COMPUTATIONAL METHODS IN APPLIED MATHEMATICS, Vol. 10 (2010), No. 1, pp. 87–94

# POTENTIALITIES OF THE ROBUST MULTIGRID TECHNIQUE

S. I. MARTYNENKO[1]

**Abstract** — The present paper discusses the parallelization of the robust multigrid technique (R$_{\mathbb{M}}$T) and the possible way of applying this to unstructured grids. As opposed to the classical multigrid methods, the R$_{\mathbb{M}}$T is a trivial method of parallelization on coarse grids independent of the smoothing iterations. Estimates of the minimum speed-up and parallelism efficiency are given. An almost perfect load balance is demonstrated in a 3D illustrative test. To overcome the geometric nature of the technique, the R$_{\mathbb{M}}$T is used as a preconditioner in solving PDEs on unstructured grids. The procedure of auxiliary structured grids generation is considered in details.

**2000 Mathematics Subject Classification:** 65N06, 65N55.

**Keywords:** robust multigrid methods, multigrid structure, parallelisation.

## Introduction

Multigrid algorithms are a fast and efficient method for solving a wide class of integral and partial differential equations. The main field of application of multigrid solvers is large-scale problems where the computational efficiency is critical. For many problems, it is possible to prove that the execution time of the classical multigrid methods (C$_{\mathbb{M}}$M) is asymptotically optimal.

Parallelization of C$_{\mathbb{M}}$M is carried out in a standard fashion by dividing the domain into subdomains (one for each processor). Each processor is then responsible for updating the unknowns associated within its subdomain only. However, the parallel efficiency of C$_{\mathbb{M}}$M can degrade due to the coarse grid smoothing. Situations can be reached where the number of processors exceeds the number of coarse grid points. As a result, some processors stand idle during these computations. One way of overcoming the poor computation-to-communication ratio on coarse grids is multiple coarse grid corrections. The best known algorithms were proposed by P. Frederickson and O. McBryan [2], W. Hackbusch [3] and W. Mulder [7]. Nevertheless the choice of appropriate multigrid components and their efficient parallel implementation is highly problem-dependent in C$_{\mathbb{M}}$M.

Another multigrid method based on multiple coarse grid corrections is the robust multigrid technique (R$_{\mathbb{M}}$T) where the transfer operators are problem-independent for black box applications [5]. Although the R$_{\mathbb{M}}$T is not optimal in computational work, the absence of interpolation and presmoothing jointly with the powerful coarse grid correction strategy makes it possible to use the R$_{\mathbb{M}}$T for solving many applied (non)linear problems in the black box manner.

[1] *P.I. Baranov Central Institute of Aviation Motors*, 2 Aviamotornaya St., 111116 Moscow, Russia. E-mail: martyn_s@mail.ru

The present paper gives details of RMT parallelization on coarse levels. The multigrid structure of the RMT gives a direct way of obtaining full parallelism independent of the smoothing procedure in addition to the black box robustness.

Another important problem is the development of multigrid methods for unstructured grids. Although now algebraic multigrid algorithms (AMG) are used for such applications, it is desirable to solve PDEs on (un)structured grids in a unified manner.

## 1. Parallelization of the robust multigrid technique

The following properties of the coarse grids of the RMT are attractive for parallel implementation [5]:

1) all coarse grids of the same level have no common vertices and faces

$$G_n^L \cap G_m^L = \varnothing\,, \quad n \neq m\,, \quad L = 1, \ldots, L^+,$$

therefore, the smoothing iterations can be performed on coarse grids in parallel;

2) the fixed number of coarse grids on each level allows to predict the number of processors for efficient parallelization of the RMT;

3) the almost the same number of vertices and faces on the coarse grids ($\approx 3^{-NL}\bar{\mathbf{N}}$) results in a perfect load balance;

4) the absence of interpolation errors makes it possible to use well-parallelizable (weak) smoothers on the finest grid.

Since the $L$-th level consists of $3^{NL}$ grids, the number of processors should be $3^{Nk}$, where $k = 1, \ldots, L^+$ is the order of parallelism. Consequently, we can consider the simplest case $k = 1$ or the first order parallelism. In this case, the number of processors is $p = 3^N, N = 2, 3$.

