Adaptive Solution of the Wave Equation

Václav Valenta[1], Gabriela Nečasová[1], Jiří Kunovský[1], Václav Šátek[1,2] and Filip Kocina[1]
[1]*Department of Intelligent Systems, Brno University of Technology, Božetěchova 2, 612 66 Brno, Czech Republic*
[2]*IT4Innovations, VŠB Technical University of Ostrava, 17. listopadu 15/2172, 708 33 Ostrava-Poruba, Czech Republic*

Keywords: Wave equation, A posteriori error estimation, Triangulation, Gradient, Modern Taylor Series Method, Finite difference formula.

Abstract: The paper focuses on the adaptive solution of two-dimensional wave equation using an adaptive triangulation update based on a posteriori error estimation. The a posteriori error estimation is based on the Gradient super-approximation method which is based on works of J. Dalík et al that is briefly explained. The Modern Taylor Series Method (MTSM) used for solving a set of ordinary differential equations is also explained. The MTSM adapts to the required accuracy by using a variable number of Taylor Series terms. It possible to use the MTSM to solve wave equation in conjunction with Finite Difference Method (FDM).

1 INTRODUCTION

The wave equation is widely used in real technical problems. It can be used to simulate AC electric circuits, strings, optics or electromagnetism. The wave equation is a partial differential equation of the second order. This article focuses on the wave equation with two dimensions and one time variable with Dirichlet boundary values (1).

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial^2 u}{\partial t^2} \tag{1}$$

When simulating a system with the wave equation, it is necessary to quantize continuous space to a discrete space. This operation brings an error into the solution. If smaller number of discrete points is used, the quantization error is big, while the solution is calculated very quickly with small truncation error. On the other hand, if more discrete points are used, the quantization error decreases. However the calculation is very time consuming and truncation error is bigger. The solution tends to diverge when using ordinary methods.

One of the possible solutions to reduce the errors is to use an a posteriori error estimation which identifies the areas which should be covered with more points than the other areas. This balances between these two extremes. In most cases the result of the estimation is the non-symmetric triangulation which is more difficult to solve. This approach also requires iterative calculations of the same problem, but they can be faster than the uniform dense grids.

2 FINITE DIFFERENCE METHOD

Finite difference method (FDM) is a method for solving partial differential equations. The idea behind this method is to replace all partial derivatives by finite differences in predefined points. The values in these points approximate the solution of the equation (Strikwerda, 1989).

The finite difference formulas are defined. These formulas approximate the values of the derivatives and replace all the space derivatives by these formulas. The formulas can then be used to obtain a system of ordinary differential equations which can then be solved using the Modern Taylor Series Method (MTSM) (Kunovský, 1995).

2.1 Modern Taylor Series Method

The Modern Taylor Series Method (MTSM) is used for numerical solution of differential equations. The main idea is to calculate Taylor series terms recurrently for each time interval. An important property of MTSM is an automatic integration order setting, which means using as many Taylor series terms as the defined accuracy requires. During the computation, different number of Taylor series terms is used for

Valenta V., Necasová G., Kunovský J., Šátek V. and Kocina F..
Adaptive Solution of the Wave Equation.
DOI: 10.5220/0005539401540162
In *Proceedings of the 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2015), pages 154-162
ISBN: 978-989-758-120-5

different steps that have the same length. According to experiments and theoretical analyses, the accuracy and stability of the Taylor series method is better than algorithms that are used at present (Kunovský, 1995) or (Satek et al., 2009). The MTSM has been implemented in TKSL software (Kunovský, 2014). Some practical usage of MTSM can be found in (Fuchs et al., 2013).

