

Parallel Computation of the Functions Constructed with R-operations using CUDA

Roman A. Uvarov

*Podgorny Institute for Mechanical Engineering Problems of NAS of Ukraine,
2/10 Dm. Pozharsky str., Kharkiv, Ukraine*

uvarov@ipmach.kharkov.ua

Abstract. *The developed software system "AutoOmega" for automation of modeling various physical fields in two-dimensional domains of complex geometrical shapes is represented. Automation is achieved by solving the direct and inverse problems of automatic translation of geometric information into analytical and programming form, supporting the meshless natural elements method with R-functions method as well as using CUDA technology for parallel computing the single analytical function describing the geometrical object in two- and three-dimensional domains.*

Keywords

Parallel Computing, the Theory of R-functions, CUDA Programming, Natural Elements Method.

1 Introduction

The problem of adequate implementation of parallelism is acute mainly for the purpose of receiving the results of solving computational problems in time as less as possible. It is based on the fact that the task is divided into a set of smaller tasks that can be solved simultaneously. There are many different types of solutions, both physical and software, and one of them is CUDA technology featured in Nvidia's GPUs of 8 series and above.

The **aim of this work** is to parallelize the computation in the software package that implementing the automation process enhanced by R-functions for researching the fields of different physical nature in two-dimensional domains with complex geometries using the meshless method of natural elements (Natural Elements Method) for solving boundary value problems.

2 Theoretical Part

The mathematical models of physical and mechanical fields are the problems for partial differential equations under certain boundary and initial conditions. A specific feature of such fields is their dependence not only on the nature of physical laws taken into account by the relevant equations, but also on the shape and the relative position of objects, in which the fields are invoked. The presence of two different kinds of information (analytical and geometrical) in the formulation of boundary value problems is the actual problem, while creating the methods and algorithms for their solutions [1, 2].

Diversity, complexity and uniqueness of the forms of geometric objects determined the relevance of the inverse problem of analytical geometry, formulated by R. Descartes: given a geometrical object, its equation should be written. The theory of R-functions developed by Academician V. L. Rvachev [1] allows to solve this problem. It makes possible to describe the geometrical objects by functions whose values determine the boundary, interior and geometrical properties of the object. Construction of equations of the geometrical object's boundary requires the definition of support functions and a logical formula, which allows to obtain the equation analytically with an appropriate choice of the R-operations.

In general, the following set of R-operations is used for constructing the functions ω_i

$$\begin{aligned}
 x \wedge_{\alpha} y &= \frac{1}{1+\alpha} \left(x + y - \sqrt{x^2 + y^2 - 2\alpha xy} \right), \\
 x \vee_{\alpha} y &= \frac{1}{1+\alpha} \left(x + y + \sqrt{x^2 + y^2 - 2\alpha xy} \right), \\
 \bar{x} &= -x,
 \end{aligned} \tag{1}$$

where $\alpha = \alpha(x, y)$, $-1 < \alpha \leq 1$.

In addition, the property of function's normalization may be implemented:

$$\left. \frac{\partial \omega_i}{\partial \nu_i} \right|_{\omega_i=0} = \pm 1, \tag{2}$$

where ν_i is a normal vector to $\mathcal{A}\Omega_i$.

At present times, the R-functions method (RFM) is actively developed and applied. The method [1] for constructing the equations of a complex geometrical object is a good technology foundation for the automation of the composition process for these equations. A universal logical formula describing a composite geometrical form with the R-operations (1) and property of function's normalization (2) is proposed [3]:

$$\omega = \left(\bigvee_{i=1}^N \omega_i^{in} \right) \wedge_{\alpha} \left(\bigwedge_{i=N+1}^{N+K} \bar{\omega}_i^{ex} \right), \tag{3}$$

where ω_i^{in} is a function of geometrical object's boundary that describes its interior, while $\bar{\omega}_i^{ex}$ is a function that describes its exterior.

It is sufficient to indicate kinds of used standard geometric objects and parameters that define their positions and sizes.

2.1 «AutoOmega» system

The software system «AutoOmega» was developed with a constructive apparatus of the theory of R-functions. Input information in this system includes kinds of used standard geometrical objects, geometric parameters that determine their positions and sizes, rotation angles, and the widths of "the rings", in a case of a doubly connected domain. Support functions in form of normalized equations as well as predicate and analytical functions of the composite object (3) are automatically generated with this information. This method has been called the standard primitives method [3].