The parallel implementation of the RMT is subdivided into incomplete and complete parallelization. The incomplete parallelization uses the parallel properties of the multigrid structure without parallelization of the smoothing iterations on the finest grid. Actually, the incomplete parallelization is intended for estimating the minimum speed-up and efficiency.

**1.1. Incomplete parallelization.** For the sake of simplicity, we design a hypothetical multiprocessor for incomplete parallelization of the RMT. The multiprocessor consists of $3^N$ processors for parallelization of the smoothing iterations of the coarse grids and the main processor for the smoothing iterations of the finest grid (Fig. 1.1). In fact, each of the $3^N$ processors can be used instead of the main processor. As a result, the main processor will not accounted at minimum speed-up and efficiency estimation.

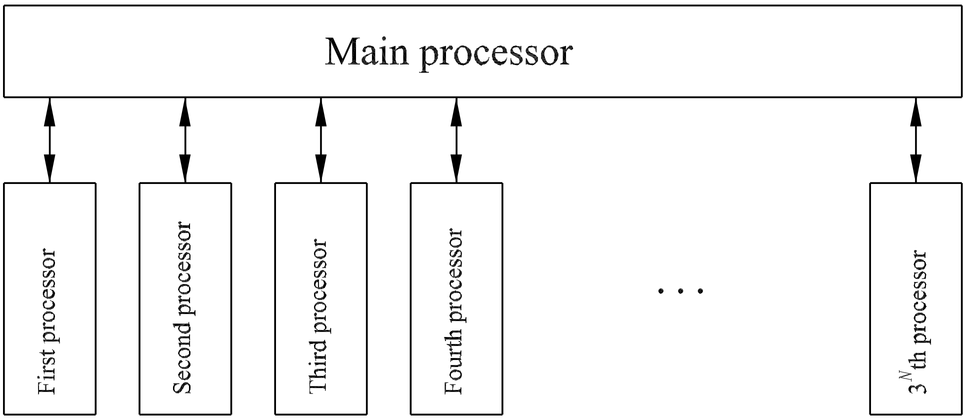

Fig. 1.1. Hypothetical multiprocessor for incomplete parallelization

Since all coarse grids of the first level have no common vertices and faces, the grid distribution between the processors in the first order parallelism is shown in Fig. 1.2. Figure 1.3 shows the incomplete parallelization of the multigrid iterations of the RMT. Efficient parallelization of the CMM is hampered by the poor computation-to-communication ratio on the coarse grids. As opposed to the CMM, the RMT is a trivial method for parallelization on coarse grids.

