Inexact matrix exponential preconditioner for implicitly restarted Arnoldi method in fluid dynamics stability problems for parallel heterogeneous architecture

Nikolay M. Evstigneev
Federal Research Center "Informatics and Control", Institute for System Analysis, Russian Academy of Science, 117312, Moscow, pr. 60-letiya Oktyabrya, 9, Russia.
E-mail: evstigneevnm@yandex.ru

Abstract. We wish to solve the problem of linear stability analysis of stationary solutions for a fluid dynamics problem governed by 3D Navier Stokes equations. The discrete problem is of order $O(10^6) - O(10^7)$ so we use parallel computations to perform linear analysis. We aim at the heterogeneous computational architecture based on central (CPU) and graphical (GPU) processing units. In the paper we use the implicitly restarted Arnoldi method (IRAM) to recover the most dangerous eigenvalues. The IRAM has poor convergence without specific transformation of the matrix spectrum. Standard methods require either long computations or inverse of the transformed matrix. This can be troublesome in fluid dynamics problems where matrix is not available explicitly and is non-symmetric, ill-conditioned and can be fully filled. We propose the usage of the inexact matrix exponential with shift-inverse transformation using only matrix-vector product. The inverse is performed using Krylov methods using only matrix-vector product. We prove convergence bounds for the GMRES method and show that this type of matrix transformation accelerates GMRES convergence. We demonstrate application of the method in recovering eigenvalues close to the imaginary axis in some model problems on some matrices from the Matrix Market. We also demonstrate the application of the method for some fluid dynamics problems. We demonstrate convergence, acceleration and efficiency on the multiple GPU cluster.

1. Introduction
During the bifurcation analysis of fluid dynamics problem we need to solve the eigenvalue problem for the most dangerous part of the spectrum at a certain point in parameter space. We assume that by some procedure we derive the finite dimensional system of equations that is solved on a computer:

$$\mathbb{M} \mathbf{v}_t = \mathbf{F}(\mathbf{v}, \mathbf{R}) + \mathbf{f}.$$  \hspace{1cm} (1)

where $\mathbf{v}(t) \in \mathbb{R}^n$ is a vector of a discrete solution of the original system, $n$ is large (typically from $1 \cdot 10^6$ to $1 \cdot 10^8$), $\mathbb{M}$ is a linear operator (also called mass matrix, can be an identity $\mathbb{E}$) and $\mathbf{F}$ is a discrete operator, that is usually combined of linear and nonlinear parts (such operator $\mathbf{F}$ is called Ozeen operators for incompressible flow equations), $\mathbf{f} \in \mathbb{R}^n$ is a discrete source term and $\mathbf{R}$ is the parameter vector. Boundary conditions are included in $\mathbb{M}$ and $\mathbf{F}$ operators. For
some solution $v_0$ at $R_0$ of (1) we derive the linearized system for ininitialismal perturbations $y = v - v_0$:
\[ M y_t = A(y, R_0 + \delta R)|_{v_0}, \]
where $A|_{v_0}$ is the linear operator of $F$ at solution $v_0$ and $\delta R$ is the variation of parameter vector. Operator $A$ can be represented in operator form (as operator–vector application) or in matrix form, in this case $A \in \mathbb{R}^{n \times n}$. We would like to know the most dangerous part of the spectrum $\text{sp}(A)$ that is responsible for the linear stability loss for $\delta R$ variation. So have two problems of linear stability. The first one is related to periodic and quasi-periodic solutions and in this case the left hand term is not zero. This problem usually poses no computational challenge for eigenvalue solvers. In case of the zero left hand side we interested in the stability of the stationary solution by finding the part of the spectrum closest to the imaginary axis. This problem poses substational computational challenges for the eigenvalue solvers especially in HPC implementation. In order to solve (1) and (2) with high precision i.e. perform Direct Numerical Simulation (DNS), one must use High Performance Computations (HPC). In this paper we are aiming at the heterogeneous computational architecture that consists of Central Processing Units (CPUs) and Graphical Processing Units (GPUs) in configuration where number of GPUs and CPUs are installed in a single node and these nodes are connected by fast interconnect. This type of HPC cluster configuration is a standard configuration for almost all Top500 machines. We are mostly interested in GPU dominated machines (up to 8 GPUs installed on one node) because GPUs decrease power consumption and machine cost.

