# USING STRUCTURED ADAPTIVE COMPUTATIONAL GRID FOR SOLVING MULTIDIMENSIONAL COMPUTATIONAL PHYSICS TASKS

Prof. Boris Rybakin, Institute of Mathematics and Computer Science
rybakin@math.md
Dr. Peter Bogatencov, RENAM Association
bogatencov@renam.md

HP-SEE

High-Performance Computing Infrastructure for South East Europe's Research Communities

**HP-SEE** - www.hp-see.eu

The HP-SEE initiative is co-funded by the European Commission under the FP7 Research Infrastructures contract no. 261499

## In this presentation we consider:

- The algorithm and a program for solving multidimensional problems represented by differential equations with partial derivatives adopted for using SEE regional HPC resources.

- The algorithm is based on the AMR method - Adaptive Mesh Refinement of the computational grid.

- Utilization of AMR method can significantly improve the resolution of the difference grid in areas of high interest, and from other side to accelerate the processes of the multi-dimensional problems calculating. To improve the accuracy of the calculations is necessary to choose a sufficiently small grid (with a small cell size). This leads to the drawback of a substantial increase of computation time.

## Importance of the proposed method:

- Despite the permanent computer power growth the finding of effective algorithms and approaches for elaborating adequate models remain the key factor for solving of complex practical problems of large dimensions.
- We use one of the effective methods that allow developing optimized applications and speeding up the process of complicated models execution. The method based on adaptive refinement of computational mesh – AMR (Adaptive Mesh Refinement). This approach was initially proposed for organization of consecutive computations (James M. Stone, Michael L. Norman. ZEUS-2D: A Radiation Magnetohydrodynamics Code for Astrophysical Flows in Two Space Dimensions. I. The Hydrodynamic Algorithms and Tests. The Astrophysical Journal Supplement Series, 80:753-790, 1992 June) and only in 2001- 2002 parallelized realizations became available (D.S. Balsara, C.D. Norton. Highly parallel structure adaptive mesh refinement using parallel language-based approaches. Elsevier, Parallel Computing (27), 2001)
- The grid refinement is performed only in the areas of interest of the structure, where e.g. the shock waves are generated, or a complex geometry or other such features exist. Applying AMR the computing time is greatly reducing and the execution of the application on the resulting sequence of nested, decreasing nests can be parallelized.

We review using of AMR method for the solution of two- and three- dimensional tasks of gas dynamics that have obvious practical interest. These solutions can be applied to many nowadays problems the calculation of the aerodynamics of aircraft and the air flow of cars, a large number of other problems of mathematical modeling – the calculation of the flow of blood through the vessels, the calculations of the heart valves, etc.

## Description of the solving problem :

The elaborated algorithm and application we consider to used for computer simulation of the gravitational collapse of stars with masses ranging from 7 to 70 solar masses. This process leads to the formation of a supernova and requires using of non-oscillating schemes of high resolution.

It should be taken into account that the density of a collapsing star changes by many orders - from $10^{14}$ g/sm$^3$ at the center of a neutron star to a density of a rarefied gas on the boundary of the stellar envelope. It is therefore necessary to create a grid, the size of which depends on the density, that is, the cells in the center should have a minimum size, and should increase with distance from the center.

## Formulation of the problem

The system of equations of gravitational gas dynamics, which describe the process of collapse of a star, can be written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho v_i) = 0,$$

$$\frac{\partial (\rho v_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho v_i v_j + P\delta_{ij}) = -\rho \frac{\partial \varphi}{\partial x_i},$$

$$\frac{\partial e}{\partial t} + \frac{\partial}{\partial x_i}[(e+P)v_i] = -\rho v_i \frac{\partial \varphi}{\partial x_i}.$$

In these equations, the value of the gravitational potential is determined from the Poisson equation

$$\Delta\varphi = 4\pi G\rho$$

The equation of state is used in the form of:

$$e = 1/2\rho v^2 + \varepsilon$$

In the above equations: ρ - the density, v - velocity, P - pressure, ε - specific internal energy, e - full energy, t - time, $x_i$ - spatial coordinates, G = $6{,}67 \cdot 10^{-11}$ $kg^{-1}м^3s^{-2}$ - constant gravitational potential.

