
THEORETICAL AND MATHEMATICAL
PHYSICS

Graphic Processor Realization of the Formal Solution of the Problem of Radiation Transfer in the Neutron Star Atmosphere

A. Danilenko

*Ioffe Institute, Russian Academy of Sciences,
St. Petersburg, 194021 Russia
e-mail: danila@astro.ioffe.ru*

Received July 3, 2017

Abstract—The realization (on the CUDA platform) of the search for the formal solution to the problem of radiation transfer in the strongly magnetized plasma in the atmosphere of a neutron star on graphic processors with the CUDA architecture. The solution is obtained using the Galerkin method with finite elements. The realization of graphic processors makes it possible to substantially accelerate computations in constructing models for the atmospheres of neutron stars.

DOI: 10.1134/S1063784218040084

INTRODUCTION

Neutron stars are born hot as a result of collapse of massive stars at the last stages of their evolution. During the first few seconds after their generation, isolated neutron stars become transparent to neutrinos that are formed in the inner layers of a star due to various nuclear transformations. For this reason, during the next 10^5 years, neutron stars lose their energy due to emission of neutrinos and are gradually cooled. The rate of such cooling depends on the composition, properties, and equations of state of matter in neutron stars (see, for example, [1]). Another way for obtaining information about the superdense matter of neutron stars lies in the possibility in principle to determine their radii and masses. The dependence of the mass of a neutron star on its radius can be obtained from the solution of hydrostatic balance equations for a preset equation of state of the matter in the bulk of the star. Analyzing thermal radiation from a neutron star, it is possible in principle to determine its mass, radius, and temperature and therefore to find the limits of the composition and state of the superdense matter (see, for example, [2] for the review of the methods for investigating the superdense matter by observations).

As a rule, thermal radiation of neutron stars is observed in the soft X-ray range. Thermal radiation is formed in a thin (about a centimeter) layer on the surface. Accordingly, the thermal radiation spectrum is determined by such properties of the surface as its phase composition, the magnetic field magnitude and direction, chemical composition, and temperature. To interpret the observed thermal radiation from a neu-

tron star correctly, it is important to take into account the magnetic field structure. Since the radiation transfer in a magnetic field is anisotropic, the observed spectrum also depends on the direction to the observer; since a neutron star rotates, this direction changes continuously relative to the magnetic field configuration, and the radiation spectrum changes together with the phase of rotation even for a uniform distribution of temperature over the surface (in other words, radiation pulsates). In addition, the thermal conductivity in the outer layers of magnetized neutron stars is anisotropic (thermal conductivity along field lines is higher); therefore, the temperature distribution over the surface is also nonuniform (see, for example, [3]).

The model of a magnetized hydrogen atmosphere was initially constructed in [4–8]. It was shown in these publications how the radiation intensity per unit volume of a small surface element of a neutron star can be calculated for preset values of the effective temperature, the magnetic field, and the angle between the magnetic field direction and the normal to the surface. For calculating the radiation flux from the entire surface of a neutron star, which is detected by a distant observer, the surface can be divided into small elements so that the temperature and magnetic field can be assumed to be constant within an element; then the intensity for each element can be calculated, and the contributions from each element must be summed taking into account the bending of light in the gravity field of the star.

For interpreting the results of observations correctly, a multiparametric model of thermal radiation from the surface of the neutron star is required; using this model, the observed spectrum can be described depending on the rotation phase, the shape of the light curve depending on the photon energy (or dynamic spectrum), as well as polarization of thermal radiation. The presumptive model must depend on the following parameters, which can be varied so as to obtain the best description of the observed data (magnetic field strength, temperature, mass, radius, angle between the magnetic moment and the rotational axis, angle between the rotational axis and the direction to the observer, and acceleration due to gravity on the surface. In practice, it is possible to calculate first the radiation intensity from a surface element on the grid of various values of the photon energy, the effective temperature, the magnetic field, and the angle between the magnetic field and the normal to the surface. Such a grid must obviously be very large. For example, in the Xspec data analysis package, the nsa model [6], which actually describes the radiation emitted by a single element of the surface with a fixed magnetic field, is specified by a table with intensities for 14 values of temperature and 1000 values of the photon energy (14 000 values in all). If the grid also contains at least 10 values of the magnetic field and 10 values of the angle, it means that we must calculate radiation from a single element of the surface about a million times. One of the main obstacles for the realization of this plan is a huge time expenditure for calculation of thermal radiation from an element of the surface.

