Technical Report

TR-2014-001

Parameter Estimates for the Relaxed Dimensional Factorization Preconditioner and Application to Hemodynamics

by

Michele Benzi, Simone Deparis, Gwenol Grandperrin, Alfio Quarteroni

MATHEMATICS AND COMPUTER SCIENCE

EMORY UNIVERSITY

PARAMETER ESTIMATES FOR THE RELAXED DIMENSIONAL FACTORIZATION PRECONDITIONER AND APPLICATION TO HEMODYNAMICS*

MICHELE BENZI†, SIMONE DEPARIS‡, GWENOL GRANDPERRIN‡ §, AND ALFIO QUARTERONI‡

Abstract. We present new results on the Relaxed Dimensional Factorization (RDF) preconditioner for solving saddle point problems, first introduced in [5]. This method contains a parameter $\alpha > 0$, to be chosen by the user. Previous works provided an estimate of α in the 2D case using Local Fourier Analysis. Novel algebraic estimation techniques for finding a suitable value of the RDF parameter in both the 2D and the 3D case with arbitrary geometries are proposed. These techniques are tested on a variety of discrete saddle point problems arising from the approximation of the Navier–Stokes equations using a Marker-and-Cell scheme and a finite element one. We also show results for a large-scale problem relevant for hemodynamics simulation that we solve in parallel using up to 8196 cores.

Key words. scalable parallel preconditioners, finite element method, high performance computing, Navier–Stokes equations, hemodynamics applications, dimensional splitting preconditioner, relaxed dimensional factorization preconditioner

AMS subject classifications. 65M60, 65F08, 65Y05, 76Z05

1. Introduction.

In the last decade, many techniques have been proposed for preconditioning linear systems of equations in saddle point form, like those arising from the discretization of the Navier–Stokes equations. Among the most successful, we mention the Pressure Convection–Diffusion (PCD) preconditioner [32, 26, 19], which makes the GMRES iterations converge with a rate independent of the mesh, at least when the viscosity is sufficiently large.

An alternative to PCD is represented by the so-called Least–Squares Commutator (LSC) preconditioner [15, 16, 19], which can be built automatically, albeit with higher computational cost. The convergence rate of LSC is independent of the mesh size and mildly dependent on the viscosity. A version for stabilized finite element discretizations has been introduced in [16].

In [6], an Augmented Lagrangian (AL) preconditioner is introduced starting from the augmented Lagrangian formulation of the underlying saddle point problem. The corresponding convergence rate is independent of the mesh size [7], mildly dependent on the viscosity, and robust when anisotropic meshes are considered.

In [8], the Modified Augmented Lagrangian (MAL) preconditioner is introduced to make the action of the AL method cheaper and easier to implement, particularly on unstructured grids. When using the MAL preconditioner, the rate of convergence shows a mild dependence on the viscosity and is independent of the mesh size [7]. Like the AL preconditioner, MAL is robust when anisotropic meshes are used [10].

More recently, the so–called Relaxed Dimensional Factorization (RDF) preconditioner has been introduced in [5] as an improvement to the Dimensional Splitting (DS) preconditioner [4]. Although this method was mainly intended to solve steady

* This work is supported by the HP2C project lead by the Swiss National Supercomputer Centre (CSCS) in Lugano, Switzerland.

†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA.

‡MATHICSE - Chair of Modelling and Scientific Computing (CMCS), EPFL, CH - 1015 Lausanne, Switzerland.

§Corresponding author. E-mail address: gwenol.grandperrin@epfl.ch

Stokes and Oseen problems, it can also handle the unsteady case. Experimental results indicate independence of its convergence rate of the mesh size and a mild dependence on the viscosity. This preconditioner relies on a parameter that, in simple 2D geometries, can be estimated using Local Fourier Analysis (LFA). A comparison of the performance of the RDF preconditioner and other preconditioners such as PCD or LSC can be found in [5]. Those results showed that RDF can be an attractive alternative to PCD and LSC, especially for low values of the viscosity and anisotropic meshes. In this paper, we develop a new approach to estimating the RDF parameter for both 2D and 3D problems in arbitrary geometries. We test our technique on a few numerical benchmarks, as well as on a large 3D problem originating from the simulation of blood flow in large arteries. In particular, we investigate the performance of the preconditioner in terms of strong scalability using up to 8192 cores.

The remainder of this paper is organized as follows. The mathematical model and the strategy to solve saddle point problems using the RDF preconditioner are presented in Sections 2 and 3, respectively. In Section 4, we propose original techniques to estimate the parameter of the RDF preconditioner. Then in Sections 5, we test the RDF preconditioner on simple 2D cases, on a 3D driven cavity problem, and on a benchmark relevant to hemodynamic simulations using up to 8192 cores. Finally, some conclusions are drawn in Section 6.

2. Mathematical model. We consider an incompressible Newtonian fluid with constant density ρ and viscosity μ in a bounded domain Ω of $\mathbb{R}^d$ $(d = 2, 3)$. The Navier–Stokes equations read

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega, \quad t > t_0, \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \quad t > t_0, \tag{2.2}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity vector field, $p = (\mathbf{x}, t)$ the pressure scalar field, $\mathbf{f}_{ext}$ the external force per mass unit, and $\nu = \frac{\mu}{\rho}$ the kinematic viscosity. These partial differential equations are complemented with an initial solution and boundary conditions:

$$\mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x}) \qquad \forall \mathbf{x} \in \Omega,$$

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{g}_D(\mathbf{x}, t) \qquad \forall \mathbf{x} \in \Gamma_D, \quad t > t_0, \tag{2.3}$$

$$\left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n}\right)(\mathbf{x}, t) = \mathbf{g}_N(\mathbf{x}, t) \qquad \forall \mathbf{x} \in \Gamma_N, \quad t > t_0, \tag{2.4}$$

where Γ_D and Γ_N refer to the Dirichlet and Neumann part of the boundary, respectively, $\Gamma_D \cup \Gamma_N = \partial\Omega$, $\Gamma_D \cap \Gamma_N = \emptyset$, and $\mathbf{u}_0$, $\mathbf{g}_D$, and $\mathbf{g}_N$ are assigned functions.

We consider a fully implicit scheme, see, e.g. [18, 28], to discretize in time the equations:

$$\begin{aligned} \frac{1}{\Delta t}\mathbf{u}^{n+1} - \nu \Delta \mathbf{u}^{n+1} + \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1} + \nabla p^{n+1} &= \mathbf{f}^{n+1} + \frac{1}{\Delta t}\mathbf{u}^n, \\ \nabla \cdot \mathbf{u}^{n+1} &= 0. \end{aligned}$$

Picard iteration is used to solve the nonlinearity; see, e.g., [18]. The weak formulation of the resulting equations is discretized in space using the Finite Element Method

(FEM) yielding at each timestep a linear system of the form $\mathcal{A}_{NS}\mathbf{x} = \mathbf{b}$ with

$$\mathcal{A}_{NS} = \begin{pmatrix} F_1 & 0 & 0 & B_1^T \\ 0 & F_2 & 0 & B_2^T \\ 0 & 0 & F_3 & B_3^T \\ -B_1 & -B_2 & -B_3 & 0 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} U_1^{n+1} \\ U_2^{n+1} \\ U_3^{n+1} \\ P^{n+1} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ 0 \end{pmatrix}. \tag{2.5}$$

Here F_i $(i = 1, 2, 3)$ are discrete operators associated with the Laplace operator, the (linearized) convection operator and one part of the time derivative for the different components of the velocity, and B_i, B_i^T $(i = 1, 2, 3)$ are discretized versions of the first partial derivatives (and their adjoints) for the different components of the velocity, respectively. Note that we are using $-B_i$ instead of B_i, $i = 1, 2, 3$; this choice guarantees that all the eigenvalues of $\mathcal{A}_{NS}$ lie in the right half-plane [3, 29]. Finally, G_i $(i = 1, 2, 3)$ contain the discretized source forces and the second part of the time derivatives which depend on $\mathbf{u}^n$. Both the matrix and the right hand side of the system are modified to take the boundary conditions into account.

