Fully implicit multiple graphics processing units’ schemes for hypersonic flows with lower upper symmetric Gauss-Seidel preconditioner on unstructured non-orthogonal grids

A N Bocharov¹, N M Evstigneev², O I Ryabkov ¹,²

¹Joint Institute for High Temperatures of the Russian Academy of Sciences, Izhorskaya 13 Bldg 2, Moscow 125412, Russia.
²Federal Research Center "Informatics and Control", Institute for System Analysis, Russian Academy of Sciences, Moscow, pr. 60-letiya Oktyabrya, 9, 117312, Russia.
E-mail: evstigneevnm@ya.ru

Abstract The governing equations are the Navier-Stokes equations for viscous calorically perfect gas. Second order finite volume discretization is applied to the problem on arbitrary unstructured grids with implicit temporal treatment. The problem is solved using the Newton-Raphson method. Such methods require the solution of large linear systems with iterative solvers and require a preconditioning operator to converge. In this paper a well known Lower-Upper Symmetric Gauss Seidel (LU-SGS) preconditioner is applied. The new method is based on the tricky reordering for the factored Jacobian matrix that allows one to execute block triangular matrix solvers on GPUs in parallel without the loss of algebraic properties of the original non-factored operator. Good convergence properties are demonstrated for large scale problems of external aerodynamics with Courant number around 1000 - 10000 for flows with Mach number around 13-25. The influence of the non-orthogonal corrections on the fluxes for highly skewed anisotropic grids is demonstrated at Mach numbers around 20. It is shown that the non-orthogonality corrections are of the second infinitesimal order compared to the effects of the shock fix influence on the heat fluxes on the body.

Introduction
The paper is focused on a particular interest in hypersonic flows when the discretization is carried out using unstructured non-orthogonal grids on hybrid computational architecture. Since the flow is hypersonic, the characteristics of the hyperbolic part of the governing equations are dominated by upwinding configuration hence the efficient solver must utilize this feature of the flow in terms of numerical methods. It is easy to show that a Gauss Seidel (GS) method applied to the implicit method for the advection equations with uniform velocity and upwind discretization converges with one iteration [1]. Similar result can be obtained for the uniformal supersonic inviscid gas flow with Riemann solvers that can switch to upwinding (e.g. HLL and AUSM family solvers). Hence the usage of this idea is beneficial for high speed hypersonic flows. The application of this idea is implemented in Lower-Upper Symmetric Gauss Seidel (LU-SGS)
preconditioner that can be applied to accelerate the convergence of the linear solver in implicit methods. There are many papers in this field, e.g. [2-5]. The difficulty is the usage of the LU-SGS method on the parallel computational architecture where the inversion of the upper and lower triangular systems must be carried out in parallel. Various methods are suggested, including coloring [3,6] and reordering [7,8]. The former method is easier to implement but in this case the beneficial algebraic properties of the serial SGS method are undermined, since such procedure no longer respects upwinding. It is tolerated for two pattern coloring (for structured grids) but becomes a problem for unstructured grids where number of colors increases to 4. In this case the preconditioner operator \( P \) loses its main purpose and the property of \( P^{-1}A \approx E \) is lost, where \( A \) is the linearized discrete operator of the problem at hand and \( E \) is the identity. Reordering, on the other hand, respects upwinding and the reordered LU-SGS operator becomes a good preconditioner. More details are given in the very good review papers [9,10] dealing with the multithreaded OpenMP implementation of the LU-SGS method on unstructured mesh. The difficulty is to construct reordering for multiple GPU computational architecture, where reordering must be done on two levels: in each GPU and between GPUs. The idea of this reordering is described in this paper (previously given in [11] for supersonic flows) and the application of the reordered LU-SGS is demonstrated for a hypersonic flow problem.

In addition, a correction to the viscous fluxes being calculated must be carried out for the unstructured non-orthogonal grids. Such corrections are provided in variety of literature, see [12] for example. But their influence on the secondary properties, i.e. on heat fluxes and stress tensor on the surface of the body, has not being investigated so far in connection with the hypersonic flow regimes.

The paper is laid out as follows. First a brief discription of the governing equations and numerical method is presented that leads to the solution of the large sparse linear systems of equations. The method of the gradient discretization for the unstructured non-orthogonal grids is presented and analyzed next. The method of solving systems of equations with LU-SGS preconditioning and reordering process are described. Finally, the results are provided that demonstrate convergence of the iterative processes and influence of the non-orthogonal corrections on the obtained numerical heat fluxes on the body surface that is subject to hypersonic flow by the viscous gas.

