

Article

Numerical Solution of the Navier–Stokes Equations Using Multigrid Methods with HSS-Based and STS-Based Smoother

Galina Muratova ^{1,*}, Tatiana Martynova ^{1,†}, Evgeniya Andreeva ^{1,‡}, Vadim Bavin ^{1,†} and Zeng-Qi Wang ²

¹ Mechanics and Computer Science, Vorovich Institute of Mathematics, Southern Federal University, Rostov-on-Don 344000, Russia; martynova@sfedu.ru (T.M.); andreeva@sfedu.ru (E.A.); tefp9999@mail.ru (V.B.)

² School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200000, China; wangzengqi@sjtu.edu.cn

* Correspondence: muratova@sfedu.ru; Tel.: +7-863-219-9736

† Current address: 200/1 Stachki Ave., Bld. 2, Rostov-on-Don 344090, Russia.

‡ These authors contributed equally to this work.

Received: 29 November 2019; Accepted: 21 January 2020; Published: 4 February 2020



Abstract: Multigrid methods (MGMs) are used for discretized systems of partial differential equations (PDEs) which arise from finite difference approximation of the incompressible Navier–Stokes equations. After discretization and linearization of the equations, systems of linear algebraic equations (SLAEs) with a strongly non-Hermitian matrix appear. Hermitian/skew-Hermitian splitting (HSS) and skew-Hermitian triangular splitting (STS) methods are considered as smoothers in the MGM for solving the SLAE. Numerical results for an algebraic multigrid (AMG) method with HSS-based smoothers are presented.

Keywords: multigrid methods; Hermitian/skew-Hermitian splitting method; skew-Hermitian triangular splitting method; strongly non-Hermitian matrix

1. Introduction

Mathematical modeling of hydrodynamics is the base for research of various natural phenomena, technological processes, and environmental problems. The main equations describing this problem are the Navier–Stokes equations. Development and research of effective numerical algorithms for solving these equations and their practical realization is an actual task. The use of the MGM for the numerical solution of the Navier–Stokes equations describing the motion of an incompressible viscous fluid is discussed. Currently, various discretization methods for the corresponding differential model are known. However, with any choice of the discretizing method, the problem of constructing effective methods for solving large systems of algebraic equations—to which the discrete model is reduced—arises. This problem is especially relevant in the nonstationary case, when multiple solutions of the systems of algebraic equations are required at each discrete time step.

To discretize the system of two-dimensional Navier–Stokes equations on regular grids, we use the finite difference method. The equations are considered in the natural variables “velocity–pressure”:

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\nabla P + \nu \Delta \mathbf{V}, \quad \operatorname{div} \mathbf{V} = 0, \quad (1)$$

where P/ρ is replaced by P (i.e., ρ is normalized at 1), P is the static pressure, \mathbf{V} is the velocity vector, and ν is the kinematic viscosity coefficient. At the initial moment of time and at the boundary of the domain, the initial and boundary conditions are set, respectively.

Flow simulating is accompanied by a number of mathematical difficulties. One of the problems in solving this system is the nonlinearity associated with convective terms in the equations, which can lead to the appearance of oscillations of the solution in regions with large gradients. The main efforts of the researchers were directed at overcoming the difficulties associated with the nonlinearity of the Navier–Stokes system of equations.

One of the most time-consuming stages of the computational procedure is finding the solution of the system of linear algebraic equations (SLAE). Modern application packages usually use the linearization of the original equations, and Krylov subspace methods are used to solve the resulting SLAEs. Despite the fact that these methods have proven themselves well, they have some problems in cases of significant nonsymmetry of the SLAEs—associated, for example, with variable coefficients in differential equations or using complex numerical boundary conditions. For time discretization of the unsteady problem, we use an implicit difference scheme. Here, we do not specifically consider the stages of discretization and linearization of the Navier–Stokes equations, but focus on solving SLAEs. Given that the SLAEs resulting from the use of the implicit time schemes have a large dimension and a sparse nonsymmetric matrix, we propose using the MGM to solve them.

Thus, we consider the iterative solution of the large sparse SLAE

$$Av = b, \quad v, b \in \mathbb{C}^n, \quad (2)$$

where $A \in \mathbb{C}^{n \times n}$ is a non-Hermitian and positive definite matrix.

Naturally, the matrix A can be split as

$$A = A_0 + A_1, \quad (3)$$

where

$$A_0 = \frac{1}{2}(A + A^*), \quad A_1 = \frac{1}{2}(A - A^*) \quad (4)$$

and A^* denotes the conjugate transpose of the matrix A . Positive definiteness of the matrix A means that for all $x \in \mathbb{C}^n \setminus \{0\}$, $x^* A_0 x > 0$. Here, x^* denotes the conjugate transpose of the complex vector x . Let in some matrix norm $\|\cdot\|$, $\|A_0\| \ll \|A_1\|$, then the matrix A is called a strongly non-Hermitian one. This situation occurs in many real applications, such as the discretization of the Navier–Stokes equations.

The Hermitian and skew-Hermitian splitting (HSS) iteration methods, based on HS splitting (3) and (4), for solving large sparse non-Hermitian positive definite SLAE were firstly proposed in [1]. The HSS iteration method has been widely developed in [2–5] and others.