Let us give a brief survey of methods that are already developed and successfully used during the linear stability analysis of systems (1), where $F$ is the Ozeen discrete operator. The solution of an eigenvalue problem for (1) in case of incompressible flow is conducted using Krylov subspace methods with exponential estimation preconditioner [20] or shift and inverse preconditioner [1]. One of the first such papers was [7] (and some other papers by these authors), where a 3D Floquet stability analysis was applied to a circular cylinder 2D wake stability problem. The fluid flow solver was a spectral element method for 2D equations where Floquet analysis was used through the introduction of infinitesimal three-dimensional perturbations and derivation of a state-transition periodic monodromy matrix. Such approach can be successfully used for the analysis of first bifurcations but the condition number becomes large still for higher Reynolds numbers. Using this approach bifurcation analysis for incompressible flow problems were successfully conducted in papers, for example in [2, 3]. Another successful approach is developed by Sanchez and co-authors, some results can be seen in [4, 5, 6] also using Krylov methods, namely Arnoldi method. The linear stability of a 3D compressible subsonic flow in a cavity is analyzed in [8] using ARPACK implementation of the implicitly restarted Arnoldi method and a sixth-order compact finite-difference scheme for spatial discretization. Other methods exist that can be successfully applied to these problems. For example the novel method based on two sided Newton method [22, 23] is successfully applied to find linear stability and optimal disturbances in many fluid dynamics problems. Such approach can be used for HPC application for systems where a basic Newton–Raphson method is tuned and efficiently implemented.

In all cases we can see that there are no methods of eigenvalue analysis that are specifically aimed on multiple GPU architecture. Our goal is to present a method that is used in many our papers to perform eigenvalue analysis on multiple GPU orientated architecture. For the solution of the eigenvalue problem for (2) we use the use Implicitly Restarted Arnoldi Method (IRAM) [20, 21, 36]. We chose this method for its efficiency and need only to have a matrix–vector linear operator to execute Arnoldi process. We solve the following eigenvalue problem:
\[ f(A)u = f(\lambda)u, \]
where $f()$ is a function that transforms matrix spectrum in some manner in order to accelerate
convergence of the IRAM. If IRAM converges, we obtain diagonal matrix \( f(\Lambda) \) and eigenvector matrix \( V \) of the desired part of the spectrum (defined separately for particular \( f() \)). Original eigenvalues are recovered, say, by the Galerkin projection:

\[
R = V^H A V, \tag{4}
\]

where desired eigenvalues of the original problem are stored in matrix \( R \). The IRAM is based on the construction of the Krylov subspace \( K \), dimension \( p \):

\[
K_p(f(A), u^0) := \text{span}\{ f(A)^{p-1} u^0, \ldots, f(A) u^0, u^0 \}, \tag{5}
\]

where \( u^0 \) is a starting vector with \( \|u^0\| = 1 \) and we designate \( u^n := f(A) u^{n-1} \). Every basis vector in (5) is orthogonalized with respect to all other basis vectors by the Arnoldi process. More information on the IRAM is given in [20] and currently used multiple GPU implementation is described in [36]. The method convergence well if desired part of the spectrum is well separated from the rest of the spectrum and is located on the boundary of the convex hall constructed over the spectrum. In other words, by applying \( f() \) we wish to transform the desired part of the spectrum to the boundary and separate it. One usually applies the following transformations:

- **Exponential transformation**: \( f(\Lambda) = e^{\mu \Lambda} \).
- **Polynomial transformation**: \( f(\Lambda) = \sum_j c_j \Lambda^j \).
- **Shift–inverse transformation**: \( f(\Lambda) = (A - \sigma E)^{-1} \).
- **Cayley transformation**: \( f(\Lambda) = (A - \sigma E)^{-1} (A + \sigma_0 E) \).
- **Inverse Kronecker sum transformation**: \( f(\Lambda) = 2 (A \otimes E + E \otimes A)^{-1} \).

Let us assume that we have two leading real eigenvalues \( \lambda_1 \) and \( \lambda_2 \) close to zero with \( 0 > \lambda_1 > \lambda_2, |\lambda_2| < 1 \). The matrix exponential transformation (6) would be ideal for the IRAM preconditioning only if it was explicitly available. One would seek transformed eigenvalues with maximum magnitude and recover original eigenvalues with maximum real part. However it is difficult to apply this transformation to separate eigenvalues close to the origin, so, the method is not used in stability problems that we are interested at.

A polynomial transformation (7) applies coefficients \( c_j \) such that a desired part of the spectrum is increased and the rest is damped. This method is used in ARPACK optionally. However it is difficult to apply this transformation to separate eigenvalues close to the origin, so, the method is not used in stability problems that we are interested at.

Shift and inverse (8) and Cayley (9) transformations are, probably, the best candidate to precondition eigenvalues located near a specific point \( (\sigma = 0 \) for those near the origin\) for the IRAM. The error will be reduced by the small factor \( \lambda_1/\lambda_2 \) for \( |\lambda_1| << 1 \). One, then, seeks eigenvalues with maximum magnitude for transformed spectrum and reovers eigenvalues, closest to point \( \sigma \). Direct inverse in (8) is never used in practice and one has to solve a system of linear equations in the form \( (A - \sigma E) u^{n+1} = u^n \) for the IRAM in order to find a new Krylov vector \( u^{n+1} \) in (5). However, it is well known [9, 10] that the condition of \( A \) can be very poor, especially for high Reynolds numbers and compressible flows. So the inversion in (8) becomes a huge problem that is to be solved in order to solve the eigenvalue problem. The same is true for (10). The condition of such matrix is worse than that for the Jacobi matrix and problem with preconditioners emerges. This problem is worsen for HPC GPU oriented
calculations because GPUs have limited amount of device memory and efficient preconditioners for advection-dominated flows (like ILU factorization) require explicit matrix storage. We present the method that allows us to solve the problem of eigenvalue analysis efficiently on HPC multi GPU computational architecture without explicit matrix storage or matrix graph reconstructions.