3. Relaxed Dimensional Factorization (RDF) preconditioner.

The Relaxed Dimensional Factorization (RDF) preconditioner was introduced in [5] as an improved version of the Dimensional Splitting (DS) preconditioner [4]. It was originally designed for steady Oseen problems with small viscosity ν, possibly using anisotropic meshes. On such problems, most of the preconditioners fail to converge at all or converge very slowly as showed in [5]. The RDF preconditioner exploits the structure of the matrix of the linear system (2.5) and reads:

$$\mathcal{P}_{RDF} = \frac{1}{\alpha^2} \begin{pmatrix} F_1 & 0 & 0 & B_1^T \\ 0 & \alpha I & 0 & 0 \\ 0 & 0 & \alpha I & 0 \\ -B_1 & 0 & 0 & \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & 0 & 0 & 0 \\ 0 & F_2 & 0 & B_2^T \\ 0 & 0 & \alpha I & 0 \\ 0 & -B_2 & 0 & \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & 0 & 0 & 0 \\ 0 & \alpha I & 0 & 0 \\ 0 & 0 & F_3 & B_3^T \\ 0 & 0 & -B_3 & \alpha I \end{pmatrix},$$

where $\alpha > 0$ is a parameter to be chosen. By expanding the product, we can observe that $\mathcal{P}_{RDF}$ coincides with $\mathcal{A}_{NS}$ up to additional terms (which depend on α) showing up on the upper triangular part:

$$\mathcal{P_{RDF}} = \begin{pmatrix} F_1 & -\frac{1}{\alpha}B_1^T B_2 & -\frac{1}{\alpha}B_1^T B_3 & B_1^T \\ 0 & F_2 & -\frac{1}{\alpha}B_2^T B_3 & B_2^T \\ 0 & 0 & F_3 & B_3^T \\ -B_1 & -B_2 & -B_3 & \alpha I \end{pmatrix}.$$

Numerical experiments conducted in [5] on 2D problems using $\mathbb{Q}_2-\mathbb{Q}_1$ and $\mathbb{Q}_2-\mathbb{P}_1$ FE on a structured grid show that the RDF preconditioner leads to better results than the DS preconditioner for Stokes problems, and generalized Oseen problems, when the dynamic viscosity ν is relatively small. For unsteady problems, the convergence rate of RDF preconditioned iterations is independent of h and ν. Experiments with inexact variants of the RDF preconditioner also indicate h-independent convergence rates. However, a moderate dependency on ν is noticed. In [5], it is shown that RDF is generally more robust and effective than PCD, in particular for small ν.

4. Estimates for the parameter of the RDF preconditioner.

When the preconditioner RDF is used in combination with GMRES, the rate of convergence is not overly sensitive with respect to the parameter α, as pointed out in [5]; for a value of α close enough to the optimal value, the convergence rate remains acceptable. The optimal value α_{opt} of is the one that minimizes the number of iterations and was determined in [5] by LFA for 2D problems; this technique is recalled in Section 4.1 of this paper. For 3D problems, empirical search was used to find a good estimate for α_{opt}. In this section, we introduce two new search strategies for α_{opt} for both 2D and 3D problems. The first one is based on the minimization of a suitable norm of the difference between $\mathcal{P}_{RDF}$ and $\mathcal{A}_{NS}$, and is described in Section 4.2. The second one is based on an analysis of the trace of the preconditioned matrix, and is presented in Section 4.3.

4.1. Local Fourier analysis for the determination of α_{opt}. LFA is a classical tool for parameter estimation in iterative methods, see, e.g., [16, 9, 33]. In particular, estimates for the α parameter of the RDF preconditioner in the context of a fluid cavity problem have been found using this technique in [5] in the 2D case. This type of analysis is based on the following assumptions:

- the viscosity ν and $\boldsymbol{\beta}$ in the convective term $\boldsymbol{\beta} \cdot \nabla$ are constant;
- periodic boundary conditions are imposed;
- centered finite differences are used to discretize the problem;
- the discrete problem is extended to an infinite uniform structured grid;
- F_1, F_2, B_1, B_2, are all square of the same order and commute.

The first assumption may seem quite restrictive. However, we note that when ν is large the diffusion term $-\nu\Delta$ dominates the convection term $\mathbf{v} \cdot \nabla$ such that we can assume that $\boldsymbol{\beta} = \mathbf{v}$ has almost no impact. The Laplacian term is also dominating the convection term for h small enough. Indeed, for finite difference discretization the entries of the stiffness matrix associated to the Laplacian scale as $\mathcal{O}(h^{-2})$ and the entries of the matrix associated to the convection as $\mathcal{O}(h^{-1})$. Under these assumptions, F_1, F_2, are replaced by discrete unsteady convection-diffusion operators of the form

$$\frac{1}{\Delta t} I + \nu L_x + N_x, \quad \frac{1}{\Delta t} I + \nu L_y + N_y,$$

where I is the identity operator, L_x, and L_y are discrete one-dimensional Laplacians obtained by centered differences and N_x, and N_y are the one-dimensional convection operators obtained by centered differences in the x, y, and z directions respectively. B_1, B_2 are replaced with the one-dimensional differentiation operators S_x and S_y obtained by one-sided differences in x, and y directions, respectively. Finally, to mimic the scaling of the entries of the finite element mass matrix, we multiply these operators by h^2 in 2D. For the sake of comparison, we assume, as in [5], that $\Delta t \approx h$. The symbols corresponding to the operators are the following

$$\begin{aligned}
\frac{1}{\Delta t} I + \nu L_x + N_x \quad &: \quad \nu(1 - e^{i2\pi h\theta_x} - e^{-i2\pi h\theta_x}) + h(1 + \beta_{\max,x}(e^{i2\pi h\theta_x} - e^{-2p\pi h\theta_x})) \\
\frac{1}{\Delta t} I + \nu L_y + N_y : \quad & \nu(1 - e^{i2\pi h\theta_y} - e^{-i2\pi h\theta_y}) + h(1 + \beta_{\max,y}(e^{i2\pi h\theta_y} - e^{-2p\pi h\theta_y})) \\
S_x : \quad & h(1 - e^{-i2\pi h\theta_x}) \\
S_y : \quad & h(1 - e^{-i2\pi h\theta_y}),
\end{aligned}$$

where $\beta_{\max,x}$ and $\beta_{\max,y}$ denote the maximum of the x and y components of the convective field $\boldsymbol{\beta}$. For arbitrary meshes, we choose h to be the average of the diameter of the tetrahedra.

In this case, it has been shown in [5] that a good estimate for α_{opt} is given by

$$\alpha_* = \arg\min_{\alpha} |\mu(\alpha) - 1|, \tag{4.1}$$

where $\mu(\alpha)$ is defined as follows:

$$\mu(\alpha) = \frac{1}{\alpha}(s_1 + s_2) - \frac{2}{\alpha^2} s_1 s_2. \tag{4.2}$$

Here s_1 and s_2 are the eigenvalues of $S_1 = B_1(F_1 + \frac{1}{\alpha}B_1^T B_1)^{-1} B_1^T$ and $S_2 = B_2(F_2 + \frac{1}{\alpha}B_2^T B_2)^{-1} B_2^T$, respectively. We observe that the quantity $\mu(\alpha)$ is to be minimized not only with respect to α but also with respect to all the frequencies θ_x, θ_y that appear in the expression for the eigenvalues of S_1 and S_2; we refer to [5] for details. In practice, the computational cost for computing α_* for a given value of h and ν is small compared to the solution of the saddle point problem, and the computation can be performed off-line for a broad range of values of h and ν.

Since in this paper we are using a FE method to discretize the equations, the entries of the mass matrix M_u scale as $\mathcal{O}(h^{-2})$. Therefore, we expect small values for the parameter α.

4.2. Estimation using error minimization. As previously noted, the difference between $\mathcal{A}_{NS}$ and $\mathcal{P}_{RDF}$ is given by

$$\mathcal{R}_\alpha = \mathcal{P_{RDF}} - \mathcal{A_{NS}} = \begin{pmatrix} 0 & -\frac{1}{\alpha}B_1^T B_2 & -\frac{1}{\alpha}B_1^T B_3 & 0 \\ 0 & 0 & -\frac{1}{\alpha}B_2^T B_3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha I \end{pmatrix}. \tag{4.3}$$

We want to choose $\alpha > 0$ by minimizing the function $\Psi(\alpha) = \|\mathcal{R}_\alpha\|$, where $\|\cdot\|$ represents a suitable norm to be chosen. We refer to this approach as *global error minimization* (GEM) method. We consider three versions: $\Psi_F(\alpha)$, $\Psi_1(\alpha)$, and $\Psi_\infty(\alpha)$ corresponding to the Frobenius norm, the 1-norm, and the infinity norm, respectively:

$$\begin{aligned}
\Psi_F(\alpha) &= \sqrt{\alpha^2 N_p + \frac{1}{\alpha^2}\left(\left\|B_1^T B_2\right\|_F^2 + \left\|B_1^T B_3\right\|_F^2 + \left\|B_2^T B_3\right\|_F^2\right)}, \\
\Psi_1(\alpha) &= \frac{1}{\alpha}\max\left(\alpha, \left\|B_1^T B_2\right\|_1, \left\|\begin{pmatrix} B_1^T B_3 \\ B_2^T B_3\end{pmatrix}\right\|_1\right), \\
\Psi_\infty(\alpha) &= \frac{1}{\alpha}\max\left(\alpha, \left\|\begin{pmatrix} B_1^T B_2 & B_1^T B_3\end{pmatrix}\right\|_\infty, \left\|B_2^T B_3\right\|_\infty\right).
\end{aligned}$$

