VII International Conference on Computational Methods in Marine Engineering
MARINE 2017
M. Visonneau, P. Queutey and D. Le Touzé (Eds)

ON THE DESIGN OF BLOCK PRECONDITIONERS FOR MARITIME ENGINEERING

C.M. KLAIJ[*], **X. HE**[†] **and C. VUIK**[‡]

[*] Maritime Research Institute Netherlands
P.O.Box 28, 6700AA Wageningen, The Netherlands
email: c.klaij@marin.nl, web page: http://www.marin.nl/

[†] Institute of Computing Technology, Chinese Academy of Sciences
No.6 Kexueyuan South Road Zhongguancun Haidian District Beijing, P.R.China 100190
email: hexin2016@ict.ac.cn, web page: http://english.ict.cas.cn/

[‡] Delft University of Technology
Mekelweg 4, 2628CD Delft, The Netherlands
email: c.vuik@tudelft.nl, web page: http://www.ewi.tudelft.nl/

Keywords: incompressible flow, Reynolds-averaged Navier-Stokes, segregated and coupled solvers, block preconditioners, Schur complement approximation

Abstract. The iterative error can be an important part of the total numerical error of any Computational Fluid Dynamics simulation when the iterative convergence stagnates or when loose convergence criteria are used. In the quest for better iterative convergence of CFD simulations, we consider the design of iterative methods for the Reynolds-averaged Navier-Stokes equations, discretized by finite-volume methods with cell-centered, co-located variables. The central point is the approximation of the Schur complement (pressure matrix) in the block factorization of the discrete system of mass and momentum equations. We show particular approximations of these blocks that yield either segregated solvers or block preconditioners for fully coupled solvers. The performance of these solvers are then demonstrated by computing the flow over a flat plate and around a tanker on both structured and unstructured grids. We find that iterative convergence to machine precision is attainable despite the high Reynolds numbers and mesh aspect ratio's. Improved approximations of the Schur complement do result in improved convergence rates, but do not seem to pay-off in terms of total cost compared to the basic SIMPLE-type approximation.

1 INTRODUCTION

Applications of Computational Fluid Dynamics (CFD) in maritime engineering typically involves incompressible, turbulent flow around complex geometries. Most commercial and open-source CFD packages rely on the Reynolds-averaged Navier-Stokes (RaNS) equations to model such flows, on finite-volume methods with co-located variables to discretize these equations, and on SIMPLE-type methods to solve the resulting discrete systems [1, 2, 3]. Various procedures can be applied to quantify the total numerical error of a CFD simulation before making design choices [4, 5, 6]. The total numerical error consists of a discretization error that can be reduced by grid refinement and of an iterative error that can be reduced by tightening the convergence criteria. Unfortunately, as the grid is refined, the convergence rate of methods such as SIMPLE typically deteriorates to the point where even loose criteria are out of reach. Therefore, in this paper, we focus on alternative iterative methods.

Two alternatives emerge from the vast literature on scalable methods for the incompressible Navier-Stokes equations. Both share the idea of solving the coupled mass-momentum system (instead of solving a SIMPLE-type decomposition), but differ in the way this is achieved. One way is to interlace the velocity and pressure unknowns to form a system of small 4-by-4 blocks. The small blocks on the diagonal are easily inverted, which can be exploited in multigrid methods [7, 8, 9] or in Krylov subspace methods [10]. The other way is to separate the velocity and pressure unknowns to form a system with 4 large blocks and exploit this structure in Krylov subspace methods with block preconditioners. This seems to be a popular choice in the finite-element context (see [11] for an overview), but less so in the finite-volume context, although some results are found in [12] for RaNS on co-located grids. On staggered grids, we find results for the Stokes and Oseen equations in [13, 14, 15].

Our research is focused on the design of efficient preconditioners for industrial CFD applications. We have shown in [16, 17] that SIMPLE-type methods can effectively be used as preconditioners, resulting in significant savings of CPU time compared to their standard usage as solvers. In [18], we have shown that the stabilization which is typically used for finite-volume methods with co-located variables to avoid spurious pressure oscillations is particularly favorable for SIMPLE-type methods. Furthermore, we have ported the Augmented Lagrangian Preconditioner [19], which is popular in finite-element context, to the finite-volume context in [20] with promising results for academic test cases.

In this paper, we consider Schur complement preconditioners that are readily available in PETSc [21] once the coupled mass-momentum matrix is provided. Many variations are possible, ranging from the exact Schur complement preconditioner to SIMPLE-type preconditioners and any approximation in-between. We apply three variations to test cases that are relevant for maritime engineering and report iteration counts which, together with a basic cost model, serve as a measure for potential savings in CPU time.

The paper is thus organized in two parts: first a description of the overall algorithm, the mass-momentum matrix, the preconditioners and their relative cost, then followed by examples of the preconditioners' performance for the three test case.