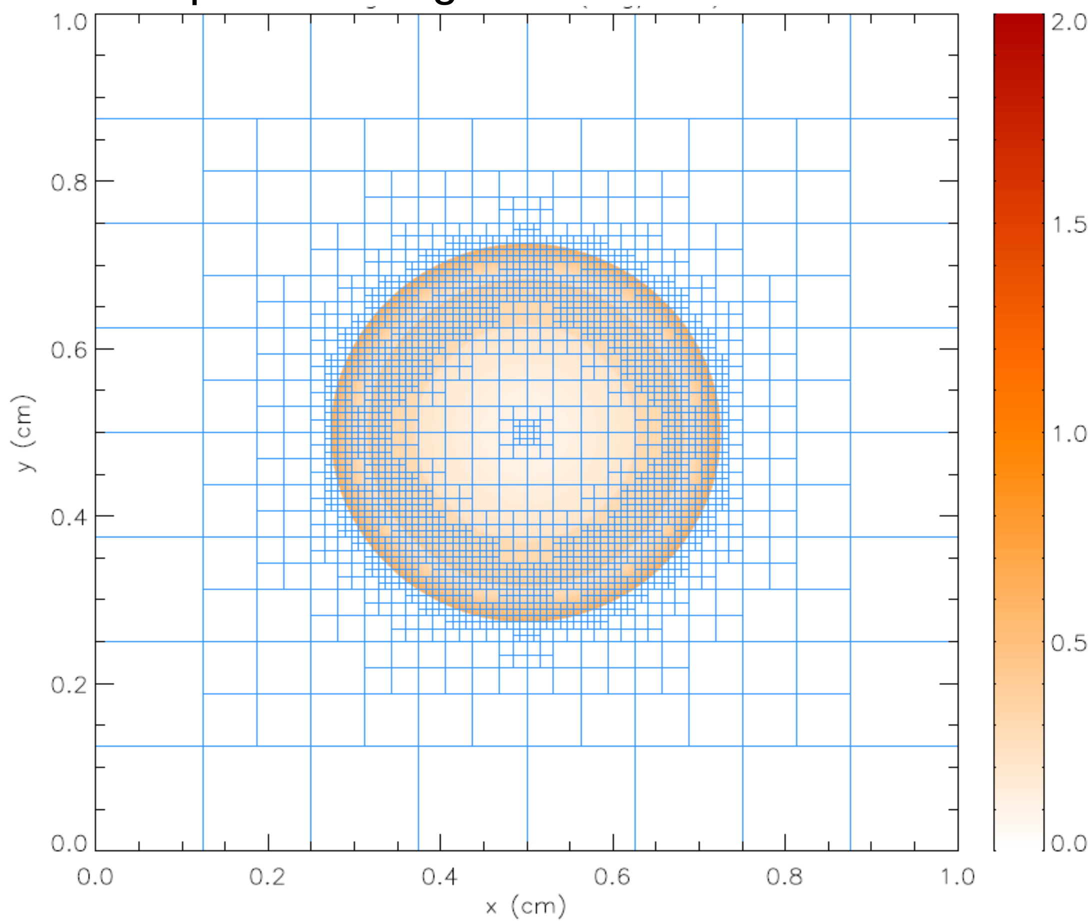
**Using of the AMR Method.**

In the resolving problem, the density of matter in a collapsing star is changed by many orders. On the surface density of the material is almost equal to the density of interstellar matter. At the center of a neutron star the order of matter density increases up to g/ $10^{14}$ It $sm^3$ herefore necessary to create a grid, the size of which depends on the density, that is, the cells in the center should have a minimum size, and should increase with distance from the center.

Such kind of a grid can be constructed by using the mechanism of AMR (building nested "fine" mesh). In the center of the computational area (level L0) allocate cube with mesh size is 2 times smaller (in each coordinate) than the initial size of the cell. In the center of the cube level L1 built cube with reduced cell size L2, etc. The initial computational area and each subcube has the same dimensions M3, where M varies in the range from 64 to 1024 cells. Size of the computational area and the number of nested levels is determined by the parameters of the problem, the size of available memory and computing installation performance.

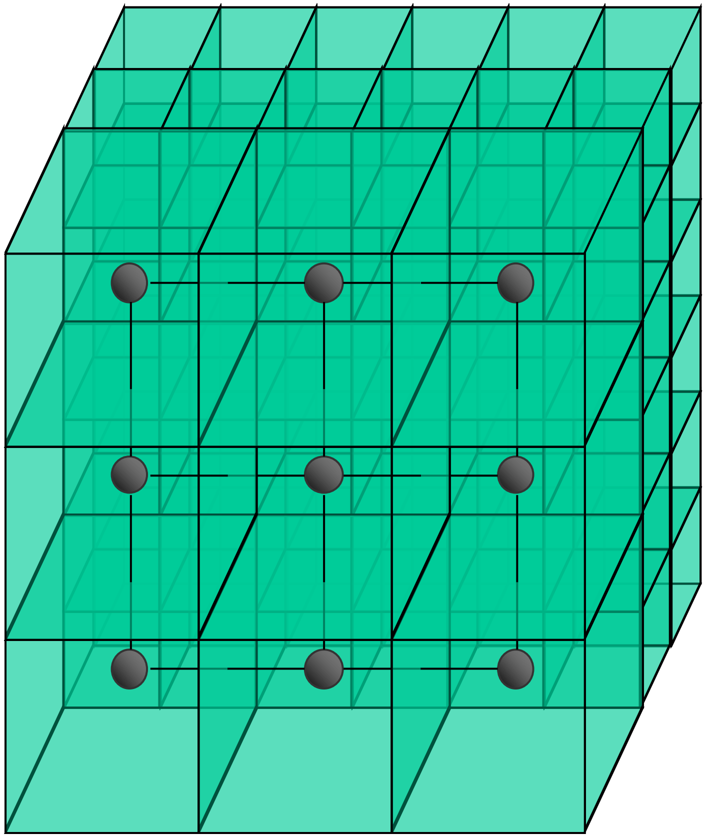
## AMR Method.