The paper is laid out as follows - first we give the preconditioner theory and theoretical convergence estimates. We also include information on the execution algorithm. Then we present the method that allows us to solve the problem of eigenvalue analysis efficiently on Ivy Bridge and up to six k40 NVIDIA GPUs assembled in one chassis.

2. Inexact matrix exponent preconditioning for Arnoldi eigensolver

We denote the left half-plane as $\mathbb{C}^- : \{\forall (a, b) : a + ib, a \leq 0\}$ and right half-plane as $\mathbb{C}^+ : \{\forall (a, b) : a + ib, a > 0\}$. Assume that the spectrum $\text{sp}(A) \subset \mathbb{C}^-$. Then we apply the following two stage transformation for the Krylov basis vector $(u^{n+1})$:

$$((E + \mu A)^m - \sigma E) u^{n+1} = u^n.$$  \hspace{1cm} (11)

The first transformation $f_1(A) = (E + \mu A)^m$ is the inexact exponent transformation (6) which is executed using explicit Euler’s method (or another timestepping method used during the solution of (1)) with $m$ timesteps and $\mu$ such that these iterations are stable. The contraction is controlled by number of iterations $m$. Eigenvalues $\lambda_1 > \lambda_2$ are transformed to region $(1 - \epsilon, 0)$ on the complex plane, where $\epsilon = (1 + \Delta t \lambda_1)^m$. If $\lambda_1 = 0$, then it is transformed exactly to $(1, 0)$. By this transformation we have $\text{sp}(f_1(A)) \subset B(1)(0, 0)$ (unit 2D ball with the center at the origin). The next transformation is a shift in the manner of shift and inverse transformation. The shift $\sigma$ defines the part of the spectrum to be sought: $\sigma = 1 - \epsilon + \delta$. We are interested in the spectrum near the imaginary axis, so $\delta$ is selected such that the solution of the linear system (11) convergence rapidly and the spectrum of the resulting mapping is well separated.

To justify our approach may use the following

**Proposition 1.** Let $A \in \mathbb{R}^{N \times N}$, $B = (E + \mu A)^m - \sigma E$, $B = VD^m$ and $k(V) = \|V\| \|V^{-1}\| = M < +\infty$ with $\sigma = 1 + \delta$, $0 < \delta < 1$, $0 < \mu < 1$ and $m = \mathcal{O}(1)$. Let the GMRES method applied to the solution of $Bx = b$ generate residual $r_q$ on the $q$-th step. Then the GMRES method with $q$ steps convergence to the, at least, given precision $\frac{\|r_q\|}{\|r_0\|} \leq \epsilon > 0$, where:

$$q \geq \log_{1+\delta} \left( \frac{M}{\epsilon} \right).$$  \hspace{1cm} (12)

**Proof.** Let us consider the $q$-th residual $r_q = b - Bx_q$ of the GMRES method [11]:

$$\|r_q\| = \min_{p \in P_q} \|p(B)r_0\|,$$  \hspace{1cm} (13)

where $P_q$ denotes the set of polynomials of degree at most $q$ and with value one at the origin. Following [11] we derive:

$$\|r_q\| \leq \min_{p \in P_q} \|p(VD^m)\| \leq k(V) \min_{p \in P_q} \|p(\lambda_k)\|.$$  \hspace{1cm} (14)
Now, by our assumption $\text{sp}(A) \subseteq \mathbb{C}^{-0}$ so mapping (11) will transfer the whole spectrum of the problem to the $B_1(-1 - \delta, 0)$. Analytical complex function reaches its maximum at the boundary, i.e. \( \max_k |p(\lambda_k)| = \max_{\lambda \in S_1(-1 - \delta, 0)} |p(\lambda)|. \) Then we can use trial polynomial \( \hat{C}_q \) and Chebyshev polynomials \( C_q \) for the boundary $S_1(-1 - \delta, 0)$ [12] and arrive at:

\[
\min_{p \in P_q} \max_{\lambda \in S_1(-1 - \delta, 0)} |p(\lambda)| \leq \min_{p \in P_q} \max_{\lambda \in S_1(-1 - \delta, 0)} |\hat{C}_q(\lambda)| = \frac{C_q(1)}{C_q(1 + \delta)} \leq \left( \frac{2}{2 + 2\delta} \right)^q. \tag{15}
\]

Using (14) and (15) we have:

\[
\varepsilon \leq k(V) \left( \frac{2}{2 + 2\delta} \right)^q, \tag{16}
\]

followed by (12) after reduction by 2. This completes the proof. \( \square \)

Technically it means that the convergence doesn't explicitly depend on the size of the matrix $A$. The dependence on the $k(V)$, as it is said in [13] makes the estimate (13) bad. However the bound of $k(V) = M$ is in the logarithm and logarithmically weak and dependence on $\delta$ is in the base of the logarithm and very intensive. We demonstrate this fact in figure 1.