Then, we can split the skew-Hermitian part A_1 of the matrix $A \in \mathbb{C}^{n \times n}$ into

$$A_1 = K_L + K_U, \quad (5)$$

where K_L and K_U are the strictly lower and the strictly upper triangular parts of A_1 , respectively. Obviously, that $K_L = -K_U^*$.

Based on the splitting (3)–(5) in [6–8] classes of skew-Hermitian triangular splitting (STS), iteration methods for solving SLAE (2) have been proposed. The triangular operator of the STS uses only the skew-Hermitian part of the coefficient matrix A . These methods have been further developed in [9–12].

The use of the multigrid method (MGM) with the STS-based smoothers for solving convection–diffusion problems has been studied in [13]. The convergence of the MGM with the STS-based smoothers has also been proved in this research. The local Fourier analysis of the MGM with the triangular skew-symmetric smoothers has been performed in [14]. The results of numerical

experiments for convection–diffusion problems with large Peclet numbers by the geometric MGM have been presented in both researches.

In [15], it was shown that the MGMs with the HSS-based smoothers converge uniformly for second-order nonselfadjoint elliptic boundary value problems. This happens if the mesh size of the coarsest grid is sufficiently small, but independent of the number of the multigrid levels.

2. Multigrid Methods

The MGMs are proving themselves as very successful tools for solving the SLAE associated with discretization of partial differential equations (PDEs).

The main idea of the MGM has been proposed by R.P. Fedorenko in [16]. Then, A. Brandt [17], W. Hackbusch [18], and other researchers showed the efficiency of the multigrid approach and extended Fedorenko's idea.

The multigrid technique is based on two principles: error smoothing and coarse grid correction. The smoothing property is fundamental for the MGM. It is connected with fast damping high-frequency Fourier components of an initial error in decomposition on the basis from eigenvectors.

There exist two approaches in the MGM: geometric multigrid and algebraic multigrid methods.

Geometric multigrid methods were critical to the early development of the MGM and still play an important role today. Nevertheless, there are classes of problems for which geometric techniques are too difficult to apply or cannot be used at all. These classes can be solved by the algebraic multigrid (AMG) methods, as introduced in [19,20].

The MGM is not a fixed algorithm. Rather, there is a multigrid technique that defines its scope. The efficiency of the MGM depends on the adjustment of its components to the considered problem [21]. The key to this is the correct choice of its components and effective interaction between smoothing and coarse-grid correction [22]. We need to use special iteration methods as smoothers for the MGM and nonstandard coarse-grid correction to a good approximation of the smooth error components.

The smoothing method is the central component of the multigrid algorithm; it is the most dependent part of the MGM on the type of the problem being solved. The role of smoothing methods is that they should not so much reduce the total error as smooth it (namely, suppress the high-frequency harmonics of the error) so that the error can be well approximated on a coarse grid.

Standard smoothing methods are linear iteration methods, for example, the Gauss–Seidel method. An alternative is the following methods:

- Richardson's Iterative method;
- Gauss–Jacobi method;
- Symmetric Gauss–Seidel method;
- Gauss–Seidel Alternate Direction method;
- Gauss–Seidel method with black and white ordering;
- Four-color Gauss–Seidel method;
- Iteration zebra method;
- Incomplete factorization method;
- Specially adapted SOR.

The MGMs can be used as solvers as well as preconditioners. The MGMs have been widely used for complicated nonsymmetric and nonlinear systems, like the Lamé equations of elasticity or the Navier–Stokes problems.

3. Smoothers Based on the HSS and the STS Iteration Methods

A particular problem when using the MGM is the choice of smoothers. There are a number of iteration methods that can be used as smoothers, but not all of them are effective for solving strongly non-Hermitian SLAEs. The behavior of the HSS and the STS iteration methods is similar to the behavior of the Gauss–Seidel method, which quickly damps the high-frequency harmonics of the error, slowing down in the future. We give the formulas of these iteration methods.

The HSS iteration method [1]: Given an initial guess $v^{(0)}$, for $k = 0, 1, 2, \dots$ until $\{v^{(k)}\}$ convergence, compute

$$\begin{cases} (\alpha I + A_0)v^{(k+\frac{1}{2})} = (\alpha I - A_1)v^{(k)} + b, \\ (\alpha I + A_1)v^{(k+1)} = (\alpha I - A_0)v^{(k+\frac{1}{2})} + b, \end{cases}$$

where α is a given positive constant and I is an identity matrix.

Bai, Golub, and Ng [1] proved that the HSS iteration method converges unconditionally to the exact solution of the SLAE (2). Moreover, the upper bound of the contraction factor depends on the spectrum of A_0 but is independent of the spectrum of A_1 .

We can rewrite the HSS iteration method in the following form:

$$v^{(k+1)} = G(\alpha)v^{(k)} + B(\alpha)^{-1}b,$$

where

$$G(\alpha) = B(\alpha)^{-1}(B(\alpha) - A)$$

and

$$B(\alpha) = \frac{1}{2\alpha}(\alpha I + A_0)(\alpha I + A_1).$$

The STS iteration method [6,8]: Given an initial guess $v^{(0)}$ and two positive parameters ω and τ . For $k = 0, 1, 2, \dots$ until $\{v^{(k)}\}$ convergence, compute

$$v^{(k+1)} = G(\omega, \tau)v^{(k)} + \tau B(\omega)^{-1}b,$$

where

$$G(\omega, \tau) = B(\omega)^{-1}(B(\omega) - \tau A),$$

ω and τ are two acceleration parameters, and $B(\omega)$ is defined by

$$B(\omega) = B_c + \omega((1+j)K_L + (1-j)K_U), \quad j = \pm 1$$

with $B_c \in \mathbb{C}^{n \times n}$ a prescribed Hermitian matrix.