2 ITERATIVE SOLUTION OF THE RANS EQUATIONS

2.1 Non-linear iteration and coupling

To solve the Reynolds-averaged Navier-Stokes equations we start by linearizing the momentum equations. Since the non-linearity comes from the convective term $\mathrm{div}(\rho\tilde{\mathbf{v}}\mathbf{v})$, it can be linearized by assuming that the mass flux $\rho\tilde{\mathbf{v}}$ is known from a previous iteration. For finite-volume discretization with co-located variables, computing this mass flux involves the pressure-weighted interpolation (PWI) of the velocity, symbolized here by writing $\tilde{\mathbf{v}}$ instead of $\mathbf{v}$. The PWI effectively couples the mass and momentum equations and thereby avoids spurious pressure oscillations. Furthermore, the coupling between the Navier-Stokes equations and the turbulence model takes place through the convective terms $\operatorname{div}(\rho\tilde{\mathbf{v}}\phi)$ in the turbulence transport equations where ϕ is a quantity such as turbulent kinetic energy, and through the diffusive terms $\operatorname{div}(\mu_{\text{eff}}(\operatorname{grad}\mathbf{v} + (\operatorname{grad}\mathbf{v})^T)$ in the momentum equations. In these terms, the effective viscosity μ_{eff} is the sum of the (constant) dynamic viscosity μ and the (variable) turbulent eddy viscosity μ_t provided by the turbulence model. The effective viscosity is assumed to be known when solving the momentum equations. The iterative process of solving the mass and momentum equations with known effective viscosity, computing the mass flux and solving the turbulence equations is repeated until a convergence criterion is met. The solution of the system of mass and momentum equations within this procedure is of the form

$$\begin{aligned} x^{k+1} &= x^k + x' \\ &= x^k + (A^k)^{-1}(b^k - A^k x^k) \qquad k = 0, 1, \dots \end{aligned} \tag{1}$$

with unknowns $x = [\mathbf{v}, p]^T$. Thus, at non-linear iteration k, the value x^k is corrected by x', which is the solution of $A^k x' = r^k$ with $r^k = b^k - A^k x^k$ the residual of the mass and momentum equations. We are interested in solvers that can exploit the structure of A, which we discuss first.

2.2 Structure of the mass-momentum matrix

We consider a finite-volume discretization where the variables are co-located in the cell centers. On a three-dimensional computational mesh with N cells, we order the $4N$ unknowns by equation as $[(v_1)_1, \dots, (v_1)_N, (v_2)_1, \dots, (v_2)_N, (v_3)_1 \dots (v_3)_N, p_1, \dots, p_N]^T$. The system of mass and momentum equations $A^k x' = r^k$ then has the following form

$$\begin{bmatrix} Q_1 & 0 & 0 & G_1 \\ 0 & Q_2 & 0 & G_2 \\ 0 & 0 & Q_3 & G_3 \\ D_1 & D_2 & D_3 & C \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ p \end{bmatrix}' = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_p \end{bmatrix}$$

where each block is a sparse N-by-N matrix. Here, Q contains the contribution from the convection and diffusion, G from the pressure gradient, D from the velocity divergence and C from the PWI.

The block diagonal form of Q follows not only from the linearization that takes the mass flux $\rho\tilde{\mathbf{v}}$ from the previous iteration, but also requires the second part of the diffusion term, $(\operatorname{grad}\mathbf{v})^T$, to be taken from the previous iteration. Notice that the matrices Q_i are equal because they only contain the convection term $\operatorname{div}(\rho\tilde{\mathbf{v}}v_i)$ and the remaining diffusion term $\operatorname{div}(\mu_{\text{eff}}(\operatorname{grad} v_i))$. To avoid the order barrier [3], higher-order convection schemes are non-linear even though the convection term is linear. Such schemes are implemented in defect-correction form [3, 22] and combined with iteration (1)

The gradient matrix G and divergence matrix D are obtained by Gauss' theorem and linear interpolation. Eccentricity and non-orthogonality corrections [2] are implemented in defect-correction form and combined with (1). The stabilization matrix C follows from the pressure-weighted interpolation and has the particular form [18]

$$C = -\frac{1}{dQ}L + D\frac{1}{dQ}G$$

with L the N-by-N Laplacian matrix and dQ the diagonal of Q. This stabilization is particularly favorable for SIMPLE-type solvers. Notice that C, as the difference between two ways of discretizing second-order derivatives, is in some sense close to zero. Solving $A^k x' = r^k$ by the Jacobi iterative method is impossible, because the diagonal elements of C can be zero. For this reason we investigate linear solvers which can be applied even if $C = 0$.