Since calculation of matrix exponent is a real operation this method can only be applied to pure real eigenvalue detection. To correct this deficiency we introduce the rotation parameter $\varphi = \sigma \cos(\varphi) + i \sin(\varphi)$. Then the transformation (11) is replaced with the following system:

\[
\begin{pmatrix}
(\mathbf{E} + \mu A)^n - \text{Re}(\theta) \mathbf{E} \\
\text{Re}(\theta) \mathbf{E}
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}^n_{\text{Re}} \\
\mathbf{u}^n_{\text{Im}}
\end{pmatrix}
= 
\begin{pmatrix}
\mathbf{E} + \mu A)^n - \text{Im}(\theta) \mathbf{E} \\
\text{Im}(\theta) \mathbf{E}
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}^{n+1}_{\text{Re}} \\
\mathbf{u}^{n+1}_{\text{Im}}
\end{pmatrix}, \tag{17}
\]

where we have additional basis vectors responsible for real $\mathbf{u}_{\text{Re}}$ and imaginary $\mathbf{u}_{\text{Im}}$ mapping. The angle $\varphi \in [0, \varphi_{\max}]$, and maximum angle of rotation $\varphi_{\max}$ is known due to the following

**Proposition 2.** Let a timestepper that is used to integrate (1) has no dominant stability limit due to diffusion. Let $C$ be a Courant number from the timestepper and let us use the transformation (11). Then:

\[
\pi \geq \varphi_{\max} \geq \left| \text{Im} \left[ \sum_{k=0}^{m} \left( \frac{m}{k} \right) i^k(\mu C)^k \right] \right| \tag{18}
\]

in transformation (17).

**Proof.** Let us note that $\mu$ is selected such, that $\mu C \leq 1$. If the numerical method is stable, then the spectrum of the operator is bounded by the stability region of the method. Maximum is reached on the imaginary part of eigenvalues since diffusion is not a limiting factor. Knowing the $C$ constant we know a bounding region on an imaginary axis. This bounding region is roughly approximated by a bounding box. The most unstable spectrum is located at the neighbourhood of the imaginary axis. Taking all this into account we consider the boundary eigenvalue $\lambda_1$ such, that $|\text{Re}(\lambda_1)| << 1, \text{Re}(\lambda_1) < 0, |\text{Re}(\lambda_1)| << |\text{Im}(\lambda_1)|$. Let us apply the transformation (11). In this case the location of a transformed eigenvalue $f(\lambda_1)$ on the imaginary axis can be bounded as:

\[
\left| \text{Im}(f(\lambda_1)) \right| \leq \left| \text{Im} \left[ \sum_{k=0}^{m} \left( \frac{m}{k} \right) (1 + \mu \text{Re}(\lambda_1))^{m-k} (\mu \text{Im}(\lambda_1)^k) \right] \right| \leq \left| \text{Im} \left[ \sum_{k=0}^{m} \left( \frac{m}{k} \right) i^k(\mu \text{Im}(\lambda_1)^k) \right] \right| < \left| \text{Im} \left[ \sum_{k=0}^{m} \left( \frac{m}{k} \right) i^k(\mu C)^k \right] \right|. \tag{19}
\]

So the maximum angle of rotation $\varphi_{\max}$ that brings this corner eigenvalue to the real axis by transformation (17) is bounded by (19) from bellow and by $\pi$ from above. \( \square \)
Figure 1. Illustration of the $q$ vs $\delta$ dependence by (12) for a matrix with $M = 1 \cdot 10^{12}$ and $\varepsilon = 1 \cdot 10^{11}$.

The system increases its size twofold resulting in additional storage space which is a significant drawback. Another drawback results in the introduction of rotational grid that must be used in order to find all leading eigenvalues with nonzero imaginary parts. However this approach, in authors experience, is more stable in practice than the Cayley transformation. The convergence properties don’t change since the mapping (17) has no affect on $k(V)$.

In order to recover the original eigenvalues of the problem from the mapped operator (17) we perform the following projections:

$$
R_R = V_R^H A V_R,
R_I = V_I^H A V_I,
R = R_R + R_I.
$$

Desired eigenvalues of the original problem are contained in the matrix $R$. Notice, that pure real eigenvalues are doubled by this projection.