For the STS method a convergence analysis, optimal choice of parameters and an accelerating procedure have presented in [8]. As it was mentioned above, smoothers in the MGMs should have a smoothing effect on the error of approximation. It was shown in [14] that the skew-Hermitian triangular iteration methods have such properties. Therefore, these methods can be used as smoothers in the MGMs.

4. Numerical Experiments

A wide class of CFD (Computational Fluid Dynamics) problems is associated with solving the equations of motion of a viscous incompressible fluid with a predominance of convective transfer. As a model, we consider the problem of internal single-phase chemically homogeneous flows, which are described by the unsteady Navier–Stokes equations in the domain Ω with a solid boundary Γ . At the initial stages of the development of CFD, preference was given to explicit methods that were used to solve stationary and nonstationary Navier–Stokes equations. Recently, increased attention has been paid to implicit methods. This is primarily due to the insufficient computational efficiency of explicit methods in solving the equations of motion of a viscous fluid using small difference grids. From the

point of view of computational linear algebra, the matrices obtained at each time step when integrating unsteady equations using implicit schemes (after linearization) are nonselfadjoint and require special iterative methods for their effective solution. Therefore, in this research, we suggest using the AMG with special smoothers to solve such SLAEs.

So, we consider the model unsteady Navier–Stokes problem

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\nabla P + \nu \Delta \mathbf{V}, \quad (6)$$

$$\operatorname{div} \mathbf{V} = 0, \quad (7)$$

or

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial P}{\partial x} - \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0, \quad (8)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial P}{\partial y} - \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) = 0, \quad (9)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (10)$$

$$\begin{aligned} u(x, y, t) &= 0, & v(x, y, t) &= 0 \quad \text{on } \Gamma, \\ u(x, y, 0) &= 0, & v(x, y, 0) &= 0, \\ P(x, y, 0) &= \zeta x - \frac{1}{2} \zeta, & \zeta &= \text{const}, \end{aligned}$$

where ν is the kinematic viscosity coefficient; $Re = UL/\nu$ is the Reynolds number, where U is a characteristic velocity of the flow and L is a characteristic length scale; $\mathbf{V} = (u(x, y, t), v(x, y, t))$ is the velocity vector; P is the static pressure; the initial pressure distribution is given by a linear function. The initial conditions are taken to be zero. At the boundary, no-slip conditions are accepted. It means that at a solid boundary, the fluid will have zero velocity relative to the boundary. There are no mass forces in the formulation; motion is determined only by the boundary and initial conditions for the velocity field as well as the initial pressure distribution. For convenience, only square domain $\Omega = (0, 1) \times (0, 1)$ will be considered. We assume that the fluid motion occurs in the time interval $[0, T]$. Therefore, the equations are considered in the domain $\Omega \times (0, T)$ with the boundary $\Gamma \times [0, T]$. The Navier–Stokes equations with the introduced boundary conditions have a solution determined up to an arbitrary constant for pressure, therefore, an agreement was adopted on the next normalization $\int_{\Omega} P(x, y, t) dx dy = 0, \forall t$.

The most common approach to solving the Navier–Stokes equations in natural variables essentially uses the replacement of the difference continuity equation by the difference Poisson equation for pressure. Following this approach, first the difference equations are constructed that approximate the mass and momentum conservation equations and then, by algebraic transformations, the Poisson equation for determining the pressure is derived. This equation is used in the calculations instead of the continuity equation.

First, the equations of motion and continuity (6) and (7) are rewritten in schematic form [23]:

$$\frac{\partial \mathbf{V}}{\partial t} + \nabla P = \mathbf{R}, \quad (11)$$

where \mathbf{R} contains all convective and diffusive forces,

$$\mathbf{R} = -(\mathbf{V} \cdot \nabla) \mathbf{V} + \frac{1}{Re} \Delta \mathbf{V}, \quad (12)$$

$$\operatorname{div} \mathbf{V} = 0. \quad (13)$$

We fix the time step δt and introduce a discrete time grid $t^n = n\delta t$, $n \geq 0$ and denote the approximation to $f(t^n, x, y)$ as $f^{(n)}$. Then, the fully implicit scheme will have the form

$$\frac{1}{\delta t}(\mathbf{V}^{(n+1)} - \mathbf{V}^{(n)}) + \nabla P^{(n+1)} = \mathbf{R}^{(n+1)}, \quad (14)$$

$$\mathbf{R}^{(n+1)} = -(\mathbf{V}^{(n+1)} \cdot \nabla)\mathbf{V}^{(n+1)} + \frac{1}{Re}\Delta\mathbf{V}^{(n+1)}, \quad (15)$$

$$div\mathbf{V}^{(n+1)} = 0, \quad (16)$$

$$\mathbf{V}^{(n+1)}|_{\Gamma} = 0. \quad (17)$$

The Poisson equation for pressure is obtained by taking the divergence from both sides of the Equation (14), taking into account the continuity Equation (16):

$$\Delta P^{(n+1)} - div\mathbf{R}^{(n+1)} = div\frac{\mathbf{V}^{(n)}}{\delta t}. \quad (18)$$