If necessary, in the areas with large gradients of pressure, temperature, etc., using the AMR method we can build sophisticated grid:

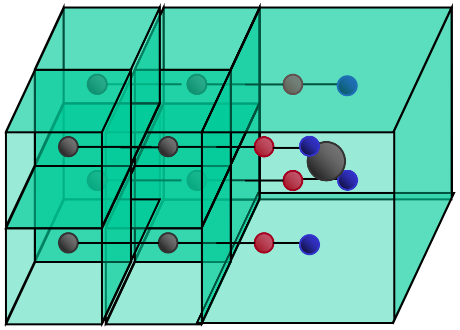


**AMR Method.**

At the foreground a coarse grid in the centers - balls. In the background is a fine mesh, which is 2 times less in each coordinate

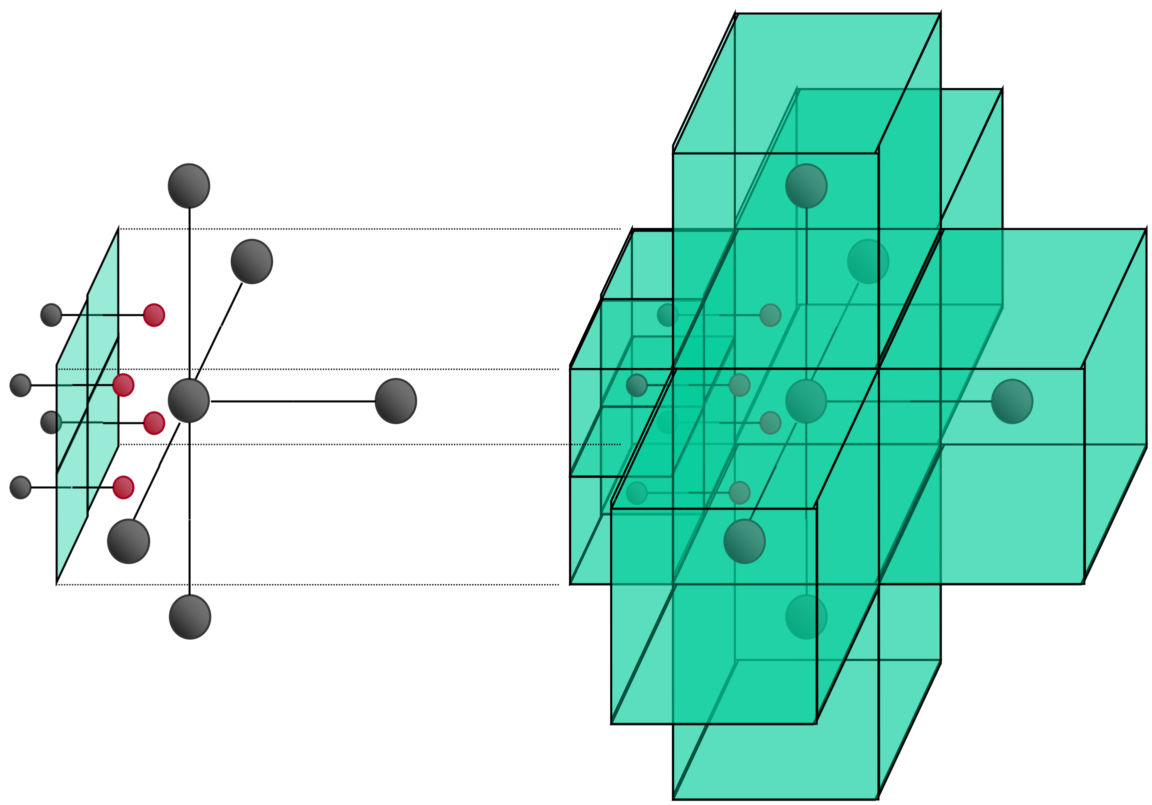


Interpolation of red dots with black and gray dots.



# USING STRUCTURED ADAPTIVE COMPUTATIONAL GRID FOR SOLVING MULTIDIMENSIONAL COMPUTATIONAL PHYSICS TASKS

Three-dimensional pattern to convert a "coarse" grid to the "fine"



**Program fragment:**

The solution of Poisson's equation was found by the Gauss-Seidel and successive overrelaxation methods.