To avoid explicitly forming $B_1^T B_2$, $B_1^T B_3$, and $B_2^T B_3$, we consider instead

$$\begin{aligned}
\hat{\Psi}_F(\alpha) &= \sqrt{\alpha^2 N_p + \frac{1}{\alpha^2}\left(\left\|B_1^T\right\|_F^2 \|B_2\|_F^2 + \left\|B_1^T\right\|_F^2 \|B_3\|_F^2 + \left\|B_2^T\right\|_F^2 \|B_3\|_F^2\right)}, \\
\hat{\Psi}_1(\alpha) &= \frac{1}{\alpha}\max\left(\alpha, \left\|B_1^T\right\|_1 \|B_2\|_1, \left(\left\|B_1^T\right\|_1 + \left\|B_2^T\right\|_1\right)\|B_3\|_1\right), \\
\hat{\Psi}_\infty(\alpha) &= \frac{1}{\alpha}\max\left(\alpha, \left\|B_1^T\right\|_\infty \left(\|B_2\|_\infty + \|B_3\|_\infty\right), \left\|B_2^T\right\|_\infty \|B_3\|_\infty\right).
\end{aligned}$$

Since the three norms are submultiplicative, $\Psi(\alpha) \leq \hat{\Psi}(\alpha)$ for each one of the norms. The minimization of these functions leads to

$$\alpha_{GEM,F} = \left(\frac{\left\|B_1^T\right\|_F^2 \|B_2\|_F^2 + \left\|B_1^T\right\|_F^2 \|B_3\|_F^2 + \left\|B_2^T\right\|_F^2 \|B_3\|_F^2}{N_p} \right)^{\frac{1}{4}},$$
$$\alpha_{GEM,1} \geq \max\left(\left\|B_1^T\right\|_1 \|B_2\|_1, \left(\left\|B_1^T\right\|_1 + \left\|B_2^T\right\|_1 \right) \|B_3\|_1 \right),$$
$$\alpha_{GEM,\infty} \geq \max\left(\left\|B_1^T\right\|_\infty \left(\|B_2\|_\infty + \|B_3\|_\infty \right), \left\|B_2^T\right\|_\infty \|B_3\|_\infty \right).$$

The estimates $\alpha_{GEM,1}$ and $\alpha_{GEM,\infty}$ are not very informative since they hardly provide a value close to α_{opt}. In particular, the numerical results presented in [5] show examples for which values of α away from α_{opt} deteriorate the convergence rate of preconditioned GMRES iterations. Therefore, for these two estimates we only take the lower bound. We now consider a slightly different approach from GEM; we try to minimize the error in R_α by minimizing the function

$$\Phi(\alpha) = \alpha \|I\| + \frac{1}{\alpha} \left(\left\|B_1^T B_2\right\| + \left\|B_1^T B_3\right\| + \left\|B_2^T B_3\right\| \right),$$

where $\|\cdot\|$ represents again a generic norm. We note that clearly $\Phi(\alpha) \geq 0$ and that if $\Phi(\alpha) = 0$ then R_α is the null matrix. We refer to this approach as *block error minimization* (BEM) method.

To evaluate $\Phi(\alpha)$ without explicitly forming $B_1^T B_2$, $B_1^T B_3$, and $B_2^T B_3$ we consider its approximation

$$\hat{\Phi}(\alpha) = \alpha \|I\| + \frac{1}{\alpha} \left(\left\|B_1^T\right\| \|B_2\| + \left\|B_1^T\right\| \|B_3\| + \left\|B_2^T\right\| \|B_3\| \right).$$

Let $\hat{\Phi}_F(\alpha)$, $\hat{\Phi}_1(\alpha)$, and $\hat{\Phi}_\infty(\alpha)$, denote the versions of $\hat{\Phi}(\alpha)$ where the Frobenius norm, the one norm, and the infinity norm is used, respectively. Because of submultiplicativity, $\Phi(\alpha) \leq \hat{\Phi}(\alpha)$. We observe that $\hat{\Phi}(\alpha)$ has a minimum given by

$$\alpha_{BEM} = \sqrt{\frac{\left\|B_1^T\right\| \|B_2\| + \left\|B_1^T\right\| \|B_3\| + \left\|B_2^T\right\| \|B_3\|}{\|I\|}}. \tag{4.4}$$

With respect to α_{GEM}, the estimates are no longer given in terms of inequalities.

It should be mentioned that when $\|B_i\|_1 = \|B_i\|_\infty$ $(i = 1, 2, 3)$, the estimates for the optimal α obtained using the 1-norm and the infinity norm are identical. This is indeed the case for most saddle point problems arising from PDEs, and in particular for most discretizations of the Navier–Stokes equations. For more general saddle point problems, however, the two norms may lead to different estimates.

Finally, we observe that neither α_{BEM}, nor α_{GEM} depend on the viscosity ν. To amend this deficiency, we develop in Section 4.3 a different approach.

4.3. Estimation using matrix traces. Experience shows that the number of preconditioned GMRES iterations is usually lower when the eigenvalues of the right preconditioned matrix $\mathcal{T}_\alpha = \mathcal{A}_{NS}\mathcal{P}_{RDF}^{-1}$ (or, equivalently, of the left preconditioned matrix $\mathcal{T}'_\alpha = \mathcal{P}_{RDF}^{-1}\mathcal{A}_{NS}$) are clustered around one. For this reason, we propose to use the following value for α:

$$\alpha_T = \arg\min_\alpha \left| \sum_i (\lambda_i - 1) \right| = \arg\min_\alpha |\operatorname{Tr}(\mathcal{T}_\alpha) - N|,$$

where the λ_i's denote the eigenvalues of $\mathcal{T}_\alpha$ (and $\mathcal{T}'_\alpha$). As proved in [5] all these eignvalues have positive real part. Computing the trace of $\mathcal{T}_\alpha$ (or of $\mathcal{T}'_\alpha$, which is the same) is an expensive task. To reduce its cost, we use an explicit expressions of $\mathcal{T}'_\alpha$ that is provided by Lemma 4.1.

LEMMA 4.1. *Let $\hat{F}_i - F_i + \frac{1}{\alpha}B_i^T B_i$ and $S_i = B_i \hat{F}_i B_i^T$, $i = 1, 2, 3$. Then*

$$\mathcal{T}'_\alpha = I - \mathcal{P}_{RDF}^{-1}\mathcal{R}_\alpha = I - \begin{pmatrix} 0 & -\frac{1}{\alpha}\hat{F}_1^{-1}B_1^T B_2 & -\frac{1}{\alpha}\hat{F}_1^{-1}B_1^T B_3 & -\hat{F}_1^{-1}B_1^T \\ 0 & \frac{1}{\alpha^2}\hat{F}_2^{-1}B_2^T S_1 B_2 & -\frac{1}{\alpha^2}\hat{F}_2^{-1}B_2^T(\alpha I - S_1)B_3 & -\frac{1}{\alpha}\hat{F}_2^{-1}B_2^T(\alpha I - S_1) \\ 0 & \frac{1}{\alpha^3}\hat{F}_3^{-1}B_3^T(\alpha I - S_2)S_1 B_2 & \frac{1}{\alpha^3}\hat{F}_3^{-1}B_3^T[(\alpha I - S_2)S_1 + \alpha S_2]B_3 & -\frac{1}{\alpha^2}\hat{F}_3^{-1}B_3^T(\alpha I - S_2)(\alpha I - S_1) \\ 0 & -\frac{1}{\alpha^4}(\alpha I - S_3)(\alpha I - S_2)S_1 B_2 & -\frac{1}{\alpha^4}(\alpha I - S_3)[(\alpha I - S_2)S_1 + \alpha S_2]B_3 & \frac{1}{\alpha^3}(\alpha I - S_3)(\alpha I - S_2)(\alpha I - S_1) \end{pmatrix}$$

Proof. Let $\mathcal{P}_{RDF} = \frac{1}{\alpha^2}\mathcal{M}_1\mathcal{M}_2\mathcal{M}_3$ with

$$\mathcal{M}_1 = \begin{pmatrix} F_1 & 0 & 0 & B_1^T \\ 0 & \alpha I & 0 & 0 \\ 0 & 0 & \alpha I & 0 \\ -B_1 & 0 & 0 & \alpha I \end{pmatrix}, \quad \mathcal{M}_2 = \begin{pmatrix} \alpha I & 0 & 0 & 0 \\ 0 & F_2 & 0 & B_2^T \\ 0 & 0 & \alpha I & 0 \\ 0 & -B_2 & 0 & \alpha I \end{pmatrix}, \quad \mathcal{M}_3 = \begin{pmatrix} \alpha I & 0 & 0 & 0 \\ 0 & \alpha I & 0 & 0 \\ 0 & 0 & F_3 & B_3^T \\ 0 & 0 & -B_3 & \alpha I \end{pmatrix}, \text{ then we have}$$

$$\mathcal{M}_1^{-1} = \begin{pmatrix} \hat{F}_1^{-1} & 0 & 0 & -\frac{1}{\alpha}\hat{F}_1^{-1}B_1^T \\ 0 & \frac{1}{\alpha}I & 0 & 0 \\ 0 & 0 & \frac{1}{\alpha}I & 0 \\ \frac{1}{\alpha}B_1\hat{F}_1^{-1} & 0 & 0 & \frac{1}{\alpha}I - \frac{1}{\alpha^2}S_1 \end{pmatrix}, \ \mathcal{M}_2^{-1} = \begin{pmatrix} \frac{1}{\alpha}I & 0 & 0 & 0 \\ 0 & \hat{F}_2^{-1} & 0 & -\frac{1}{\alpha}\hat{F}_2^{-1}B_2^T \\ 0 & 0 & \frac{1}{\alpha}I & 0 \\ 0 & \frac{1}{\alpha}B_2\hat{F}_2^{-1} & 0 & \frac{1}{\alpha}I - \frac{1}{\alpha^2}S_2 \end{pmatrix}, \ \mathcal{M}_3^{-1} = \begin{pmatrix} \frac{1}{\alpha}I & 0 & 0 & 0 \\ 0 & \frac{1}{\alpha}I & 0 & 0 \\ 0 & 0 & \hat{F}_3^{-1} & -\frac{1}{\alpha}\hat{F}_3^{-1}B_3^T \\ 0 & 0 & \frac{1}{\alpha}B_3\hat{F}_3^{-1} & \frac{1}{\alpha}I - \frac{1}{\alpha^2}S_3 \end{pmatrix}$$