**Problem formulation**

The governing equations for compressible viscous calorically perfect gas in domain \( \Omega \) with boundary \( \partial \Omega \) are defined as:

\[
\begin{align*}
(\rho)_t + \nabla \cdot (\rho \mathbf{u}) &= s_1, \\
(\rho \mathbf{u})_t + \nabla \cdot (\mathbf{u} \otimes (\rho \mathbf{u})) + \nabla p &= \nabla \cdot \mathbf{P} + (s_2, s_3, s_4)^T, \\
(E)_t + \nabla \cdot (\mathbf{u}(E + p)) &= \nabla \cdot \mathbf{G} + \nabla \cdot (\mathbf{P} \cdot \mathbf{u}) + s_5.
\end{align*}
\]

Here, \( t \) is time; \( \rho \) is the gas density; \( E = \rho \mathbf{u}^2/2 + \rho e \) is the total specific gas energy, \( e = C_v T \) is an internal specific gas energy; \( p = \rho RT \) is a pressure; \( T \) is a gas temperature; \( R \) is a specific gas constant; \( \mathbf{P} \) is a viscous stress tensor with dynamic viscosity \( \mu \) defined by the Sutherland's law; \( C_v \) is the heat capacity at constant volume; operation \( \otimes \) designates tensor product. The source terms are given by the vector \( \mathbf{s} = (s_1, s_2, s_3, s_4, s_5)^T \). Heat flux is given by:

\[
\mathbf{G} = k \nabla T,
\]

where \( k \) is the gas heat conductivity coefficient. The viscous stress tensor can be written explicitly as:

\[
\mathbf{P} = (2\mu/3 - \xi)(\nabla \cdot \mathbf{u})\mathbf{I} + \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T),
\]

where \( \xi \) is a bulk viscosity.
Discretization and solution

The domain $\Omega$ is discretized into computational elements by the semi-automated procedure in the SALOME open source grid generation software. The resulting discrete problem can be formulated as:

$$U_r - \vec{V} \cdot (F(U)) = 0,$$

(4)

where conservative variables are grouped into vector $U$, inviscid and viscous parts are grouped into $F(U)$ and discrete divergence operator is given as $\vec{V} \cdot$. The temporal discretization is carried out using backward Euler (BE) method if one is mainly interested in the stationary solutions:

$$U^{n+1} - U^n = \tau F(U^{n+1}),$$

(5)

where $n$ is the temporal slice number and $\tau$ is the fictitious time step parameter. The solution process is conducted using Newton-Raphson method by substituting linearization near $U^{n+1}$. On each iteration one solves a linear system, reformulated into the standard form:

$$A\delta U = b,$$

(6)

where the main matrix is the Jacobian, $\delta U$ is the solution vector update and $b$ is the right-hand side.

The solution of the linear system (6) is performed using Krylov iterative method (BiCGStabL) with the left preconditioner ($P$) in the form:

$$r = Ax - b,$$

$$z = P^{-1}r.$$

(7)

This residual $z$ is supplied to the Krylov subspace method and is no longer directly related to the original system. We apply LU-SGS factorization [2] for the preconditioner matrix which is basically two triangular sparse system solutions: one is for the lower triangular system and one is for the upper triangular system. This approach can be referenced to [2], briefly given as the split of the main matrix into block diagonal, block lower and block upper triangular parts, each block is sized $5 \times 5$. The Jacobi matrix is assembled using automatic differentiation (AD) method [13]. This allows one to use complex flux solvers and various gas models without explicitly defining their derivatives w.r.t. to the vector of conservative variables. The first order inviscid reconstruction is used in the Jacobi matrix as well as diagonal entries to the viscous flux tensor to minimize the stencil. The obtained Jacobian is an approximation to the real one. The whole computational mesh is divided into local regions that belong to each GPU. Two regions share common faces of elements. It is important to have only neighbours of level $l = 0$ on these faces. The coloring is introduced for elements that share common faces from different GPUs. The colors of these elements are such, that neighbouring elements have different colors. Only boundary elements are assigned with coloring hence the number of colors depends on the domain decomposition but not on the number of GPUs. Usually there are no more than two colors. Next, the boundary colored elements are reordered in such a way, that those elements are the last entries in the vector $U$. In this case the suggested parallel algorithm executes the LU-SGS preconditioning process on this reordered matrix which is equivalent to the serial LU-SGS algorithm. No penalty due to parallel implementation in algebraic properties of the matrix is introduced.