Program is written in Fortran 90 (95) in MS Visual Studio 2008 (2010). For all meshes was created structure SingleGrid, which allowed to use a uniform way to store and process all the grids, independently of the level of nesting. Relative constructor and destructor were created. Thus in Fortran class that allows uniform process for all of the grids was created.

```
type single_grid
    integer:: level
      integer:: maxIters
      real(4):: tolerance, epsilon, dx
      real(4),dimension (nx,nx,nx):: u, dFi
      real(4),dimension (nx,nx)::    b1, b2, b3, b4, b5, b6
  end type single_grid
     interface new
   module procedure single_grid_init
  end interface
```

**Program fragment (**continuation)**:**

```
interface delete
    module procedure single_grid_delete
  end interface
      contains
   subroutine single_grid_init(this,level,maxIters,tolerance,  &
             epsilon, dx, u, dFi, b1, b2, b3, b4, b5, b6)
       real(4),dimension (nx,nx,nx):: u, dFi
       real(4),dimension (nx,nx)::   b1, b2, b3, b4, b5, b6
        type (single_grid),pointer::grid,this
       type (single_grid) sg(10)
       allocate (grid)
       grid%level        = level
       grid%maxIters  = maxIters
       grid%tolerance  = tolerance
       grid%epsilon     = epsilon
       grid%dx            = dx  !1.0/(nx-1)
       grid%u              = u
       grid%dFi           = dFi
```

**Program fragment (**continuation)**:**

```
      grid%b1        = b1
      grid%b2        = b2
      grid%b3        = b3
      grid%b4        = b4
      grid%b5        = b5
      grid%b6        = b6
      this => grid
   end subroutine single_grid_init
```

## Calculation algorithm

The solution of the three-dimensional Poisson equation has been tested on 5 levels of AMR nesting.

In the test a homogeneous sphere of radius R and density ρ for the equation for the gravitational potential (4) is considered:

(3)

$$\Phi(r) = \begin{cases} 2\pi G\rho_0(R^2 - r^2/3) & \text{if} \quad r \le R \\ \frac{4}{3}\pi G\rho_0 R^3/r & \text{if} \quad r > R \end{cases},$$

(4)

$$\nabla\Phi(r) = \begin{cases} -\frac{4}{3}\pi G\rho_0 r & \text{if} \quad r \le R \\ -\frac{4}{3}\pi G\rho_0 R^3/r^2 & \text{if} \quad r > R \end{cases};$$

The computational area is filled with values $\Phi$ from (3) in the sphere of radius R, outside sphere values are (3) are specified with bottom line. Then the values of the gravitational potential are calculating in the three-dimensional formulation. The numerical solution was calculated for AMR hierarchy levels from one to five.

Figure 1 shows the analytical solution for the gravitational potential, obtained from equation (4).
Figure 2 - the numerical results,
Figure 3 - two-dimensional cross-section of the X-axis of Fig.2

Figure 1.

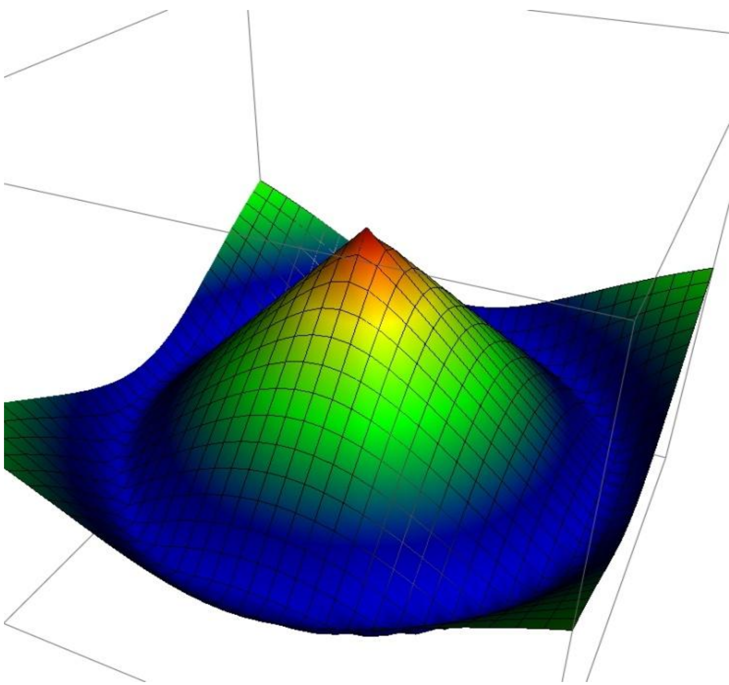


Figure 2.