Primitives in «AutoOmega» are divided into groups characterized by a growing number of defining parameters and interface to set them. So, to set the function of:

- a circle, 3 parameters are needed (center coordinates and radius);
- an ellipse, 4 parameters are needed (center coordinates and half sizes of axes);
- a rectangle, 4 parameters are needed (center coordinates and half sizes of sides);
- a triangle, a segment of circle, and a sector of circle, 6 parameters are needed (coordinates of three vertices);
- a convex quadrangle, 8 parameters are needed (coordinates of four vertices);
- a regular polygon, 4 parameters are needed (coordinates of circle's center and its radius as well as a number of vertices).

Constructive apparatus of the theory of R-functions is used for construction of each primitive. Let's consider some of the primitives. For example, the regular polygon. It will be built with parameters of its circumcircle and a number of polygon vertices.

Then, during the transition to polar coordinates we use the following formulas

$$\rho = \sqrt{(x - x_{rp})^2 + (y - y_{rp})^2},$$

$$\theta = \begin{cases} \arctan \frac{y - y_{rp}}{x - x_{rp}}, \text{ when } x > 0, y > 0 \\ \arctan \frac{y - y_{rp}}{x - x_{rp}} + \pi, \text{ when } x < 0, y > 0 \\ \arctan \frac{y - y_{rp}}{x - x_{rp}} + \pi, \text{ when } x < 0, y < 0 \\ \arctan \frac{y - y_{rp}}{x - x_{rp}} + 2\pi, \text{ when } x > 0, y < 0 \end{cases}$$

where x_{rp} , y_{rp} are coordinates of regular polygon's circumcenter.

Let's introduce the parameter

$$\mu = \frac{8}{\pi \cdot n} \cdot \sum_{i=1}^m (-1)^{i+1} \sin \frac{\theta \cdot n \cdot (2 \cdot i - 1)}{2 \cdot (2 \cdot i - 1)^2}$$

where n is a number of polygon's vertices, and m is a number of terms.

Normalized function of a regular polygon is described by:

$$\omega_{rp} = -(\rho \cdot \cos \mu - r_{rp}), \tag{4}$$

where r_{rp} is a radius of polygon's circumcircle.

Normalized function of "the ring" of a regular polygon is described by:

$$f_1 = -(\rho \cdot \cos \mu - r_{rp}), \quad f_2 = -(\rho \cdot \cos \mu - (r_{rp} - d_{rp})),$$

$$\omega_{rrp} = f_1 \wedge_{\alpha} \overline{f_2}, \tag{5}$$

where d_{rp} is a width of "the ring" of a regular polygon.

To rotate the angle φ_{rp} of a regular polygon it is needed to replace variables x and y involved in the calculation of the parameters used in equations (4) - (5) with variables x_{rrp} and y_{rrp} by the following formula:

$$x_{rrp} = \cos \varphi_{rp} \cdot (x - x_{rp}) - \sin \varphi_{rp} \cdot (y - y_{rp}) + x_{rp},$$

$$y_{rrp} = \sin \varphi_{rp} \cdot (x - x_{rp}) + \cos \varphi_{rp} \cdot (y - y_{rp}) + y_{rp}.$$

The dialog for the regular polygon (Figure 1) allows the user to automatically construct a regular polygon by given coordinates of the center and the radius of the circle.