But following [23], at this moment, the Poisson Equation (18) does not need to be created. Instead, we need to do a discretization. In addition, the continuity Equation (16) is first discretized before substituting the discrete version of (14). To approximate the problem in space, the finite difference method is used. Let the equations in discrete form be given by

$$D_h\mathbf{V}_h^{(n+1)} = 0, \quad (19)$$

$$\frac{1}{\delta t}(\mathbf{V}_h^{(n+1)} - \mathbf{V}_h^{(n)}) + G_h P_h^{(n+1)} = \mathbf{R}_h^{(n+1)}, \quad (20)$$

$$\mathbf{V}_h^{(n+1)}|_{\Gamma} = 0, \quad (21)$$

where D_h and G_h are the discrete div and ∇ operator, respectively. Then, \mathbf{V}_h , P_h and \mathbf{R}_h are the discrete grid functions corresponding with \mathbf{V} , P , and \mathbf{R} . After discretization of (16), the number of velocity unknowns equals the number of discrete momentum equations. The number of pressure unknowns is equal to the number of discrete continuity equations, since both are equal to the number of grid cells [23]. Our approach uses the idea of [23], but it differs in implementation.

The uniform grid Ω is introduced in the domain Ω with steps h_1 and h_2 ; $h_1 = 1/N_1$, $h_2 = 1/N_2$, where N_1 , N_2 are the number of cells in each direction. The grid cells are positioned such that the cell faces coincide with the boundary Γ of Ω . The discretization in space of the Navier–Stokes equations is performed on MAC (Marker and Cell) [24] (staggered) grids when pressure P and velocities in two-dimensional problems are determined on three grids shifted relative to each other. So, P is located in the center of each cell, the x -component velocity u is on the middle points of vertical faces, the y -component velocity v is on the middle points of horizontal faces. For the MAC-method, the solution advanced in time by solving the momentum equation with the best current estimate of pressure distribution. Such a solution initially would not satisfy the continuity equation unless the correct pressure distribution was used. The pressure is improved by numerically solving the Poisson equation with estimated velocity field. We rewrite the equation for pressure in the following form:

$$\begin{aligned} \Delta P = & \frac{d}{dx} \left(- \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right) + \\ & + \frac{d}{dy} \left(- \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \right) - \frac{\partial}{\partial t} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right). \end{aligned} \quad (22)$$

We now introduce the grid sets and the corresponding spaces:

$$\bar{D}_1 = \{x_{ij} = ((i + 1/2)h_1, jh_2) : i = 0, \dots, N_1 - 1, j = 0, \dots, N_2\},$$

$$\bar{D}_2 = \{x_{ij} = (ih_1, (j + 1/2)h_2) : i = 0, \dots, N_1, j = 0, \dots, N_2 - 1\},$$

$$D_3 = \{x_{ij} = (ih_1, jh_2) : i = 1, \dots, N_1 - 1, j = 1, \dots, N_2 - 1\}.$$

Let $V_h = V_{1,h} \times V_{2,h}$ be the linear space of vector functions defined on $\bar{D}_1 \times \bar{D}_2$ and vanishing at the corresponding grid boundaries, and P_h is the space of functions defined on D_3 and orthogonal to unity. Thus,

$$V_{1,h} = \{u_{ij} = u(x_{ij}) : x_{ij} \in \bar{D}_1, u_{0,j} = u_{N_1-1,j} = u_{i,0} = u_{i,N_2} = 0\},$$

$$V_{2,h} = \{v_{ij} = v(x_{ij}) : x_{ij} \in \bar{D}_2, v_{0,j} = v_{N_1,j} = v_{i,0} = v_{i,N_2-1} = 0\},$$

$$P_h = \{P_{ij} = P(x_{ij}) : x_{ij} \in D_3, \sum_{ij} h_1 h_2 P_{ij} = 0\}.$$

Variables are denoted by a single set of indices, despite the fact that different variables are calculated at different grid nodes. As a result, the indices i, j refer to a set of three mismatched points.

The term $R^{(n+1)}$ in (15) contains the nonlinear terms. So, for treating this nonlinearity, Newton linearization around the old time level is used. For example, we want to linearize a nonlinear term $u^{(n+1)}\phi_x^{(n+1)}$, then

$$u^{(n+1)}\phi_x^{(n+1)} = u^{(n)}\phi_x^{(n+1)} + u^{(n+1)}\phi_x^{(n)} - u^{(n)}\phi_x^{(n)} + O(\delta t^2). \tag{23}$$

The expression in the right-hand side of (23) is linear in the variables at the new time level and possesses a discretization error $O(\delta t^2)$.

Let $D = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ in (22) be the local dilation term, and other terms with velocity field determined from the solution of momentum equation with a provisional estimate of pressure $P' = \begin{pmatrix} f'_1 \\ f'_2 \end{pmatrix}$, counter, and $D_{ij}^{(n+1)}$ be set equal to zero. That is, the correction of pressure is required to compensate for nonzero dilation at the n iterative level. The Poisson equation is then solved for the revised pressure field. The improved pressure is then used in the momentum equation for better solution at time step. If the dilation (divergence of velocity field) is not zero, the cyclic process of solving the momentum equation and Poisson equation is repeated until the velocity field is divergence free.