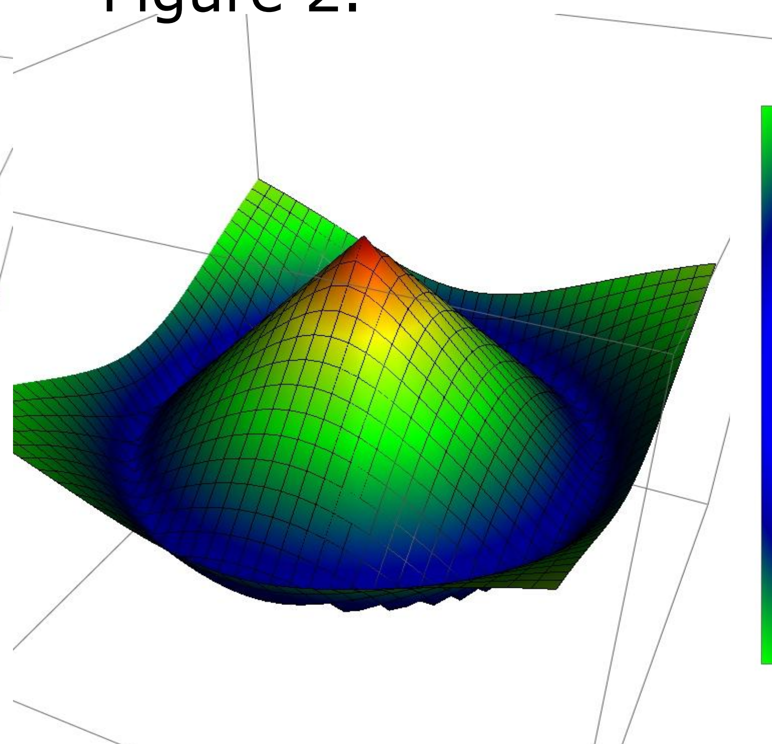
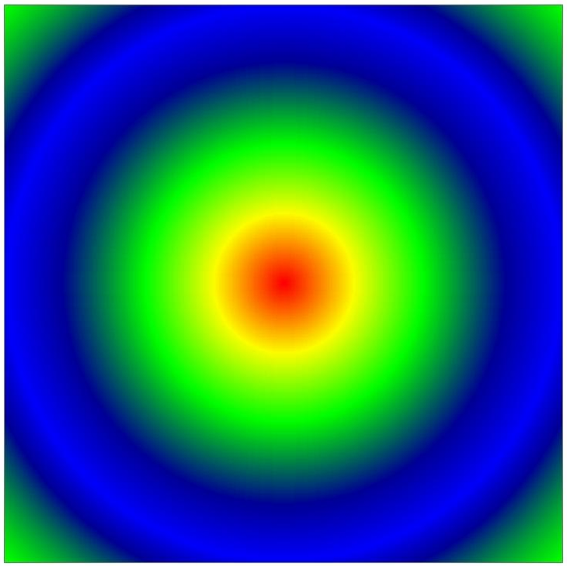


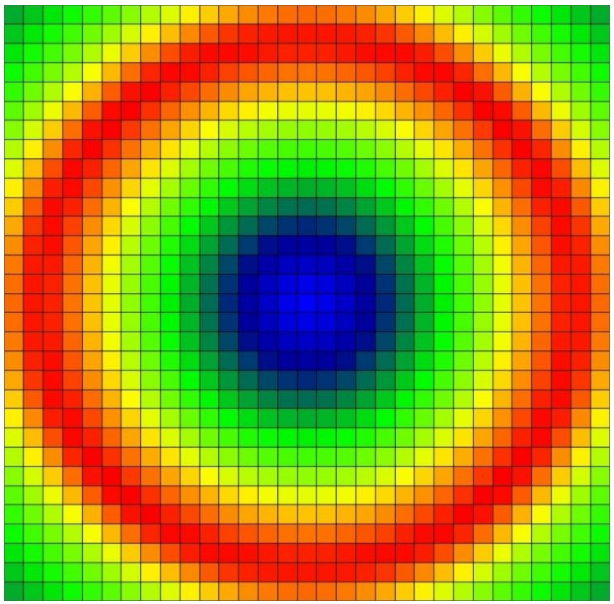
Figure 3



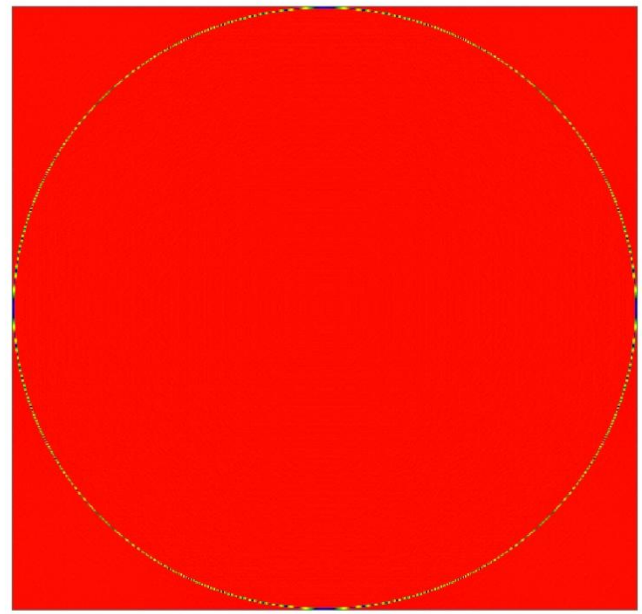
## Graphic representation of the error

The difference between the analytical and numerical solutions:

for the grid of 32x32 and two levels of grid



for the grid of 512x512 and 5-level grid



It should be mentioned the significant refinement of result and consequently decreasing of the error by four orders. These results show the importance of applying the AMR method for raising accuracy of the solutions of multidimensional partial differential equations.