Consider the following ordinary differential equation (ODE) with initial condition.

$$y' = f(t,y), \quad y(t_0) = y_0 \tag{2}$$

Taylor series approximation (3) can be constructed to calculate a new value of an ODE numerical solution.

$$y_{n+1} = y_n + h \cdot f(t_n,y_n) + \frac{h^2}{2!} \cdot f^{[1]}(t_n,y_n) + \cdots \\ \cdots + \frac{h^p}{p!} \cdot f^{[p-1]}(t_n,y_n), \tag{3}$$

where $h \in \mathbb{R}$ is the integration time step and $p \in \mathbb{N}$ is the order of the approximation. More information about the properties of the Taylor series can be found in (R. Barrio and Lara, 2005) and (R. Barrio and Blesa, 2011). The MTSM adapts the order automatically, it means, that the values of the terms (4) are computed for increasing values of p. The computation is stopped when the last term of the Taylor series is smaller than a predefined threshold ε.

$$\frac{h^p}{p!} \cdot f^{[p-1]}(t_n,y_n) < \varepsilon \tag{4}$$

2.2 Taylor Series Based Formulas

This section focuses on the wave equation which is one of the most common hyperbolic partial differential equation (PDE) (Gabriela Nečasová and Veigend, 2015), (Filip Kocina and Šátek, 2014)

$$\frac{\partial^2 V(x,t)}{\partial x^2} = \frac{\partial^2 V(x,t)}{\partial t^2} \tag{5}$$

with the following domain Ω and Dirichlet boundary values on Ω.

$$\Omega = \langle 0,1\rangle \times \langle 0,1\rangle \tag{6}$$
$$V(0,t) = 0 \tag{7}$$
$$V(1,t) = 0 \tag{8}$$

Cauchy initial values follow.

$$V(x,0) = \sin(\pi x) \tag{9}$$
$$\frac{\partial V(x,0)}{\partial t} = 0 \tag{10}$$

The wave equation (5) describes the oscillations of an ideal string of unit length. Both ends of the string are fixed, boundary values are fullfiled. The string has a non-zero original velocity and it is released at time $t = 0$.

Numerical methods for solving PDE's are based on approximations of the derivatives by differences. One variable remains continuous and the others are replaced by differences, the result is the method of lines. Using this method it is possible to transform PDE into the system of ordinary differential equations. This system of ODEs can be solved using the MTSM. Using central difference formula for the three-point approximation it is possible to replace left side of the equation (5) as follows.

$$\frac{\partial^2 V(x,t)}{\partial x^2} = \frac{y_{k-1} - 2y_k + y_{k+1}}{(\Delta x)^2} \tag{11}$$

The following sections of the paper (2.3 and 2.4) show an easy way of very high order difference formulas construction using Taylor series terms.s

2.3 Forward Difference Formula

The forward method uses points from the right side of the current point. Note, only positive steps are used (Figure 1). For each point of the string, four Taylor series terms are constructed, because five-point approximation is considered. Using five-point approximation it is possible to obtain derivatives from the first to the fourth order in each point of the string. The following equations are valid for the point u_1 using forward formula (with respect to Figure 1).

$$u_2 = u_1 + hu_1' + \frac{h^2}{2!}u_1'' + \\ + \frac{h^3}{3!}u_1''' + \frac{h^4}{4!}u_1'''' \tag{12}$$
$$u_3 = u_1 + 2hu_1' + \frac{(2h)^2}{2!}u_1'' + \\ + \frac{(2h)^3}{3!}u_1''' + \frac{(2h)^4}{4!}u_1'''' \tag{13}$$
$$u_4 = u_1 + 3hu_1' + \frac{(3h)^2}{2!}u_1'' + \\ + \frac{(3h)^3}{3!}u_1''' + \frac{(3h)^4}{4!}u_1'''' \tag{14}$$
$$u_5 = u_1 + 4hu_1' + \frac{(4h)^2}{2!}u_1'' + \\ + \frac{(4h)^3}{3!}u_1''' + \frac{(4h)^4}{4!}u_1'''' \tag{15}$$

It is possible to denote Taylor series terms as DU1, DU2, DU3 and DU4, where $DUi = \frac{u_1^{(i)}}{i!}h^i$.

