PAPER • OPEN ACCESS

Coupled gas heat and mass transfer problem solution using fully implicit method on graphics processing units

To cite this article: A N Bocharov et al 2019 J. Phys.: Conf. Ser. 1147 012053

View the article online for updates and enhancements.

You may also like

- Time-resolved temperature characterization of a hypersonic shock layer using a single high-speed color camera for aerospace design applications Sneh Deep, Yedhu Krishna and Gopalan Jagadeesh
- <u>Computational analysis in underexpanded</u> jets simulations J R Martins, J V Barbosa, F P Santos et al.
- <u>A Preliminary Study on Numerical Waves</u> and Its Impact on Horizontal Plate Based on Fluent

Jinpeng Hu, Shihui Liu and Ling Zhu





DISCOVER how sustainability intersects with electrochemistry & solid state science research



This content was downloaded from IP address 3.135.187.106 on 16/05/2024 at 22:34

Coupled gas heat and mass transfer problem solution using fully implicit method on graphics processing units

A N Bocharov¹, V A Bityurin¹, N M Evstigneev³, V E Fortov¹, N N Golovin², V P Petrovskiy¹, O I Ryabkov^{1,3}, Yu S Solomonov², A A Shustov² and I O Teplyakov¹

¹ Joint Institute for High Temperatures of the Russian Academy of Sciences, Izhorskaya 13 Bldg 2, Moscow 125412, Russia

² Open Joint Stock Company "Corporation Moscow Institute of Heat Technology", Beryozovaya Avenue 10, Moscow 127273, Russia

³ Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Vavilova 44 Bldg 2, Moscow 119333, Russia

E-mail: bocharov@ihed.ras.ru

Abstract. We present benchmarking results from a solver for hypersonic flow problems. The aim of the work is to verify new numerical methods designed for solving coupled heat and mass transfer problem with explicit or implicit time marching schemes using multiple graphics processing units and central processing units hybrid computational architecture without any usage of ANSYS Fluent. The verification is performed on the problem of a flow over the sphere. We consider hypersonic flow regimes of Mach number M = 20-30. New results are compared with results of the previous version of the code, where ANSYS Fluent was used to solve body heat and ablation problems. We consider the body shape after surface ablation, heat loads on the surface and aerodynamic loads on the whole sphere. We also compare timings between ANSYS Fluent and new heat and mass solver; compare time difference between explicit and implicit methods for gas dynamics part and present total wall time for convergence using different methods.

1. Introduction

The main problem of the hypersonic flow regime for computations is the stiff conditions that are laid on numerical methods. In our previous papers [1-3], we presented methods that are aimed at heterogeneous high performance computational architecture using general purpose computations on graphics processing units (GPUs). The previous code used explicit time integration and commercial software (ANSYS Fluent) to solve solid body problem of heat transfer and ablation. Here we present new methods that have implicit time stepping procedure for gas dynamical part and new high performance GPU orientated solid body solver for heat transfer and ablation. The main goal of the paper is to perform benchmarking of physical and numerical quantities and confirm the validity of newly developed numerical codes.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI. Published under licence by IOP Publishing Ltd 1



Figure 1. Spherical shell in the gas flow, the geometry and the mesh of the problem: I is the inflow gas boundary and O is the outflow gas boundary.

2. Problem formulation

We are considering a spherical shell (shown in figure 1) of external diameter 0.1 m and constant thickness 0.025 m (internal diameter is 0.05 m) moving with hypersonic speed in the air. The gas rest properties are given as density ρ_0 and temperature T_0 . The pressure is defined as $P_0 = \rho_0 R_{\text{spec}} T_0$. Inflow velocity on the boundary I is defined as $\mathbf{u}_0 = u(1,0,0)^T$, where constant u is set by selected Mach numbers of the problem. The outflow boundary O uses sponge zone in order to perform correct outflow for subsonic and supersonic flow conditions, see [4]. Gas viscosity is found, using Sutherland's law [1]. Solid body part consits of graphite with common properties.

The spherical shell in the volume Ω is meshed with unstructured grid. The geometry and mesh are divided into two parts by different physical properties—gas dynamics part Ω_1 containing all surrounding air (zone G in figure 2) and solid body part Ω_2 (zone T in figure 2). Conditions on the interface surface $\partial \Omega_w$ between gas and solid body and numerical values for initial and boundary conditions are described below. Boundary condition for the heat equation on the internal spherical shell boundary $\partial \Omega_i$ is given by the Neumann conditions.

