OpenCL Realization of Algorithm for Modelling of Heat Conducting Processes

A. Ayriyan[1], E. Ayryan[1], J. Buša[2], J. Buša Jr.[2], H. Grigorian[1,3], E. Donets[4], J. Pribiš[2]

[1]Laboratory of Information Technologies, JINR, Russia

[2]Department of Mathematics and Theoretical Informatics, Technical University of Košice, Slovakia

[3]Yerevan State University, Armenia

[4]Veksler and Baldin Laboratory of High Energy Physics, JINR, Russia

Introduction

We describe the OpenCL realization of the algorithms for solving the direct problem for the heat equation with a periodic source for modelling the thermal processes inside a pulsed cryogenic cell. This cell periodically injects working gases to the electron-string ion source. The cell is represented as a multilayer cylindrical object in which the thermal processes drove by periodic electric current passing through one of the conductive layers.

Problem Formulation

The direct problem for the heat equation in cylindrical coordinates can be formulated as following:

$$\begin{aligned} \rho_m C_{V\,m}(T)\frac{\partial T}{\partial t} &= \frac{1}{r}\frac{\partial}{\partial r}\left(r\lambda_m(T)\frac{\partial T}{\partial r}\right) \\ &+ \frac{\partial}{\partial z}\left(\lambda_m(T)\frac{\partial T}{\partial z}\right) + X_m(T), \end{aligned} \tag{1}$$

where $r \in [0, r_{\max}(z)]$, $z \in [0, z_{\max}(r)]$ and $t \geq 0$. The m is index of the layers (materials). The source function is nonzero only in layer $m = 2$ ($X_2(T) \neq 0$). The functions of heat capacities and thermal conductivities and source function of materials are given from [**?**] and fitted as describe in [**?**].

The boundary conditions are taken as

$$\begin{cases} \left.\dfrac{\partial T}{\partial r}\right|_{(r,z)\in\delta\Omega} = 0, \\ \left.\dfrac{\partial T}{\partial z}\right|_{(r,z)\in\delta\Omega} = 0, \end{cases} \tag{2}$$

where $\delta\Omega$ is the bound of the T domain (Ω), and the initial condition is

$$T(r, z, t = 0) = T_0, \tag{3}$$

where $T_0 \equiv 4.2$ (liquid helium temperature). Conjugation condition between materials considered to be ideal.

Numerical Scheme

For numerical solution the initial-boundary-value problem (1)–(3), the equation (1) was approximated by the following finite difference scheme at the special non-uniform grid [**?**]:

$$\rho\, C_{V\,i,j}\frac{\widehat{T}_{i,j} - T_{i,j}}{\tau} = \Lambda_i\,[\,\widehat{T}_{i,j}\,] + \Lambda_j\,[\,T_{i,j}\,] + X_{i,j}, \tag{4}$$

here the finite difference operators are:

$$\begin{aligned} \Lambda_i\,[\,\widehat{T}_{i,j}\,] &= \frac{1}{r_i}\frac{1}{\hbar_i}\left[r_{i+\frac{1}{2}}\lambda_{i+\frac{1}{2},j}\frac{\widehat{T}_{i+1,j} - \widehat{T}_{i,j}}{h_{i+1}}\right. \\ &- \left. r_{i-\frac{1}{2}}\lambda_{i-\frac{1}{2},j}\frac{\widehat{T}_{i,j} - \widehat{T}_{i-1,j}}{h_i}\right], \end{aligned} \tag{5}$$

$$\begin{aligned} \Lambda_j\,[\,T_{i,j}\,] &= \frac{1}{\eta_j}\left[\lambda_{i,j+\frac{1}{2}}\frac{T_{i,j+1} - T_{i,j}}{\eta_{j+1}}\right. \\ &- \left.\lambda_{i,j-\frac{1}{2}}\frac{T_{i,j} - T_{i,j-1}}{\eta_j}\right], \end{aligned} \tag{6}$$

where $r_{i\pm\frac{1}{2}} = \dfrac{r_i + r_{i\pm1}}{2}$; $i = 1 \ldots N_j - 1$, $j = 1 \ldots M - 1$ are indexes of the grid points; $h_i = r_i - r_{i-1}$, $\eta_j = z_j - z_{j-1}$, $\hbar_i = 0.5\,(h_{i+1} + h_i)$, $\eta_j = 0.5\,(\eta_{j+1} + \eta_j)$ are the space steps; $T_{i,j} = T(r_i, z_j, t_k)$ – the sought-for function on current time of evolution, always known starting from initial conditions; $\widehat{T}_{i,j} = T(r_i, z_j, t_{k+1})$ – the sought-for function at the next in the time; $\lambda_{i,j} = \lambda_m(T_{i,j})$, $\lambda_{i\pm\frac{1}{2},j} = \lambda_m\left(\dfrac{T_{i,j} + T_{i\pm1,j}}{2}\right)$, $\lambda_{i,j\pm\frac{1}{2}} = \lambda_m\left(\dfrac{T_{i,j} + T_{i,j\pm1}}{2}\right)$, $C_{V\,i,j} = C_{V\,m}(T_{i,j})$ – corresponding values of the coefficients. $X_{i,j} = X_m(T_{i,j})$ – source function.

The finite difference scheme (4) approximate the equation (1) implicitly in r direction and explicitly in z direction. To be short we don't describe here the approximation of the boundary conditions.

OpenCL Realization

The OpenCL realization of the numerical algorithm described in the previous section based on the following idea. In each time step the cycle for j-index from 1 to $M - 1$ is parallelized. Each called thread simultaneously calculates the sought-for function by the Thomas algorithm, see Fig. 1.