Thus, our computational scheme can be represented as follows:

1. Velocity field components $u' = u^{(n+1)}$ and $v' = v^{(n+1)}$ are determined by solving the implicit momentum equation with P' , and for treating nonlinearity, the Newton linearization around the old time level is used.

$$\begin{aligned} & \frac{u_{ij}^{(n+1)} - u_{ij}^{(n)}}{\delta t} + \left(u_{ij}^{(n)} \left(\frac{u_{ij}^{(n+1)} - u_{i-1,j}^{(n+1)}}{h_1} \right) + u_{ij}^{(n+1)} \left(\frac{u_{ij}^{(n)} - u_{i-1,j}^{(n)}}{h_1} \right) - u_{ij}^{(n)} \left(\frac{u_{ij}^{(n)} - u_{i-1,j}^{(n)}}{h_1} \right) \right) + \\ & + \left(v_{ij}^{(n)} \left(\frac{u_{ij}^{(n+1)} - u_{i,j-1}^{(n+1)}}{h_2} \right) + v_{ij}^{(n+1)} \left(\frac{u_{ij}^{(n)} - u_{i,j-1}^{(n)}}{h_2} \right) - v_{ij}^{(n)} \left(\frac{u_{ij}^{(n)} - u_{i,j-1}^{(n)}}{h_2} \right) \right) - \\ & - \frac{1}{Re} \left(\frac{u_{i+1,j}^{(n+1)} - 2u_{ij}^{(n+1)} + u_{i-1,j}^{(n+1)}}{h_1^2} + \frac{u_{i,j+1}^{(n+1)} - 2u_{ij}^{(n+1)} + u_{i,j-1}^{(n+1)}}{h_2^2} \right) = f'_1, \end{aligned} \tag{24}$$

$$\begin{aligned} & \frac{v_{ij}^{(n+1)} - v_{ij}^{(n)}}{\delta t} + \left(u_{ij}^{(n)} \left(\frac{v_{ij}^{(n+1)} - v_{i-1,j}^{(n+1)}}{h_1} \right) + u_{ij}^{(n+1)} \left(\frac{v_{ij}^{(n)} - v_{i-1,j}^{(n)}}{h_1} \right) - u_{ij}^{(n)} \left(\frac{v_{ij}^{(n)} - v_{i-1,j}^{(n)}}{h_1} \right) \right) + \\ & + \left(v_{ij}^{(n)} \left(\frac{v_{ij}^{(n+1)} - v_{i,j-1}^{(n+1)}}{h_2} \right) + v_{ij}^{(n+1)} \left(\frac{v_{ij}^{(n)} - v_{i,j-1}^{(n)}}{h_2} \right) - v_{ij}^{(n)} \left(\frac{v_{ij}^{(n)} - v_{i,j-1}^{(n)}}{h_2} \right) \right) - \quad (25) \\ & - \frac{1}{Re} \left(\frac{v_{i+1,j}^{(n+1)} - 2v_{ij}^{(n+1)} + v_{i-1,j}^{(n+1)}}{h_1^2} + \frac{v_{i,j+1}^{(n+1)} - 2v_{ij}^{(n+1)} + v_{i,j-1}^{(n+1)}}{h_2^2} \right) = f_2', \end{aligned}$$

$$\frac{u_{ij}^{(n+1)} - u_{i-1,j}^{(n+1)}}{h_1} + \frac{v_{ij}^{(n+1)} - v_{i,j-1}^{(n+1)}}{h_2} = 0. \quad (26)$$

2. The Poisson equation with estimated velocity field components $u' = u^{(n+1)}$ and $v' = v^{(n+1)}$ is solved for the revised pressure field $P = P^{(n+1)}$.

$$\begin{aligned} & \frac{1}{h_1^2} \left(P_{i+1,j}^{(n+1)} - 2P_{ij}^{(n+1)} + P_{i-1,j}^{(n+1)} \right) + \frac{1}{h_2^2} \left(P_{i,j+1}^{(n+1)} - 2P_{ij}^{(n+1)} + P_{i,j-1}^{(n+1)} \right) = \\ & = \frac{1}{\delta t} \left(\frac{u_{ij}^{(n+1)} - u_{i-1,j}^{(n+1)}}{h_1} + \frac{v_{ij}^{(n+1)} - v_{i,j-1}^{(n+1)}}{h_2} \right) + \\ & + \frac{1}{h_1} \left(-u_{ij}^{(n+1)} \left(\frac{v_{ij}^{(n+1)} - v_{i-1,j}^{(n+1)}}{h_1} \right) - v_{ij}^{(n+1)} \left(\frac{v_{ij}^{(n+1)} - v_{i,j-1}^{(n+1)}}{h_2} \right) \right) + \\ & + \frac{1}{Re h_1} \left(\frac{u_{i+1,j}^{(n+1)} - 2u_{ij}^{(n+1)} + u_{i-1,j}^{(n+1)}}{h_1^2} + \frac{u_{i,j+1}^{(n+1)} - 2u_{ij}^{(n+1)} + u_{i,j-1}^{(n+1)}}{h_2^2} \right) + \\ & + \frac{1}{h_2} \left(-u_{ij}^{(n+1)} \left(\frac{v_{ij}^{(n+1)} - v_{i-1,j}^{(n+1)}}{h_1} \right) - v_{ij}^{(n+1)} \left(\frac{v_{ij}^{(n+1)} - v_{i,j-1}^{(n+1)}}{h_2} \right) \right) + \\ & + \frac{1}{Re h_2} \left(\frac{v_{i+1,j}^{(n+1)} - 2v_{ij}^{(n+1)} + v_{i-1,j}^{(n+1)}}{h_1^2} + \frac{v_{i,j+1}^{(n+1)} - 2v_{ij}^{(n+1)} + v_{i,j-1}^{(n+1)}}{h_2^2} \right). \end{aligned}$$