The gas dynamics mesh part contains exponentially contracting boundary layer in the direction to the external surface of the spherical shell, denoted b (minimum length in the normal direction is 10^{-6} m) while solid body part is meshed with either prismatic or tetrahedral grid. We consider two grid densities: *Coarse grid* consists of 150 000 elements in the gas dynamics part and 14 900 elements in the solid body part (tetrahedral grid) while *Fine grid* consists of 850 000 elements in the G part and 38 500 elements (prism grid) in the T part.



Figure 2. Spherical shell in the gas flow calculation zones: G stands for the gas dynamics mesh part, b stands for the gas boundary layer and T stands for the thermal energy and ablation mesh part.

3. Governing equations

We assume that the gas flow can be governed by continuous mechanics, which is similar to [1-3]. The governing equations in the gas zone are defined as follows:

$$\begin{aligned} (\rho)_t + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ (\rho \mathbf{u})_t + \nabla \cdot (\mathbf{u} \otimes (\rho \mathbf{u})) + \nabla p &= \nabla \cdot \mathbf{\Pi}, \\ (E)_t + \nabla \cdot (\mathbf{u}(E+p)) &= \nabla \cdot \mathbf{G} + \nabla (\mathbf{\Pi} \cdot \mathbf{u}). \end{aligned}$$
(1)

Heat equations that are solved for the solid body zone of the spherical shell are

$$\rho_m C_p(T_m)_t = \nabla \cdot (\lambda \nabla T_m) \,. \tag{2}$$

Here, t is time; ()_t is a time derivative; operation \cdot is a dot product, E is a full gas energy; T is a gas temperature; **Π** is a viscous stress tensor; p is a pressure; **G** is a heat flux; **u** is a velocity vector; ρ is a gas density; \otimes designates tensor product; ρ_m is a density of the solid body material, C_p is a heat capacity of the material, λ is a heat conduction in the material, T_m is a material temperature. The code solves two problems with iterative coupling: the solution of (1) for external flow problem on Ω_1 and (2) for internal solid body on Ω_2 . The flow–wall interface $\partial \Omega_w$ is the exchange boundary between two problems. Calculation of heat and mass flux, estimation of the surface recession rate and re-shaping of surface is performed on the $\partial \Omega_w$. The ablating surface $\partial \Omega_w$ is considered as moving, the mass flux $(m)_t$ is found on this surface, thus a normal velocity of surface element $v_n = (m)_t/\rho_m$ is defined through the solution of equations

$$q_w = f(T, p, q_0, \ldots), \qquad \dot{m}_w = f(T, p, q_0, \ldots).$$
 (3)

Here, q_0 is the heat flux from the flow (it can include the radiation flux as well). In the paper, well known model [5] was applied as an example. Shapard's method [6] was used to deform grid

XXXIII International Conference on Equations of State for Matter

IOP Conf. Series: Journal of Physics: Conf. Series 1147 (2019) 012053 doi:10.1088/1742-6596/1147/1/012053

in both computational domains Ω_1 and Ω_2 : in the air and in the solid body. More details are found in [1]. We now present new methods for the solution of the problem.

3.1. Solution of the heat equation

The heat equation (2) is equiped with boundary conditions as described in Problem formulation. The boundary for $\partial \Omega_w$ is the Robbin condition defined as

$$q_w = \alpha \left(T_{m,w} - T_{m,\text{eff}} \right),\tag{4}$$

IOP Publishing

where $q_w = -\lambda \partial T_m / \partial \mathbf{n}_w$ is the heat flux upon the wall that is defied by (3). The temperature $T_{m,w}$ in (4) is found using iterative procedure forming external iterations over the linear solver. We define $\alpha = -\mathrm{d}q/\mathrm{d}T$ and $T_{m,\mathrm{eff}} = T_{m,w} - q_w (\mathrm{d}T/\mathrm{d}q)$.

