 Initial experiments

Our initial experiments are based on the square-hole geometry shown in Figure 3(b). The author of [7] used this to benchmark their algorithms, thus it is a suitable toy problem on which to test our implementation.

For the first experiment we fix a mesh size of 64×64, let $L = 1$ mm, and $R = 0.4$ mm. We use microscopic permeabilities $k_s \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}\}$ mm^2. The loading condition is a macroscopic pressure gradient $G = 1$ MPa in the positive x-direction, and $\mu = \mu_e = 1$ MPa$\cdot$s. The FFT tolerance is $\varepsilon = 10^{-6}$, and the FEM tolerance is RTOL $= 10^{-6}$ for the preconditioned residual (26). We run the solvers to convergence, and use (3) to compute $\mathbf{K}$ which we then rescale by k_s.

Figure 4 shows our results and those of the FFT solver. We immediately see that the two solvers in their current states are not converging to the same solution. We hypothesise that this is due to two reasons:

1. The stopping criterions introduced in Section 4.2 are not equivalent, namely, choosing $\varepsilon =$ RTOL does not mean the solvers are stopping at the same solution.
2. Error due to the numerical discretisation. In particular, our solver uses a finite element discretisation whereas the FFT solver uses a staggered finite difference [7].

Crucially, this initial experiment suggests further analysis into the convergence behavior of the FEM solvers is necessary before a fair performance comparison with the FFT solver can be made.

Furthermore, there is concerning numerical behaviour at the boundary between the fluid and solid region for both the FEM and FFT solvers. An example using BS with multigrid is given in Figure 5, where $k_s = 10^{-6}$. We see that the solution becomes numerically unstable at the interface but is otherwise smooth. We expect global continuity of our solution so

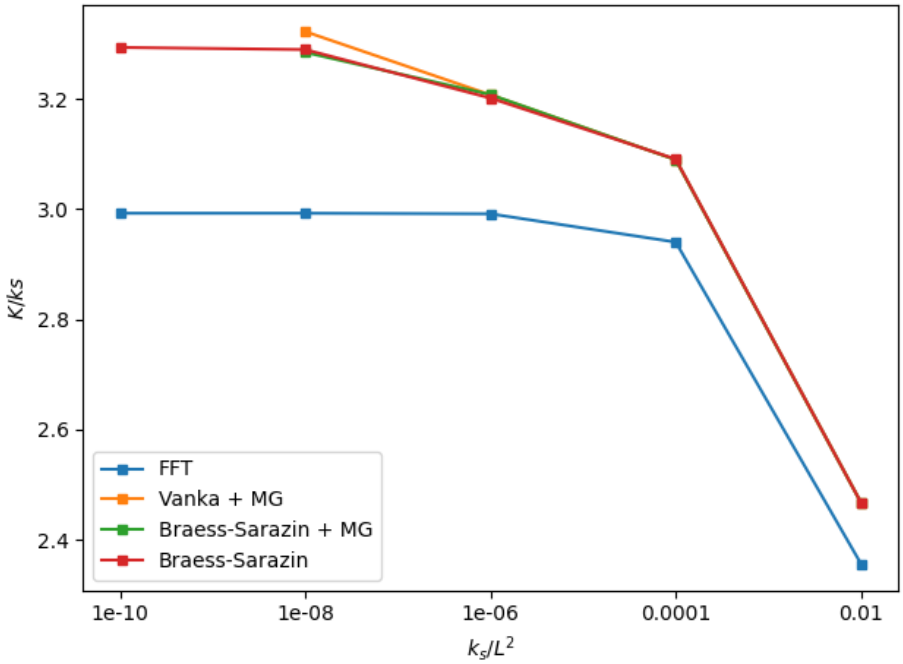
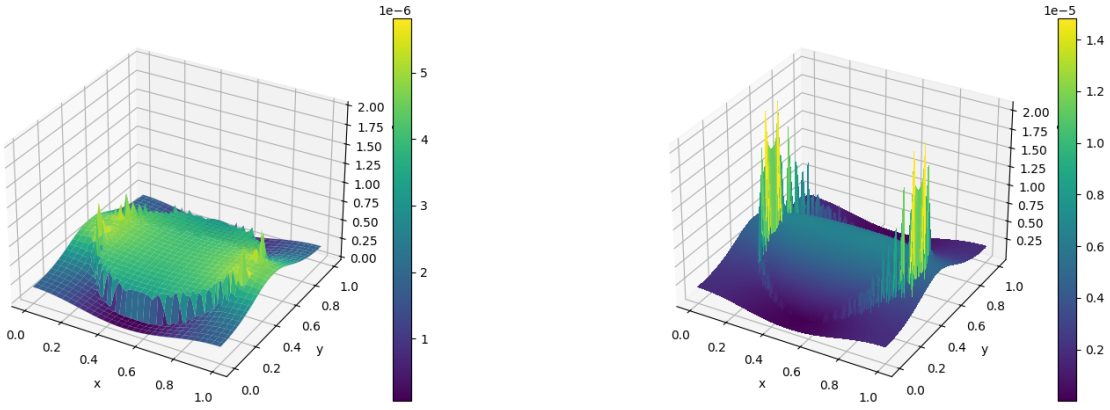