Let now there exist the part of the spectrum $sp_1(A) : sp(A) \supset sp_1(A) \subseteq \mathbb{C}^+$ with leading eigenvalues $Re(\lambda_1) > Re(\lambda_2) > 0$. It means that the main solution becomes unstable. Then we apply inexact exponential transformation (6) and eigenvalues are transformed as $(1 + \Delta t \lambda_j)^m, j = \{1, 2\}$. So we contrll the separation of the desired spectrum part $sp_1(A)$ by increasing $\Delta t$ or increasing $m$. This approach for multiple GPU architecture and stability analysis is illuustrated in [36].

For the linear system solution we use restarted GMRES(n) or BiCGstab(L) methods (the latter has better convergence in practice). Schematically, the described algorithm for inexact matrix exponent preconditioner is presented in figure 2.

From this figure we can estimate number of matrix–vector multiplications as:

$$
op(MV) = 2mK_{IRAM} I_{Arnoldi} K_{GMRES} I_{GMRES},
$$

where $K_{IRAM}$ is the number of vectors in Krylov subspace of the IRAM, $I_{Arnoldi}$ is the number of Arnoldi iterations, $K_{GMRES}$ is the number of basis vectors in GMRES, $I_{GMRES}$ is the number of restarts for GMRES method and $m$ is number of iterations in the inexact exponent mapping (11). Typical value of $\nop(MV)$ is about 60000–80000 from our experience. This is compareable with the typical $MV$ count for the linear system solution during the standard shift and inverse process with good preconditioning. However, this approach is way more efficient on the multiple
Figure 2. Schematic description of the inexact exponent shift and inverse IRAM algorithm

GPU systems since minimal PCIeExpress transfer is used during iterations and only matrix vector applications are used by calling linearised timestep for the given vector.

3. Shift-inverse preconditioning on multiple GPUs

We also implemented shift-inverse and Cayley transformations (8) for multiple GPU architecture using simplified approximate preconditioner for the solution of the linear system.

Survey and analysis of preconditioners for fluid dynamics problem or GPU architecture is given in [14, 15, 16] and for shift inverse methods – in [24, 25]. One can observe that the iterative convergence properties of the matrix \( f(A) \) are worsen by the decrease of diagonal elements in (8). The preconditioned shift–inverse method is solved iteratively using Krylov–type method (GMRES or BiCGStab(L)). Let \( x^j \) be an approximation of the eigenvector and \( B := (A−σE) \).

An iteration of the Krylov method can be formulated with the right preconditioner \( P \) as:

\[
BP^{-1} \tilde{u} = x^j, \quad u = P^{-1} \tilde{u}.
\]

(22)

Olshanskii 2012 We construct the preconditioner by using the linearisation on more simple operator with close spectral properties (pseudo-compressibility method for incompressible flows or first order method for compressible flows) with simple Lax–Friedrichs fluxes and simple diffusion discrete operator (7 point Laplace operator in 3D). It is easy to show that the these methods generate diagonally dominant matrices, even for large Reynolds numbers. The Jacobi matrix is found using analytical differentiation of the Lax–Friedrichs flux. Then the matrix can be inverseed by the application of the Block Jacobi or Block Symmetric Gauss–Seidel methods and applied as the preconditioner that gives a rough estimate of the linear operator spectrum. We use Geometric Muligrid (GMG) method [17, 18, 19, 32] to accelerate convergence of the preconditioner. It was tested in many applications on GPU architecture and shown good parallel efficiency.

Let us denote \( GMG(A, b) \) as a GMG set of V or W cycles for the solution of the linear system with matrix \( A \) and the RHS \( b \). Now we construct the preconditioner. First we split our operator \( F(u) = N(u, u) + Cu \) to nonlinear part \( N() \) and linear part \( C \). The nonlinear part is found using Lax–Friedrichs flux. Then we apply linearisation that gives us linear operator \( P|_{u_0} := N_{u_0} + C \) near the solution \( u_0 \). The linearisation of \( N() \) and linear matrix \( C \) becomes split in diagonal and off-diagonal parts \( P := L + D + U \):

\[
U = L := C_n + \partial N/\partial u - \lambda_{\text{max}} E, \\
D := C_d + \partial N/\partial u + \lambda_{\text{max}} E.
\]

(23)

Here \( \lambda_{\text{max}} \) is the spectral radius of the Jacobi matrix that can be estimated by the maximum absolute value of the spectrum in the linearized system. Observe that matrices \( L, U \) and \( D \) consist
of blocks defined by the system (1) dimension. For gas dynamics problems these are blocks of $5 \times 5$ and $3 \times 3$ for incompressible flows. First we add shift to the system by the scalar parameter $\sigma$ that influences only diagonal block matrix $D$, obtaining modified diagonal $\tilde{D} = D - \sigma E$.