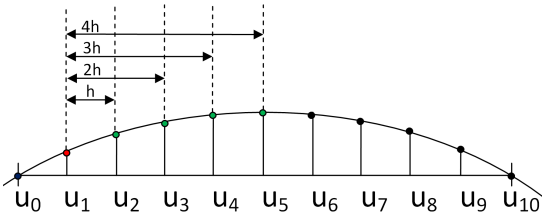

Figure 1: Forward difference formula, point u_1.

$$u_2 - u_1 = DU1 + DU2 + + DU3 + DU4 \quad (16)$$

$$u_3 - u_1 = 2DU1 + 2^2DU2 + + 2^3DU3 + 2^4DU4 \quad (17)$$

$$u_4 - u_1 = 3DU1 + 3^2DU2 + + 3^3DU3 + 3^4DU4 \quad (18)$$

$$u_5 - u_1 = 4DU1 + 4^2DU2 + + 4^3DU3 + 4^4DU4 \quad (19)$$

Backward difference formula can be constructed similarly.

2.4 Symmetrical Difference Formula

Symmetrical difference formula uses the same number of points from both sides of the current point (Figure 2). In this case positive and negative steps are used. Let's suppose the five-point approximation, then the Taylor series for the point u_2 can be constructed as follows.

$$u_3 = u_2 + hu_2' + \frac{h^2}{2!}u_2'' + \frac{h^3}{3!}u_2''' + \frac{h^4}{4!}u_2'''' \quad (20)$$

$$u_4 = u_2 + 2hu_2' + \frac{2h^2}{2!}u_2'' + \frac{2h^3}{3!}u_2''' + \frac{2h^4}{4!}u_2'''' \quad (21)$$

$$u_1 = u_2 + (-h)u_2' + \frac{(-h)^2}{2!}u_2'' + + \frac{(-h)^3}{3!}u_2''' + \frac{(-h)^4}{4!}u_2'''' \quad (22)$$

$$u_0 = u_2 + (-2h)u_2' + \frac{(-2h)^2}{2!}u_2'' + + \frac{(-2h)^3}{3!}u_2''' + \frac{(-2h)^4}{4!}u_2'''' \quad (23)$$

It is also possible to denote Taylor series terms as DU1, DU2, DU3 and DU4.

$$u_0 - u_2 = -2DU1 + 2^2DU2 + - 2^3DU3 + 2^4DU4 \quad (24)$$

$$u_1 - u_2 = -DU1 + DU2 + - DU3 + DU4 \quad (25)$$

$$u_3 - u_2 = DU1 + DU2 + + DU3 + DU4 \quad (26)$$

$$u_4 - u_2 = 2DU1 + 2^2DU2 + + 2^3DU3 + 2^4DU4 \quad (27)$$

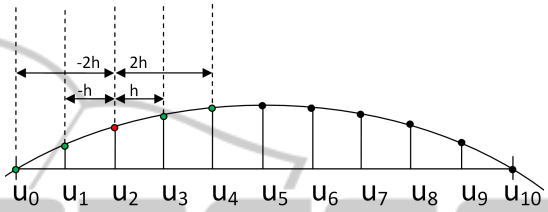

Figure 2: Symmetrical difference formula, point u_2.

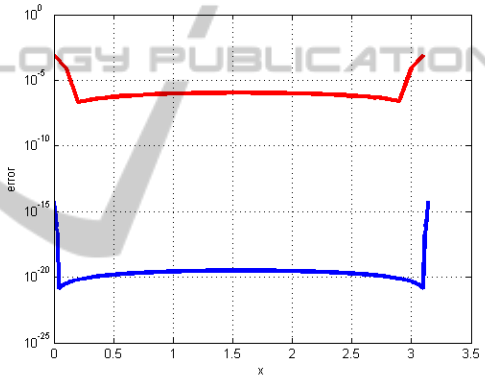

Figure 3: Symmetrical difference formulas.

2.5 Accuracy of the Calculation

There are two parameters affecting the accuracy. The first parameter is the integration step. If the smaller step is used, the resulting solution is more accurate. The second parameter is the order of the difference formulas. The higher order is set, the more accurate solution is obtained.