The algorithm of the Jacobi matrix construction is the following:

- Assemble diagonal blocks on each GPU locally.
- Assemble off-diagonal blocks on each GPU locally.
- Assemble diagonal and off-diagonal blocks of colored boundary elements, store them
in each GPU.
  - Assemble off-diagonal blocks of fluxes that depends on colored boundary elements when the derivative is taken by the local GPU elements, store them in each GPU.
  - Assemble off-diagonal blocks of fluxes that depend on local GPU elements when derivative is taken by the colored boundary elements, store them in each GPU.

After the matrix is assembled the diagonal block is inverted and the preconditioner is factored. The resulting matrices are stored in block compressed row storage (BSR) sparse format that can be used in sparse linear algebra (cuSPARSE) library from compute unified device architecture (CUDA) toolkit on NVIDIA GPUs. The application of (7) is performed by the following steps. Before the main iterative cycle the matrix is analyzed by running NVIDIA cuSPARSE library cusparseXbsrsv2_analysis function on each GPU for each local matrix. Next, the following algorithm is executed in the main iterative cycle:
  - Apply lower triangular block matrix factorization.
    - Solve local lower triangular matrices using cuSPARSE library cusparseXbsrsv2_solve function on each GPU locally.
    - Use the obtained vector values for the lower part of the off-diagonal blocks (that correspond to the colored boundary elements when derivative is taken by the local GPU elements) and solve the system by back substitution locally on each GPU. The vector $r^*$ is obtained.
  - Apply upper triangular block matrix factorization.
    - Solve the system for the boundary colored elements (that correspond to the local GPU elements when the derivative is taken by the colored boundary elements).
    - Use obtained values of the boundary elements to substitute these values in each GPU local matrix. Thus these entries are regrouped to the RHS.
    - Solve local upper triangular matrices using cuSPARSE library cusparseXbsrsv2_solve function on each GPU locally. The vector $z$ is obtained.

This procedure is executed on each call to the matrix vector application in the Krylov subspace solver.

The definition of the heat flux (2) and viscous stress tensor (3) requires the calculations of gradients in order to obtain the desired flux on the face of each element before using resulting finite volume formula. This approach is based on a well known least squares method with non-orthogonal corrections. In doing so we partly follow [12]. We test three approaches regarding the correction method: a minimum correction, where the correction vector has a lesser magnitude, then it's orthogonal projection, an orthogonal correction and over-relaxed correction where the correction vector has a higher magnitude, then it's orthogonal projection, see [12].

Results
Two tests are considered. First, the simple problem of the natural convection in rectangular domain is considered. Its purpose is to check the effects of the non-orthogonal corrections for the diffusion dominated flows. Next, the test related to the hypersonic flow is performed to verify the influence of the correction on the secondary flow properties.
Natural convection in rectangular domain

The problem is set up in accordance with [14]. Two parameters are introduced, namely, the Prandtl number $\Pr = \nu/k$ and Rayleigh number $\Ra = \Pr g \rho_0^2 (T_h - T_c) H^3 / (T_0 \mu_0)$ where $\nu$ is kinematic viscosity, $g$ is the magnitude of the gravity vector, $T_h$ and $T_c$ is the temperature values on hot and cold walls, $T_0$ is the mean reference temperature and $H$ is the height of the domain. A 3D domain is considered with symmetry boundary conditions in $z$-direction to mimic the 2D problem formulation. The source term is given as $s = (0,0, -\rho g, 0, -u_2 \rho g)^T$. The following constants are used: $p_0 = 101325$ Pa, $T_0 = 600$ K, $R = 287$ J kg$^{-1}$ K$^{-1}$, $\Pr = 0.71$, $\gamma = 1.4$, $(T_h - T_c) = 10$ K and $\Ra = 1000$. Such low Rayleigh number was chosen to test the non-orthogonal corrections at the dominant diffusive flow.

The testing grids are totalling $400 \times 400 \times 5$ elements for hexahedral orthogonal reference mesh and 112500 elements for the tetrahedral non-orthogonal mesh (corresponds to $75 \times 75 \times 5$ hexahedra, each subdivided to four tetrahedra), middle sections are presented in figures 1 and 2. The result of the steady state simulations for orthogonal grid is presented in figure 3 and for non-orthogonal grid in figure 4. The results for the different correction methods are presented in figures 5 - 7, all figures are plotted with no smoothing using second order approximation.

One can observe that the non-orthogonal corrections indeed allow one to obtain solutions closer to the orthogonal grid. The section graph in figure 8 clearly shows that the corrected solutions are closer to the reference solution. The difference between different corrections is minor and cannot be traced on these figures in favour of particular correction method.