Figure 4: Rescaled macroscopic permeabilities for the square hole geometry on a 64×64 mesh. Comparison between FEM and FFT solvers. No marker means the method failed to converge. Parameters: $k_s \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}\}, \mu = \mu_e = 1, G = 1, R/L = 0.4, \varepsilon = 10^{-6}$ and RTOL $= 10^{-6}$ (preconditioned).

this is another problem that needs addressing before the solvers can be compared. We also observe that the "blow-up" is an order of magnitude larger for the FEM solver.



(a) FFT

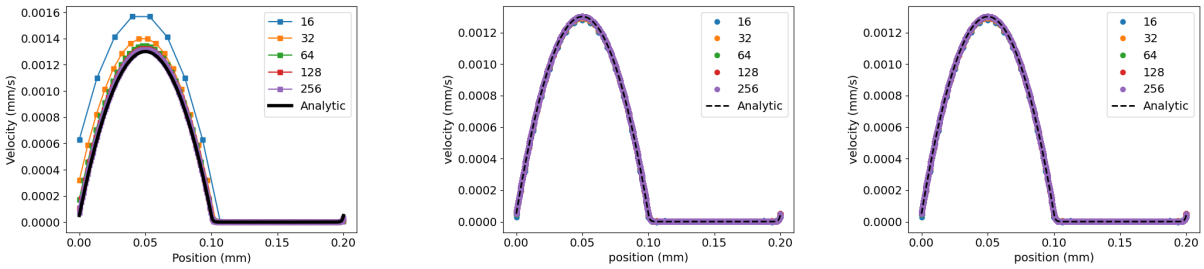
(b) FEM: BS with multigrid

Figure 5: Local velocity field computed for the square hole geometry using FFT, and BS with multigrid on a 64×64 mesh. Parameters: $k_s = 10^{-6}, \mu = \mu_e = 1, G = 1, R/L = 0.4, \varepsilon = 10^{-6}$ and RTOL $= 10^{-6}$ (preconditioned).

4.5 Parallel channel geometry

In light of Section 4.4, we need to validate the FEM solver. We use the parallel channel geometry (Figure 3) introduced in [7] which has a known analytical solution.

The FFT solver uses a mesh-size of $1 \times n_y$. Due to technical limitations, the FEM solver has to be implememted on a mesh of size $n_x \times n_y$. Full details of the differences in implementation are given in Appendix D. We set $a = b = 0.1$ mm, $\mu = \mu_e = 1$ MPa $\cdot$ s, and $= 1$ MPa. The FFT tolerance is $\varepsilon = 10^{-6}$, and the FEM tolerance is RTOL $= 10^{-6}$ for the preconditioned residual (26).