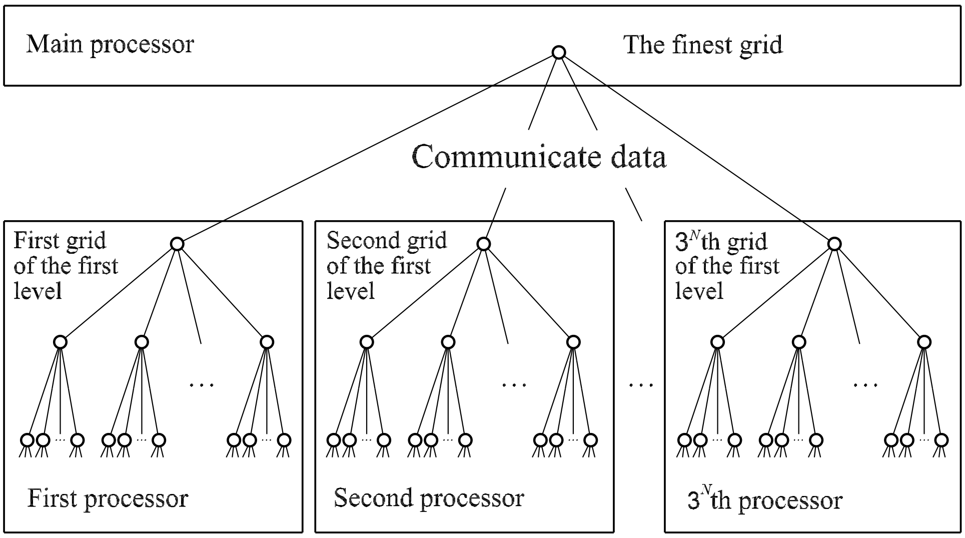

Fig. 1.2. Coarse grid distribution between the processors

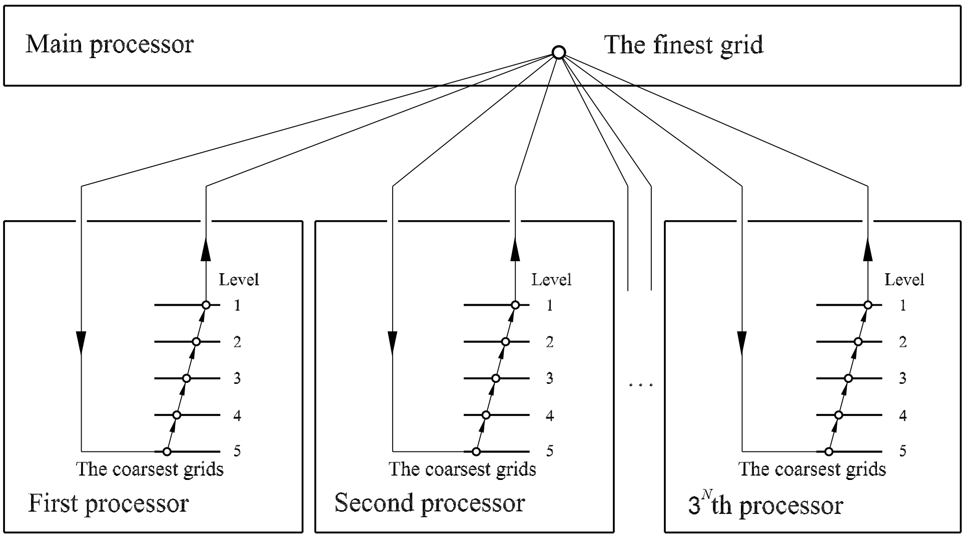

Fig. 1.3. Incomplete parallelization of the multigrid iteration

The numerical test is intended to illustrate a perfect load balance in the parallelization of the smoothing iterations of the coarse grids. We start from the 3D Dirichlet boundary value problem ($N = 3$) for the Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \frac{1}{3} \exp\left(\frac{x+y+z}{3}\right)$$

with the exact solution $u_a(x, y, z) = \exp((x+y+z)/3)$ in a unit cube. The five-level structure ($L^+ = 4$) with a $201 \times 201 \times 201$ finest grid ($h = 1/200$) is used in the test, the smoother is a variant of the preconditioned conjugate gradient (PCG) method offered in [4].

Let $T_i$ be the execution time for the PCG-smoothing on the $i$th processor ($i = 1, 2, \dots, 3^N$). We define the average execution time $\langle T \rangle$ in the first order parallelism as

$$\langle T \rangle = \frac{1}{3^N} \sum_{i=1}^{3^N} T_i \, .$$

The load unbalance ($\delta_{\max}$) can be defined by

$$\delta_{\max} = \max_{i=1,\dots,3^N} |\delta_i| \, , \qquad \text{where} \quad \delta_i = \frac{\langle T \rangle - T_i}{\langle T \rangle} \, .$$

Figure 1.4 shows the distribution of the relative execution time ($\delta_i$) on the coarse levels ($L = 1, 2, 3, 4$) in the first multigrid iteration starting from iterand zero. The results of the test show that $\delta_{\max} \approx 1\%$, i.e., the RMT has an almost perfect load balance. It should be emphasized that the parallelization is smoother-independent.