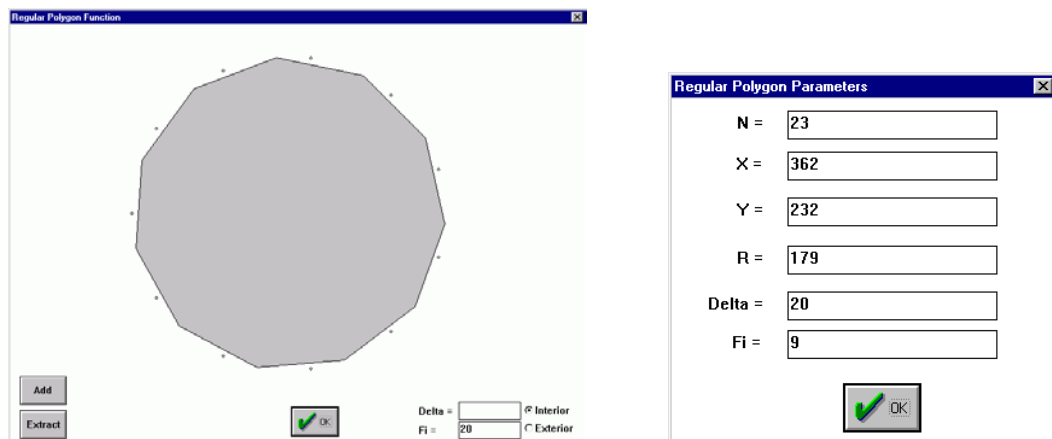


Fig.1. Dialog and pre-dialog for editing the regular polygon

The default number of vertices of the polygon is set to three, but can be changed to a higher number (limited to 50, as a visual polygon becomes a circle) in the pre-dialog (Figure 1).

The translator included in «AutoOmega» system allows to automatically generate the lines of code in problem-oriented language for a specific scientific system such as MATLAB or POLYE-RL based on the parameters, and to run it to execution for solving a specific boundary value problem. Translator dialog box is divided into two panels. The left pane shows the contents of the AO-file in format of «AutoOmega» system, which stores settings of primitives. The right pane is filled with the content of the automatically generated file in a problem-oriented language. Parameters of primitives are used for construction of the support functions and the functions of a complex domain. The result is a picture of the level lines of the field in the domain, built in the «AutoOmega» system.

This system was customized for the problems of researching the temperature fields on the board with heat sources emitting the heat with known intensity [4], and problems of calculating the metal products with a known profile of complex shape, such as a rail (Figure 2), Larsen rabbet.



Fig.2. Model of a rail constructed in «AutoOmega» and calculated in POLYE-RL

2.2 Natural Elements Method

The meshless method of natural elements (NEM) was selected as a method for solving boundary value problems in «AutoOmega». It can be described as a method that uses Sibson and non-Sibsonian functions as interpolants. The mesh is not required for the interpolants construction.

Initially, natural neighbor interpolation of the elements were introduced by Sibson for smoothing the scattered data[5]. It is based on the Voronoi cells T_i defined as

$$T_i = \{x \in R : d(x, x_i) < d(x, x_j) \forall j \neq i\},$$

where $d(x_i, x_j)$ is a distance (a Euclidean norm) between x_i and x_j .

Sibson functions or functions of natural neighbor elements are defined as ratio of the polygonal areas of the Voronoi diagram. Hence,

$$\Phi_i(x) = \frac{A_i(x)}{A(x)},$$

where $A(x) = T_x$ is a total area of the Voronoi cell for x and $A_i(x) = T_i \cap T_x$ is an area of partial coverage of i -node by Voronoi cell (Figure 3).

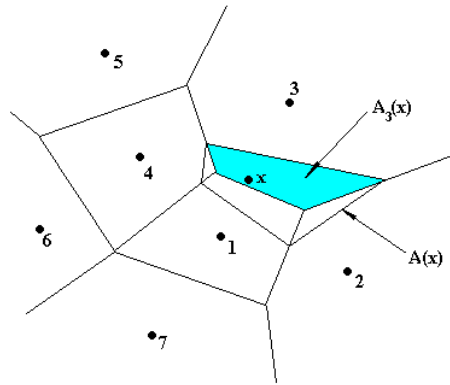


Fig.3. Construction of Sibson function

Key differences that separate the NEM from the finite elements method (FEM) lie in the design and numerical implementation of shape functions, and in the methodology used for assembling the stiffness matrix. The algorithm proposed by Watson [6] for constructing the shape function was implemented. All other steps are common to both methods.

2.3 CUDA appliance

CUDA (an acronym for Compute Unified Device Architecture) for Nvidia's graphics processing units of 8-series or higher was used for parallel computing in the «AutoOmega» system.

The following parameters were used:

- CPU – Intel Core i7-3770K;
- GPU – NVIDIA GeForce GTX 680 (8 Multiprocessors x 192 CUDA Cores/MP = 1536 CUDA Cores);
- Microsoft Visual Studio 2010 Express Edition (C++);
- NVIDIA CUDA Toolkit v5.0.