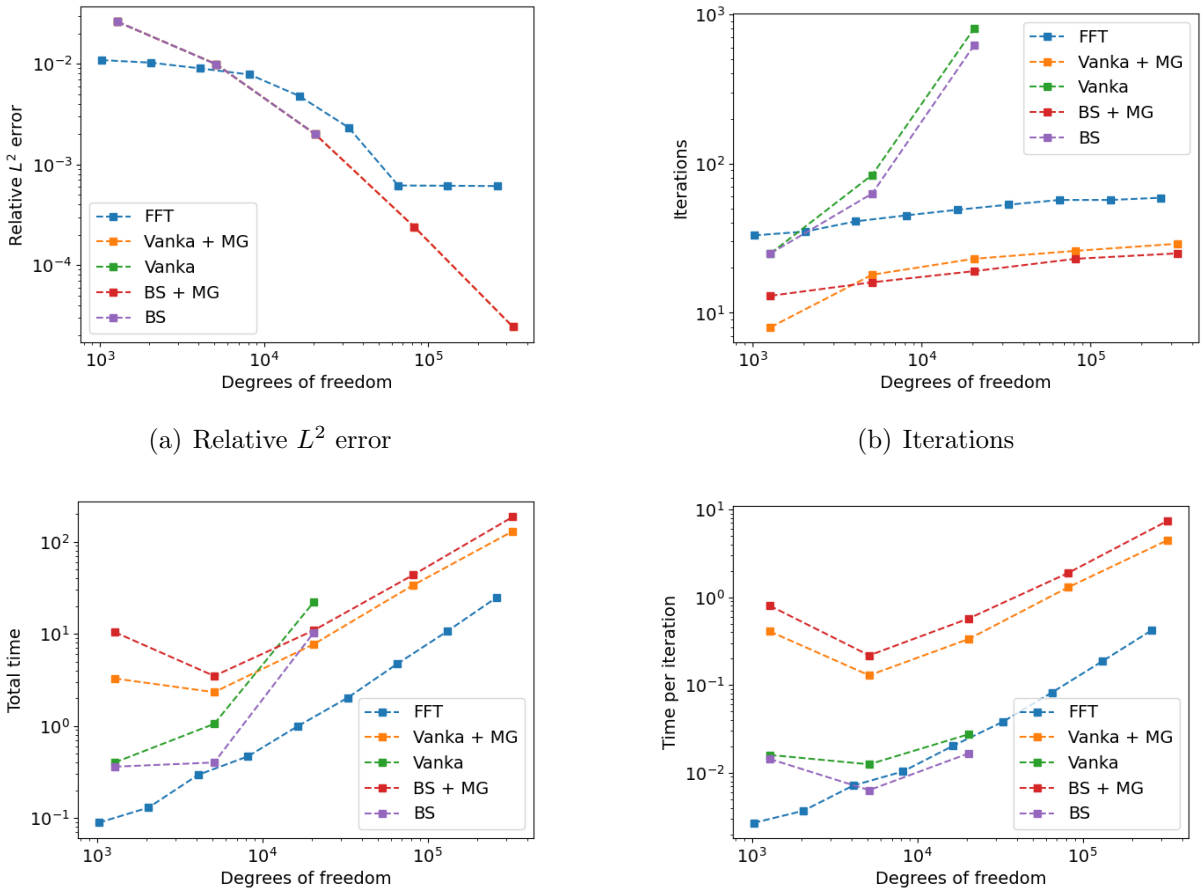


(a) FFT (b) BS + multigrid (c) Vanka + multigrid

Figure 6: Local velocity field for the parallel channel toy problem. Parameters: $a = 0.1, b = 0.1, \text{Lx} = a + b, \text{Ly} = 1, \mu = \mu_e = 1, G = 1, k_s \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}\}, \varepsilon = 10^{-6}, \text{RTOL} = 10^{-6}$(preconditioned).

For the first experiment we consider $n_x = n_y \in \{16, 32, 64, 128, 256\}$. We run both solvers to convergence, and plot the local velocity field against the analytic solution. The results are given in Figure 6. We see that both solvers are converging to the correct solution. Moreover, the solution is continuous at the interface between the fluid and solid region. This is a promising result however the numerical instabilities observed in Figure 5 make it clear that further testing for other toy problems is still required.