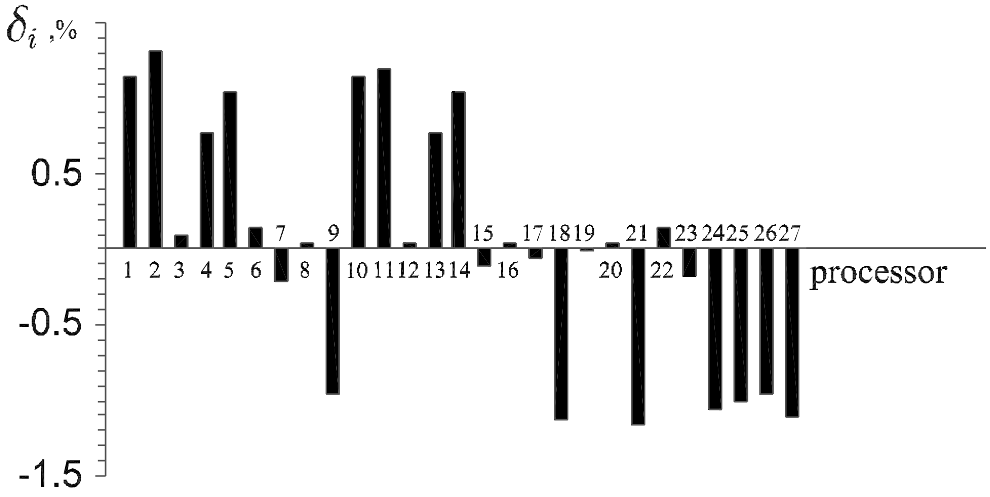

Fig. 1.4. Relative execution time ($\delta_i$) in the numerical test

The first order of incomplete parallelism supposes that $p$ processors are used for parallelization of the smoothing iterations on the coarse grids and a single processor is used for parallelization of the smoothing iterations on the finest grid (i.e., $p-1$ processors stay idle). Neglecting the efforts for data communication, we obtain the following expression for the minimum speed-up ($\mathcal{S}^p_{\min}$) and efficiency of parallelism ($\mathcal{E}^p_{\min}$)

$$\mathcal{S}^p_{\min} = \frac{\sum_{l=0}^{L^+} T_l}{T_0 + p^{-1} \sum_{l=1}^{L^+} T_l} \, , \qquad \mathcal{E}^p_{\min} = \frac{1}{p} \frac{\sum_{l=0}^{L^+} T_l}{T_0 + p^{-1} \sum_{l=1}^{L^+} T_l} \, ,$$

where $T_l$ is the execution time on $l$th level. If the same smoother is used on all levels, we obtain $T_l = \text{const}$ and the minimum speed-up and efficiency are given by

$$\mathcal{S}^p_{\min} = \frac{L^+ + 1}{1 + L^+/p} = p \, \mathcal{E}^p_{\min} \, .$$

For the first order of incomplete parallelism ($p = 3^N$) the estimate takes the form

$$\mathcal{S}^p_{\min} = 3^N \, \frac{L^+ + 1}{L^+ + 3^N} < \mathcal{S}^p < 3^N \, , \quad \mathcal{E}^p_{\min} = \frac{L^+ + 1}{L^+ + 3^N} < \mathcal{E}^p < 1 \, .$$

The estimates show that the total parallelism efficiency strongly depends on the parallelization of the smoothing iterations on the finest grid.

**1.2. Complete parallelization.** Parallelization of the smoothing iterations on the finest grid in the RMT can be performed in a traditional manner. Since the prolongation operator of the RMT does not put errors to the coarse grid correction, it makes the task of the smoother the least demanding. As a result, well-parallelizable (weak) smoothers can be used on the finest grid jointly with the RMT.

## 2. Unstructured grids

The computational fluid dynamics is a discipline to which multigrid has been applied most widely and shown to be useful [8,9]. Numerical simulation of incompressible fluid flows results in a large-scale saddle point problem [1]. In spite of intensive researches, many challenges remain.

Some promising approaches can be developed for solving the Navier — Stokes equations on structured grids. For example, the pressure splitting

$$p(t, x, y, z) = p^x(t, x) + p^y(t, y) + p^z(t, z) + p^{xyz}(t, x, y, z)$$

is proposed in [6] for efficient pressure computation. One-dimensional (in spatial directions) components $p^x(t, x)$, $p^y(t, y)$ and $p^z(t, z)$ can be computed using mass conservation equations by the effective numerical methods developed for simplified Navier — Stokes equations. However, the approach cannot be generalized directly for unstructured grids.

On the other hand, unstructured grids seem to be more preferable for complex domains. Constructing coarse grids from fine grids is easy when the fine grid is structured, but not if the fine grid is unstructured [9].