On the figure we show the discretization of the function domain. Particularly we group a set of points corresponding to one j^{th} thread. We also show the points involved in calculation of the given (i, j) point (bold point and crossed points on the Fig. 1). Note, that for each (i, j) point the two of the involving points used by neighbor threads ($j-1$ and $j+1$ threads), this demonstrates possibly memory sharing problem in our algorithm.

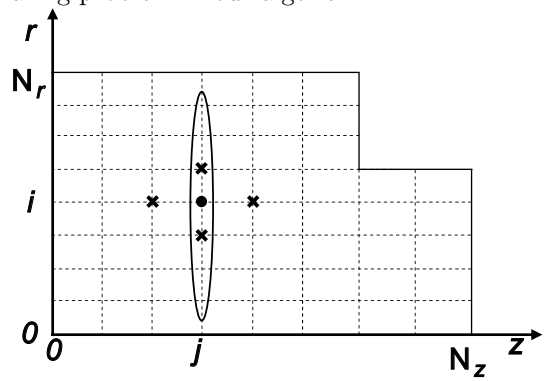

Figure 1: Schematic representation of discretization of the function domain.

Results and Discussion

The time calculations for different $N \times M$ are given in the Table **??**. The calculation has been done for evolution up to $t = 0.0765$ sec. with time step $\tau = 10^{-6}$ sec. **CPU** – Core i7 3517U (Ubuntu 11.0). **GPU** – GF GTX 470 (Core 2 Duo, Debian 6.0). For compilation of programs -O2 optimization flag have been used. GTG 470 has 14 streaming multiprocessors (SM), and each SM contains 32 computational cores. It is shown, that since GPU allows to run a lot of threads parallel the calculation time remains approximately the same on GPU in contrast with CPU, when M increases.

Table 1: Time of calculation

$N \times M$	CPU[1]	GPU[2]
431×101	309 sec.	345 sec.
431×201	607 sec.	349 sec.
431×401	1315 sec.	357 sec.
631×401	1968 sec.	509 sec.

For the further minimization of calculation time on GPU for described algorithm we assume an optimization of memory usage. For instance, removing the possible conflict of sharing GPU memory between parallel running threads. As well as minimizing number of calls to global memory in each threads.

Further Application

One of important developments of the above described applications for use calculations on GPU is the parallelization of the numerical algorithm for solving the heat transport equation

$$c_v e^{\Phi} \frac{\partial T}{\partial t} + \vec{\nabla} \cdot (e^{2\Phi} \vec{F}) = e^{2\Phi} Q \qquad (7)$$

for the simulation of cooling of compact stars with strong homogeneous magnetic field [**?**, **?**]. In this equation the flux reads:

$$\vec{F} = -e^{-\Phi} \hat{\kappa} \cdot \vec{\nabla} (e^{\Phi} T) \qquad (8)$$

where $\hat{\kappa}$ is the thermal conductivity tensor and Φ is metric function describing the geometry of spacetime manifold and it is time independent.

This problem is similar to the above considered problem with a different geometry. In this case the axial symmetric temperature distribution evaluated in the spherical symmetric distributed matter, therefore the most relevant choice of coordinate system is the spherical one (r, θ).

The other specific difference of this problem is that the integrating scheme is more effective if one uses the implicit-explicit scheme with change of directions (alternating direction method). So to present the method let us approximate the differential equation (**??**) by the following finite-difference scheme:

$$\begin{aligned} \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t_n} &+ \sigma \Lambda_i \left[u_{i,j}^{n+1}\right] + (\sigma - 1) \Lambda_j \left[u_{i,j}^{n+1}\right] \\ &= (\sigma - 1) \Lambda_i \left[u_{i,j}^{n}\right] - \sigma \Lambda_j \left[u_{i,j}^{n}\right] \\ &+ S_{i,j}\left[u_{i,j}^{n}\right] + S_{j,i}\left[u_{i,j}^{n}\right] + Q; \end{aligned} \qquad (9)$$

for unknown function $u = \log(T)$. Index n accounts the time steps, Δt_n is the time step for corresponding n-moment of the time, (i, j) are the indexes of the point on (r, θ) plane. Λ_i and Λ_j the operators representing the second order derivatives in the finite-deference scheme in the one and the other directions, $S_{i,j}$ and $S_{j,i}$ corresponding to mixed derivatives, and Q is the source term. Here we introduced parameter σ, which defines the change of the direction of the implicit and/or explicit method; when $\sigma = 1$ the direction i is solving implicitly, when $\sigma = 0$ the method is implicit in the direction j, note that it concern both angle and radius.

The realistic simulation of the physical problem of cooling of magnetars [**?**] needs a long time of calculations therefore it is important to use more effective methods for simulation. The discussed above approach realized in OpenCL can be considered as a half step of the alternating direction method.

Acknowledgment

The research was supported by JINR grant No. 14-602-01 and RFBR grants 11-01-00278, 13-01-00595 and 14-01-31227.

References

[1] National Institute Of Standards And Technology, `http://www.nist.com`

[2] A.S. Ayriyan. Model with One Spatial Variable for Design of a Technical Device // Bulletin of PFUR. Series Mathematics. Information Sciences. Physics (2013), no. 2, pp. 89-93.

[3] A.A. Samarsky. The Theory of Difference Schemes. Marcel Dekker Inc., New York (2001).

[4] C.W. Misner, K.S. Thorne, J.A. Wheeler. Gravitation. San Francisco: W.H. Freeman and Co. (1973).

[5] D.N. Aguilera, J.A. Pons, J.A. Miralles. 2D Cooling of magnetized neutron stars. A&A 486, 255271 (2008).

[6] D. Blaschke, H. Grigorian, D.N. Voskresensky, F. Weber. Cooling of the neutron star in Cassiopeia A. Phys. Rev. C 85, 022802(R) (2012).