Therefore,

$$\mathcal{P}_{RDF}^{-1} = \begin{pmatrix} \hat{F}_1^{-1} & 0 & 0 & -\frac{1}{\alpha}\hat{F}_1^{-1}B_1^T \\ -\frac{1}{\alpha}\hat{F}_2^{-1}B_2^T B_1\hat{F}_1^{-1} & \hat{F}_2^{-1} & 0 & -\frac{1}{\alpha^2}\hat{F}_2^{-1}B_2^T(\alpha I - S_1) \\ -\frac{1}{\alpha^2}\hat{F}_3^{-1}B_3^T(\alpha I - S_2)B_1\hat{F}_1^{-1} & -\frac{1}{\alpha}\hat{F}_3^{-1}B_3^T B_2\hat{F}_2^{-1} & \hat{F}_3^{-1} & -\frac{1}{\alpha^3}\hat{F}_3^{-1}B_3^T(\alpha I - S_2)(\alpha I - S_1) \\ \frac{1}{\alpha^3}(\alpha I - S_3)(\alpha I - S_2)B_1\hat{F}_1^{-1} & \frac{1}{\alpha^2}(\alpha I - S_3)B_2\hat{F}_2^{-1} & \frac{1}{\alpha}B_3\hat{F}_3^{-1} & \frac{1}{\alpha^4}(\alpha I - S_3)(\alpha I - S_2)(\alpha I - S_1) \end{pmatrix}.$$

Finally, direct computations give the result. □

We now state the following result:

LEMMA 4.2. *The trace of $\mathcal{T}_\alpha$ and $\mathcal{T}'_\alpha$ is given by*

$$\mathrm{Tr}(\mathcal{T}_\alpha) = \mathrm{Tr}(\mathcal{T}'_\alpha) = N_{\mathbf{u}} + \frac{1}{\alpha}\,\mathrm{Tr}(S_1+S_2+S_3) - \frac{2}{\alpha^2}\,\mathrm{Tr}(S_1S_2+S_1S_3+S_2S_3) + \frac{2}{\alpha^3}\,\mathrm{Tr}(S_1S_2S_3),$$

where $S_i = B_i\hat{F}_{ii}^{-1}B_i^T$, $i = 1,2,3$ and $N_{\mathbf{u}}$ is the number of degrees of freedom for the discrete velocity.

Proof. It is obvious that $\mathcal{T}_\alpha$ and $\mathcal{T}'_\alpha$ have the same trace. From Lemma 4.1, we find

$$\mathrm{Tr}(\mathcal{T}'_\alpha) = \mathrm{Tr}(I) - \frac{1}{\alpha^2}\,\mathrm{Tr}(F_2^{-1}B_2^TS_1B_2) - \frac{1}{\alpha^3}\,\mathrm{Tr}\left(F_3^{-1}B_3^T\left((\alpha I_p - S_2)S_1 + \alpha S_2\right)B_3\right)$$
$$-\frac{1}{\alpha^3}\,\mathrm{Tr}\left((\alpha I_p - S_3)(\alpha I_p - S_2)(\alpha I_p - S_1)\right).$$

Thanks to the property $\mathrm{Tr}(XY) = \mathrm{Tr}(YX)$ for $X \in \mathbb{R}^{N\times M}$ and $Y \in \mathbb{R}^{M\times N}$, simple computations lead to

$$\mathrm{Tr}(F_2^{-1}B_2^TS_1B_2) = \mathrm{Tr}(B_2F_2^{-1}B_2^TS_1), \text{ and}$$
$$\mathrm{Tr}\left(F_3^{-1}B_3^T\left((\alpha I_p - S_2)S_1 + \alpha S_2\right)B_3\right) = \mathrm{Tr}\left(B_3F_3^{-1}B_3^T\left((\alpha I_p - S_2)S_1 + \alpha S_2\right)\right),$$

and this concludes the proof. ☐

The computation of $\mathrm{Tr}(S_i)$, $\mathrm{Tr}(S_iS_j)$ or $\mathrm{Tr}(S_1S_2S_3)$ for $i = 1,2,3$ in Lemma 4.2 is the most expensive part of this approach and we would like to avoid it. More generally, computing or estimating the trace of a product of matrices or the trace of the inverse of a matrix are challenging problems with no easy solution. To simplify the computation, we approximate $\hat{F}_i = F_i + \frac{1}{\alpha}B_i^TB_i$ by F_i and S_i by $\tilde{S}_i := B_i\mathrm{diag}(F_i)^{-1}B_i^T$. Then, we can explicitly form $\tilde{S}_i$ for $i = 1,2,3$ and find α_T by minimizing the function

$$\phi(\alpha) := N_{\mathbf{u}} + \frac{1}{\alpha}a - \frac{2}{\alpha^2}b + \frac{2}{\alpha^3}c,$$

where

$$a := \mathrm{Tr}(\tilde{S}_1 + \tilde{S}_2 + \tilde{S}_3), \quad b := \mathrm{Tr}(\tilde{S}_1\tilde{S}_2 + \tilde{S}_1\tilde{S}_3 + \tilde{S}_2\tilde{S}_3), \quad c := \mathrm{Tr}(\tilde{S}_1\tilde{S}_2\tilde{S}_3),$$

are independent of α. To further reduce the cost of this computation, we make use of the two following formulas valid for $A, B \in \mathbb{R}^{N\times N}$:

$$\mathrm{Tr}(A+B) = \mathrm{Tr}(A) + \mathrm{Tr}(B),$$
$$\mathrm{Tr}(AB) = \sum_{i,j=1}^{N}(A \circ B^T)_{ij}, \tag{4.5}$$

where $\circ$ denotes the Hadamard product.

Finally, we observe that it is inexpensive to evaluate $\phi(\alpha)$ once a, b, and c have been computed. Indeed, one can simply evaluate the function for α in a given range to find an approximation of the minimum value, i.e., α_T.

5. Numerical results. We now focus on testing the efficiency of the RDF preconditioner. In particular, we study the different techniques that we introduced in Section 4 to estimate α_{opt}. The numerical experiments are split into two parts. The first one shows some Matlab experiments that allow us to compare how the new estimates for α compete against the Fourier analysis estimate introduced in [5]. The second one shows some numerical experiments using up to 8192 cores on the aneurysm benchmark problem introduced in [14].

5.1. Three simple benchmark tests. In this section, we want to compare the iterations count using the new estimates with those obtained using the parameter (4.1) yielded by Fourier Analysis (FA). We consider the same settings as in [5]: we solve the Oseen equations using two test problems. First, we consider the steady 2D lid–driven cavity problem discretized by $\mathbb{Q}_2 - \mathbb{P}_1$ finite elements, which satisfy the Babuška–Brezzi (inf–sup) condition [11], on uniform grids. In particular, we consider three values for the viscosity $\nu = 0.1$, $\nu = 0.01$, and $\nu = 0.001$, and four different space discretizations corresponding to 16×16, 32×32, 64×64, and 128×128 structured meshes. Next, we consider a 3D Marker–and–Cell (MAC) discretization of the Oseen problem [20].

Unless otherwise specified, the linear system arising from the Oseen problem is solved using a right preconditioned restarted GMRES; the maximum subspace dimensions is set to 20. We use a zero initial guess and the stopping criterion is based on the norm of the residual scaled by the right hand side, i.e.,

$$\frac{\|\mathbf{r}\|_2}{\|\mathbf{b}\|_2} \leq 10^{-6}.$$

All the simulations are carried out with Matlab [27]. The problems are generated using the IFISS 3.0 software package [17].

For each problem, we report the number of iterations for different choices of α stemming from the strategies presented in Section 4, as well as an optimal value obtained experimentally by an expensive "brute force search". In all the examples, the subproblems involved in the application of the RDF preconditioner are solved by direct methods. Note that we are only interested in a good choice for α to lower the number of iterations. For that, we only report the iterations count for GMRES. We refer to [5] for numerical experiments involving the use of subiterations to apply $\hat{F}_i^{-1}$, $i = 1, 2, 3$, instead of direct methods.