For solving this problem, the realization of the formal solution to transport equations in a strongly magnetized plasma on graphic maps has been developed. The formal solution is a basic and resource consuming step in calculating thermal radiation from a surface element. In this paper, radiation transport is considered in the diffusion approximation [9] because any practical calculation begins with the diffusion approximation [4]. Radiation transport in the diffusion approximation is briefly described in Section 1. For solving corresponding equations, the Galerkin method with finite elements is used, which is briefly described in Section 2. The realization of the Galerkin method on graphic maps is described in Section 3. The realization is verified for the problem of transport in an isothermal atmosphere, which has a simple analytic solution. Further prospects of constructing the model of radiation from a neutron star are briefly discussed in Section 4.

1. RADIATION TRANSPORT IN A STRONGLY MAGNETIZED PLASMA

Let us consider the system of radiation transport equations for two normal (ordinary and extraordinary) modes in the diffusion approximation [9, 10]:

$$\frac{d}{d\tau} D_j \frac{dJ_j}{d\tau} - s(J_j - J_{1-j}) = k_j(J_j - B_v/2), \quad (1)$$

$$j = 0.1.$$

Here, $d\tau = n_e \sigma_T dz$ is the Thompson optical depth, $(4\pi/c)J_j$ is the spectral density of the radiation energy for the j th mode, $B_v = B_v(T)$ is the Planck function, and D_j and k_j are the diffusion and absorption coefficients that depend on frequency and are expressed in Thompson units $(n_e \sigma_T)^{-1}$ and $n_e \sigma_T$. The diffusion coefficient also depends on angle Θ_B between the magnetic field and the normal to the surface:

$$D_j = D_j^{\parallel} \cos^2 \Theta_B + D_j^{\perp} \sin^2 \Theta_B, \quad (2)$$

$$D_j^{\parallel} = \int_0^1 \frac{\mu^2 d\mu}{K_j(\mu)}, \quad D_j^{\perp} = \int_0^1 \frac{(1-\mu^2) d\mu}{K_j(\mu)}.$$

Here, μ is the cosine of the angle between the wavevector \mathbf{n} and the magnetic field direction and K_j is the sum of the absorption and scattering coefficients,

$$K_j(\mathbf{n}) = K_j^a(\mathbf{n}) + K_j^s(\mathbf{n}). \quad (3)$$

Integral emissivity K_j^s can be expressed as follows:

$$K_j^s(\mathbf{n}) = \sum_{i=0}^1 \int d\mathbf{n}' k_{ji}^s(\mathbf{n}, \mathbf{n}'). \quad (4)$$

The scattering coefficient and the emissivity averaged over angles have the form

$$k_j = \frac{1}{4\pi} \int d\mathbf{n} K_j^a(\mathbf{n}), \quad (5)$$

$$s = \frac{1}{4\pi} \int d\mathbf{n} d\mathbf{n}' K_{01}^s(\mathbf{n}, \mathbf{n}').$$

The system of equations must be solved under the following boundary conditions [4]:

$$2D_j \frac{dJ_j}{d\tau} = J_j \quad (\tau = 0), \quad (6)$$

$$J_j \rightarrow \frac{1}{2} B_v(T) \quad (\tau \rightarrow \infty).$$

System (1) must also be supplemented with the conditions of conservation of the flux and local thermodynamic equilibrium. System of equations (1) is solved by the method of successive approximations [4]. First, we must specify the temperature profile. Usually, the profile for the gray atmosphere (or Eddington profile [11]) is chosen,

$$T(\tau_R) = T_{\text{eff}} \left[\frac{3}{4} \left(\tau_R + \frac{2}{3} \right) \right]^{1/4}, \quad (7)$$

where T_{eff} is the effective temperature, $d\tau_R = K_R d\tau$ is the Rosseland optical depth, and K_R is the Rosseland mean of the absorption coefficient (see [11]),

$$K_R^{-1} = \frac{3\pi}{4\sigma T^3} \int d\nu \frac{D_0 + D_1}{2} \frac{\partial B_\nu}{\partial T}. \quad (8)$$