2.3 Segregated and coupled solvers

For brevity, we will continue with the following shorthand notation for $Ax' = r$ and consider the LDU factorization of the matrix A:

$$\begin{bmatrix} Q & G \\ D & C \end{bmatrix} \begin{bmatrix} v \\ p \end{bmatrix}' = \begin{bmatrix} r_v \\ r_p \end{bmatrix}, \qquad \begin{bmatrix} Q & G \\ D & C \end{bmatrix} = \begin{bmatrix} I & 0 \\ DQ^{-1} & I \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & Q^{-1}G \\ 0 & I \end{bmatrix}$$

with $S = C - DQ^{-1}G$ the Schur complement. The corresponding inverse

$$\begin{bmatrix} Q & G \\ D & C \end{bmatrix}^{-1} = \begin{bmatrix} I & -Q^{-1}G \\ 0 & I \end{bmatrix} \begin{bmatrix} Q^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -DQ^{-1} & I \end{bmatrix}$$

shows that the solution of $Ax' = r$ involves the solution of momentum sub-systems (Q^{-1}) and pressure sub-systems (S^{-1}). The main problem is that S is a dense matrix that cannot be explicitly constructed in real-life applications. Therefore, it needs to be approximated, for example by the SIMPLE approximation $\tilde{S} = C - D\frac{1}{dQ}G = -\frac{1}{dQ}L$ that exploits the particular form of C. If we disregard the upper block, we obtain the following solver $x' = P^{-1}r$ with

$$\begin{bmatrix} v \\ p \end{bmatrix}' = \begin{bmatrix} Q^{-1} & 0 \\ 0 & \tilde{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -DQ^{-1} & I \end{bmatrix} \begin{bmatrix} r_v \\ r_p \end{bmatrix} = \begin{bmatrix} Q^{-1}r_v \\ \tilde{S}^{-1}(r_p - DQ^{-1}r_v) \end{bmatrix}$$

Here, we recognize the first two steps of the SIMPLE method: 1) momentum solve $Qu' = r_v$ and 2) pressure solve $\tilde{S}p' = r_p - Du'$. The third step of SIMPLE is obtained when we also include the upper block in the approximation together with the approximation $Q^{-1} \approx \frac{1}{dQ}$ that was already used for the Schur complement. Other solvers can be devised by making other approximations, the trade-off being that a better approximation is usually more expensive.

Note that $x' = P^{-1}r$ is referred to as a *segregated* solver: it does not involve the matrix A. The matching *coupled* solver $x' = A^{-1}r$ is obtained by using P^{-1} as a preconditioner, i.e. by solving $P^{-1}Ax' = P^{-1}r$ with a Krylov subspace method such as GMRES. Since the coupled solver computes the corrections x' with the original matrix A instead of the cheaper approximation P, we expect faster convergence of the sequence $\{x^k\}$ but also a higher cost per iteration.

Once the matrix A is provided, PETSc can readily construct preconditioners based on the inverse of the LDU factorization with any desired approximation. Here, we will consider three variations [1]:

- 'exact Schur': full factorization with momentum and pressure sub-systems solved to a relative tolerance of 10^{-12}. Preconditioner for S based on the SIMPLE approximation $\tilde{S}$.

[1] Be careful, similar names in literature can have slightly different meaning.

- ‘approx Schur’: same as exact Schur but with a relative tolerance of 10^{-2} instead of 10^{-12}.
- ‘SIMPLE’: same as approx Schur but with $\tilde{S}$ instead of S.

For the mass-momentum system we use FGMRES with any of these preconditioners, for the subsystems we use GMRES with ILU(0) preconditioning. The exact Schur preconditioner should lead to convergence in two steps [23], which is the main motivation for considering cheaper approximations that are feasible in real-life applications.

2.4 Solver cost model

Wall-clock time may be the obvious measure to discriminate between different solvers but since it highly depends on machine and implementation, we get more insight from a basic cost model.

First consider the cost of the segregated SIMPLE solver. This forms the baseline for evaluating the cost of other solvers. At each non-linear iteration, SIMPLE solves the three momentum equations (‘mom-u’) and the pressure equation (‘mass-p’) to a certain relative tolerance, which we assume to be fixed. Thus, the total cost expressed in solves per non-linear iteration is

- segregated SIMPLE: 3 mom-u + 1 mass-p.

Now consider the cost of using SIMPLE as a preconditioner for a Krylov subspace method that solves the coupled mass-momentum system to a certain relative tolerance in n_x iterations. The preconditioner is applied at each Krylov iteration. Therefore, if all other costs are assumed negligible compared to the cost of the preconditioner, we arrive at a total cost of

- coupled, SIMPLE preconditioner: $n_x \times (3 \text{ mom-u} + 1 \text{ mass-p})$

per non-linear iteration. Clearly, this higher cost per non-linear iteration only pays-off if the number of non-linear iterations required to achieve convergence is significantly reduced compared to the baseline solver.