Revised velocity field components u and v are determined by solving the implicit momentum equation with revised pressure P . Process of solving the momentum equation and Poisson equation is repeated until the velocity field is divergence free. Thus, at each time step in solving the Navier–Stokes equation, we need to solve SLAE with nonsymmetric matrices that are solved by the AMG method with HSS smoothers.

There are two coarsening approaches in the AMG: RS and PMIS algorithms. Coarsening splits initial grid on C-points and F-points—coarse and fine grid points, respectively. The RS (Ruge-Stuben) algorithm [25] is a traditional coarsening approach. The RS algorithm is based on two heuristic criteria that achieve optimal convergence and minimal computational cost. The first criterion provides the achievement of good convergence, as the effective coarsening scheme should allow to accurately interpolate a smooth error. Then, it is desirable that each F-point (Fine-grid point) has as many strongly influencing C-points (Coarse-grid points) as possible [26]. The criterion, provided minimal computational cost for different levels of V-cycle, requires that the set of C-points is the maximum subset of all F-points, to obtain more accurate interpolation, provided that no C-point is strongly dependent on another C-point (the set is maximum and independent), since such points would have

increased the computational costs without providing visible benefits of interpolation [26]. In general, as the convergence is increased, the computational costs of the V-cycle decrease. Therefore, the first criterion is strictly observed and the second one is guidance. The RS algorithm has two passes. The first pass splits the full grid in C and F points; the second one ensures strict implementation of the first criterion [26]. PMIS (parallel changes independent set), the algorithm of coarsening, is based on the same principles as the RS algorithm except that a heuristic criterion is not strictly observed, i.e., F-F connections without a common C-point are permitted. Unlike the RS coarsening, the PMIS is not sequential. However, the precision may be deteriorated because an insufficient number of points reduces the accuracy of interpolation [26].

Numerical experiments have been done using the PMIS-algorithm. In Tables 1 and 2, we give the number of AMG-iterations with the HSS-based smoother on the different grids, where α is the parameter of the HSS iteration method. For comparison, we give the AMG calculations when the Gauss–Seidel method is used as the smoothing procedure. In our implementations, all iterations are started from the zero vector, and terminated when

$$\frac{\|r^{(p)}\|_2}{\|r^{(0)}\|_2} \leq 10^{-6},$$

where $r^{(p)} = b - Av^{(p)}$ is the residual vector of the SLAE (2) at the current iterate $v^{(p)}$ and $r^{(0)}$ is the initial residual. Our comparisons are done for the number of iteration steps and the elapsed CPU time (in seconds, in parentheses). The abbreviation “n.c.” in Table 2 means “no convergence”.

The experiments are run in MATLAB (version R2018b) with a machine precision of 10^{-16} .

From Tables 1 and 2, it follows that the AMG methods with the HSS-smoother have fast convergence speed for all tested values of the viscosity coefficient ($\nu = 10^{-1} \div 10^{-5}$) on all used grids, while the AMG with the Gauss–Seidel smoother does not converge for $\nu = 10^{-4}, 10^{-5}$ on all considered grids, and does not converge on the grids 260×260 and 520×520 nodes for all values of the viscosity coefficient. For all tests, the AMG+HSS (Algebraic multigrid method with Hermitian/Skew-Hermitian Splitting smoother) outperforms the AMG+GS (Algebraic multigrid method with Gauss–Seidel smoother) with respect to both number of iteration steps and CPU time. Moreover, the number of iteration steps and CPU time increase with increasing grid size for both methods.

From the data shown in the Table 1, an increase in the number of iterations with an increase in the mesh size follows. However, this relates to some features of the algebraic approach in MGM (more precisely, the PMIS algorithm in the AMG). The traditional (a scalable) approach in the AMG (RS algorithm) works well for problems arising from the discretization of PDEs in two spatial dimensions. For many two-dimensional problems, a solver can be obtained with the number of iterations, regardless of the size of the problem n , as well as the solution time per iteration, linearly proportional to n . For the RS algorithm, the convergence factor is separated from unity and does not depend on the size of the problem n . But when using regular AMG interpolation in combination with PMIS, AMG convergence worsens depending on the size of the problem. This results in a loss of scalability [27]. However, when traditional AMG algorithms are applied to three-dimensional (3D) problems, numerical tests show [27] that in many cases scalability is lost. However, the number of iterations may remain constant. The computational complexity and size of the stencil can increase significantly, which will lead to an increase in execution time and memory usage. In addition, the PMIS algorithm allows for natural parallelization, unlike the RS algorithm. These properties of the PMIS algorithm seem promising to us for the further study of the three-dimensional Navier–Stokes equations using parallel computing.

Table 1. Algebraic multigrid (AMG)+(HSS) Hermitian/skew-Hermitian splitting iterations with different ν .