Then for each smoother in the $GMG(P, r)$ we perform the following sweeps:

$$\begin{align*}
(\tilde{D} + L) \tilde{u} &= r, \\
(\tilde{D} + U) u &= \tilde{D} \tilde{u}.
\end{align*}$$

In this case Krylov residuals $r^{n+1}$ are constructed using preconditioner:

$$\begin{align*}
\tilde{r}^{n+1} &= u^n - B u^n, \\
r^{n+1} &= GMG(P, \tilde{r}^{n+1}),
\end{align*}$$

Sweeps (24) are performed in the manner of LU-SGS method analogous to the suggested approach in [26, 27]. First we decompose our computational domain into separate subdomains for each computational device (GPU or CPU) and color these domains in such manner, that adjusted boundary elements belong to domains with different coloring, say ”blue” and ”green”. Next, we select internal and boundary elements for these domains. First, we perform sweeps in ”blue” internal cells, then we perform sweeps in half of ”green” internal and all boundary cells, then we perform sweeps in ”blue” boundary cells and, finally, use the rest of the ”green” cells. Inside every cell subdomain we use red-black coloring to perform GPU or OpenMP parallelization. Boundary cells are exchanged using CUDA-aware MPI. After that we perform either restriction or prolongation on GMG and perform sweeps again on different grids.

It is noted in [24] that a standard preconditioner to $B$ is not very efficient so one must apply a tuned preconditioner formed by Sherman–Morrison formula. However we were unable to find any implementation of such preconditioner for HPC architecture and this problem is the topic of another research.

4. Testing problems and efficiency

In this section we give results on testing problems, efficiency estimates and examples of the method application.

4.1. Matrix Market examples

First we present some test on standard matrices. Small matrix $\text{rdb450}$ from matrix market collection [31] is used to illustrate the application of the method. This non–symmetric matrix is originated from the discretization of reaction-diffusion brusselator model. Different stages of the method application are illustrated in figure 3. Green and red circles represent eigenvalues that are found by the application of the described inexact matrix exponent preconditioner and IRAM. We seek least stable part of the spectrum, i.e. eigenvalues with maximum real parts. Original spectrum of the matrix and desired eigenvalues are depicted in figure 3a. First we perform inexact matrix exponent transformation, result is presented in figure 3b. Observe, that the desired eigenvalues are now largest magnitude eigenvalues (LM) and are located close to the line $\text{Re}(z) = 1$. Then we perform shift with $\sigma = 1.05$ transforming the whole spectrum close to the line $\text{Re}(z) = 0.05$. Then we inverse the transformed matrix using GMRES method (in this case it converges to machine epsilon with 4 iterations) and the rest of the spectrum is inverseed analytically for illustrative purpose. The result is presented in figure 3c. We can observe, that the desired eigenvalues are now largest magnitude eigenvalues (LM) and are located close to the line $\text{Re}(z) = 1$. Then we perform shift with $\sigma = 1.05$ transforming the whole spectrum close to the line $\text{Re}(z) = 0.05$. Then we inverse the transformed matrix using GMRES method (in this case it converges to machine epsilon with 4 iterations) and the rest of the spectrum is inverseed analytically for illustrative purpose. The result is presented in figure 3c. We can observe, that the method successfully recovered eigenvalues near the origin. However, one more complex–conjugate eigenvalue is not recovered (marked with red circles in figure 3c). In order to recover this eigenvalue we apply rotation to the matrix transformation using (17) with rotation angle $0 < \varphi < \pi/8$ and obtain missed eigenvalues at $\varphi = \pi/11$, shown in figure 3d with the transformed spectrum.
Figure 3. Process of spectrum transformation during inexact exponent shift and inverse IRAM algorithm for matrix rdb450, left to right: (a) - original spectrum, (b) - inexact matrix exponent spectrum, (c) - shifted and inverse spectrum, (d) - rotated shifted spectrum. Green circles represent eigenvalues with largest real part (LR) in original and transformed spectrum.

Next we consider the stress test problem of finding four eigenvalues with largest real part for non–symmetric matrix S80PI_n1, sized 4025 × 4025. This matrix has two complex–conjugate desired eigenvalues \( \lambda_{1,2} = 2.0 \cdot 10^{-12} \pm 1.87i \) and \( \lambda_{3,4} = 2.0 \cdot 10^{-12} \pm 1.68i \). We compare the suggested transformation (11) with Cayley transformation (9). For this test we chose \( \sigma = 0.01, \sigma_0 = -0.01, 0 < \varphi < \pi/3 \) with step \( d\varphi = \pi/50 \) and use \( m = 3 \) with \( \mu = 0.3 \). We add 10 additional Krylov vectors for filtering in the IRAM, thus Krylov subspace dimension is 14. Results for Cayley transformation are provided in figure 4. One can see that it is impossible to recover desired spectrum with this transformation, despite good convergence of Ritz eigenvalues. The mapped spectrum (middle figure) is tuning Ritz eigenvectors to the boundary. On the other hand we are able to recover desired spectrum using suggested transformation, see figure 5. One can see, that the mapped spectrum of the problem is transformed in such way, that the IRAM recovers selected eigenvalues very efficiently. Red circles correspond to \( \varphi = \pi/6.5 \) and cyan circle - to \( \varphi = \pi/7.5 \). This process takes some time, because we need to solve the eigenvalue problem for each value of \( \varphi \) in accordance with the introduced step size \( d\varphi \).