For a given temperature profile, coefficients k_j and s , as well as $B_v(T)$ in Eq. (1) are preset. In such a formulation, this is the problem of determining a formal solution to transport equations. Solving this problem, we determine $J_j(\tau)$ and calculate the correction to the temperature profile:

$$\Delta T = -\frac{f}{df/dT}; \quad f = \int dv \sum_{j=0}^1 k_j \left(J_j - \frac{1}{2} B_v \right). \quad (9)$$

This gives a new temperature profile $T(\tau)$, and we again solve system of equations (1) until relative error $\Delta T/T$ becomes smaller than the preset value. The obtaining of the formal solution is the most time consuming stage of computations.

Further, we consider the construction of the formal solution by the Galerkin method with finite elements.

2. GALERKIN METHOD WITH FINITE ELEMENTS

Let us first consider a particular case of system (1), in which $k_0 = k_1 = k$, $s = 0$, and $D_0 = D_1 \equiv D$. Accordingly, $J_0 = J_1 \equiv J$ in this case, and the equation for J assumes the form

$$\frac{d}{d\tau} D \frac{dJ}{d\tau} = k \left(J - \frac{B_v}{2} \right). \quad (10)$$

In the Galerkin method with finite elements, the solution to Eq. (10) on segment $[\tau_0, \tau_N]$ is sought in the form (see, for example, [12])

$$J(\tau) = \sum_{i=0}^N \psi_i(\tau) \tilde{J}_i, \quad (11)$$

where $\tilde{J}_i = J(\tau_i)$ is the value of the sought function at nodal points of the grid, N is the number of elements, and $\psi_i(\tau)$ are piecewise smooth polynomials. For simplicity, we first consider linear polynomials. All arguments can easily be generalized for higher-order polynomials. On element $[\tau_i, \tau_{i+1}]$, linear polynomials have the following form in local coordinates of element ξ (Fig. 1):

$$\begin{aligned} \psi_i(\xi) &= \frac{1}{2}(1 - \xi), & \psi_{i+1}(\xi) &= \frac{1}{2}(1 + \xi), \\ \xi &= 2 \frac{\tau - 0.5(\tau_i + \tau_{i+1})}{\tau_{i+1} - \tau_i}, & \xi &\in [-1, 1]. \end{aligned} \quad (12)$$

Further, we denote the operator in Eq. (10) by $L(J)$ and write the equation in weak form:

$$\int_{\tau_0}^{\tau_N} \psi_j(\tau) L(J(\tau)) d\tau = 0. \quad (13)$$

Substituting expansion (11) into this equation, integrating the term with the second derivative by parts, and transposing summation and integration, we obtain

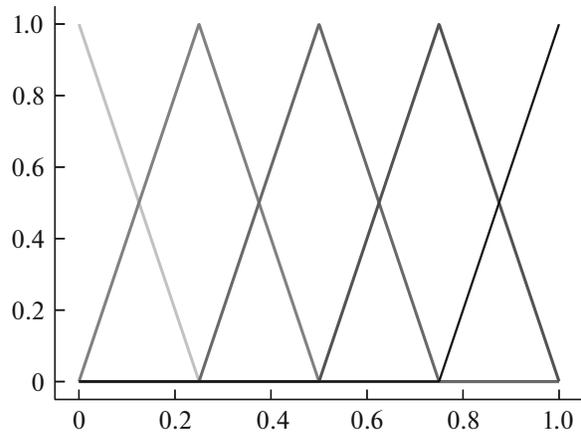


Fig. 1. Linear functions $\psi_i(\tau)$ ($i = 0-5$) for a four-element grid on segment $\tau \in [0, 1]$.