Grid	$\nu = 10^{-1}$	$\nu = 10^{-2}$	$\nu = 10^{-3}$	$\nu = 10^{-4}$	$\nu = 10^{-5}$
60 × 60	25 (21.20)	26 (21.59)	29 (26.20)	30 (26.54)	21 (14.15)
120 × 120	40 (64.50)	45 (67.70)	54 (94.60)	40 (59.20)	37 (51.50)
180 × 180	54 (152.61)	50 (151.52)	64 (161.82)	49 (126.85)	35 (97.51)
260 × 260	85 (192.70)	93 (197.20)	82 (191.52)	83 (197.26)	58 (126.7)
520 × 520	90 (282.51)	97 (290.58)	90 (286.26)	92 (286.85)	85 (252.21)

Table 2. Algebraic multigrid (AMG)+(GS) Gauss–Seidel iterations with different ν .

Grid	$\nu = 10^{-1}$	$\nu = 10^{-2}$	$\nu = 10^{-3}$	$\nu = 10^{-4}$	$\nu = 10^{-5}$
60 × 60	26 (32.57)	46 (42.51)	54 (114.85)	n.c.	n.c.
120 × 120	57 (83.82)	64 (122.61)	83 (160.50)	n.c.	n.c.
180 × 180	59 (162.36)	82 (185.38)	85 (192.20)	n.c.	n.c.
260 × 260	n.c.	n.c.	n.c.	n.c.	n.c.
520 × 520	n.c.	n.c.	n.c.	n.c.	n.c.

Table 3 shows the number of iteration steps and CPU time of the AMG+HSS method depending on the value α , when $\nu = 10^{-5}$. For the AMG+HSS method, the optimal (experimental) parameter value that reduces the number of iterations depends on the size of the grid. As the grid size increases, the value of α , which provides the best convergence, decreases. Numerical experiments showed that for parameter values less than 0.2, the AMG+HSS method diverges.

Table 3. (AMG)+(HSS) iterations with different α , $\nu = 10^{-5}$.

Grid	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1.0$
60 × 60	29 (26.84)	24 (21.51)	21 (14.15)	42 (40.61)	54 (58.86)	56 (68.22)	65 (84.20)
120 × 120	40 (61.50)	39 (64.67)	37 (51.50)	45 (67.52)	56 (94.60)	57 (94.20)	82 (162.85)
180 × 180	52 (114.2)	35 (97.51)	42 (129.20)	67 (14.82)	84 (165.84)	86 (175.21)	91 (196.21)
260 × 260	58 (126.7)	65(171.58)	65 (187.21)	82 (192.64)	84 (194.54)	91 (194.60)	95 (197.22)
520 × 520	82 (251.26)	84(251.84)	85 (252.21)	92 (260.52)	93 (262.42)	94 (282.52)	97 (290.21)

Thus, the numerical experiments have showed that the HSS-based smoothers can be effectively used for the AMG, in which the stage of coarse-grid correction can be considered as a kind of accelerating procedure of the HSS methods.

5. Conclusions

In our previous theoretical and numerical studies of the MGM with the STS-based smoothers, the stationary (and nonstationary) linear diffusion–convection equation with dominant convection was considered as a test problem [13,14]. All theoretical results and calculations were performed using geometric MGM. Here, we first use the HSS-method as the smoother in the algebraic MGM for solving the unsteady Navier–Stokes equations. It is supposed to further prove the theoretically smoothing properties of the HSS iteration methods and to prove the convergence of the MGM with the corresponding smoothers. In addition, theoretical and numerical results should be obtained for the MGM with the STS-based smoothers for the Navier–Stokes problem. The PMIS algorithm was not chosen by us by chance. Preliminary testing of it on this model problem showed its robustness. In addition, the PMIS algorithm allows for natural parallelization, unlike the RS algorithm. These properties of the PMIS algorithm seem promising to us for the further study of the three-dimensional Navier–Stokes equations using parallel computing.

Author Contributions: Conceptualization, G.M.; methodology, G.M., T.M.; validation, E.A.; writing—original draft preparation, T.M., Z.-Q.W.; writing—review and editing, T.M.; software, V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by RFBR, grant N19-51-53013 GFENa, and Ministry of Science and Higher Education of the Russian Federation (basic part, project N1.5169.2017/8.9).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MGM	Multigrid Method
SLAE	Systems of Linear Algebraic Equations
AMG	Algebraic Multigrid
HSS	Hermitian/Skew-Hermitian Splitting
STS	Skew-Hermitian Triangular Splitting
PDE	Partial Differential Equations
CFD	Computational Fluid Dynamics