The convergence history is provided in figure 9. The convergence slope is independent on the mesh size. The convergence is better for the finer grids that can be explained by the better approximation of the original problem.
**Figure 3.** Temperature distribution in the middle section on orthogonal grid.

**Figure 4.** Temperature distribution in the middle section on non-orthogonal grid with no correction.

**Figure 5.** Temperature distribution in the middle section on non-orthogonal grid with minimum correction.

**Figure 6.** Temperature distribution in the middle section on non-orthogonal grid with orthogonal correction.
Figure 7. Temperature distribution in the middle section on non-orthogonal grid with over-relaxed correction.

Figure 8. Temperature distribution middle section by \( z = z_{\text{middle}} \) and \( y = y_{\text{middle}} \) planes depending on the correction.

However, asymptotic of the wall time is not optimal, see figure 10. This happens due to the fact that the LU-SGS preconditioner is no longer an optimal preconditioner for the diffusion dominated problems and, basically, becomes a block Jacobi preconditioner.

Figure 9. Convergence history of the natural convection problem for different grid sizes (\( z \) direction has uniformal 5 elements), starting from the first iteration. Left figure shows convergence with respect to the...
wall time, right figure shows convergence with respect to the problem physical time, \( R \) is the nonlinear residual vector, \( R_0 \) is the initial residual vector.

![Graph showing convergence](image)

**Figure 10.** Convergence wall time as function of number of elements \( N \) compared to different asymptotics, namely, \( O(N) \) for linear, \( O(N^{3/2}) \) for optimal timestepper asymptotic and \( O(N^2) \) for quadratic asymptotic.

**Hypersonic flow over the sphere-cone body**

The hypersonic flow over the sphere-cone body is considered. This problem was used in [15]. The sphere-cone body has a length of 1 m with 0.01 m radius sphere on the cone tip and the cone angle of 8°. Inflow gas is attaching the body with zero angle, gas is the air, parameters correspond to the height of 10 km with Mach number equals 24. We used previously generated unstructured mesh from [15] with \( 10.2 \times 10^6 \) elements. Minimum element length in normal direction to the wall of the body surface is \( 1.298 \times 10^{-5} \) m, the element ratio near the boundary layer is about 1000. The slice of the computational mesh is presented in figure 11 and velocity and pressure distribution example is presented in figure 12.

![Mesh slice](image)

**Figure 11.** Slice of the computational mesh for the hypersonic flow over the sphere-cone problem.

![Velocity vectors and pressure distribution](image)

**Figure 12.** Velocity vectors and pressure distribution around the sphere-cone at Mach 24.
The distribution of the heat flux on the surface of the sphere-cone, projected to the front plane is presented in figures 13 - 16. Introduced corrections for this problem have either zero or negative effect. One can notice that the minimum and orthogonal corrections in figures 14, 15 are even less symmetric than the results that were obtained with no correction in figure 13.

**Figure 13.** Heat flux on the surface of the sphere-cone at Mach 24, no correction.

**Figure 14.** Heat flux on the surface of the sphere-cone at Mach 24, minimum correction.

**Figure 15.** Heat flux on the surface of the sphere-cone at Mach 24, orthogonal correction.

**Figure 16.** Heat flux on the surface of the sphere-cone at Mach 24, over-relaxed correction.
Figure 17. Sections of heat flux distributions along the sphere-cone surface as function of distance on the frontal plane.

Graphs of heat fluxes as function of distances in six planes sections along the cone generatrix for different corrections are shown in figure 17. The sections are generated by six planes that contain the central axis and are rotated around the axis by the $n\pi/3$ angle, where $n$ is the plane number.

One can see in these section graphs that the orthogonal corrections can't introduce improvements for hypersonic flows, when boundary elements have sufficient aspect ratios. In some cases these correction have negative impact on the distribution of the secondary flow properties, as it is seen in figure 17, top two
figures. The distributions with no correction are more symmetric than those with minimal and orthogonal corrections.

![Figure 18](image)

**Figure 18.** Convergence of the hypersonic flow problem as function of different non-orthogonal corrections; calculation wall time on the left, physical problem time on the right.

The convergence history for the hypersonic problem as function of different non-orthogonal corrections is provided in figure 18. No strong dependence on the correction method is observed. The residual reduction is $5 \cdot 10^{-5}$ that is sufficient for the heat fluxes to converge on the solid boundary of the sphere-cone, see [11] for more details. The timestep during the globalization process is automatically selected and reaches values that correspond to $10,000$ CFL, maximum.