Combining the advantages of structured and unstructured approaches, it is possible to use different methods for approximating the left and right sides of the $\Sigma$-modified boundary value problems [5], i.e., to use the RMT as a preconditioner. Let us reduce the problem to

$$\mathcal{L}(u) = f \, ,$$

where $\mathcal{L}$ is the linear operator. The $\Sigma$-modification of the solution

$$u = c + \hat{u}$$

leads to the $\Sigma$-modified form of the initial problem

$$\mathcal{L}(c) = f - \mathcal{L}(\hat{u}) \, . \tag{2.1}$$

Assume that the auxiliary structured grid ($G_{\boxplus}$) is generated for the approximation of the left side of the $\Sigma$-modified boundary value problem (2.1) and the unstructured grid ($G_{\boxtimes}$) is generated for the approximation of the right side. It results in a system of linear equations denoted by

$$Wc = \mathcal{R}\hat{r}, \tag{2.2}$$

where $\hat{r} = \hat{b} - \hat{A}\hat{u}$ is a residual computed on $G_\boxtimes$, $\mathcal{R}$ is a transfer operator from $G_\boxtimes$ onto $G_\boxplus$, the matrices $W$ and $A$ have arisen from the approximation of the linear operator $\mathcal{L}$ on $G_\boxplus$ and $G_\boxtimes$, respectively. Hence, $\hat{\alpha}$ means that $\alpha$ is defined on $G_\boxtimes$. Then the solution of (2.2) (correction)

$$c = W^{-1}\mathcal{R}\hat{r}$$

is interpolated from $G_\boxplus$ to $G_\boxtimes$

$$\hat{c} = \mathcal{Q}c\,,$$

where $\mathcal{Q}$ is a transfer operator from $G_\boxplus$ onto $G_\boxtimes$. The interpolated correction is added to the approximation of the solution

$$\hat{u} := \hat{u} + \hat{c}\,, \quad c = 0\,.$$

The algorithm can be rewritten in the matrix form as follows:

$$\hat{u}^{(n+1)} = (I - \mathcal{Q}W^{-1}\mathcal{R}\hat{A})\hat{u}^{(n)} + \mathcal{Q}W^{-1}\mathcal{R}\hat{b}\,.$$

Unfortunately, all multigrid methods for unstructured grids should include such a problem-dependent component as interpolation. In a nonlinear case (for example, the Navier — Stokes equations), not only the residual ($\hat{r}$), but also approximation of the solution ($\hat{u}$) should be interpolated from $G_\boxtimes$ to $G_\boxplus$. The more advanced algorithm includes smoothing on the unstructured grid for reducing the interpolation error.

Note that the grid mapping can result in an increase in the condition number of matrix $W$. This leads to a decrease in the multigrid efficiency. An example of the R$_\mathrm{M}$T convergence deterioration on nonuniform grids is given in [5, test 4].

**Generation of auxiliary structured grids.** The initial unstructured grid and the auxiliary structured grid should have approximately the same number of nodes and the same refinement in the subdomains. Assume that a domain $\Omega$ is submerged in an $N$-dimensional unit cube. An example of triangulation of a 2D domain is shown in Fig. 2.1. The set of grid vertices is denoted by $\Upsilon(x_k, y_k)$, $k = 1, 2, \ldots, \mathrm{K}$. We can form an ordered set

$$\tilde{\Upsilon}^x = \{x_i \,|\, 0 = x_1 < x_2 < \ldots < x_i < x_{i+1} < \ldots < x_{\tilde{\mathrm{K}}_\mathrm{x}} = 1\}, \quad \tilde{\mathrm{K}}_\mathrm{x} \leqslant \mathrm{K}\,.$$

by deleting the coincident vertex abscissas and ordering the remaining abscissas. Further a uniform grid is generated as

$$\hat{\Upsilon}^x = \{\hat{x}_i \,|\, \hat{x}_i = (i-1)/(\tilde{\mathrm{K}} - 1), \quad i = 1, 2, \ldots, \tilde{\mathrm{K}}_\mathrm{x}\}.$$