Provided examples show that the suggested preconditioning method can be used to recover the most dangerous part of the spectrum from the linear stability point of view.

4.2. Fluid dynamics examples

Now let us consider the acceleration achieved for multiple GPU architecture and compare acceleration and efficiency with shift and inverse transformation using preconditioners (25).

Results of convergence for different methods are provided in table 1. One can see, that the choice of parameters \( \sigma, m \) and \( \mu \) can heavily influence the method. For example for the same problem size (say 6.4E6) we can obtain convergence for the restarted GMRES method from 1100 to 3600 iterations. Unfortunately, there’s no a priori recipe for selecting these parameters. This we leave for future work. Second, we can see, that the BiCGStav(L) method converges more rapidly, so in practice we use this method more often.

Verification of the method on fluid dynamics applications is performed in all problems we solved, e.g. [33], [34], [35]. For example we present here the comparison of growth rates as
Figure 4. Cayley transformation. Convergence of two Ritz eigenvectors in the IRAM, mapped spectrum with desired and filtered Ritz eigenvalues and recovered largest real part eigenvalues (red circles) of matrix S80PI_n1 with true largest real part eigenvalues (blue asterisk).

Table 1. Averaged number of iterations ($It \cdot 10^2$, best/worst choice of parameters) for GMRES(n) (with n being Krylov basis dimension, if $It > n$, then restarting applied) and BiCGStab(L) methods vs system size for different problems applied to inexact exponential preconditioned system with target tolerance $1 \cdot 10^{-10}$.

| Problem size | Incompressible, Spectral Method | Compressible, Finite Volume Method |
|--------------|---------------------------------|-----------------------------------|
|              | GMRES(50)                       | BiCGStab(5)                       | BiCGStab(7)                      |
|              | 4/7                             | 3/9                               | 2/4                              |
|              | 10/34                           | 4/12                              | 3/7                              |
|              | 11/32                           | 5/15                              | 4/7                              |
|              | 11/36                           | 5/17                              | 4/7                              |
|              | 14/41                           | 5/19                              | 5/21                             |
|              | 21/43                           | 2/12                              | 3/12                             |
|              | 18/45                           | 6/12                              | 6/24                             |
|              | 24/56                           | 7/9                               | 5/22                             |
|              | 32/120                          | 8/21                              | 7/31                             |

functions of the Mach number $M$ for 2D compressible Kelvin–Helmholtz instability problem test in table 2. We can observe satisfactory results compared to the available in literature.

Result of wall time parameters for the solution of the eigenvalue problem for the incompressible fluid problem [35] are presented in figure 6 and wall time for compressible flow problems [34] are presented in figure 7. We can see, that only four iterations of the full IRAM are required to achieve desired tolerance. The same rapid convergence was observed in figure 5. This is achieved by adjusting regulating parameters $\mu$ and $\sigma$ for the particular problem.
Figure 5. Inexact exponent shift inverse transformation. Convergence of two Ritz eigenvectors in the IRAM, mapped spectrum with desired and filtered Ritz eigenvalues and recovered largest real part eigenvalues (red and cyan circles) of matrix $S80PI_{n1}$ with true largest real part eigenvalues (blue asterisk).

Table 2. Results of the validation for growth rates $\alpha$ as functions of Mach number

| $M$ | computed $\alpha$ | $\alpha_{exact}$, [28] | $\alpha$, [29] | $\alpha$, [30] |
|-----|-------------------|-----------------------|----------------|----------------|
| 0.1 | 0.187312          | 0.187                 | 0.188          | 0.189          |
| 0.5 | 0.1411            | 0.141                 | 0.1411         | 0.1411         |
| 0.9 | 0.054743          | 0.055                 | 0.054723       | 0.0547         |

We noticed that the computational cost of the suggested method is $O(N)$ for compressible flow problem and $O(N \log_2 N)$ for incompressible problem using spectral method. It means that the suggested method doesn’t introduce additional difficulties during computation compared to the original time stepping method proportional to the problem size. Additionally one can see that the initial stagnation is related to the incomplete GPU loading.

Comparison of the preconditioned shift and inverse method with the suggested method is shown in right figure 7 and done on the compressible problem of the Kelvin–Helmholtz instability [34]. We can notice that preconditioned shift and inverse method is more efficient for small matrices, however, starting with problem size around $1 \cdot 10^6$ the presented method becomes more efficient. This can partially be explained by less efficient preconditioner for the shift.
and inverse method. One can also check that the efficiency of parallel implementation for the preconditioned shift and inverse method becomes lower with the increase of parallel nodes. This is due to the increase of PCIExpress communications.

Convergence of eigenvalues to true ones in the physical problem is the function of problem discretization. In figure 8 we show the convergence of first four leading eigenvalues to the true ones in the Kelvin–Helmholtz problem as function of domain discretization and show modulo of the first unstable eigenvector.