**Discussion**

The utilization of non-orthogonal corrections for hypersonic flows is pointless, at best. The boundary layer grid is constructed out of almost orthogonal elements with large aspect ratios. In this case the corrections introduce either no improvements or even worsen the situation as can be seen from the hypersonic flow over the sphere-cone problem in figure 17. It is impossible to test the hypersonic flow with elements of unit aspect ratio because this would lead to a very large computational mesh, which is unpractical.

The assumption is that the asymmetry in heat fluxes at hypersonic flow is due to the utilization of the shockfix stabilization designed to cure the carbuncle effect, see [16,17]. An attempt to use other approaches to cure this asymmetry as well as carbuncle (rotated solvers, low order local flux reconstruction, switches etc) led to no improvements. In one of such attempts the material in paper [18] was used, where the author claims that he constructed AUSM-family combined solvers to cure the carbuncle effect. Unfortunately we were unable to confirm his claims and found number of misprints and mistakes. Original paper contained asymmetric fluxes formula that lead to the violation of the conservation laws. Next, the Sod tube test revealed that the suggested solvers are not monotone, even for the first order of approximation. Finally, the
corrected (by our understanding) solvers were able to solve the hypersonic problem but the obtained heat fluxes and shear stresses on the body were sufficiently lower, then the benchmark data. We hope that the author of the paper [18] will be able to comment on these results. All data is available from the corresponding author at request. It is assumed that the solution to the asymmetric heat flux distribution as well as the carbuncle effect must utilize artificial viscosity damping combined with higher order boundary elements. Such combination can be applied in the discontinuous Galerkin method.

The convergence rates of the proposed LU-SGS preconditioner are problem dependant, as expected. The preconditioning of the hypersonic problems (as well as supersonic problems, see [11]) is efficient and the suggested preconditioner can be used for such problems with success. The application of this preconditioner for diffusion dominated problem (natural convection) shows lower efficiency and the convergence rate becomes dependent on the grid spacing, see figure 10. The application of the multigrid preconditioning is suggested for this kind of problems.

The future work will be focused on the implementation of the implicit scheme with artificial viscosity carbuncle cure. Next, multigrid preconditioner must be incorporated into the preconditioning procedure. Finally, the discontinuous Galerkin method with second order boundary elements will be implemented.

Computations were performed on the high performance computing mini-cluster owned by one of the authors.

References
[1] Evstigneev N M 2019 Journal of Physics: Conference Series 1391 012080
[2] Jameson A and Yoon S 1987 Aiaa Journal - AIAA J 25 929-935
[3] Sun Y, Wang Z, Liu Y and Chen C 2007 45th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings 313
[4] Borisov V E, Davydov A A, Kudryashov I Y, Lutsky A E and Men'shov I S 2015 Mathematical Models and Computer Simulations 7 222-232
[5] Sharov D, Luo H, Baum J and Loehner R 2000 38th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings 927
[6] Sharov D, Luo H, Joseph D B and Lohner R 2000 AIAA-2000-927-Aerospace Sciences Meeting and Exhibit
[7] Kim J and Kwon O 2002 40th AIAA Aerospace Sciences Meeting and Exhibit
[8] Menshov I and Pavlukhin P 2016 The Journal of Supercomputing
[9] Petrov M, Titarev V, Utyuzhnikov S and Chikitkin A 2017 Comput. Math. and Math. Phys. 57 1895-1905
[10] Chikitkin A, Petrov M, Titarev V and Utyuzhnikov S 2018 Lobachevskii Journal of Mathematics 39(4) 503-512
[11] Bocharov A, Evstigneev N, Petrovskiy V, Ryabkov O and Teplyakov I 2020 Journal of Computational Physics 406 109189
[12] Moukalled F, Mangani L and Darwish M 2016 The Finite Volume Method in Computational Fluid Dynamics (Springer International Publishing)
[13] M Tadjouddine S A F and Qin N 2005 Int. J. Num. Meth. in Fluids 47 1315-1321
[14] Yu Y, Liu F, Zhou T, Gao C and Liu Y 2019 Acta Mechanica Sinica 35 401-410
[15] Bocharov A N, Evstigneev N M and Ryabkov O I 2015 J. Phys.: Conf. Ser. 653 012119
[16] LeFloch P G and Mishra S 2014 Acta Numerica 23 743-816
[17] Garicano-Mena J, Lani A and Deconinck H 2016 Computers & Fluids 133 43-54
[18] Phongthanapanich S 2019 Shock Waves 29 755-768