$$\begin{aligned} -\psi_j D \frac{dJ}{d\tau} \Big|_{\tau_0}^{\tau_N} + \sum_{i=1}^N \tilde{J}_i \left[\int_{\tau_0}^{\tau_N} D \frac{d\psi_i}{d\tau} \frac{d\psi_j}{d\tau} d\tau + \int_{\tau_0}^{\tau_N} k \psi_i \psi_j d\tau \right] \\ = \int_{\tau_0}^{\tau_N} k \frac{B_v}{2} \psi_i d\tau. \end{aligned} \quad (14)$$

It is also convenient to represent coefficients k , D , and B_v in form (11). Substituting the expansions for the coefficients and integrating, we obtain the system of algebraic equations for \tilde{J}_i :

$$\mathbf{M}\tilde{\mathbf{J}} = \mathbf{g} \equiv \mathbf{G}\mathbf{f}, \quad (15)$$

where \mathbf{f} is a vector with components $1/2\tilde{k}_i\tilde{B}_i$, and \tilde{k}_i and \tilde{B}_i are the values of coefficient k and the Planck function at the nodal points of the grid. Let us disregard for the time being the boundary conditions and the extra-integral term in Eq. (14). Then, matrix \mathbf{M} for a four-element grid has the form

$$\begin{bmatrix} M_{00}^0 & M_{01}^0 & 0 & 0 & 0 \\ M_{10}^0 & M_{11}^0 + M_{11}^1 M_{12}^1 & 0 & 0 & \\ 0 & M_{21}^1 & M_{22}^2 + M_{22}^2 & M_{23}^2 & 0 \\ 0 & 0 & M_{32}^2 & M_{33}^3 + M_{33}^3 & M_{34}^3 \\ 0 & 0 & 0 & M_{43}^3 & M_{44}^3 \end{bmatrix}. \quad (16)$$

Matrix \mathbf{G} can be constructed analogously. Superscripts denote the number of the element in the grid on which the given matrix element is calculated; matrices \mathbf{M}^e and \mathbf{G}^e for element e have the form

$$\begin{aligned} \mathbf{M}^e &= \frac{2}{\Delta\tau_e} [\tilde{D}_e \mathbf{g}^0 + \tilde{D}_{e+1} \mathbf{g}^1] + \frac{\Delta\tau_e}{2} [\tilde{k}_e \mathbf{n}^0 + \tilde{k}_{e+1} \mathbf{n}^1], \\ \mathbf{G}^e &= \frac{\Delta\tau_e}{2} \mathbf{n}, \end{aligned} \quad (17)$$

$$\Delta\tau_e = \tau_{e+1} - \tau_e,$$

where \tilde{D}_e is the value of function D at nodal point e , and elements of matrices $\mathbf{g}^{0,1}$, $\mathbf{n}^{0,1}$, and \mathbf{n} can be expressed in terms of the integrals

$$\int_{-1}^1 \psi_i' \psi_j' \psi_k d\xi, \quad \int_{-1}^1 \psi_i \psi_j \psi_k d\xi, \quad \int_{-1}^1 \psi_i \psi_j d\xi. \quad (18)$$

The integrals can be evaluated easily, and the matrices have the form

$$\begin{aligned} \mathbf{g}^0 &= \frac{1}{6} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, & \mathbf{g}^1 &= \frac{1}{3} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \\ \mathbf{n}^0 &= \frac{1}{2} \begin{bmatrix} 1 & 1/3 \\ 1/3 & 1/3 \end{bmatrix}, & \mathbf{n}^1 &= \frac{1}{2} \begin{bmatrix} 1/3 & -1/3 \\ 1/3 & 1 \end{bmatrix}, \\ \mathbf{n} &= \frac{1}{3} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}. \end{aligned} \quad (19)$$