For the second experiment, we produce a preliminary comparison between the FEM and FFT solvers. The aim is to compare their performance on a problem for which they have both been validated, and to compare the stopping criteria. We set $a = b = 0.1$ mm, $\mu = \mu_e = 1$ MPa$\cdot$s, and $= 1$ MPa. The FFT tolerance is $\varepsilon = 10^{-6}$, and the FEM tolerance is RTOL $= 10^{-6}$ for the preconditioned residual (26). Due to technical limitations explained in Appendix D, for a given choice of n_y, FFT solves for n_y DoFs, whereas the FEM solves for $5n_y^2$ DoFs. To make the comparison fair, we consider how quantities scale with DoFs instead of n_y. We measure iterations, total time, time per iteration, and relative L^2 error with the analytical solution. Results are given in Figure 7.



(a) Relative L^2 error

(b) Iterations

(c) Total time

(d) Time per iteration

Figure 7: Comparison between FFT and FEM solvers on the parallel channel toy problem. Parameters: $a = 0.1, b = 0.1, \mathrm{Lx} = a + b, \mathrm{Ly} = 1, \mu = \mu_e = 1, G = 1, k_s = 10^{-6}, \varepsilon = 10^{-6}, \mathrm{RTOL} = 10^{-6}$(preconditioned).

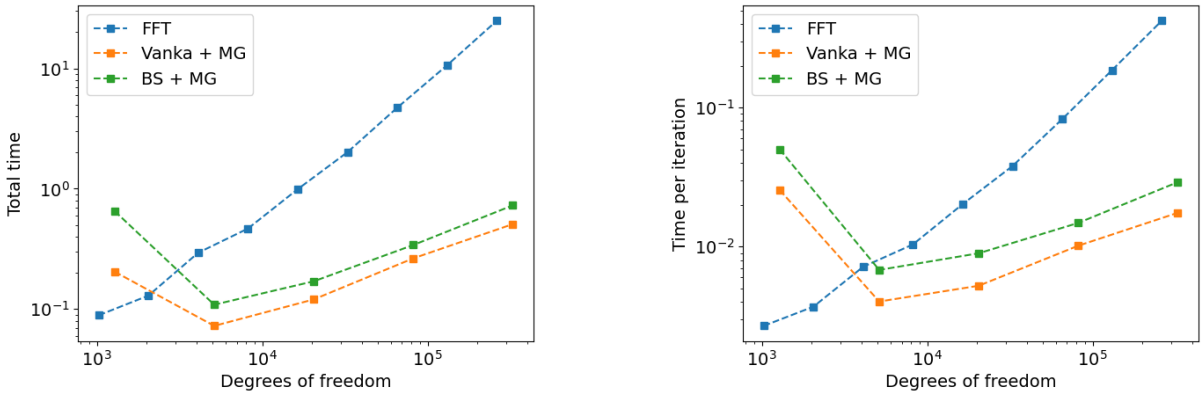
Observe, that for $\varepsilon = \mathrm{RTOL}$, the FEM solver leads to a greater reduction in the relative L^2 error, and that the FFT solver plateaus. This suggests that a residual based stopping criterion is a better choice. This is a delicate point that we discuss further in Section 5.1. As predicted by the theory, Vanka and BS as standalone preconditioners do not perform well. In particular, the number of iterations to converge quickly blows up and, in fact, both fail to converge when the number of DoFs reaches approximately $80,000$. Due to this, it is reasonable to no longer consider them as standalone methods going forward. When coupled with multigrid, both smoothers converge for all the experiments. Morevoer, the number of iterations almost stay constant as the problem size increases. Note that the theory predicts they should remain constant, and it is believed that this behaviour can be achieved by finetuning of the multigrid cycles.

5 Discussion

5.1 Future work

We now discuss the different avenues that can be explored in the future.