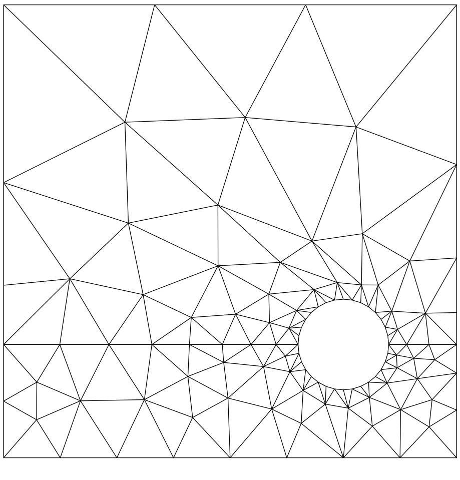

Fig. 2.1. Example of initial triangulation

The manager function $\hat{\mathfrak{F}} : \hat{x}_i \rightarrow x_i$ is a uniform grid in the computational domain mapped onto ordered set of abscissas as shown in Fig. 2.2. However, the number of vertices on the auxiliary structured grid will be $\sim \mathrm{K}^2$, which requires an unpracticable amount of computational work.

To avoid this problem, a second uniform structured grid $\bar{\Upsilon}^x$ is generated

$$\bar{\Upsilon}^x = \{\bar{x}_i \mid \bar{x}_i = (i-1)(\bar{\mathrm{K}}_{\mathrm{x}} - 1),\ i = 1, 2, \ldots, \bar{\mathrm{K}}_{\mathrm{x}}\},$$

where $\bar{\mathrm{K}}_{\mathrm{x}} = [\sqrt{\tilde{\mathrm{K}}_{\mathrm{x}}}] + 1$, square brackets mean the integer part. The spatial position of the second grid vertices can be determined using a spline interpolant of the manager function $\hat{\mathfrak{F}}$ as shown in Fig. 2.3.

Structured grids in other spatial directions are generated in the same manner. For the given example the number of auxiliary structured grid vertices is somewhat larger than the number of vertices of the initial triangulation. Fig. 2.4 presents the generated auxiliary structured grid.