After the discretization of the heat equation and introduction of time slices n and n + 1 with timestep τ and discrete operator ∇_h , one gets the following iterative procedure over the trapezoidal time implicit method:

$$T_m^{r+1,n+1} - T_m^n = \left[\tau / (2\rho_m C_p) \right] \left(\nabla_h \cdot \left(\lambda \nabla_h T_m^{r,n+1} \right) + \nabla_h \cdot \left(\lambda \nabla_h T_m^n \right) \right), q_r^{r+1}|_{\partial \Omega_i} = 0, q_w^{r+1} = q_w^r + (\mathrm{d} q_w^n) / (\mathrm{d} T) \left(T_{m,w}^{r+1,n+1} - T_{m,w}^{r,n+1} \right),$$
(5)

where r is the external iteration index. Initial conditions are set for some realistic temperature value T_m^0 . The heat equation in (5) is solved using either the geometric multigrid method [7] or the Jacobi method. Both can be executed on multiple GPUs.

3.2. Implicit method for gas dynamics equations

Assume that the discretization procedure is applied to (1) and equations are translated into the discrete form for the given tessellation of Ω_1 . This procedure (including used numerical schemes and Riemann solvers) is presented in [2]. Let us rewrite discrete analog of equations (1) for all mesh elements in the vector form, grouping conservative variables into vector **U**, inviscid part into vector **F** and viscous part into vector **H** with discrete operator ∇_h :

$$\mathbf{U}_t - \nabla_h \cdot (\mathbf{F} + \mathbf{H}) = \mathbf{0}. \tag{6}$$

Time explicit method for solving the problem (6) is discribed in [2]. We introduce backwards Euler time stepping method with time step τ and time slices n and n+1. Then (6) are rewritten as

$$\mathbf{U}^{n+1} - \mathbf{U}^n = \tau \nabla_h \cdot \left(\mathbf{F}(\mathbf{U}^{n+1}) + \mathbf{H}(\mathbf{U}^{n+1}) \right).$$
(7)

Let us designate $\mathcal{F}(\mathbf{U}) \triangleq \nabla_h \cdot (\mathbf{F}(\mathbf{U}) + \mathbf{H}(\mathbf{U}))$ and introduce nonlinear iterations r as

$$\mathbf{U}^{n+1,r+1} = \mathbf{U}^{n+1,r} + \delta \mathbf{U}.$$
(8)

Sign \triangleq means "equality by definition". Plugging (8) into (7) and linearising near $\mathbf{U}^{n+1,r}$ leads to the following Newton–Raphson method:

$$\begin{cases} \left(\mathbf{E}/\tau_r - \mathbf{J} \right) \delta \mathbf{U} = \left(\mathbf{U}^n - \mathbf{U}^{n+1,r} \right) / (\tau_r) + \mathcal{F}(\mathbf{U}^{n+1,r}), \\ \mathbf{U}^{n+1,r+1} = \mathbf{U}^{n+1,r} + \delta \mathbf{U}, \end{cases}$$
(9)

where **E** is an identity matrix and $\mathbf{J} \triangleq (\partial \mathcal{F}(\mathbf{U})) / (\partial \mathbf{U}) |_{\mathbf{U}^{n+1,r}}$ is the Jacobi matrix that we estimate analytically. We omit off-diagonal viscous tensor elements for the Jacobi matrix. This does not influence order of approximation but only modifies the inclanation of the tangent hyperplane. We can use (9) to find time dependant solution (by setting $\tau_r = \tau$ as a real physical timestep) or steady state solution. In the latter case τ is used for the continuation procedure (pseudo-transient continuation) and has no real physical meaning, asymptotically $\tau_r \to \infty, r \to \infty$, see [8]. The main problem of (9) from computational point of view is the ill

conditioning of the J matrix that results in extremely poor convergence (or even divergence) of the iterative solution for the linear system. Such poor conditioning of the J matrix is the result of the nonlinearity of the governing equations and relitively big ratio of the mesh characteristic length in the boundary layer regions l_b to the mesh characteristic length in the free flow regions l_f , yielding $l_f/l_b \sim 10^7$. We use two different preconditioners: block Lower Upper Symmetric Gauss–Seidel (LU–SGS) and block Incomplete Lower Upper factorization with 0 fill-in (ILU(0)). For the first method we use a decomposition on lower triangular, diagonal, and upper triangular matrices $E/\tau_r - J = L + D + U$ and obtain the preconditioner matrix as $M = (L+D)D^{-1}(U+D)$. Two full block Gauss–Seidel sweeps are required to get the result of application of inverted matrix on a vector **b** defined as $\mathbf{m} = M^{-1}\mathbf{b}$: forward $\mathbf{g} = (L+D)^{-1}\mathbf{b}$ and backward $\mathbf{m} = (U+D)^{-1}D\mathbf{g}$. The inversion of a 5×5 block is done using Gauss–Jordan elimination. For ILU(0) cases we obtain a preconditioner matrix $M^{-1} = (\hat{L}\hat{U})^{-1}$. Here \hat{L} and \hat{U} are incomplete lower and upper factors. These factorizations require a solution of the block-tridiagonal linear system. It is solved using cuSPARSE library from CUDA libraries toolkit. Finaly, the linear system residual **r** is formed as