Workflow of the program with CUDA had the following specifics:

- Copying the data from the main memory to the memory of the GPU;
- CPU allows the GPU to perform the task;
- GPU executes in parallel on each of its cores;
- Copying the result from the GPU memory to the main memory.

A simplest program sample to calculate the R-conjunction of two real arguments using CUDA is stated below.

```
#include <iostream>

__device__ float R_ANDem(float a, float b ) {
    float alpha = 0.5;
    return 1.0/(1+alpha)*(a+b-sqrt((a*a+b*b-2*alpha*a*b)*1.0));
}

__global__ void R_AND( float a, float b, float *c ) {
    *c = R_ANDem( a, b );
}

int main( void ) {
    float c;
    int *dev_c;

    HANDLE_ERROR( cudaMalloc( (void**)&dev_c, sizeof(int) ) );

    R_AND <<<1,1>>> ( 2, 7, dev_c );
```

```

HANDLE_ERROR(cudaMemcpy( &c, dev_c, sizeof(float), cudaMemcpyDeviceToHost ) );
printf( "Result = %5.2f\n", c );
cudaFree( dev_c );

return 0;
}

```

The calculation of a single analytic function, describing the geometrical object and consisting of a number of support functions, in each point of specified area needs the parallelization. It is obvious that the efforts for computing the function of a complex geometrical object will increase with the number of the support functions if such a process is sequential.

The idea of parallel computing in this case is based on the fact that the problem of computing the function of a complex geometrical object is divided into a set of tasks for computing the support functions, which can be solved simultaneously. In this case, the idea was implemented with the Nvidia's CUDA.

For comparison purposes, calculations of various functions with R-operations and without them were carried out in the two-dimensional domain (suitable for «AutoOmega» system) and in three-dimensional domain (suitable for "currently in development" «AutoOmega 3D» system that is a successor of «AutoOmega» system's ideas). Calculations were performed using both the CPU and GPU. The results are presented in Table 1.

Tab.1. Total time of calculation of the functions in each point of the three-dimensional domain of 4000 x 4000 x 40 points in size, in seconds

| <i>Function</i> | <i>CPU</i> | <i>GPU</i> |
|--|------------|------------|
| Unbounded cylinder | 2.44 | 0.01 |
| R-conjunction of scalar values (alpha = 0.5) | 11.89 | 0.01 |
| R-conjunction of two cylinders | 14.65 | 0.01 |
| Prism with a convex quadrangle as a base | 12.32 | 0.01 |
| R-disjunction of two spheres | 13.9 | 0.01 |
| Parallelepiped | 14.5 | 0.01 |

3 Conclusion

In general, Nvidia's CUDA technology is oriented on solving the dynamical problems of graphical and computational kinds. In this article its appliance to stage of computing the single analytic function constructed based on the constructive apparatus of the theory of R-functions was analyzed. Obtained results allow highlight its benefits and limitations. In general, the use of CUDA technology has significantly reduced the time spent on computing.

References

- [1] V. L. Rvachev Theory of R-functions and its several applications. – Kiev: Nauk. dumka, 1982. – 552 p. [in Russian]
- [2] V. L. Rvachev, A. P. Slesarenko Algebra of logic and integral transforms in boundary value problems – Kiev: Nauk. dumka, 1977. – 287 p. [in Russian]
- [3] K. V. Maksimenko-Sheyko, A. M. Matsevity, A. V. Tolok, T. I. Sheyko Constructive apparatus of R-functions method for automation of constructing the equations of complex geometrical objects. *Bulletin of Zaporozhye State University*, 2: 66-76, 2005. [in Russian]
- [4] K. V. Maksimenko-Sheyko, R. A. Uvarov, T. I. Sheyko Mathematical and computer modelling of heat modes of radioelectronics tools by R-functions method. *Electronic modeling*. 31(4): 79-87, 2009. [in Russian]
- [5] R. Sibson A vector identity for the Dirichlet tessellation. *Math. Proc. Cambridge Philos. Soc.* 87: 151-155, 1980.
- [6] N. Sukumar, B. Moran, T. Belytschko The Natural Element Method in Solid Mechanics. *Int. J. Numer. Meth. Engng.* 43: 839-887, 1998.