1. Figure 7 suggests that the FFT solver is quicker. However, with the current stopping criteria it achieves a worse relative L^2 error than the seemingly slower FEM solvers. In fact, this is not a particularly fair comparison due to the technical limitations (Appendix D) meaning that the FEM solver has to be setup on a 2D grid. We hypothesise that if the FEM solver can be set up on a $n_x \times 1$ mesh, then it will outperform the FFT solver. It is reasonable to assume that on this mesh the total time will be reduced by a factor of n_y, and the number of iterations will remain the same. Under these assumptions we would attain the results given in Figure 8.



(a) Relative L^2 error

(b) Iterations

Figure 8: Predicted results for the FEM solver (vs the FFT solver) for the parallel channel geometry if it were run on a $n_x \times 1$ mesh. Parameters: $a = 0.1, b = 0.1, \mathrm{Lx} = a + b, \mathrm{Ly} = 1, \mu = \mu_e = 1, G = 1, k_s = 10^{-6}, \varepsilon = 10^{-6}, \mathrm{RTOL} = 10^{-6}$(preconditioned).

   These results suggest that the FEM solver would outperform the FFT solver giving us a first insight into which has the better performance. However, these are only estimated results, and we cannot make informed conclusions without doing these experiments. Unfortunately, this is currently not possible due to the discussed technical limitations.

2. We propose a selection of further numerical experiments:

   (a) Repeat the experiment for Figure 4 with a tight tolerance (for example 10^{-14}) to see if the solutions get closer together.

   (b) Repeat the experiment for Figure 5 but cut out a small region around the interface. One should then check if the solutions coincide away from the region of numerical instability.

(c) Repeat the experiments of Section 4.5 with different tolerances. In particular, consider Figure 7(a), and look at how these results change.

(d) Repeat the experiments for Figure 6 but with meshes that do not coincide with the fluid-solid interface, and check if the numerical solution exhibits instability.

3. As discussed in Section 4.5, a consistent stopping criterion is necessary before a fair (and useful) performance comparison between the FFT and FEM solvers can be made. It is worth noting that the author of [7] has recently updated their FFT implementation to use a residual based solver. The first step would be to repeat the experiments of Section 4.5, and re-analyse the results. Additionally, a decision must be made on whether to use the preconditioned or unpreconditioned residual for the FEM solver. We propose to first perform further numerical experiments comparing both choices for different tolerances, but if the results do not shed much light on this issue a deeper analysis of the underlying mathematics may be necessary.

4. In general, the literature surrounding Stokes-type problems focuses on the standard Stokes equation or a generalised Stokes equation (16). It is important to note that the Stokes-Brinkman problem contains the discontinuous Brinkman term $\beta(x)$, which can take large values depending on the local permeability k_s. Our initial implementations were based off of convergence guarantees given in the literature for problems of the former type [12, 4]. It is not immediately obvious whether such claims hold in our case. Moreover, in Figure 5, we observed numerical instabilities at the interface containing the discontinuity. This is a clear issue that needs resolving before moving forward. Possible approaches include

   (a) Further numerical experiments

   (b) A different finite element choice. For example, would discontinuous Galerkin be a better choice?

   (c) A different solver. One such example is the GenEO framework [16].

5. A key reason for designing a FEM solver is its superior parallel scalability compared to the FFT solver. Firedrake is now available on the Nimbus high-performance computing (HPC) system at the University of Bath. This will allow for numerical experiments to verify the expected scalability for the Stokes-Brinkman problem, and to compare this with the corresponding results given in [7] for the FFT solver.

5.2 Connections with the other strands

The design of a FEM solver is just one part of an interdisciplinary research project looking at multiscale flow in porous media. The general idea is as follows:

We wish to create a large training dataset of bundle geometries and macroscopic permeabilities, $(\beta(x), \mathbf{K})$, to be used by the machine learning model. This would be done by generating physically viable images. After post-processing these would be passed through a numerical solver to obtain the corresponding macroscopic permeability. It is expected that this will be a computationally expensive task hence we wish to use the most efficient and accurate solver

at our disposal. The machine learning model can be trained on this dataset, and then used in an industrial context to make predictions of macroscopic permeability. Ultimately, this will allow for the optimisation and streamlining of the manufacturing process of composite materials formed via resin injection. The connections between the strands are summarised in Figure 9.