$$\hat{\mathbf{r}} = (\mathbf{E}/\tau_r - \mathbf{J})\,\delta\mathbf{U} - \left(\left(\mathbf{U}^n - \mathbf{U}^{n+1,r}\right)/\tau_r + \mathcal{F}(\mathbf{U}^{n+1,r})\right),$$

$$\mathbf{r} = \mathbf{M}^{-1}\hat{\mathbf{r}}.$$
(10)

The linear system residual is supplied to the BiCGStab(L) method [9]. We use L = 3 or L = 5 as optimal dimension of biorthogonal Krylov space. Linear system iterations stop when the following condition for the norm of the weighted residual $\|\mathbf{r}_k/a_k\|_2 \leq 1 \times 10^{-6}, k = 1, .., 5$ is satisfied. Here **a** is the characteristic length of an element multiplied by the characteristic scale of conservative variables for each conservative variable (counted by index k).

4. Results

In this paper we present results of a single GPU computation because we still optimize implicit scheme for multiple GPUs with ILU(0) preconditioner.

4.1. Gas dynamics, heat fluxes and ablation

First we benchmark explicit vs implicit method accuracy on the well known test problem of Peter Gnoffo [10] of hypersonic flow over a 0.1 m radius sphere with $\rho_0 = 0.0216 \text{ kg/m}^3$, $T_0 = 300 \text{ K}$, u = 4167 m/s and $T_m = 800 \text{ K}$. Results of the problem solution are presented in figures 3 and 4. Data for the figures is obtained by merging data from multiple sections of the sphere by 12 planes. These planes contain stagnation flow and symmetric points and are orthogonal to the equator plane. The angle step between planes is $\pi/6$. This way we check the spherical symmetry of the result on the unstructured surface grid.

One can see that the results practically coincide and are almost spherically symmetric. The L_{∞} error is 2.2% for both heat flux distribution and shear flux projection modulus. These results are also very close to those obtained in [10] for regular grid on LAURA code.

We further test the solution of the heat equation and ablation process by setting the following problem: hypersonic flow over a 0.1 m radius sphere with $\rho_0 = 0.0001 \text{ kg/m}^3$, $T_0 = 200 \text{ K}$, $u = 10\,000 \text{ m/s}$ and $T_m = 700 \text{ K}$, resulting in the Mach number $M_0 = 30$. We validate the solution of the problem with the previously developed code [1–3]. The physical time of the simulation is t = 5 s.

The gas dynamics parameters of the flow around the sphere are shown in figure 5. One can see that the flow is in the extreme conditions, namely, the outflow regime is mostly hyper and supersonic. Yet there is a small filament of the flow with subsonic regime leaving the outflow boundary with no visible disturbances. This demonstrates effective application of the sponge zone at the outflow boundary conditions [4]. We also present heat flux on the surface of the



Figure 3. Averaged heat flux (H) for explicit (red) and implicit (blue) methods for the Gnoffo problem as a function of the sphere radius (R).



Figure 4. Averaged shear stress modulus (S) for explicit (red) and implicit (blue) methods for the Gnoffo problem as a function of the sphere radius (R).

sphere and projection of the stress tensor in figure 6. One can see large magnitude values of heat flux up to 6.11 MW/m^2 . Such conditions result in extensive ablation.

Results of the temperature distribution in the solid body and heat ablation are presented in figures 7 and 8. The maximum linear surface recession is 0.52 cm at the stagnation point of the flow. The 3D shape of ablation is shown in figure 8, where the initial mesh position is presented for comparison.

The results that were calculated using the previous version of the code (with utilization of ANSYS Fluent for heat equation solution and mesh deformation) are identical to the currently obtained results. Maximum relative error is 2.54×10^{-7} in temperature difference and 6.23×10^{-6} in surface position.



Figure 5. Boundary layer velocity vectors, slices of the velocity vectors and Mach number slice (log scale).