5.1.1. The 2D leaky lid driven cavity problem discretized by $\mathbb{Q}_2$-$\mathbb{P}_1$ finite elements. This test problem describes the behavior of a fluid standing into a cavity and driven by a current at the top (the lid) of the domain. For example, one can think of a cavity in the bed of a river where the fluid starts moving due to the river flowing at the top. The computational domain Ω is the square $[-1, 1]^2$. $u_x = 1$ is imposed on the upper side of the square. No–slip boundary conditions are imposed on the lower, left and right sides.

In this example $\|B_i\|_1$ and $\|B_i\|_\infty$ are equal, which is to be expected since they represent discretizations of partial derivatives. Therefore, the estimates using the two norms are identical and we only report the estimates computed with the 1-norm in the tables that follow. Tables 5.1 to 5.3 show the number of GMRES iterations to converge. When $\nu = 0.1$ (see Table 5.1) we observe that the LFA, GEM_1, and the strategy based on trace estimates provide the best results. The BEM_1 estimate give acceptable results, however the number of iterations increases as the mesh is refined. BEM_F, and GEM_F estimates are the least efficient techniques in this case. The results for $\nu = 0.01$ are reported in Table 5.2. The worst estimates are again those obtained using BEM_F and GEM_F. Then, using the 16×16 and 32×32 meshes the new methods introduced in this work perform better than the LFA estimates, while in the case of the 64×64 and 128×128 the trace and the LFA estimates are better. This is consistent with the fact that, as ν becomes small, it is only guaranteed that LFA produces physically consistent solutions with fine meshes. Finally, the result for $\nu = 0.001$ are reported in Table 5.3. In this context, only results with fine meshes,

Grid	16 × 16		32 × 32		64 × 64		128 × 128	
	α	Iter	α	Iter	α	Iter	α	Iter
RDF optimal	0.101-0.161	8	0.031-0.051	8	0.011	8	0.001-0.011	9
RDF LFA estimate	0.073	9	0.025	8	0.008	8	0.002	7
RDF trace estimate	0.119	8	0.032	8	0.008	8	0.002	7
RDF BEM$_1$ estimate	0.609	11	0.304	13	0.152	14	0.076	15
RDF BEM$_F$ estimate	0.798	12	0.579	15	0.415	16	0.295	15
RDF GEM$_1$ estimate	0.370	10	0.093	10	0.023	9	0.006	8
RDF GEM$_F$ estimate	0.798	12	0.579	15	0.415	16	0.295	15

TABLE 5.1

Preconditioned GMRES on steady Oseen problems with different grid sizes ($\mathbb{Q}_2 - \mathbb{P}_1$ FEM, uniform grids), viscosity $\nu = 0.1$

Grid	16 × 16		32 × 32		64 × 64		128 × 128	
	α	Iter	α	Iter	α	Iter	α	Iter
RDF optimal	0.401-0.731	12	0.131-0.201	11	0.041	10	0.011	10
RDF LFA estimate	0.126	21	0.057	17	0.024	13	0.010	10
RDF trace estimate	1.188	15	0.321	14	0.083	12	0.021	11
RDF BEM$_1$ estimate	0.609	12	0.304	13	0.152	17	0.076	22
RDF BEM$_F$ estimate	0.798	13	0.579	19	0.415	28	0.295	37
RDF GEM$_1$ estimate	0.370	13	0.093	13	0.023	13	0.006	12
RDF GEM$_F$ estimate	0.798	13	0.579	19	0.415	28	0.295	37

TABLE 5.2

Preconditioned GMRES on steady Oseen problems with different grid sizes ($\mathbb{Q}_2 - \mathbb{P}_1$ FEM, uniform grids), viscosity $\nu = 0.01$

i.e., with 64 × 64 and 128 × 128 are relevant for the LFA estimate. The strategy based on trace estimates do not work well in this situation; the iterations count is too large. However, the number of iterations decreases as the mesh is refined. The BEM$_1$ and GEM$_1$ estimates give iteration counts that remains stable as long as the mesh is not too fine, whereas the LFA estimate gives the best results on the finest mesh. Note that on this problem, the strategies based on the 1-norm and those based on the infinity norm coincide, since $\|B_i\|_1 = \|B_i\|_\infty$. All in all, the strategy based on the trace estimate works equally well as the one based on LFA estimate as long as $\nu \geq 0.01$. BEM$_1$ works surprisingly well in all the tested configurations. The performance of GEM$_1$ is very sensitive to the viscosity. Finally, GEM$_F$ and BEM$_F$ are the least effective techniques.

5.1.2. The 3D leaky lid driven cavity problem discretized by Marker-and-Cell method. We now consider again a leaky lid driven cavity problem but in 3D and using Marker–and–Cell (MAC) discretization [20]. As in the 2D leaky lid driven cavity problem, the 1-norm and the infinity norm of B_1, B_2 and B_3 coincide. Therefore, we report only the estimates computed with the 1-norm in the tables that follow. Tables 5.4 to 5.6 show the number of GMRES iterations to converge. The LFA technique does not work well in the 3D case and is therefore not mentioned in what follows. Using $\nu = 0.1$, all the methods result in acceptable iteration counts although the trace estimate leads to a convergence requiring half the number of iterations of the other ones. If $\nu = 0.01$, BEM$_1$ gives an estimate for α that is about twice that of α_{opt} but the number of iterations remains acceptable. On the contrary, BEM$_F$, GEM$_1$, and GEM$_F$ provide an estimate that is totally inaccurate. Overall, the approximate trace estimate gives a value for α that is very close to the optimal one. Using $\nu = 0.005$, the same observation holds. In summary, only the strategy based on the trace technique is able to provide a good approximation of α_{opt}, while the others appear to overestimate

Grid	16×16		32×32		64×64		128×128	
	α	Iter	α	Iter	α	Iter	α	Iter
RDF optimal	0.651-1	30	0.281-0.321	29	0.091-0.121	27	0.031	25
RDF LFA estimate	0.126	80	0.063	71	0.032	44	0.016	29
RDF trace estimate	11.898	238	3.212	215	0.833	135	0.212	110
RDF BEM$_1$ estimate	0.609	31	0.304	29	0.152	30	0.076	47
RDF BEM$_F$ estimate	0.798	30	0.579	43	0.415	84	0.295	281
RDF GEM$_1$ estimate	0.370	39	0.093	52	0.023	55	0.006	51
RDF GEM$_F$ estimate	0.798	30	0.579	43	0.415	84	0.295	281

TABLE 5.3

Preconditioned GMRES on steady Oseen problems with different grid sizes ($\mathbb{Q}_2 - \mathbb{P}_1$ FEM, uniform grids), viscosity $\nu = 0.001$

Grid	$16\times 16\times 16$		$32\times 32\times 32$	
	α	Iter	α	Iter
RDF optimal	1.7-2.8	11	1.5-3.1	12
RDF trace estimate	1.265	13	1.306	13
RDF BEM$_1$ estimate	55.4	22	110.9	24
RDF BEM$_F$ estimate	303.6	24	1038.0	27
RDF GEM$_1$ estimate	2048	26	8192.0	28
RDF GEM$_F$ estimate	230.7	23	788.7	26

TABLE 5.4

Preconditioned GMRES on 3D steady Oseen problems with different grid sizes (MAC discretization, uniform grids), viscosity $\nu = 0.1$

it.

5.2. Large-scale experiments. For large-scale 3D problems, exact application of the RDF preconditioner is too expensive and inexact variants must be used [5]. The approximation of the RDF preconditioner uses the factorization of the RDF preconditioner from Section 3 as a starting point. Here, inverses of the algebraic operators are replaced by some suitable approximations denoted by a "tilde" over the corresponding operators. The approximate RDF preconditioner can then be defined in factored form, as follows:

$$\mathcal{P}_{aRDF}^{-1} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & \frac{1}{\alpha}B_3 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & \tilde{\hat{F}}_3^{-1} & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & -B_3^T \\ 0 & 0 & 0 & I \end{pmatrix}$$
$$\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & \frac{1}{\alpha}B_2 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & \tilde{\hat{F}}_2^{-1} & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & -\frac{1}{\alpha}B_2^T \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & \frac{1}{\alpha}I \end{pmatrix}$$
$$\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ B_1 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} \tilde{\hat{F}}_{11}^{-1} & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & -\frac{1}{\alpha}B_1^T \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix},$$

where, as before, $\hat{F}_i = F_i + \frac{1}{\alpha}B_i^T B_i$ for $i = 1, 2, 3$. In our experiments we use GMRES with an algebraic multigrid (AMG) preconditioner to approximate the action of $\hat{F}_i^{-1}$ for $i = 1, 2, 3$. We use two sweeps of symmetric Gauss–Seidel iterations to perform