**AMR_PAR 64-bit application**

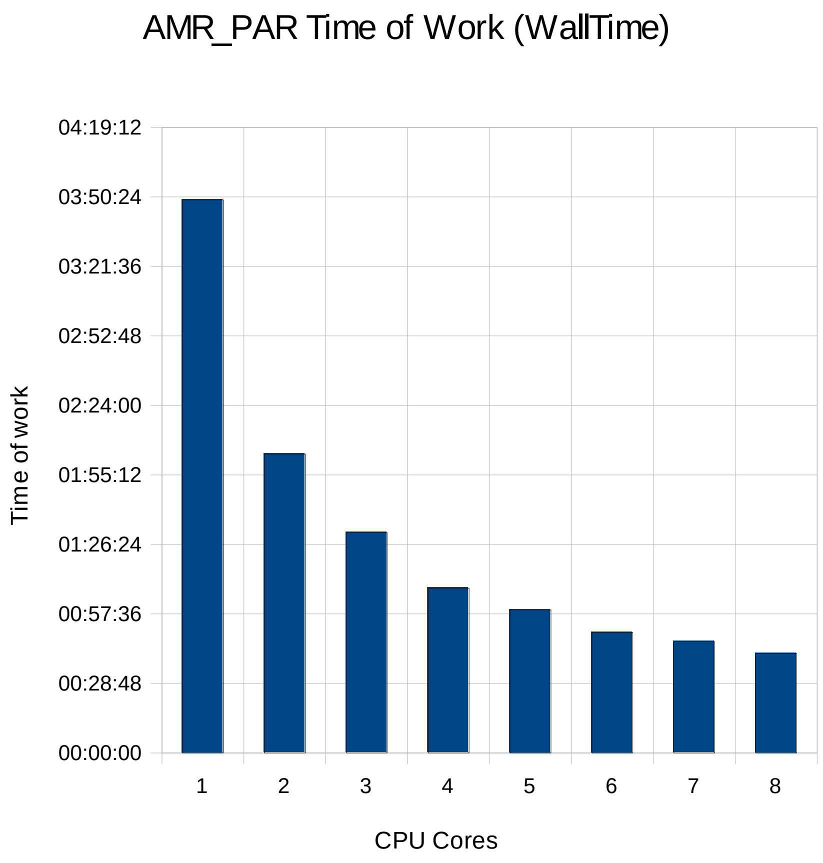
AMR_PAR application (Parallel algorithm and program for the solving of continuum mechanics equations using Adaptive Mesh Refinement), being developed in the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova.

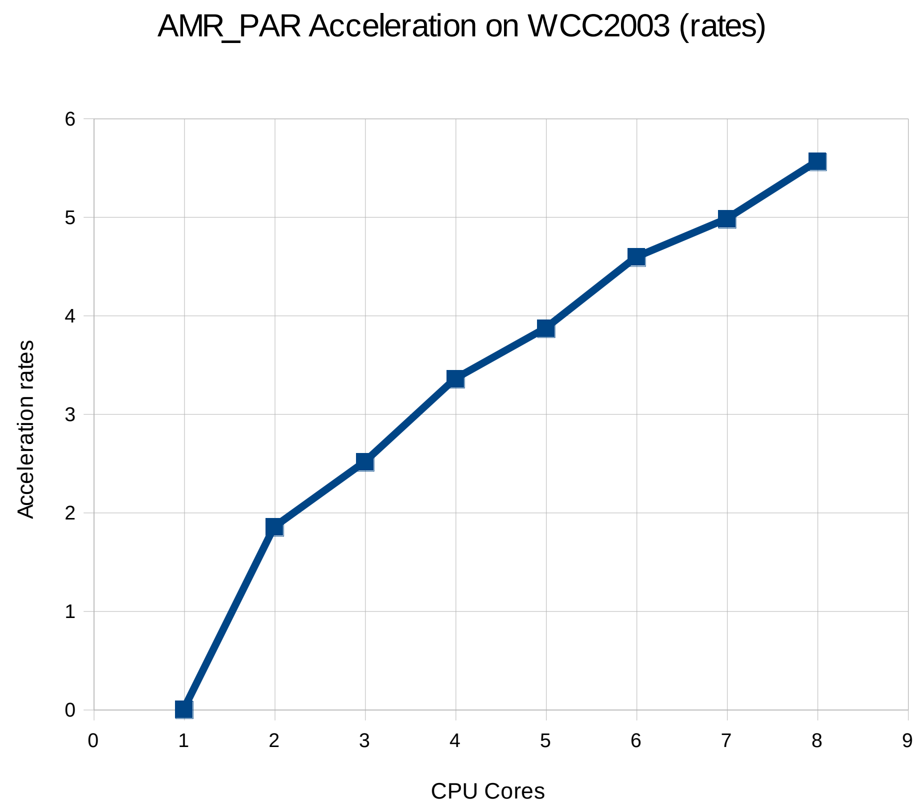
AMR_PAR 64-bit application was developed in MS Visual Studio 2010. We use the programming language Fortran 90. Program uses object-oriented approach, which is available in Fortran 90.