The above expressions were obtained disregarding the extra-integral term in Eq. (14) and correspond to the homogeneous Neuman conditions at points τ_0 and τ_N . Let us now consider boundary conditions (6). In accordance with the Dirichlet condition on the right boundary, the value of \tilde{J}_N is fixed. Accordingly, we can exclude the last equation as well as the value of the extra-integral term in expression (14) at point τ_N . Taking into account the condition on the left boundary, we can write the value of the extra-integral term at point τ_0 in the form

$$\psi_0 D \frac{dJ}{d\tau_{\tau_0}} = D \frac{dJ}{d\tau_{\tau_0}} = \frac{J}{2|_{\tau_0}} = \frac{\tilde{J}_0}{2}. \quad (20)$$

Accordingly, with allowance for the boundary conditions, matrix \mathbf{M} and vector \mathbf{g} assume the form (for a four-element grid)

$$\begin{aligned} \mathbf{M} &\rightarrow \begin{bmatrix} M_{00}^0 + 1/2 & M_{01}^0 & 0 & 0 \\ M_{10}^0 & M_1^0 + M_{11}^1 & M_{12}^1 & 0 \\ 0 & M_{21}^1 & M_{22}^1 + M_{22}^2 & M_{23}^2 \\ 0 & 0 & M_{32}^2 & M_{33}^2 + M_{33}^3 \end{bmatrix}, \\ \mathbf{g} &\rightarrow \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 - M_{34}^3 \frac{\tilde{B}_4}{2} \end{bmatrix}. \end{aligned} \quad (21)$$

It is important to note that \mathbf{M} is a symmetric and positive-definite matrix (i.e., $\mathbf{M}^T = \mathbf{M}$ and $\mathbf{x}^T \mathbf{M} \mathbf{x} > 0$ for $\forall \mathbf{x} \neq 0$).

We can now easily write the algebraic system for Eqs. (1):

$$\begin{bmatrix} \mathbf{L}_0 & \mathbf{S} \\ \mathbf{S} & \mathbf{L}_1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{J}}_0 \\ \tilde{\mathbf{J}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{Gf}_0 \\ \mathbf{Gf}_1 \end{bmatrix}, \quad (22)$$

where matrices \mathbf{L}_j and \mathbf{S} correspond to the following operators:

$$L_j(J) = -\frac{d}{d\tau} D_j \frac{dJ}{d\tau} + (k_j + s)J, \quad S(J) = -sJ. \quad (23)$$

Let us see how the construction and solution of system (22) can be realized on graphic maps.

3. REALIZATION ON CUDA PLATFORM

Let us consider the realization of the solution of the formal problem on the CUDA platform. For this purpose, the CUDA C language was used. The description of the CUDA platform and CUDA C language can be found in [13]. For developing the application for graphic maps with the CUDA technology, it is necessary to install the CUDA toolkit package that contains nvcc compiler as well as some utilities for software debugging and useful libraries.

System of equations (22) was solved by the conjugate-gradient method. This method can be formulated as the successive approximation method for solving system $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} is a symmetric positive-definite matrix. Appendix A contains the corresponding pseudocode.

It can be seen from the algorithm given in Appendix A that at each iteration, matrix \mathbf{A} must be multiplied by a vector and calculate direct products and sums of vectors. The multiplication of the matrix by a vector is the most time-consuming operation.

Let us first see how a matrix of type (21) must be multiplied to make use of advantages of multicore processors of the type of graphic maps. First, we can multiply each of the 2D matrices \mathbf{M}^e by the corresponding 2D vector. These operations can be performed independently. Further, we must compose the resultant vector from 2D vectors. The realization of this procedure in the CUDA C language for the matrix appearing when linear elements are used is described in Appendix B as an example. For operations of the scalar product and vector addition, the cuBLAS library was used, which is a part of the CUDA toolkit package.