Grid	$16 \times 16 \times 16$		$32 \times 32 \times 32$	
	α	Iter	α	Iter
RDF optimal	13.8-20.1	17	12.4-15	18
RDF trace estimate	12.651	18	13.064	18
RDF BEM$_1$ estimate	55.4	29	110.9	44
RDF BEM$_F$ estimate	303.6	63	1038.0	84
RDF GEM$_1$ estimate	2048	87	8192.0	94
RDF GEM$_F$ estimate	230.7	56	788.7	82

TABLE 5.5

Preconditioned GMRES on 3D steady Oseen problems with different grid sizes (MAC discretization, uniform grids), viscosity $\nu = 0.01$

Grid	$16 \times 16 \times 16$		$32 \times 32 \times 32$	
	α	Iter	α	Iter
RDF optimal	23.2-30.2	24	23.2-29.8	24
RDF trace estimate	25.982	24	26.133	24
RDF BEM$_1$ estimate	55.4	35	110.9	56
RDF BEM$_F$ estimate	303.6	100	1038.0	158
RDF GEM$_1$ estimate	2048	182	8192.0	199
RDF GEM$_F$ estimate	230.7	87	788.7	147

TABLE 5.6

Preconditioned GMRES on 3D steady Oseen problems with different grid sizes (MAC discretization, uniform grids), viscosity $\nu = 0.005$

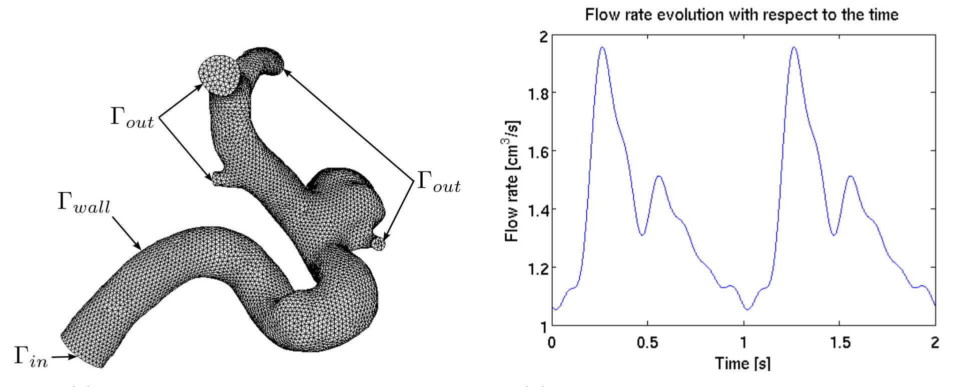
pre-smoothing only, and an LU factorization to solve the coarse problem. AMG works remarkably well in this case because $\hat{F}_i$ is dominated by its symmetric part $\frac{1}{2}(F_i + F_i^T) + \frac{1}{\alpha}B_i^T B_i$.

The three-dimensional simulations have been coded in `LifeV` [1], a C++ finite element library under the LGPL license. This library makes intensive use of `Trilinos` [23], which contains many distributed packages. In particular, the multigrid algorithm relies on the `ML` package, and GMRES is based on the `Belos` package. Whenever mesh coarsening is required, aggregates are computed using `METIS`/`ParMETIS` [24, 25]. To perform the LU factorization required by the coarse solve of multigrid, we chose to use KLU [13] rather than UMFPACK [12]. Although KLU has been developed primarily for solving sparse linear systems from circuit simulations and not the ones arising in finite element modeling, we found that the time to compute the factorization is smaller than that of UMFPACK and this makes the total costs (time to build the preconditioners and time for the preconditioned iterations) smaller. All the computations are carried out using Monte Rosa, a Cray XE6 supercomputer at the Swiss National Supercomputing Centre (CSCS), cf. Table 5.7. The reported results are obtained using as many cores as possible on each node, without multithreading enabled.

5.2.1. Flow in a cerebral aneurysm. We consider the simulation of blood flow in a cerebral aneurysm, which is a localized blood–filled deformation in a blood vessel wall. The computational domain Ω represents an artery where an aneurysm has developed (see Figure 5.1(a)). The geometry of the cerebral aneurysm was first used in the research project Aneurisk [31]. Detailed informations on the meshes are provided in Table 5.9. The diameter of the inlet Γ_{in} measures 0.35 cm and is chosen as characteristic length. Figure 5.1(a) shows the mesh used to perform the simulations, for a time step $\Delta t = 10^{-3}$. Blood flow is modeled by the Navier–Stokes equations with density $\rho = 1$ g/cm^3 and viscosity $\mu = 0.035$ g/(cm·s), i.e., $\nu = 0.035$ cm^2/s. An

Number of service nodes	40
Number of compute nodes	1'496
Number of processors per node	2x 16-core AMD Opteron Interlagos
Processors frequency	2.1 GHz
Processors shared memory per node	32 GB DDR2
Peak performance	402 Teraflop/s.
Network	Gemini 3D torus

TABLE 5.7
Monte Rosa Cray XE6 technical data



(a) Mesh of the cerebral aneurysm (b) Flux evolution with respect to the time

FIG. 5.1. *Mesh and inflow for the cerebral aneurysm*

approximation of the flow rate at the inlet Γ_{in} has been provided in [2] as

$$\varphi(t) = a_0 + \sum_{k=1}^{7} a_k \cos\left(\frac{2\pi kt}{T}\right) + b_k \sin\left(\frac{2\pi kt}{T}\right), \tag{5.1}$$

where T denotes the period of the cardiac cycle, and a_k and b_k are given in Table 5.8 (the coefficients have been scaled to correspond to our geometry). The function $\varphi(t)$ is presented in Figure 5.1(b). For our computations, we take $T = 1$ and the flow rate is imposed by a flat inlet profile on Γ_{in}. Homogenous Neumann boundary conditions are imposed on Γ_{out}, and no–slip boundary conditions are imposed on the vessel wall Γ_{wall}.

The problem is discretized in time using a fully implicit scheme linearized with Picard–Oseen iterations with $\Delta t = 10^{-3}$ and in space using the $\mathbb{P}_2 - \mathbb{P}_1$ finite elements on a tetrahedral mesh. The tolerance ϵ_{NL} for the nonlinear iterations of the implicit scheme is set to 10^{-6}. Flexible GMRES (FGMRES) without restart is used to solve the linear problem at each time step [30]. The preconditioner is recomputed at each time iteration. The stopping criterion is based on the residual scaled by the right hand side,

$$\|\mathbf{r}_k\|_2 \leq 10^{-6} \|\mathbf{b}\|_2 .$$

We evaluate four versions of the approximate RDF preconditioner: each of them corresponds to the tolerance that is used to apply $\hat{F}_i^{-1}$, $i = 1, 2, 3$ using the inner

	0	1	2	3	4	5	6	7
a_k	1.36	-0.207	-0.152	0.059	0.039	0.00286	-0.0371	-0.000759
b_k		0.176	-0.0429	-0.117	0.0385	0.0139	0.0166	-0.0359

Table 5.8
Coefficients for the flow rate function, [2]

Mesh	Velocity DoF	Pressure DoF	h_{min}	$h_{average}$	h_{max}
Coarse	597’093	27’242	0.015	0.035	0.059
Medium	4’557’963	199’031	0.005	0.018	0.051
Fine	35’604’675	1’519’321	0.0026	0.0097	0.0277

Table 5.9
Aneurysm test case: Number of Degrees of Freedom (DoF) and mesh size

AMG-preconditioned GMRES method, namely 10^{-1}, 10^{-2}, 10^{-4}, and 10^{-10}. For this reason, at each Oseen–Picard iteration, the system is solved using FGMRES. We first discuss the efficiency of the strategy based on the trace estimation technique. Table 5.10 shows the approximate value for α_{opt} computed by successive trials, the value obtained using the trace estimation technique as well as the average number of iterations over the Picard iterations, indicated in parenthesis, to solve one timestep. Using the medium mesh, we see that we are close to α_{opt}. We have not computed an estimation for α_{opt} using the fine mesh to save resources and since the iteration count is already even better than for the medium mesh. Figure 5.2 gives more details about how the iterations count behaves with respect to α using the coarse and the medium meshes.

We first report the time to build the RDF preconditioner in Figure 5.3. With the coarse mesh, strong scalability is observed up to 512 cores. With the medium and fine meshes, the time is superlinear, thanks to the LU factorization which involves the assembly of the coarse level of the multigrid preconditioners for $\hat{F}_i$, $i = 1, 2, 3$.

We report in Table 5.11 the time needed to compute α_T (in parenthesis, we report the fraction w.r.t. the total time), the time to compute $\hat{F}_1$ and the total time to build the RDF preconditioner. With all the meshes, we note that computing α_T does not scale at all with an increasing number of processors. In fact, the time is even increasing. However, the value of α_T does not vary much during the simulation. Therefore, in practice, one can compute it once and for all using a lower number of cores than the number used to do the simulations, at a negligible cost. The computation does not scale because the entries of the matrices B and B^T are stored by rows, which are owned by different parallel tasks. Therefore, when we compute the product as in Equation (4.5), a lot of communication is required to transpose the matrix, i.e., the rows of the matrix become the columns and the row entries have to be sent to the appropriate parallel task.