Finally consider the cost of the Schur preconditioners. Again these preconditioners are applied at each Krylov iteration of a coupled mass-momentum solver. However, the matrix in the pressure equation is now different. Instead of solving $\tilde{S}p' = r_p$ with $\tilde{S} = C - D\frac{1}{dQ}G$ we are now solving with $S = C - DQ^{-1}G$. To quantify the cost of this pressure solve we have to consider it in more detail. In both cases, we solve the pressure sub-system with a preconditioned Krylov subspace method up to a certain relative tolerance in n_p iterations. In both cases, we use the same preconditioner, ILU(0) based on the matrix $\tilde{S}$. Thus, the application of the preconditioner has the same cost. But preconditioned Krylov subspace methods further require the application of the matrix itself and this is where both approaches differ. In the first case, the matrix is readily available. In the second case, applying the matrix involves four steps: multiplication with G, three mom-u solves, multiplication by D and subtraction from C. If we neglect the multiplications and subtraction, we arrive at a total cost of

- coupled, Schur preconditioner: $n_x \times (3 \times (n_p + 1) \text{ mom-u} + 1 \text{ mass-p})$

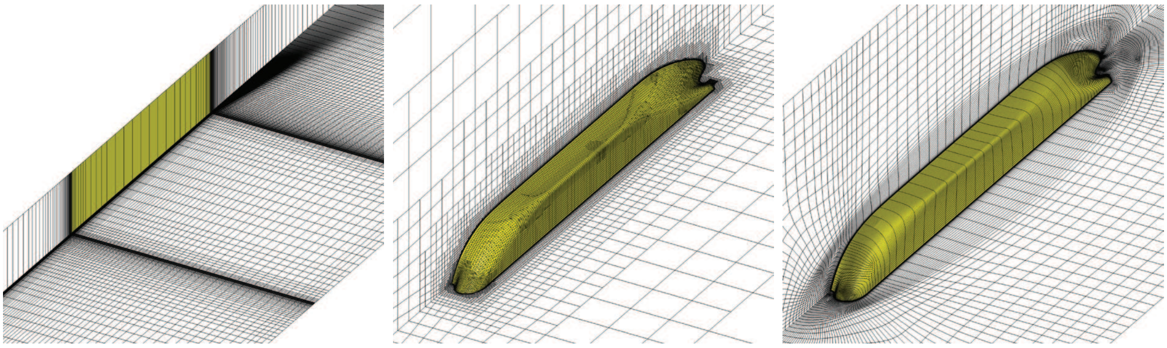

Figure 1: Impression of the grids. Flat-plate grid with 160×120 cells and max aspect ratio of order $1 : 10^4$ (left). Tanker grids, unstructured with 176K cells and max aspect ratio of order $1 : 10^2$ (middle) and structured with 256K cells and max aspect ratio of order $1 : 10^4$ (right).

per non-linear iteration. The $n_x \times n_p$ factor reflects the nesting of a momentum solve *within* the pressure solve. Clearly, this higher cost only pays-off if the number of linear iterations is significantly reduced compared to the SIMPLE preconditioner.

Thus, by quantifying the number of non-linear iterations and the number of Krylov iterations n_x and n_p in the following numerical experiments, we can already get an impression of the relative cost of the proposed solvers.

3 NUMERICAL EXPERIMENTS

The following numerical experiments were done with a pilot implementation in MARIN's CFD software package ReFRESCO [24].

3.1 Description of test cases

Flow over finite flat plate The friction resistance coefficient of an infinitely thin plate, expressed as function of the Reynolds number, is known as the friction line and plays an important role in the extrapolation of model-scale results to full scale. The flow over a finite flat plate therefore became a standard test case in maritime engineering. Numerical friction lines for various turbulence models were already proposed in [25] and compared to experimental lines such as the ITTC'57 line. Detailed results for ReFRESCO are found in [26], including a numerical uncertainty analysis with the procedure from [6]. Here, we will reconsider the fully turbulent flow at $\mathrm{Re} = 10^7$ on the same series of grids. These block-structured grids are refined near the leading and trailing edge of the plate and spread out in the wake of the plate, see Figure 1. Near the middle of the plate, the cells have an aspect ratio of order $1 : 10^4$.

Flow around model tanker Another important test case in maritime engineering is the flow around the Kriso Very Large Crude Carrier 2, see [27] for a summary of the results from literature and for a detailed study of various turbulence models with ReFRESCO. Results for different segregated and coupled solvers can be found in [17]. Here, we reconsider the model-scale case at Reynolds number $4.6 \cdot 10^6$ on two types of grids, structured and unstructured. The structured grid has 256K cells and the unstructured grid 176K cells, which is very coarse for this