For calculating the coefficients of linear system (22), we must first calculate the vectors of values of coefficients k_j , D_j , s , and B_v in system of equations (1) at points $[\tau_0, \tau_N]$ of the grid. In the simplest case, when the coefficients are defined by simple expressions, the coefficients can easily be calculated directly using CUDA. By way of example, let us consider an isothermal atmosphere. The propagation of radiation in such an atmosphere is described by Eq. (10) with coefficients $D = \text{const}$ and $k = \kappa\tau$. Then, the analytic solution to Eq. (10) can be expressed in terms of the Airy function $\phi(\tau)$ [14, 15]. We assume that $\kappa = 1$ and $D = 1$; taking in account boundary conditions (6), we can then write

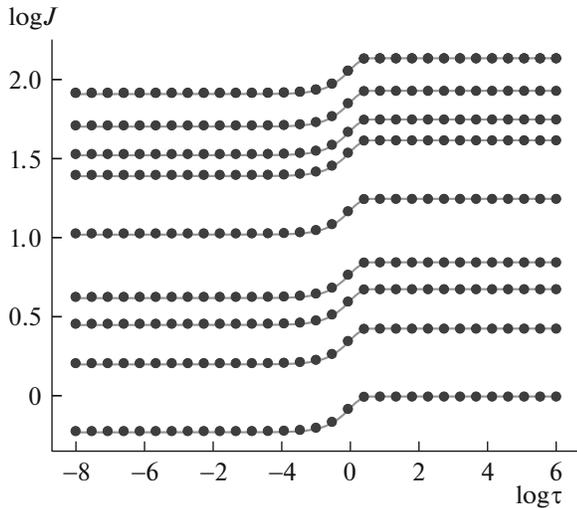


Fig. 2. Results of calculation for an isothermal atmosphere in comparison with the analytic solution (solid curves).

$$J(\tau) = \frac{B_v}{2}(1 - C_\phi(\tau)), \quad (24)$$

$$C = \phi(0) - 2\phi'(0).$$

In Fig. 2, analytic solution (24) is compared with the result of calculation for a grid of 30 points in $\tau = 10^{-8}$ to 10^6 and 9 points in energy $E = 10^{-2}-1$. The number of points was chosen for convenience of the representation. In actual calculations, it is better to use a larger number of points. In the general case, it is convenient to use tables with preliminarily calculated values in a certain grid in temperature and photon energy and then to find the required values by interpolation. The interpolation was also effectively realized on CUDA.

4. CONCLUSIONS

The CUDA-aided realization of the search for the formal solution to the equations for the radiation transport in a strongly magnetized plasma makes it possible to accelerate the computation of radiation intensity from a small element of the neutron star surface with preset values of temperature, magnetic field, and the angle between the magnetic field direction and the normal to the surface. As mentioned in the Introduction, about a million such calculations are required. Let us briefly consider the procedure that follows after such calculations. A model of the neutron star can be specified as a map of distributions of temperature and magnetic field over the surface. For comparison with the observation data, it is necessary to determine the elements of the surface the radiation from which is detected by a remote observer for the preset directions (to the observer, of the rotational axis of the star, of the magnetic moment if we are dealing with a dipole field) and the phase of rotation. Then, these directions can be varied so as to describe the

observations in the best way. The elements of the surface that are seen by a remote observer can be determined, for example, by the method of ray tracing, which can also be effectively realized on graphic maps.

ACKNOWLEDGMENTS

The author is grateful to P. Alekseev for valuable remarks that helped to improve this article.

This study was supported by the Russian Foundation for Basic Research (project no. 16-3260129 mol_a_dk).

APPENDICES

A. ALGORITHM FOR CONSTRUCTING A FORMAL SOLUTION

We describe below the algorithm for solving system of equations (1) by the Galerkin method with finite elements. The algorithm for the conjugate-gradient method used for solving system (22) is also given.

Algorithm 1. Solution of system of equations (1) on graphic maps

- 1: Calculation of k_j , D_j , s , and B_v at grid points by interpolation. The values at each point are calculated simultaneously.
- 2: Calculation of \mathbf{L}_j , \mathbf{S} , and \mathbf{G} . Calculations are performed independently for each element of the grid.
- 3: Solution of system (22) by the conjugate-gradient method.