Building the RDF preconditioner is fairly expensive, because the number of entries contained in the matrices $\hat{F}_i$, $i = 1, 2, 3$ compared to those of the matrix F_i, is high. We examined a sparse version of $\hat{F}_i$ where only the entries that are already in the pattern of F are kept. Therefore, the number of nonzero entries of the sparse $\hat{F}_i$ is the same as that of F. Unfortunately, the resulting preconditioned FGMRES method exhibits poor convergence, but this version of the RDF preconditioner can be built very quickly. Future research should be aimed at finding a version that leads to good convergence rates of FGMRES using sparse approximations of $\hat{F}_i$, $i = 1, 2, 3$. In this

Mesh	α_{opt}	Iteration count for α_{opt}	α	Iteration count for α
Coarse	$\sim 10^{-5}$	50	$7 \cdot 10^{-5}$	137.33 (6)
Medium	$\sim 5 \cdot 10^{-6}$	16.4	$1.5 \cdot 10^{-6}$	17.25 (4)
Fine	-	-	$1.76 \cdot 10^{-6}$	9.75 (4)

TABLE 5.10
Aneurysm test case: comparison between α and α_{opt}

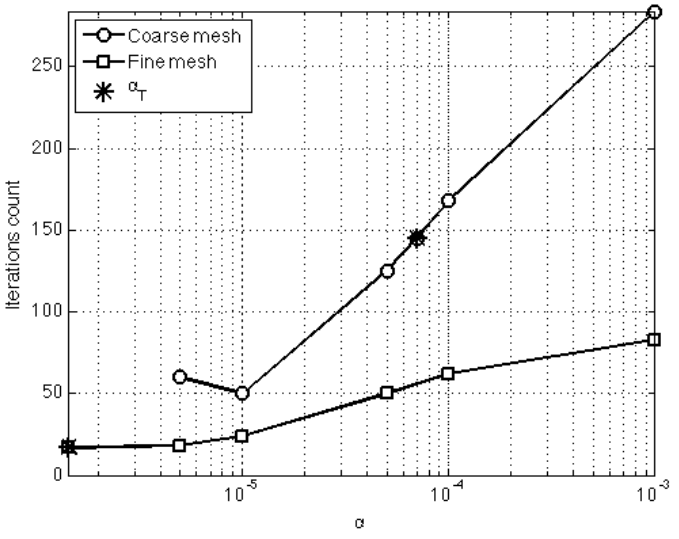
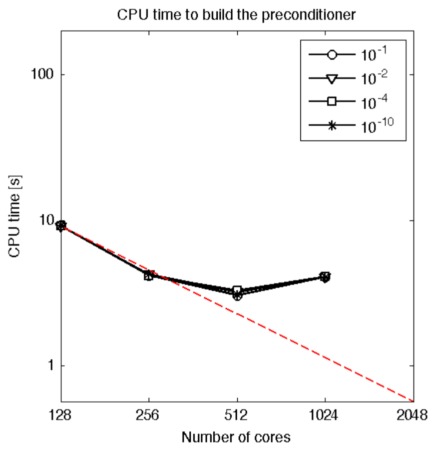


FIG. 5.2. *Aneurysm test case: number of GMRES iterations with respect to the choice of α*

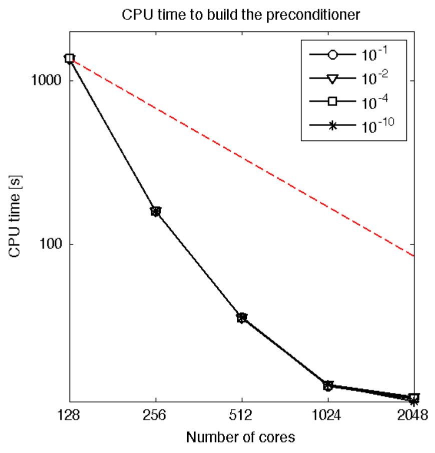
direction, we mention a promising variant of the augmented Lagrangian approach where a Grad–Div stabilization technique is used to build the augmented matrix; see [22, 21]. In this approach the augmentation is performed at the continuous level, before discretization. Experiments show that this reduces the number of entries while preserving the convergence rates. A similar approach may work well in the context of RDF preconditioning.

Figure 5.4 reports the number of iterations of FGMRES. With the coarse mesh the number of iterations remains constant for the range of number of cores that we considered. Nevertheless, the iterations count is quite large for all the precisions with which we apply the RDF. It is possible that this is due to the mesh being too coarse to yield physically meaningful solutions, and to the onset of instabilities ("wiggles") in the corresponding discrete solution. Using the medium or the fine mesh, however, the number of iterations is much lower than with the coarse mesh and remains constant. We note that applying less accurately the RDF preconditioner slightly increases the number of iterations, but in all the considered cases, the number of iterations remains almost constant.

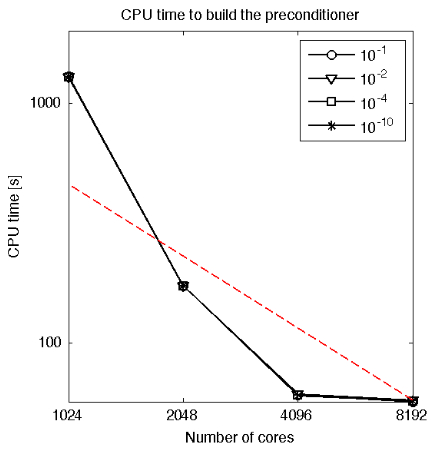
Finally, we show the scalability curves of the time needed to solve the linear system (FGMRES iterations) in Figure 5.5. With the coarse mesh the strong scalability is observed up to 512 cores, while the scalability is almost perfect with the medium and fine mesh. The accuracy used to apply the RDF preconditioner is here again not affecting the slope of the scalability curves. Therefore, it is suitable to apply the



(a) Coarse mesh



(b) Medium mesh



(c) Fine mesh

FIG. 5.3. *Aneurysm test case: CPU time needed to build the preconditioners*

inexact version using the inner tolerance 0.1, which is about 10 times faster than when using the tolerance 10^{-10}. Overall, the RDF preconditioner, while expensive to build, is very robust and is able to produce solutions even for very difficult cases.

To summarize, the RDF preconditioner built using α computed with the trace estimate leads to a robust and scalable preconditioning technique. The main issue remains the high computation time to build it due to the number of nonzero entries in $\hat{F}_i$, $i = 1, 2, 3$. Therefore, finding a sparse approximation to $\hat{F}_i$, $i = 1, 2, 3$, will be the key to obtain a more efficient version of the RDF preconditioner.

6. Conclusion.

In this paper, we considered various techniques to estimate the parameter α in the RDF preconditioner. In particular, the approach based on the trace of the preconditioned matrix results in very good convergence rates for both 2D and 3D cases. The efficiency of the preconditioner can still be improved. Numerical experiments on a large 3D hemodynamics problem show that RDF scales very well with increasing core counts. Future work should focus on improving the efficiency of

Mesh	Cores	Computation of α_T		Preconditioner for $\hat{F}_1$	Total time
Coarse	128	1.09	(0.12)	2.61	9.13
	256	0.9	(0.22)	1.0	4.1
	512	1.0	(0.34)	0.59	2.95
	1024	1.72	(0.43)	0.64	3.97
Medium	128	4.77	$(3.57 \cdot 10^{-3})$	268.3	1336.85
	256	3.16	(0.02)	52.0	158.73
	512	2.48	(0.07)	10.74	35.33
	1024	2.57	(0.19)	3.42	13.49
	2048	4.29	(0.36)	1.91	11.89
Fine	1024	6.97	$(5.38 \cdot 10^{-3})$	240.12	1296.21
	2048	7.18	(0.04)	54.48	172.55
	4096	13.0	(0.22)	14.47	60.34
	8192	23.22	(0.41)	9.41	56.25

Table 5.11
Aneurysm test case: time to build the RDF preconditioner

the preconditioner. In particular, techniques to reduce fill-in in the sub-matrices that comprise the RDF preconditioner should be investigated.

Acknowledgements. This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project IDs h02 and s392. We gratefully acknowledge the CSCS for providing us the CPU resources for our simulations. We want to thank Prof. Vaidy S. Sunderam of the Department of Mathematics and Computer Science at Emory University for facilitating this collaboration. Some preliminary parallel experiments were conducted using the Blue Gene/P system at EPFL. The financial support for CADMOS and the Blue Gene/P system is provided by the Canton of Geneva, Canton of Vaud, Hans Wilsdorf Foundation, Louis-Jeantet Foundation, University of Geneva, University of Lausanne and Ecole Polytechnique Fédérale de Lausanne. Finally, we are also thankful to Prof. Daniel Kressner for pointing us to some references about the approximations of traces of matrices.

REFERENCES

[1] LifeV user manual. http://www.lifev.org, 2010.