References

1. Bai, Z.-Z.; Golub, G.H.; Ng, M.K. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM J. Matrix Anal. Appl.* **2003**, *24*, 603–626. [[CrossRef](#)]
2. Bai, Z.-Z. Splitting iteration methods for non-Hermitian positive definite systems of linear equations. *Hokkaido Math. J.* **2007**, *36*, 801–814. [[CrossRef](#)]
3. Bai, Z.-Z.; Golub, G.H.; Ng, M.K. On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *Linear Algebra Appl.* **2008**, *428*, 413–440. [[CrossRef](#)]
4. Bai, Z.-Z.; Golub, G.H.; Pan, J.-Y. Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. *Numer. Math.* **2004**, *98*, 1–32. [[CrossRef](#)]
5. Bai, Z.-Z.; Golub, G.H.; Lu, L.-Z.; Yin, J.-F. Block triangular and skew-Hermitian splitting methods for positive definite linear systems. *SIAM J. Sci. Comput.* **2005**, *26*, 844–863. [[CrossRef](#)]
6. Krukier, L.A. Implicit Difference Schemes and an Iterative Method for Their Solution for One Class of Quasilinear Systems of Equations. *Izv. Vuzov. Math.* **1979**, *7*, 41–52, (In Russian).
7. Krukier, L.A. Convergence acceleration of triangular iterative methods based on the skew-symmetric part of the matrix. *Appl. Numer. Math.* **1999**, *30*, 281–290. [[CrossRef](#)]
8. Krukier, L.A.; Chikina, L.G.; Belokon, T.V. Triangular skew-symmetric iterative solvers for strongly nonsymmetric positive real linear system of equations. *Appl. Numer. Math.* **2002**, *41*, 89–105. [[CrossRef](#)]
9. Wang, L.; Bai, Z.-Z. Skew-Hermitian triangular splitting iteration methods for non-Hermitian positive definite linear systems of strong skew-Hermitian parts. *BIT Numer. Math.* **2004**, *44*, 363–386. [[CrossRef](#)]
10. Botchev, M.A.; Krukier, L.A. On an iterative solution of strongly nonsymmetric systems of linear algebraic equations. *J. Comput. Math. Math. Phys.* **1997**, *37*, 1283–1293. (In Russian).
11. Krukier, L.A.; Martinova, T.S.; Bai, Z.-Z. Product-Type Skew-Hermitian Triangular Splitting Iteration Methods for Strongly Non-Hermitian Positive Definite Linear Systems. *J. Comput. Appl. Math.* **2009**, *232*, 3–16. [[CrossRef](#)]
12. Bai, Z.-Z.; Krukier, L.A.; Martinova, T.S. Two-step iterative methods for solving the stationary convection-diffusion equation with a small parameter for the highest derivative on a uniform grid. *J. Comput. Math. Math. Phys.* **2006**, *46*, 295–306. (In Russian) [[CrossRef](#)]
13. Muratova, G.V.; Andreeva, E.M. Multigrid method for solving convection-diffusion problems with dominant convection. *J. Comput. Appl. Math.* **2009**, *226*, 77–83. [[CrossRef](#)]
14. Muratova, G.V.; Krukier, L.A.; Andreeva, E.M. Fourier analysis of multigrid method with triangular skew-symmetric smoothers. *Commun. Appl. Math. Comput.* **2013**, *27*, 355–362.
15. Li, S.; Huang, Z. Convergence analysis of HSS-multigrid methods for second-order nonselfadjoint elliptic problems. *BIT Numer. Math.* **2013**, *53*, 987–1012. [[CrossRef](#)]

16. Fedorenko, R.P. The relaxation method for solving difference elliptic equations. *J. Comput. Math. Math. Phys.* **1961**, *1*, 922–927. (In Russian) [[CrossRef](#)]
17. Brandt, A. Multi-level adaptive solutions to boundary-value problems. *Math. Comput.* **1977**, *31*, 333–390. [[CrossRef](#)]
18. Hackbusch, W. Convergence of multi-grid iterations applied to difference equations. *Math. Comput.* **1980**, *34*, 425–440. [[CrossRef](#)]
19. Briggs, W.; Henson, V.; McCormick, S. Algebraic Multigrid (AMG). In *A Multigrid Tutorial*, 2nd ed.; Poulson, D., Briggeman, L., Eds.; SIAM Publications: Philadelphia, PA, USA, 2000; pp. 137–159.
20. Falgout, R. An introduction to algebraic multigrid. *Comput. Sci. Eng.* **2006**, *8*, 24–33. [[CrossRef](#)]
21. Hackbusch, W. Introductory Model Problem. In *Multigrid Method and Application*; Graham, R.L., Jolla, L., Eds.; Springer: Berlin/Heidelberg, Germany, 1985; pp. 17–39.
22. Brandt, A.; Yavneh, I. Accelerated multigrid convergence and high Reynolds recirculating flows. *SIAM J. Sci. Comput.* **1993**, *14*, 151–164. [[CrossRef](#)]
23. Veldman, A.E.P. "Missing" boundary conditions? Discretize first, substitute next, and combine later. *SIAM J. Sci. Stat. Comput.* **1990**, *11*, 82–91. [[CrossRef](#)]
24. McKee, S.; Tome, M.F.; Ferreira, V.G.; Cuminato, J.A.; Castelo, A.; Sousa, F.S.; Mangiavacchi, N. The MAC method. *Comput. Fluids* **2008**, *37*, 907–930. [[CrossRef](#)]
25. Stuben, K. Algebraic multigrid (amg): Experiences and comparisons. *Appl. Math. Comput.* **1983**, *13*, 419–451. [[CrossRef](#)]
26. Yang, U.M. Parallel Algebraic Multigrid Methods—High Performance Preconditioners. In *Numerical Solution of Partial Differential Equations on Parallel Computers*; Lecture Notes in Computational Science and Engineering; Bruaset, A.M., Tveito, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 51, pp. 209–236.
27. Sterck, H.; Yang, U.M.; Heys, J. Reducing Complexity in Parallel Algebraic Multigrid Preconditioners. *SIAM J. Matrix Anal. Appl.* **2006**, *27*, 1019–1039. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).