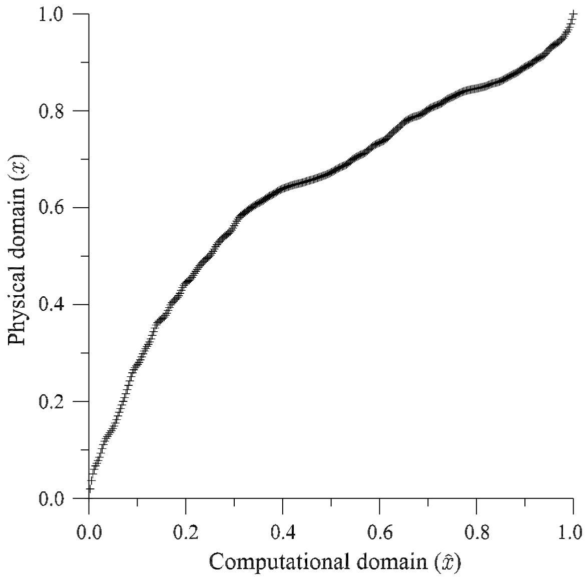

Fig. 2.2. Manager function $\hat{\mathfrak{F}} : \hat{x}_i \rightarrow x_i$

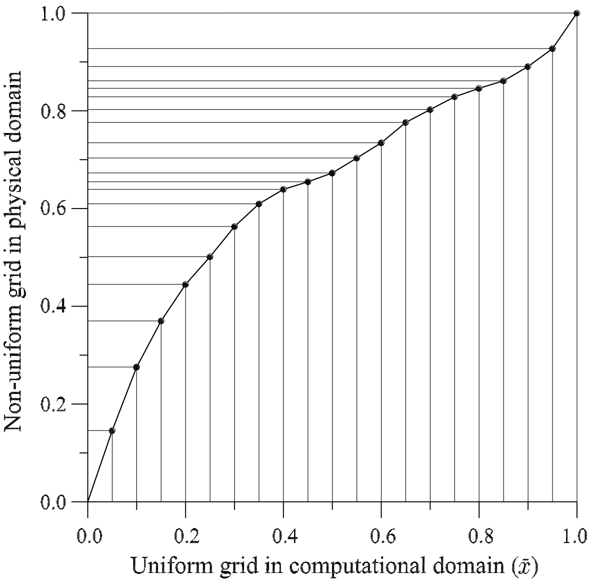

Fig. 2.3. Grids in computational and physical domains

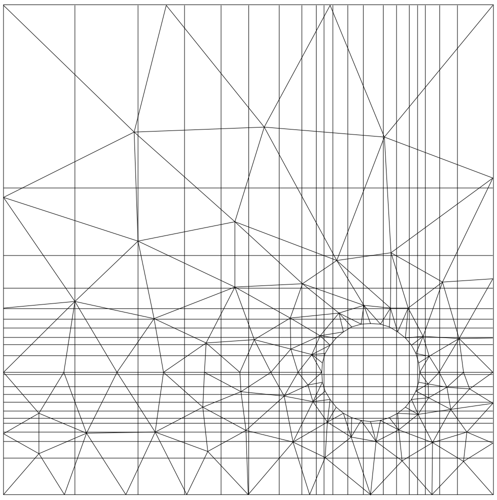

Fig. 2.4. Initial triangulation and auxiliary structured grid

As an example, the partial differential equation

$$\frac{\partial}{\partial x}\left(\lambda^x \frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\lambda^y \frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(\lambda^z \frac{\partial u}{\partial z}\right)=-f(x,y,z)$$

in the computational domain is transformed to the form

$$\frac{\partial}{\partial \bar{x}}\left(\lambda^x \frac{\bar{x}'_x}{\bar{y}'_y \bar{z}'_z}\frac{\partial u}{\partial \bar{x}}\right)+\frac{\partial}{\partial \bar{y}}\left(\lambda^y \frac{\bar{y}'_y}{\bar{x}'_x \bar{z}'_z}\frac{\partial u}{\partial \bar{y}}\right)+\frac{\partial}{\partial \bar{z}}\left(\lambda^z \frac{\bar{z}'_z}{\bar{x}'_x \bar{y}'_y}\frac{\partial u}{\partial \bar{z}}\right)=-\frac{f(\bar{x},\bar{y},\bar{z})}{\bar{x}'_x \bar{y}'_y \bar{z}'_z}\,.$$

Derivatives $\bar{x}'_x$, $\bar{y}'_y$ and $\bar{z}'_z$ can be computed by differentiating the spline interpolants of the corresponding manager functions.

## Conclusions

The multigrid structure of the RMT makes it possible to obtain full parallelism on coarse levels at an almost perfect load balance independent of the smoother procedure. The application of the RMT as a preconditioner can be considered as a promising way to extend the approach to unstructured grids.

**Acknowledgements.** Work supported by Russian Foundation for the Basic Research (project no. 09-01-00151).

## References

1. M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Num, 2005, pp. 1-137.

2. P. Frederickson and O. McBryan, *Parallel superconvergent multigrid*, In S. McCormick, editor, Proceeding of the Third Copper Mountain Conference on Multigrid Methods, NY, Marcel Dekker, 1987, pp. 195–210.

3. W. Hackbusch, *The frequency decomposition multigrid methods*, part I: Application to anisotropic equations, Numer. Math., **56**, 1989, pp. 229–245.

4. A. B. Kucherov and M. M. Makarov, *Approximate factorization method for solving difference mixed elliptic problems*, in: *Difference methods of mathematical physics*, E. S. Nikolaev (ed.), MSU, 1984, pp. 54–65.

5. S. I. Martynenko, *Robust Multigrid Technique for black box software*, Comp. Meth. in Appl. Math., **6** (2006), no. 4, pp. 413–435.

6. S. I. Martynenko, *A physical approach to development of numerical methods for solving Navier — Stokes equations in primitive variables formulation*, Int. J. Computing Science and Mathematics), **2** (2009), no. 4, pp. 291–307.

7. W. Mulder, *A new multigrid approach to convection problems*, J. Comput. Phys., **83** (1989), pp. 303–329.

8. P. Wesseling, *An Introduction to Multigrid Methods*, Wiley, Chichester, 1991.

9. P. Wesseling, C. W. Oosterlee *Geometric multigrid with applications to computational fluid dynamics*, J. of Comp. and Appl. Math., **128** (2001), pp. 311–334.