Figure 4 shows the absolute error between numerical and analytical solution. The upper red function shows this error for three-point approximation and ten segments of the string, the lower red function shows this error for three-point approximation and one hundred segments. This error between numerical and analytical solution can be effectively decreased by an increase in the order of the difference formula. The upper blue function represents five-point approximation and twelve segments. Furthermore, the lower blue function shows five-point approximation, the string is divided into one hundred segments.

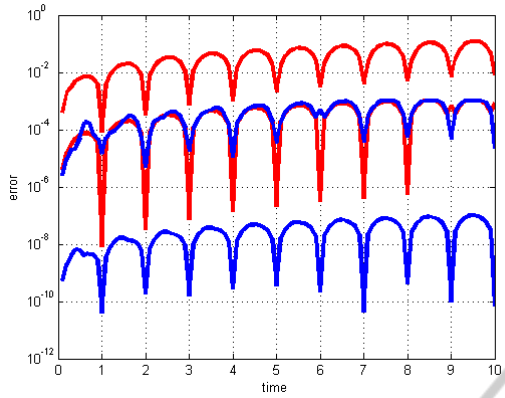

Figure 4: An absolute error between numerical and analytical solution.

There are also differences between types of difference formulas. If the forward or backward formulas are used the same calculation error is obtained, because both of these formulas are asymmetrical. This causes the error accumulation. On the other hand, symmetrical difference formula uses the same number of points from the both sides, this formula has smaller calculation error.

Figure 3 shows the absolute error between numerical and analytical solution for symmetrical formulas. The upper red function shows the absolute error for five-point approximation (integration step $h = 0.1$), the lower blue function for nine-point approximation (integration step $h = 0.01$). Note, that these functions mostly remain at the same level. Higher deviation between numerical and analytical solution at the ends of the plotted curves is caused by using asymmetrical difference formulas.

Figure 5 shows the absolute error between numerical and analytical solution for forward difference formulas, using the same settings as for symmetrical difference formulas. Note, that the deviation is considerably bigger, because mostly asymmetrical difference formulas are used.

3 GRADIENT SUPER-APPROXIMATION

The Gradient super-approximation is a method to calculate gradient of a function of two variables.

Each inner vertex $\mathbf{v}$ of a triangulation $\mathcal{T}$ is shared by multiple elements and gradient of a function $u(x,y)$ in vertex $\mathbf{v}$ can differ from element to element.

The idea of the Gradient super-approximation is to use weighted average and calculate more precise gradient of a function $u(x,y)$ in vertex $\mathbf{v}$.

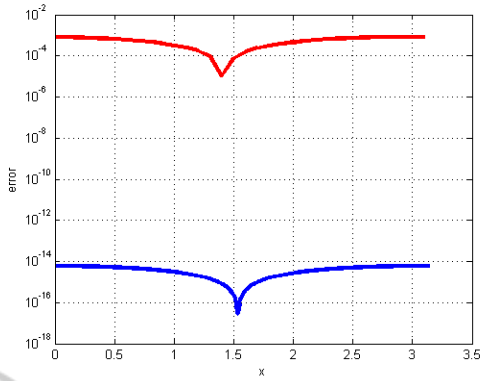

Figure 5: Forward difference formulas.

This method is well described in (Dalík, 2008), (Dalík, 2010) and (Dalík, 2012).

3.1 Triangulation

Triangulation describes a quantization of the area Ω. Basic elements of triangulation are called *elements*. The *element* can be denoted as t if it is a rectangle. Each element t is defined by four vertices $\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{a_3}$, $\mathbf{a_4}$ and contains four edges $\mathbf{a_1a_2}$, $\mathbf{a_2a_3}$, $\mathbf{a_3a_4}$ and $\mathbf{a_4a_1}$.

$\mathcal{T}_\Omega$ is called a *triangulation* of Ω of elements when the following conditions are fulfilled:

- $\bigcup t \in \mathcal{T}_\Omega = \overline{\Omega}$ (the union of all elements covers the entire area including its border),
- any two different elements have disjoint interiors, or share an edge, or share a vertex.