AMR_PAR application is using OpenMP mode and was locally tested on small AMR grids (128x128x128 cells, 5 layers) on MS Windows Compute Cluster 2003 (4-8 Nodes, 4-22 Cores (QuadCore Intel Xeon E5335, E5310 CPU)

# Results of AMR_PAR application execution

on the WCC2003 cluster of IMI ASM in OpenMP mode,
cores from 1 to 8 (2 x QuadCore Intel Xeon E5310, 1600 MHz, 8 GB of RAM)



Porting activities: Application was ported to Linux, compiled and tested on front-end computer HPCG cluster located at IICT of Bulgarian Academy of Sciences and at the front-end computer of SGI UltraViolet 1000 supercomputer at NIIFI, located in Pecs, Hungary.

Scalability studies: For HPCG cluster located at the Institute of Information and Communication Technologies of Bulgarian Academy of Sciences maximum grid dimension for 5 layers is 384x384x384, approximate time of calculations – 5 hours, optimal number of cores – 8.

Calculations for 5-7 levels and grid dimensions sizes more than 384x384x384 require up to 3 Tb of RAM.

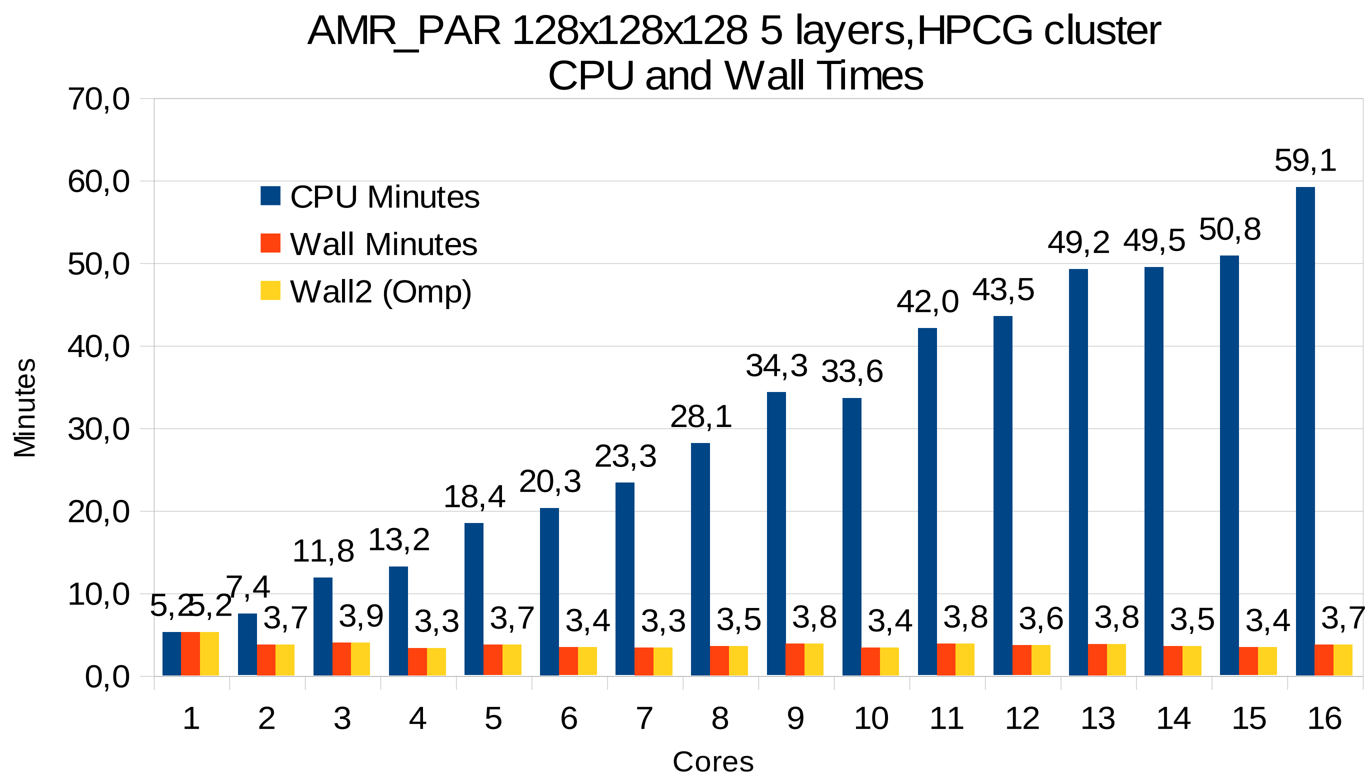
Benchmarking activities:

Acceleration and Run Time dependences from CPU cores.

For 128x128x128 dimension best number of cores — 4.

4 cores - walltime - 3,3 min, CPU time -13,2 min.