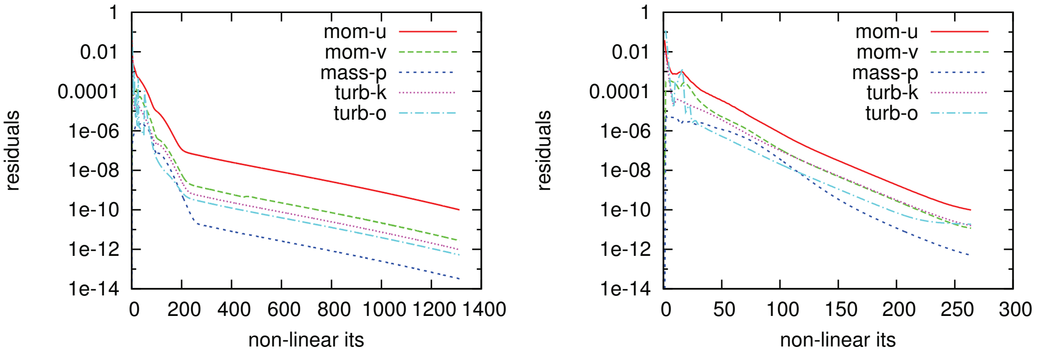

Figure 2: Non-linear convergence to steady-state for the flat plate on 80×60 grid. Segregated solver (left) and coupled solver (right).

application. The main difference is that the unstructured grid allows local refinement towards the hull using hanging nodes, see Figure 1. The structured grid has a maximum aspect ratio around $1:10^4$ and the boundary layer is fully resolved (max $y^+ \approx 1$) while the unstructured grid has a maximum aspect ratio around $1:10^2$ and part of the boundary layer is modeled with wall functions (max $y^+ \approx 350$).

3.2 Non-linear iteration

Figure 2 illustrates the benefit of the coupled solver: it requires less non-linear iterations to achieve a certain convergence criterion than the segregated solver. The same relaxation parameters were used for both solvers, so the difference in convergence is solely due to the coupled solution of the mass-momentum system. As shown in [17], this difference increases with grid refinement, leading to significant savings in CPU time on finer grids. The coupled solver also proved to be less sensitive to relaxation parameters. Note that the non-linear iterations are more expensive than those of a segregated solver, so the actual saving highly depends on the efficiency of the linear solver and preconditioner for the mass-momentum system, which is the main topic of this paper.

3.3 Linear iteration

Figure 3 shows an example of the convergence of FGMRES with the three different preconditioners discussed in Section 2.3. We see that FGMRES converges to machine accuracy in $n_x = 2$ iterations when the exact Schur preconditioner is used, as predicted by theory. When the approximate Schur preconditioner is used, $n_x = 6$ iterations are needed to reach machine accuracy for the flat plate, $n_x = 12$ for the tanker on the unstructured grid and $n_x = 13$ on the structured grid. The SIMPLE preconditioner does not converge very well in comparison, it only achieves one or two orders of convergence in $n_x = 20$ iterations. Clearly, in terms of linear iterations for the mass-momentum system, there is a lot to be gained by using better approximations than SIMPLE.

However, according to the cost model from Section 2.4 we should also consider the linear iterations of the pressure sub-system involved. For each iteration, the exact Schur preconditioner