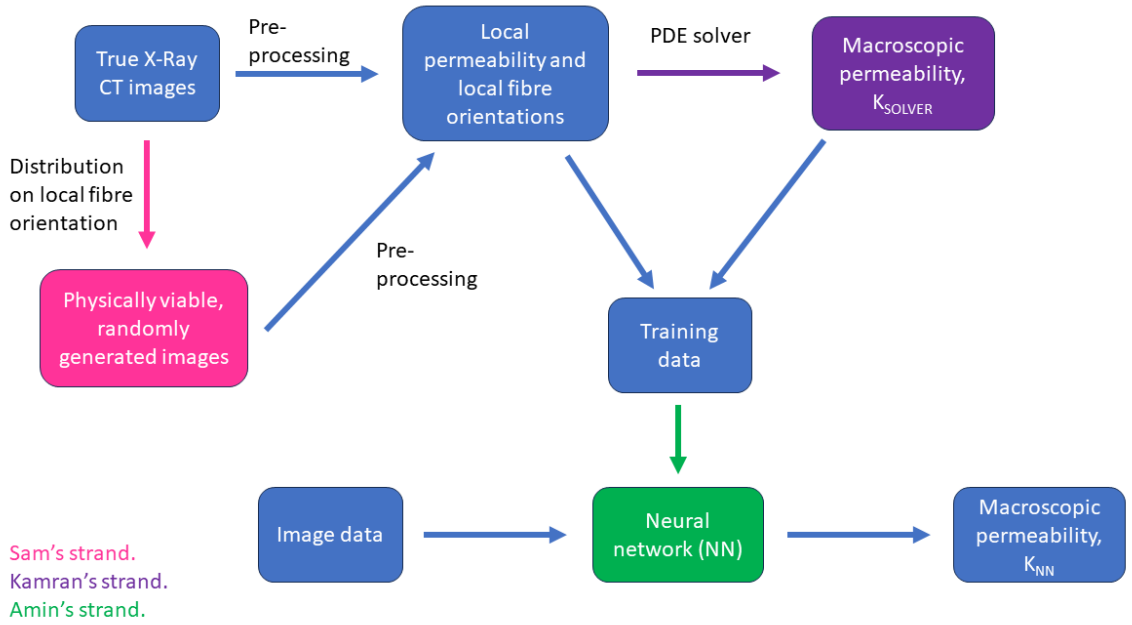


Figure 9: Connections between the different strands of the research project.

With reference to the purple strand in Figure 9, we have made progress towards the design of a FEM solver but, as discussed, we have not been able to produce a fair performance comparison with the FFT solvers. This means we are still not in a position to make an informed decision on which solver is the best to use going forward in the project which ultimately limits how well we can integrate our current work into the wider project. Despite this, the project is headed in a clear direction, and we have an idea of the next steps:

1. For the statistics strand, it is not currently know what format the randomly generated images will take after post-processing. Once this is known the FEM solver will need to be capable of translating the bundle geometries defined via these images onto a mesh and initialising the Brinkman term $\beta(x)$.

2. For the machine learning strand, experiments have been performed using Fourier Neural Operator (FNO) architecture. Currently it is being trained using macroscopic permeabilities computed using the FFT solver. Even without a performance comparison, it would be instructive to train the FNO method using the FEM solvers outputs. Perhaps this will give further insight into which method may be the better choice, or at the very least it may open up further directions for exploration.

References

[1] Michele Benzi and Gene H Golub. A preconditioner for generalized saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):20–41, 2004.

[2] Michele Benzi, Gene H Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.

[3] M Bodaghi, SV Lomov, P Simacek, NC Correia, and SG Advani. On the variability of permeability induced by reinforcement distortions and dual scale flow in liquid composite moulding: A review. *Composites Part A: Applied Science and Manufacturing*, 120:188–210, 2019.