[2] H. Baek, M. V. Jayaraman, P. D. Richardson, and G. E. Karniadakis. Flow instability and wall shear stress variation in intracranial aneurysms. *J R Soc Interface*, 7:967–988, 2010.

[3] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.

[4] M. Benzi and X.-P. Guo. A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations. *Appl. Numer. Math.*, 61(1):66–76, 2011.

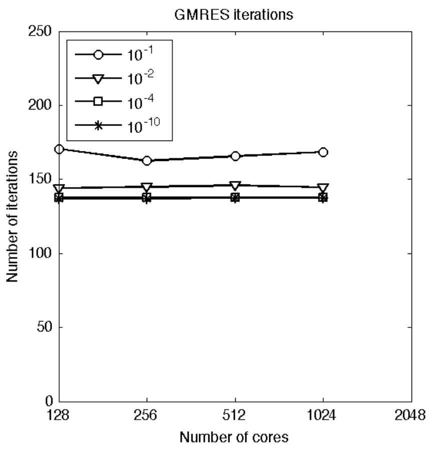
[5] M. Benzi, M. Ng, Q. Niu, and Z. Wang. A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 230(16):6185–6202, 2011.

[6] M. Benzi and M. A. Olshanskii. An augmented Lagrangian-based approach to the Oseen problem. *SIAM J. Sci. Comput.*, 28(6):2095–2113 (electronic), 2006.

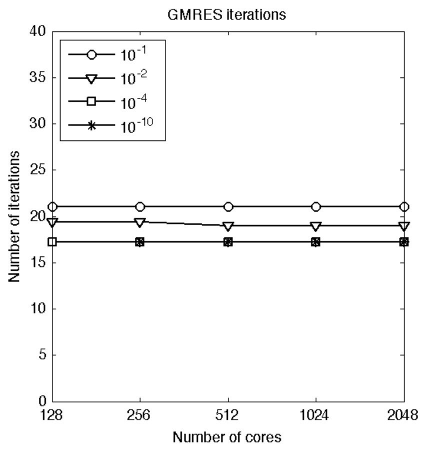
[7] M. Benzi and M. A. Olshanskii. Field-of-values convergence analysis of augmented Lagrangian preconditioners for the linearized Navier-Stokes problem. *SIAM J. Numer. Anal.*, 49(2):770–788, 2011.

[8] M. Benzi, M. A. Olshanskii, and Z. Wang. Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, (66):486–508, 2011.

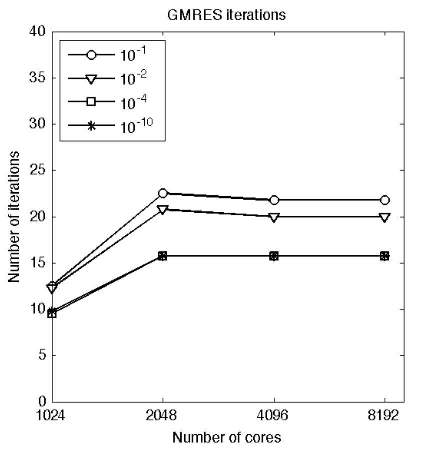
[9] M. Benzi and Z. Wang. Analysis of augmented Lagrangian-based preconditioners for the steady incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 33:2761–2784, 2011.



(a) Coarse mesh



(b) Medium mesh



(c) Fine mesh

FIG. 5.4. *Aneurysm test case: Number of FGMRES iterations*

[10] M. Benzi and Z. Wang. A parallel implementation of the modified augmented Lagrangian preconditioner for the incompressible Navier–Stokes equations. *Numer. Algorithms*, 64(1):73–84, 2013.

[11] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge*, 8(R-2):129–151, 1974.

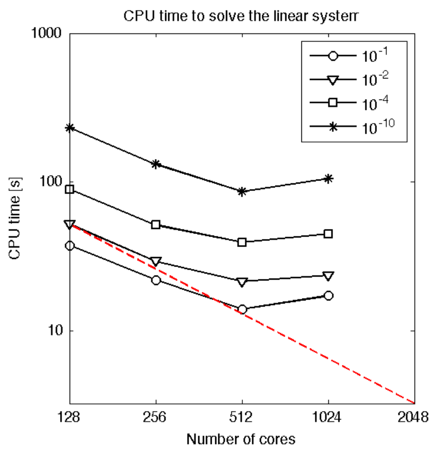
[12] T. A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, June 2004.

[13] T. A. Davis and E. Palamadai Natarajan. Algorithm 907: KLU, a direct sparse solver for circuit simulation problems. *ACM Transactions on Mathematical Software (TOMS)*, 37(3):36, 2010.

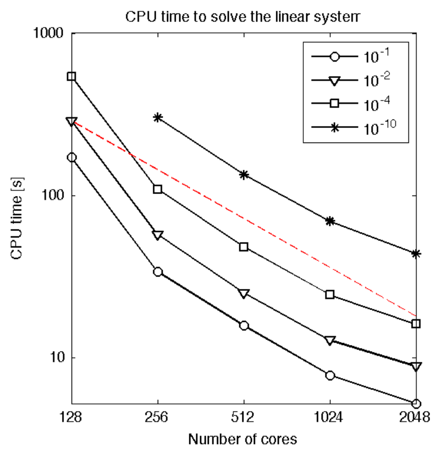
[14] S. Deparis, G. Grandperrin, and A. Quarteroni. Parallel preconditioners for the unsteady Navier–Stokes equations and applications to hemodynamics simulations. *Computers & Fluids*, (0):–, 2013.

[15] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM J. Sci. Comput.*, 27(5):1651–1668, 2006.

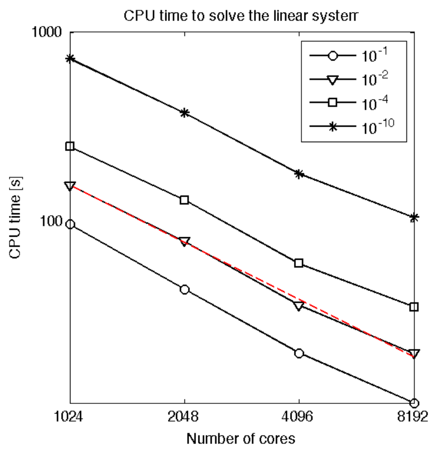
[16] H. Elman, V. E. Howle, J. Shadid, D. Silvester, and R. Tuminaro. Least squares preconditioners for stabilized discretizations of the Navier-Stokes equations. *SIAM J. Sci. Comput.*,



(a) Coarse mesh



(b) Medium mesh



(c) Fine mesh

FIG. 5.5. *Aneurysm test case: CPU time for the preconditioned iterations*

30(1):290–311, 2007.

[17] H. C. Elman, A. Ramage, and D. J. Silvester. IFISS, a Matlab toolbox for modelling incompressible flow. *ACM Trans. Math. Softw.*, 33, June 2007.

[18] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics.* Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005.

[19] H. C. Elman and R. S. Tuminaro. Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations. *Electron. Trans. Numer. Anal.*, 35:257–280, 2009.

[20] F. H. Harlow and J. E. Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*, 8(12):2182–2189, 1965.

[21] T. Heister. *A Massively Parallel Finite Element Framework with Application to Incompressible Flows.* PhD thesis, Georg-August-Universität Göttingen, 2011.

[22] T. Heister, G. Lube, and G. Rapin. On robust parallel preconditioning for incompressible flow problems. In *Numerical Mathematics and Advanced Applications 2009*, pages 443–450. Springer, 2010.

[23] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist,

R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.

[24] G. Karypis, K. Schloegel, and V. Kumar. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Technical report, Univ MN., 1998.

[25] G. Karypis, K. Schloegel, and V. Kumar. ParMETIS: Parallel Graph Partitioning and Sparse Matrix Ordering library. Technical report, Univ MN., 2003.

[26] D. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier-Stokes equations. *SIAM J. Sci. Comput.*, 24(1):237–256, 2002.

[27] MATLAB. *version 7.14.0.739 (R2012a)*. The MathWorks Inc., Natick, Massachusetts, 2012.

[28] A. Quarteroni. *Numerical models for differential problems*, volume 8 of *MS&A. Modeling, Simulation and Applications*. Springer-Verlag Italia, Milan, 2014.

[29] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations.* Numerical Mathematics and Scientific Computation. The Clarendon Press - Oxford University Press, New York, 1999. Oxford Science Publications.

[30] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.

[31] L. M. Sangalli, P. Secchi, S. Vantini, and A. Veneziani. A case study in exploratory functional data analysis: geometrical features of the internal carotid artery. *J. Amer. Statist. Assoc.*, 104(485):37–48, 2009.

[32] D. Silvester, H. Elman, D. Kay, and A. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *J. Comput. Appl. Math.*, 128(1-2):261–279, 2001. Numerical analysis 2000, Vol. VII, Partial differential equations.

[33] R. Wienands and W. Joppich. *Practical Fourier analysis for multigrid methods*, volume 4 of *Numerical Insights*. Chapman & Hall/CRC, Boca Raton, FL, 2005. With 1 CD-ROM (Windows and UNIX).