Figure 6. Heat fluxes on the outer sphere surface and stress tensor projection vectors on the surface (log scale).



Figure 7. Spherical shell surface recession in the equatorial section and temperature distribution inside the solid body of the spherical shell for t = 5 s.



Figure 8. Spherical shell surface temperature and surface recession relative to the initial surface mesh position for t = 5 s.



Figure 9. Wall time per timestep for the ANSYS Fluent and the GPU solver execution (using the Jacobi solver). Ratio is about 20–25 times.



Figure 10. Ratio of wall times to solve the gas dynamics problem to the steady state by the explicit method to the impicit method.

4.2. Computational efficiency

We divide the computational efficiency into two parts. First, we compare wall time to execute solution of the heat equation and mesh deformation between ANSYS Fluent on CPU and our GPU implementation. Results of the relative wall time used for a single time step computation with respect to the mesh size are provided in figure 9. We can see that we reach up to 23 times acceleration for a *Fine grid* setup and 19 times acceleration for *Coarse grid* setup. This can be explained by incompete GPU loading.

Second, we compare wall time ratio of the steady state gas dynamics problem solution by the explicit (using time marching) method to the implicit method. Results are provided in figure 10. We can notice that the fully implicit method demonstrates good acceleration compared to the explicit method if we are interested in the steady state solution problem. We obtain up to 12 times acceleration comparing to the explicit method. The timestep τ_p is selected as maximum possible for the linear solver to converge. Usually we obtain such timestep, that CFL is up to 3000–5000. Notice, that further tests must be conducted to draw a solid conclusion on the effectiveness of the suggested implicit method.

XXXIII International Conference on Equations of State for Matter

IOP Conf. Series: Journal of Physics: Conf. Series 1147 (2019) 012053 doi:10.1088/1742-6596/1147/1/012053

5. Conclusion

Thus, we have shown that the new implicit methods of a coupled problem solution for hypersonic gas dynamics around the spherical shell with heat transfer and ablation of the surface are more efficient for steady state problems than already benchmarked explicit method. For LU–SGS preconditioner, we obtain 7–8 times acceleration, and for ILU(0) we obtain up to 12 times acceleration. Both methods obtain correct results in terms of heat fluxes and surface stress for hypersonic flow regimes. We also show that newly developed GPU orientated solver for the solution of the heat equation and surface recession during ablation is correct (compared to the previous implementation in ANSYS Fluent) and more efficient. The acceleration of the solver for the solid body part is 25 times compared to the ANSYS Fluent implementation. More work is needed to distribute suggested implicit method across multiple GPUs.

Acknowledgments

Development and testing of physical and numerical models were performed in the Joint Institute for High Temperatures RAS under support by the Russian Science Foundation, grant No. 14-50-00124. Computations were performed on the high performance computing mini-cluster owned by one of the authors (containing 6 GPUs).

References

- [1] Bocharov A N et al 2015 J. Phys.: Conf. Ser. 653 012070
- [2] Bocharov A N, Evstigneev N M and Ryabkov O I 2015 J. Phys.: Conf. Ser. 653 012119
- [3] Bocharov A N, Bityurin V A, Golovin N N, Evstigneev N M, Fortov V E, Petrovskiy V P, Ryabkov O I, Solomonov Yu S, Shustov A A and Teplyakov I O 2016 J. Phys.: Conf. Ser. 774 012157
- $[4]\,$ Evstigneev N M and Magnitskii N A 2014 Trudy ISA RAN ${\bf 63}$ 41–52
- [5] Scala S M and Gilbert L M 1965 AIAA J. 65 12
- [6] Berens H, Schmid H and Xu Y 1995 SIAM J. Math. Anal. 26 468-87
- [7] Evstigneev N M 2009 Vychisl. Metody Program. 10 268–74
- [8] Hicken J E and Zingg D W 2009 Globalization strategies for inexact-newton solvers 19th AIAA Computational Fluid Dynamics, Fluid Dynamics and Co-located Conferences (San Antonio, Texas: AIAA) p 4139
- [9] Sleijpen G L G and Fokkema D R 1993 Electronic Transactions on Numerical Analysis 1 11–32
- [10] Gnoffo P A 2009 Multi-dimensional, inviscid flux reconstruction for simulation of hypersonic heating on tetrahedral grids 47th AIAA Aerospace Sciences Meeting (Orlando, FL: AIAA) p 599