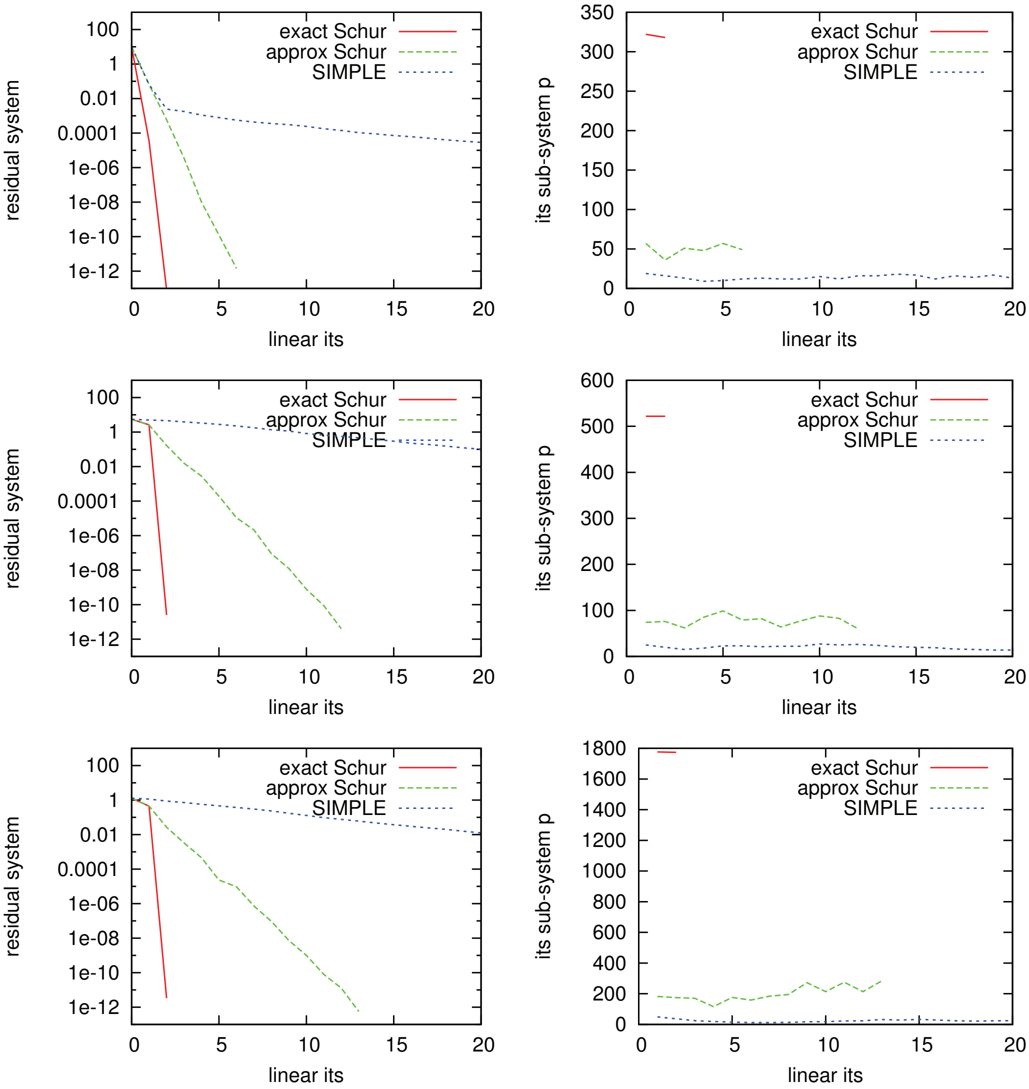

Figure 3: Linear convergence of the coupled mass-momentum system at the 10th non-linear iteration with three preconditioners (left) and corresponding number of iterations needed to solve the pressure sub-system (right). Flat plate on 80×60 grid (top), tanker on unstructured grid (middle) and on structured grid (bottom).

(which solves sub-systems to relative tolerance 10^{-12}) needs more than $n_p = 300$ iterations to solve the pressure sub-system for the flat plate, more than $n_p = 500$ for the tanker on the unstructured grid and more than $n_p = 1700$ for the structured grid. The approximate Schur preconditioner (which solves sub-systems to relative tolerance 10^{-2}) needs about $n_p = 50$ for the flat plate, $n_p = 80$ for the tanker on the unstructured grid and $n_p = 200$ for the structured grid. The SIMPLE preconditioner (which solves sub-systems to relative tolerance 10^{-2} and approximates the Schur complement) needs less than $n_p = 25$ for all cases. Clearly, the reduced number of iterations for the mass-momentum system obtained with the approximate Schur preconditioner comes with an increased number of iterations for the sub-systems.

So far, we considered the solution of the mass-momentum system arising at the tenth non-linear iteration. In the context of a non-linear solver, it is often not necessary to solve this system very accurately and a relative tolerance of 0.01 is already sufficient [17]. Therefore, we continue with this tolerance and considering the first one-hundred non-linear iterations. Figure 4 shows the number n_x of linear iterations needed to solve the mass-momentum system at each non-linear iteration and the average number of iterations n_p needed to solve the pressure sub-system.

Considering that the test cases vary from 2D to 3D, from structured to unstructured and from moderate to high aspect ratio's, we find the number n_x to be remarkably constant: around 5 for the approximate Schur preconditioner and around 20 for the SIMPLE preconditioner. The number n_p is less constant and depends both on the case and on the non-linear iteration. Extreme differences are found for the tanker on the structured grid: in the first forty non-linear iterations n_p is around 140 for the approximate Schur preconditioner and around 25 for the SIMPLE preconditioner, then these numbers drop to around 60 and 10, respectively. This makes it hard to draw any definite conclusions. Nevertheless, if we assume typical values for the approximate Schur preconditioner to be $n_x = 5$ and $n_p = 50$ we get a total cost per non-linear iteration of

- coupled, approx Schur preconditioner: 765 mom-u + 5 mass-p

while the SIMPLE preconditioner with typical value of $n_x = 20$ gives a total cost of

- coupled, SIMPLE preconditioner: 60 mom-u + 20 mass-p

From these numbers, we tentatively conclude that the reduction of iterations n_x achieved with the approximate Schur preconditioner is not nearly enough to compensate for the momentum solve needed at every iteration of the pressure solve.

One could also conclude that the SIMPLE approximation $\tilde{S}$ is a poor preconditioner for the true Schur complement S: a better preconditioner might reduce n_p and shift the cost balance. Furthermore, note that all cases presented here were done on a single process. In parallel, the preconditioner is built from $\tilde{S}$ using ILU(0) per sub-domain and Jacobi between sub-domains, which leads to a dramatic increase of n_p. A possible explanation is that $\tilde{S}$ is sparse, whereas S, if it were constructed, would be dense. With parallelization using ILU(0) per sub-domain and Jacobi between sub-domains, the preconditioner built from $\tilde{S}$ becomes even more sparse, which could explain the significant loss in performance when used for solving dense S. This drop in performance has prevented us from presenting thorough scalability studies.