A vertex $\mathbf{b}$ is a neighbor of a vertex $\mathbf{a}$ of a triangulation $\mathcal{T}$ when the segment $\overline{ab}$ is an edge of an element t of $\mathcal{T}$.

Vertex $\mathbf{a}$ is called *inner* and *outer* when $\mathbf{a} \in \Omega$ and $\mathbf{a} \in \partial\Omega$ respectively.

3.2 Interpolant

Let N_t^n be a bilinear nodal basis function defined for n-th vertex of an element t (consider only rectangles with $n = 1 \ldots 4$) of the form $N_t^n(x,y) = a + bx + cy + dxy$. This function is defined by values of all four element vertices. $N_t^i = 1$ for vertex $\mathbf{a_i}$ and $N_t^i = 0$ for all other vertices.

Let's consider following function as interpolant of a function $u(x,y)$ on an element t.

$$\Pi_t[u](x,y) = \sum_{i=1..4} u(x_i,y_i)N^i(x,y) \qquad (28)$$

where

- t is an element $t \in \mathcal{T}$,
- u is a function of two variables,

- (x_i, y_i) is a position of vertex $\mathbf{a_i}$,
- $u(x_i, y_i)$ is the value of function u in n-th vertex of element t,
- N^n is a nodal basis function of n-th vertex of element t.

3.3 Ring Around Vertex

Let's call $r_\mathbf{v} = \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ a ring around vertex $\mathbf{v}$, where $\mathbf{a_i}$ is a vertex of an element of a triangulation $\mathcal{T}$ and $\mathbf{va_i}$ is an edge of an element of a triangulation $\mathcal{T}$ and one of the two following conditions is fulfilled.

1. If $\mathbf{v}$ is an inner vertex, then equation (29) is valid.
$$\angle \mathbf{a_n v a_1} + \ldots \angle \mathbf{a_{n-1} v a_n} = 2\pi \tag{29}$$
2. If $\mathbf{v}$ is a boundary vertex, then there exists an inner vertex $\mathbf{c}$ and a ring around vertex $\mathbf{c}$ denoted as $r_\mathbf{c} = (\mathbf{a_1}, \ldots, \mathbf{v}, \ldots, \mathbf{a_n})$. The ring around $\mathbf{v}$ is created swapping $\mathbf{c}$ and $\mathbf{v}$ vertices. When this condition is not fulfilled, the Gradient super-approximation is not applicable.

When this operation is performed, new elements which are not part of the triangulation are used for calculation. More details can be found in (Dalík and Valenta, 2013).

3.4 Consistency Condition

Each ring $r_\mathbf{a}$ can be transformed into a ring $\rho_\mathbf{a}$ with its local coordinate system. The vertex $\mathbf{a}$ can be moved to origin of the ring $\rho_\mathbf{a}$ using simple translation. All calculations are performed in this local coordinate system. The gradient value is not affected.

The Gradient super-approximation method is based on the Consistency condition. This condition says that the partial derivative of a polynomial $p(x,y)$ in a vertex $\mathbf{a}$ must be equal to the weighted average of partial derivatives of interpolants of a polynomial $p(x,y)$ for all elements from the ring around vertex $\mathbf{a}$ for all polynomials $p(x,y)$ of the second order. Then the error of the gradient is $O(h^2)$, where h is the longest edge of an element t (without this operation, the error is $O(h^{\frac{3}{2}})$). Formally written in the form

$$\forall p \in \mathcal{P}^2, \frac{\partial p}{\partial x} = \sum_{\forall t \in \rho_\mathbf{a}} f_t \frac{\partial \Pi_t[p]}{\partial x} \tag{30}$$

where

- p is a polynomial of the second order,
- $\rho_\mathbf{a}$ is a ring of elements around the vertex $\mathbf{a}$,
- t is an element from the ring of elements around the vertex $\mathbf{a}$,
- f_t is a weight value for an element t from the weight vector w, and $\mathbf{f}$ is the weight vector defined for a ring $r_\mathbf{a}$.

Because all polynomials of the second order are linearly dependent on a polynomial basis, the second order polynomial basis is used instead of all polynomials of the second order.