[4] Dietrich Braess and Regina Sarazin. An efficient smoother for the stokes problem. *Applied Numerical Mathematics*, 23(1):3–19, 1997.

[5] Franco Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Publications des séminaires de mathématiques et informatique de Rennes*, (S4):1–26, 1974.

[6] Hendrik C Brinkman. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Flow, Turbulence and Combustion*, 1(1):27–34, 1949.

[7] Yang Chen. High-performance computational homogenization of stokes–brinkman flow with an anderson-accelerated fft method. *International Journal for Numerical Methods in Fluids*, 95(9):1441–1467, 2023.

[8] Howard C Elman. Preconditioners for saddle point problems arising in computational fluid dynamics. *Applied Numerical Mathematics*, 43(1-2):75–89, 2002.

[9] Patrick Farrell. Finite element methods for pdes. *Lecture notes, Unversity of Oxford*, 2021. URL: https://people.maths.ox.ac.uk/farrellp/femvideos/notes.pdf.

[10] David A. Ham, Paul H. J. Kelly, Lawrence Mitchell, Colin J. Cotter, Robert C. Kirby, Koki Sagiyama, Nacime Bouziani, Sophia Vorderwuelbecke, Thomas J. Gregory, Jack Betteridge, Daniel R. Shapero, Reuben W. Nixon-Hill, Connor J. Ward, Patrick E. Farrell, Pablo D. Brubeck, India Marsden, Thomas H. Gibson, Miklós Homolya, Tianjiao Sun, Andrew T. T. McRae, Fabio Luporini, Alastair Gregory, Michael Lange, Simon W. Funke, Florian Rathgeber, Gheorghe-Teodor Bercea, and Graham R. Markall. *Firedrake User Manual*. Imperial College London and University of Oxford and Baylor University and University of Washington, first edition edition, 5 2023.

[11] Olga Aleksandrovna Ladyzhenskaya. The mathematical theory of viscous incompressible flow. *Gordon & Breach*, 1969.

[12] Maxim Larin and Arnold Reusken. A comparative study of efficient iterative solvers for generalized stokes equations. *Numerical linear algebra with applications*, 15(1):13–34, 2008.

[13] Arnold Reusken. *Introduction to multigrid methods for elliptic boundary value problems*. Inst. für Geometrie und Praktische Mathematik, 2008.

[14] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[15] Seyed Saberi, Günther Meschke, and Andreas Vogel. A restricted additive vanka smoother for geometric multigrid. *Journal of Computational Physics*, 459:111123, 2022.

[16] Nicole Spillane. An abstract theory of domain decomposition methods with coarse spaces of the geneo family. *arXiv preprint arXiv:2104.00280*, 2021.

[17] Cedric Taylor and Paul Hood. A numerical solution of the navier-stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73–100, 1973.

[18] Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

A Well-posedness of the Stokes-Brinkman equations

Consider Brezzi's theorem for infinite-dimensional saddle point systems.

Theorem A.1 (Brezzi [5])**.** *Let V, Q be real Hilbert spaces. Let a be a bilinear form on $V \times V$. Let b be a bilinear form on $V \times Q$. Let $F \in V^*$ and $G \in Q^*$. Consider the following infinite-dimensional saddle-point problem:*

Find $(u, p) \in V \times Q$ such that

$$\begin{cases} a(u, v) + b(v, p) = F(v), \\ b(u, q) = G(q), \end{cases} \tag{27}$$

for all $(v, q) \in V \times Q$.

Assume that

1. *a and b are bounded.*
2. *The auxillary problem*

$$\text{Find} \quad u \in Ker(b) \quad \text{such that} \quad a(u, v) = F(v) \quad \forall v \in Ker(b)$$

   is well-posed.