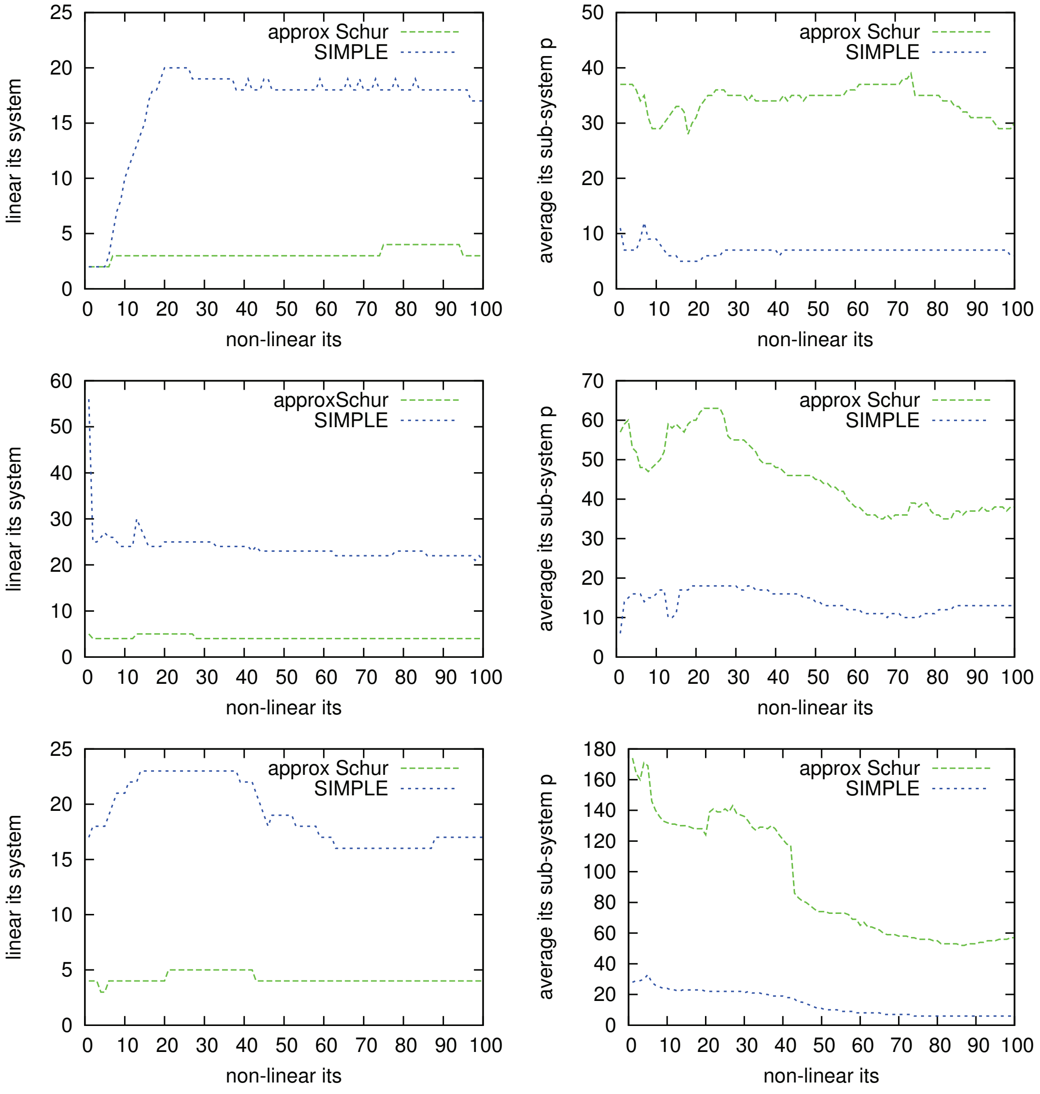

Figure 4: Linear iterations needed to solve the mass-momentum system with relative tolerance 0.01 (left) and average number of iterations needed to solve the pressure sub-system (right). Flat plate on 80×60 grid (top) and tanker on unstructured grid (middle) and on structured grid (bottom).

4 DISCUSSION

We have compared the behavior of three preconditioners for the solution of the mass-momentum system that arises from the finite-volume discretization with co-located variables of the steady, incompressible RaNS equations, using three test cases that are representative of maritime engineering.

These preconditioners are readily available in PETSc and differ only in the approximation of the Schur complement and in the relative tolerance of the sub-system solves. The main idea was to keep the true Schur complement in the pressure sub-system and use the SIMPLE-type approximation to build its preconditioner. This gives much better performance than replacing the true Schur complement altogether, at least when expressed in linear iterations for the mass-momentum system.

However, the total cost in terms of linear iterations for the sub-systems turns out to be an order higher for the three considered cases. The main reason for this difference is that each pressure sub-system iteration with the true Schur complement involves the solution of the momentum sub-system. For our test cases, this easily leads to hundreds of momentum solves within each pressure solve. Besides, the SIMPLE-type approximation turns out not to be a very good preconditioner for the true Schur complement, especially in combination with block Jacobi parallelization. This has prevented us from presenting grid studies and scalability results in this paper. Future research should therefore aim at understanding the exact causes of these problems.