For example, the following basis can be chosen (31).

$$\mathcal{B} = \left(1, x, y, xy, x^2, y^2\right) \tag{31}$$

Now let's consider the Consistency condition for both derivatives (by x and by y) and all the functions from the polynomial basis $\mathcal{B}$. The resulting system of twelve equations can be reduced to the system of three linear algebraic equations of n variables, where n is the number of elements in the ring $\rho_\mathbf{a}$.

It is well known that this system has an infinite number of solutions, let's choose the minimal norm solution.

3.5 Example

In this section a simple example of the Gradient superaproximation is presented on a very simple triangulation containing four rectangles (see Figure 6).

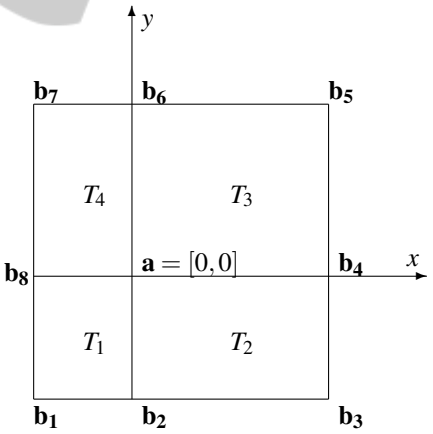

Figure 6: Example triangulation with four rectangles.

First, the important values coming out of this triangulation have to be derived.

$$x_1 = a_x - b_x^8 \tag{32}$$
$$x_3 = b_x^3 - a_x \tag{33}$$
$$y_2 = a_y - b_y^2 \tag{34}$$
$$y_6 = b_y^6 - a_y \tag{35}$$

Then the nodal functions N_t^i have to be derived for each element.

For example, nodal function for the element T_1.

$$T_1: \quad L_1^1(x,y) = \frac{1}{x_1 y_2} xy \tag{36}$$

$$L_1^2(x,y) = -\frac{1}{x_1 y_2} y(x+x_1) \tag{37}$$

$$L_1^a(x,y) = \frac{1}{x_1 y_2} (x+x_1)(y+y_2) \tag{38}$$

$$L_1^8(x,y) = -\frac{1}{x_1 y_2} x(y+y_2) \tag{39}$$

An interpolant template is created for a function u for each element, for example the element T_1

$$\begin{aligned} \Pi[u]_{T_1}(x,y) = & u_1 L_1^1(x,y) + u_2 L_1^2(x,y) + \\ & + u_0 L_1^a(x,y) + u_8 L_1^8(x,y) \end{aligned} \tag{40}$$

where u_i is a value of a function u in a vertex i.

It is necessary to find the weight vector (f_1, f_2, f_3, f_4) to fulfill the consistency condition.

$$f_1 \frac{\partial \Pi[p]_{T_1}}{\partial x}(\mathbf{a}) + \ldots + f_4 \frac{\partial \Pi[p]_{T_4}}{\partial x}(\mathbf{a}) = \frac{\partial p}{\partial x}(\mathbf{a})$$

$$f_1 \frac{\partial \Pi[p]_{T_1}}{\partial y}(\mathbf{a}) + \ldots + f_4 \frac{\partial \Pi[p]_{T_4}}{\partial y}(\mathbf{a}) = \frac{\partial p}{\partial y}(\mathbf{a})$$

Now all the functions of the basis $\mathcal{B}$ are used in both equations of consistency condition. In fact there are only four different situations.

Considering $p(x,y) = 1$, $p(x,y) = y$ or $p(x,y) = y^2$ for the derivative by x, or $p(x,y) = 1$, $p(x,y) = x$ or $p(x,y) = x^2$ for the derivative by y, equation in the form $0 = 0$ is obtained and therefore can be ignored.