3. *b satisfies an inf-sup condition, namely, there exists $\gamma \in \mathbb{R}$ such that*

$$0 < \gamma \leq \inf_{q \neq 0} \sup_{v \neq 0} \frac{b(v, q)}{\|v\|_V \|q\|_Q}$$

Then there exists a unique solution $(u, p) \in V \times Q$ to (27).

Using Brezzi's theorem, one can show that the saddle point system given by (10) has a unique solution. In particular, condition 1 follows from the Cauchy-Schwarz inequality, condition 2 follows from an application of Lax-Milgram, and condition 3 was proved by Ladyzhenskaya [11].

B The Stokes-Brinkman saddle point system arises from a constrained minimisation problem

Let $u \in H_0^1(\Omega;\mathbb{R}^3)$. Consider the constrained minimisation problem

$$\text{Minimise} \quad J(u) = \frac{1}{2}a_\beta(u,u) - L(u) \quad \text{subject to} \quad \nabla \cdot u = 0.$$

Introduce a Lagrange multiplier $p \in L_0^2(\Omega;\mathbb{R})$. We interpret this as pressure. The Lagrangian is given by

$$L(u,p) = \frac{1}{2}a_\beta(u,u) - L(u) + \int_\Omega p\nabla \cdot u \,\mathrm{d}x.$$

The optimality conditions are

$$\begin{aligned} L_u(u,p;v) &= \lim_{\varepsilon\to 0^+} \frac{L(u+\varepsilon v,p) - L(u,p)}{\varepsilon} = 0 \\ L_p(u,p;q) &= \lim_{\varepsilon\to 0^+} \frac{L(u,p+\varepsilon q) - L(u,p)}{\varepsilon} = 0. \end{aligned}$$

Computing these quantities gives the Stokes-Brinkman saddle-point system

$$\begin{cases} a_\beta(u,v) + b(v,p) = L(v), \\ b(u,q) = 0. \end{cases}$$

C Inf-Sup condition for the finite element

Consider the following theorem reproduced from [9].

Theorem C.1 ((14.7.1) [9]). *Let V_h be a closed subspace of V and Q_h be a closed subspace of Q. Consider the Galerkin approximation to the (27):*

$$\begin{cases} a(u_h,v_h) + b(v_h,p_h) = F(v_h), \\ b(u_h,q_h) = G(q_h), \end{cases} \tag{28}$$

Define

$$K_h = \{v_h \in V_h \mid b(v_h,q_h) = 0 \text{ for all } q_h \in Q_h\}. \tag{29}$$

In addition to the assumptions of Theorem A.1, suppose that

1. *The problem*

$$\text{Find} \quad u_h \in K_h \quad \text{such that} \quad a(u_h,v_h) = F(v_h) \quad \forall v_h \in K_h$$

   is well-posed.

2. *b satisfies an inf-sup condition, namely, there exists $\tilde{\gamma} \in \mathbb{R}$ such that*

$$0 < \tilde{\gamma} \le \inf_{q_h \neq 0} \sup_{v_h \neq 0} \frac{b(v_h,q_h)}{\|v_h\|_V \|q_h\|_Q}$$

Then (28) *is well-posed.*

In particular, any choice of finite element discretisation must satisfy the above inf-sup theorem in order for the discrete system to be well-posed.

D Firedrake implementation of the parallel channel toy problem

The FFT implementation of the parallel channel toy problem uses a grid of size $1 \times n_y \times 1$. Unfortunately, using the same method for the Firedrake implementation is not possible due to technical limitations. Therefore, we solve the problem on a grid of size $n_x \times n_y$. Moreover, to ensure compatability with standard Firedrake meshes, we rotate and shift the problem so that we solve on the interval $[0, a+b]$ instead of $[-b, a]$, and velocity flow is in the positive y-direction instead of the positive x-direction. This means that for each column of cells (i.e. a fixed interval of the x-axis), the solution is identical across each of these cells (since the problem is inherently 1D so the y-component of velocity remains fixed). Crucially, this dramatically increases the problem size we are solving. Therefore, in an attempt to make a more fair comparison with the FFT solver, we scaled any measured quantities with respect to the number of DoFs.