5. Conclusion
In the paper we present the method of preconditioning for the linear stability eigenvalue problem. The method is aimed at the implicitly restarted Arnoldi method and can be easily added to the existing linear analysis code for fluid dynamics problems. The main purpose of the method is to be used for stability analysis of stationary solutions on cluster based on multiple GPU parallel architecture. We show that the method has analytical convergence bound and application examples confirm this bound. We also demonstrated efficient parallel application of the method in fluid dynamics problems sized up to $2 \cdot 10^8$ elements. Our future work is to tune the method and give suggestions for the selection of tuning parameters. The method implementation is available freely under GPL licence by request to the author.

Acknowledgments
This work is supported by RFBR grant no. 17-07-00116.
Figure 8. Evolution of eigenvalues as function of problem discretization and modulus of unstable eigenvector for 3D Kelvin–Helmholtz problem [34] near the first Andronov–Hopf bifurcation, 3D view.

References
[1] Tuckerman, L.S., Bertagnolio, F., Daube, O., Le Quere, P., Barkley, D. Stokes preconditioning for the inverse Arnoldi method. In: D. Henry and A. Bergeon (Eds.), Notes on Numerical Fluid Dynamics, vol. 74, pp.241–255, (2000). doi:10.1007/bfb0107081
[2] Boronska, K. and Tuckerman, L.S.: Extreme multiplicity in cylindrical Rayleigh-Benard convection. I. Time dependence and oscillations. Physical Review E, vol. 81, iss. 3, 036320,(2010). doi:10.1103/physreve.81.036320
[3] Boronska, K., Tuckerman, L.S.: Extreme multiplicity in cylindrical Rayleigh-Benard convection. II. Bifurcation diagram and symmetry classification. Physical Review E, vol. 81, iss. 3, 036321, (2010). doi:10.1103/physreve.81.036321
[4] Sanchez, J., Marques, F. and Lopez, J. M.: A continuation and bifurcation technique for Navier-Stokes flows. J. Comput. Phys., 180, 78–98,(2002). doi:10.1006/jcph.2002.7072
[5] Sanchez, J., Net, M., Garcia-Archilla, B. and C. Simo.: Newton-Krylov continuation of periodic orbits for Navier-Stokes flows. J. Comput. Phys., 201 (1), 13–33, (2004). doi:10.1016/j.jcp.2004.04.018
[6] Sanchez, J., Net, M., Simo, C.: Computation of invariant tori by Newton-Krylov methods in large-scale dissipative systems. Physica D, 239, 123–133, (2010). doi:10.1016/j.physd.2009.10.012
[7] Barkley, D., Henderson, R.D.: Three-dimensional Floquet stability analysis of the wake of a circular cylinder. Journal of Fluid Mechanics 322, 215-241, (1996), doi:10.1017/s0022112096002777.
[8] Bres,G.A. and Colonius, T.: Three-dimensional linear stability analysis of cavity flows. AIAA 2007-1126 45th AIAA Aerospace Sciences Meeting and Exhibit 8 - 11 January 2007, Reno, Nevada, doi:10.2514/6.2007-1126.
[9] Choquet, R.: A Matrix-Free Preconditioner Applied to CFD. Research Report, RR-2605, INRIA, (1995). doi:10.1016/s0168-9274(98)00114-7
[10] Benzi, M.: Preconditioning Techniques for Large Linear Systems: A Survey. Journal of Computational Physics, 182, 418–477, (2002). doi:10.1006/jcph.2002.7176
[11] Saad, Y.: Iterative Methods for Sparse Linear Systems, Second edition, Society for Industrial and Applied Mathematics, (2003). doi:110.1137/1.9780898718003
[12] Rivlin, T.J.: The Chebyshev Polynomials: from Approximation Theory to Algebra and Number Theory. J. Wiley and Sons, New York, 1990. doi:10.2307/2153227
[13] Meurant, G., Tebbens, J.D.: The role eigenvalues play in forming GMRES residual norms with non-normal matrices. Numer. Algorithms 68, 143165 (2015). doi:10.1007/s11075-014-9891-3
[14] Rehman, M., Vuik, C., Sega, G.: A comparison of preconditioners for incompressible Navier-Stokes solvers. Int. J. Numer. Meth. Fluids, 57, 12, 1731–1751, (2008). doi:10.1002/nmta.2010.33.1
[15] Benzi, M., Olsanskii, M.A.: An augmented Lagrangian-based approach to the Oseen problem. SIAM J. Sci. Comput., 28:2095-2113, (2006). doi:10.1137/050646421
[16] Li,R., Saad,Y.: GPU-accelerated preconditioned iterative linear solvers. The Journal of Supercomputing, 63, 443–466, (2013). doi:10.1007/s11227-012-0825-3
[17] Vanka, S.P.: Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. J. Comput.