16 cores - walltime - 3,7 min, CPU time - 59,1 min



Other development/optimization issues:

# Calculated requirements of computational resources for the current OpenMP version of AMR_PAR application

| Dimension | Layers | Max Iteration per level | Cores | RAM Gb | CPU minutes | WallTime minutes |
|---|---|---|---|---|---|---|
| 128x128x128 | 5 | 200000 | 4 | 0,789 | 28 | 3,5 |
| 256x256x256 | 5 | 200000 | 4 | 5,972 | 273 | 68 |
| 256x256x256 | 5 | 200000 | 8 | 6,062 | 527 | 66 |
| 256x256x256 | 5 | 200000 | 12 | 6,068 | 807 | 68 |
| 384x384x384 | 5 | 200000 | 8 | 19,2 | 2110 | 270 |
| 448x448x448 | 5 | 200000 | 8 — 16 | 37,7 | ~ 4500 | ~ 500 |
| 512x512x512 | 5 | 200000 | 8 — 16 | ~ 55,6 | ~ 130 hours | ~ 17 hours |
| 1024x1024x1024 | 5 | 200000 | 16 — 32 | ~ 415 | ~ 2000 hours | ~ 248 hours |
| 2048x2048x2048 | 5 | 200000 | 32 — 64 | ~ 3250 | ~ 1200 days | ~ 154 days |

Foreseen activities of the application development:

- **For further optimization of AMR_PAR application, we plan collecting statistics of calculations' acceleration dependences from different number of cores - up to 64 (or more). It is necessary to produce investigations to find optimal number of cores for fastest calculations for large-scale grid dimensions. As a result of this research we plan to modify application to use OpenMP more effectively.**

- **Next step is to run application using HP-SEE regional resources for large-scale grid dimensions – up to 2048x2048x2048, 5-7 layers. After obtaining results of the modified application execution, it will be possible to make new benchmarking (due to long time of forecast calculations) and propose new recommendations for application optimization. The results of calculations will be visualized in 2-D images and 3-D models.**

# Publications:


- RYBAKIN, B.; SIDER, N. Numerical modeling of multidimensional problems of gravitational gas dynamics with high resolution schemes. Buletinul AŞM, ser. Matematica. 2010, nr. 1, 92-102. ISSN 1024-7696.
- ŞIDER, N.I. An adaptive mesh refinement technique for gas dynamics. Conferinţa Internatională a Tinerilor Cercetători, editia a VIII-a (11-12 noiembrie 2010), Chisinau, Moldova. p.36.
- B. Rybakin. “Parallel programming for graphics accelerators.” Under edition by Academician V.B. Betelin. Moscow, SIISI RAS, 2011, 261 p.
- A. Altuhov, P. Bogatencov, A. Golubev, N. Iliuha, G. Secrieru. Current State of Distributed Computing Infrastructure Deployment in Moldova. “Networking in Education and Research”, Proceedings of the 10th RoEduNet IEEE International Conference, Iasi, Romania, June 23-25, 2011, pp. 69-72. ISSN 2247-5443.;
- B. Rybakin. Modeling of III-D Problems of Gas Dynamics on Multiprocessing Computers and GPU. Book of Abstracts of the 23rd International Conference on Parallel Computational Fluid Dynamics, May 16-20, Barcelona, p.44-45.
- ILIUHA Nicolai, ALTUHOV Alexei, BOGATENCOV Peter, SECRIERU Grigore, GOLUBEV Alexandr. SEE-HP project – providing access to the Regional High Performance computing infrastructure. International Workshop on Intelligent Information Systems. Proceedings IIS. 13-14 September, Chisinau, IMI ASM, 2011, pp. 183-186. ISBN 978-9975-4237-0-0
- RYBAKIN, B.; STRELNIKOVA, E.; SECRIERU, G.; GUTULEAC, E. Computer simulation of dynamic loading of fluid-filled shell structures. În: Rezumatele Conferenţei „Mathematics & Information technologies: Research and Education (MITRE-2011)”, Chişinău, 22-25 August 2011, 102. ISBN 978-9975-71-144-9
- RYBAKIN, B.; SECRIERU, G.; GUTULEAC, E. Research of the intense-deformed condition of elastic-plastic shells under the influence of intensive dynamic loadings. In: The 19-th Edition of the Annual Conference on Applied and Industrial Mathematics- CAIM, Iaşi, September 22-25, 2011, p. 50, ISSN 1841-5512.
- B.P. RYBAKIN. Modeling of III-D problems of gas dynamics on multiprocessing computers and GPU. In: Computers & Fluids, Elsevier, January 2012, http://dx.doi.org/10.1016/j.compfluid.2012.01.016.
- G.Secrieru, P.Bogatencov, E.Gutuleac, N.Iliuha. “Access to the regional HPC resources and strategy of their development”. The 20-th conference on applied and industrial mathematics dedicated to academician Mitrofan M.Ciobanu, Chisinau, August 22-25, 2012, p.200-201

**Thank You for Your attention !**

**Prof. Boris Rybakin,**
**IMI ASM**
***rybakin@math.md***