Algorithm 2. Conjugate-gradient method for solving system $\mathbf{Ax} = \mathbf{b}$

- 1: $n \leftarrow 0$,
 - 2: $\mathbf{x}_0 \leftarrow 0$,
 - 3: $\mathbf{r}_0 \leftarrow \mathbf{b}$,
 - 4: $\mathbf{p}_0 \leftarrow \mathbf{b}$,
 - 5: $\delta_0 \leftarrow \mathbf{r}_0^\top \mathbf{r}_0$,
 - 6: while $\delta_n > \epsilon^2 \delta_0$,
 - 7: $\alpha_n \leftarrow \delta_{n-1} / (\mathbf{p}_{n-1}^\top \mathbf{A} \mathbf{p}_{n-1})$,
 - 8: $\mathbf{x}_n \leftarrow \mathbf{x}_{n-1} + \alpha_n \mathbf{p}_{n-1}$,
 - 9: $\mathbf{r}_n \leftarrow \mathbf{r}_{n-1} - \alpha_n \mathbf{A} \mathbf{p}_{n-1}$,
 - 10: $\delta_n \leftarrow \mathbf{r}_n^\top \mathbf{r}_n$,
 - 11: $\beta_n \leftarrow \delta_n / \delta_{n-1}$,
 - 12: $\mathbf{p}_n \leftarrow \mathbf{r}_n + \beta_n \mathbf{p}_{n-1}$,
 - 13: $n \leftarrow n + 1$,
 - 14: end while.
-

B. REALIZATION OF MULTIPLICATION OF THE MATRIX APPEARING IN THE METHOD OF LINEAR ELEMENTS BY A VECTOR IN THE CUDA C LANGUAGE

The code for multiplication of a matrix of type (21) by a vector is given in the CUDA C language.

```

__global__ void multMbyX(float *M, float *X,
float *Y) {
    __shared__ float cache [N_elem] [2];
    int tid=threadIdx.x+blockIdx.x*blockDim.x;
    int cache Index=threadIdx.x;
    if (tid<N_elem) {
        cache[tid][0]=M[tid][0]*X[tid]+M[tid] [1]*X[tid+1];
        cache[tid] [1] = M[tid] [1]*X[tid]+M[tid] [2]*X[tid+ 1];
    }
    __syncthreads ();
    int i = N_elem - 1;
    while (i > 0) {
        Y[i] = cache[i][0] + cache[i - 1] [1];
        i = i - 1;
    }
}

```

REFERENCES

1. D. G. Yakovlev and C. J. Pethick, *Annu. Rev. Astron. Astrophys.* **42**, 169 (2004).

2. M. C. Miller, arXiv:1312.0029 [astro-ph.HE].
3. G. Greenstein and G. Hartke, *Astrophys. J.* **271**, 283 (1983).
4. I. A. Shibanov, V. E. Zavlin, G. G. Pavlov, and J. Ventura, *Astron. Astrophys.* **266**, 313 (1992).
5. G. G. Pavlov, Y. A. Shibanov, J. Ventura, and V. E. Zavlin, *Astronomy & Astrophysics* **289**, 837 (1994).
6. G. G. Pavlov, Y. A. Shibanov, V. E. Zavlin, and R. D. Meyer, in *The Lives of the Neutron Stars*, Ed. by M. A. Alpar, U. Kizilöglu, and J. van Paradijs (Kluwer Academic, Dordrecht, 1995), p. 71.
7. V. E. Zavlin, G. G. Pavlov, Y. A. Shibanov, and J. Ventura, *Astron. Astrophys.* **297**, 441 (1995).
8. Y. A. Shibanov and V. E. Zavlin, *Astron. Lett.* **21**, 3 (1995).
9. A. D. Kaminker, G. G. Pavlov, N. A. Silantev, and I. A. Shibanov, *Astrofizika* **18**, 283 (1982).
10. W. Nagel, *Astrophys. J.* **236**, 904 (1980).
11. D. Mihalas, *Stellar Atmospheres*, 2nd ed. (W. H. Freeman and Co., San Francisco, 1978).
12. C. A. J. Fletcher, *Computational Galerkin Methods* (Springer, Berlin, 1984).
13. J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming* (Addison-Wesley, 2010).
14. Y. B. Zel'dovich and N. I. Shakura, *Astron. Zh.* **46**, 225 (1969).
15. V. V. Zhelezniakov, *Astrophys. Space Sci.* **77**, 279 (1981).

Translated by N. Wadhwa