ACKNOWLEDGEMENT

This research is partly funded by the Dutch Ministry of Economic Affairs.

REFERENCES

[1] S.V. Patankar. *Numerical heat transfer and fluid flow.* McGraw-Hill, New York, 1980.

[2] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics.* Springer, London, 3rd edition, 2002.

[3] P. Wesseling. *Principles of Computational Fluid Dynamics.* Springer, Heidelberg, 2001.

[4] P.J. Roache. *Fundamentals of Verification and Validation.* Hermosa Publishers, Albuquerque, New Mexico, 2009.

[5] L. Eça and M. Hoekstra. Evaluation of numerical error estimation based on grid refinement studies with the method of the manufactured solutions. *Computers and Fluids*, 38:1580–1591, 2009.

[6] L. Eça and M. Hoekstra. A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. *Journal of Computational Physics*, 262:104–130, 2014.

[7] S.P. Vanka. Block-implicit multigrid calculation of two-dimensional recirculating flows. *Computer Methods in Applied Mechanics and Engineering*, 59(1):29–48, 1986.

[8] M. Raw. Robustness of coupled algebraic multigrid for the navier-stokes equations. *AIAA Paper 96-0297*, 1996.

[9] M. Darwish, I. Sraj, and F. Moukalled. A coupled finite volume solver for the solution of incompressible flows on unstructured grids. *Journal of Computational Physics*, 228(1):180–201, 2009.

[10] A. van der Ploeg, L. Eça, and M. Hoekstra. Combining accuracy and efficiency with robustness in ship stern flow computation. In *Proceedings of the 23rd Symposium on Naval Hydrodynamics*, Val de Reuil, France, September 17-22 2000.

[11] H. Elman, V.E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible navier-stokes equations. *J. Comput. Phys.*, 227:1790–1808, 2008.

[12] G.B. Deng, J. Piquet, X. Vasseur, and M. Visonneau. A new fully coupled method for computing turbulent flows. *Computers & Fluids*, 30(4):445–472, 2001.

[13] A.C. de Niet and F.W. Wubs. Two preconditioners for saddle point problems in fluid flows. *Int. J. Numer. Meth. Fluids*, 54(4):355–377, 2007.

[14] B.E. Griffith. An accurate and efficient method for the incompressible navierstokes equations using the projection method as a preconditioner. *Journal of Computational Physics*, 228(20):7565–7595, 2009.

[15] M. Cai, A. Nonaka, J.B. Bell, B.E. Griffith, and A. Donev. Efficient variable-coefficient finite-volume stokes solvers. *Communications in Computational Physics*, 16(5):1263–1297, 2014.

[16] C. Vuik, A. Saghir, and G.P. Boerstoel. The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces. *International Journal for Numerical Methods in Fluids*, 33:1027–1040, 2000.

[17] C.M. Klaij and C. Vuik. SIMPLE-type preconditioners for cell-centered, colocated finite volume discretization of incompressible Reynolds-averaged Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 71(7):830–849, 2013.

[18] C.M. Klaij. On the stabilization of finite volume methods with co-located variables for incompressible flow. *Journal of Computational Physics*, 297:84–89, 2015.

[19] M. Benzi and M.A. Olshanskii. An augmented Lagrangian-based approach to the Oseen problem. *SIAM Journal on Scientific Computing 2006*, 28(6):2095–2113, 2006.

[20] X. He, C. Vuik, and C.M. Klaij. Block-preconditioners for the incompressible Navier-Stokes equations discretized by a finite volume method. *Journal of Numerical Mathematics*, 2016.

[21] J. Brown, M.G. Knepley, D.A. May, L.C. McInnes, and B.F. Smith. Composable linear solvers for multiphysics. In *Proceedings of the 11th International Symposium on Parallel and Distributed Computing (ISPDC 2012)*, pages 55–62. IEEE Computer Society, 2012.

[22] N.P. Waterson and H. Deconinck. Design principles for bounded higher-order convection schemes – a unified approach. *J. Comput. Phys.*, 224:182–207, 2007.

[23] M.F. Murphy, G.H. Golub, and A.J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing*, 21(6):1969–1972, 2000.

[24] Maritime Research Institute Netherlands. ReFRESCO Web page. `http://www.refresco.org`, 2015.

[25] L. Eça and M. Hoekstra. The numerical friction line. *Journal of Marine Science and Technology*, 13(4):328–345, 2008.

[26] D.R. Rijpkema. Flat plate in turbulent flow. Technical Report 23279-1-RD, Maritime Research Institute Netherlands, 2009.

[27] F.S. Pereira, L. Ea, and G. Vaz. Verification and validation exercises for the flow around the KVLCC2 tanker at model and full-scale Reynolds numbers. *Ocean Engineering*, 129:133–148, 2017.