Another situation is for $p(x,y) = x$ for the derivative by x and $p(x,y) = y$ for the derivative by y. Bilinear interpolant $\Pi_t[p] = p$ and four same equations can be created.

$$\frac{\partial p}{\partial x}(\mathbf{a}) = f_1 \frac{\partial p}{\partial x}(\mathbf{a}) + \ldots + f_4 \frac{\partial p}{\partial x}(\mathbf{a}) \tag{41}$$

$$\frac{\partial p}{\partial x}(\mathbf{a}) = \frac{\partial p}{\partial x}(\mathbf{a})\,(f_1 + f_2 + f_3 + f_4) \tag{42}$$

$$1 = f_1 + f_2 + f_3 + f_4 \tag{43}$$

Another situation is for $p(x,y) = xy$ for both derivatives (by x and y), where the derivatives are y and x. The value of the gradient in point $\mathbf{a} = [0,0]$, which means that the resulting equation can be ignored.

Different situation is for $p(x,y) = x^2$ for the derivative by x and $p(x,y) = y^2$ for the derivative by y. It is necessary to calculate all the interpolants for all four rectangles.

$$\Pi[x^2]_{T_1} = \Pi[x^2]_{T_4} = -x_1 x \tag{44}$$

$$\Pi[x^2]_{T_2} = \Pi[x^2]_{T_3} = x_3 x \tag{45}$$

$$\Pi[y^2]_{T_1} = \Pi[y^2]_{T_2} = -y_2 y \tag{46}$$

$$\Pi[y^2]_{T_3} = \Pi[y^2]_{T_4} = y_6 y \tag{47}$$

Their appropriate partial derivatives in the point $\mathbf{a}$.

$$\frac{\partial \Pi[x^2]_{T_1}}{\partial x}(\mathbf{a}) = \frac{\partial \Pi[x^2]_{T_4}}{\partial x}(\mathbf{a}) = -x_1 \tag{48}$$

$$\frac{\partial \Pi[x^2]_{T_2}}{\partial x}(\mathbf{a}) = \frac{\partial \Pi[x^2]_{T_3}}{\partial x}(\mathbf{a}) = x_3 \tag{49}$$

$$\frac{\partial \Pi[y^2]_{T_1}}{\partial y}(\mathbf{a}) = \frac{\partial \Pi[y^2]_{T_2}}{\partial y}(\mathbf{a}) = -y_2 \tag{50}$$

$$\frac{\partial \Pi[y^2]_{T_3}}{\partial y}(\mathbf{a}) = \frac{\partial \Pi[y^2]_{T_4}}{\partial y}(\mathbf{a}) = y_6 \tag{51}$$

Using these derivatives, two equations of the following form are obtained

$$f_1 \frac{\partial \Pi[x^2]_1}{\partial x}(\mathbf{a}) + \ldots + f_4 \frac{\partial \Pi[x^2]_4}{\partial x}(\mathbf{a}) = 0 \tag{52}$$

$$f_1(-x_1) + f_2 x_3 + f_3 x_3 + f_4(-x_1) = 0 \tag{53}$$

$$f_1 \frac{\partial \Pi[y^2]_1}{\partial y}(\mathbf{a}) + \ldots + f_4 \frac{\partial \Pi[y^2]_4}{\partial y}(\mathbf{a}) = 0 \tag{54}$$

$$f_1(-x_2) + f_2(-x_2) + f_3 y_6 + f_4 y_6 = 0 \tag{55}$$

Finally, it is necessary to solve the following system of three linear algebraic equations of four variables and find the solution of the minimal form.

$$f_1 + f_2 + f_3 + f_4 = 1 \tag{56}$$

$$-f_1 x_1 + f_2 x_3 + f_3 x_3 - f_4 x_1 = 0 \tag{57}$$

$$-f_1 y_2 - f_2 y_2 + f_3 y_6 + f_4 y_6 = 0 \tag{58}$$

4 A POSTERIORI ERROR ESTIMATION

The a posteriori error estimation is an approach of calculating the error of a solution of the partial differential equation without knowledge of the analytical solution.

4.1 Calculating a Posteriori Error Estimation

Because there is no particular formula available, custom a posteriori error estimation method had to be created using the Gradient super-approximation.

A posteriori error estimation τ_t on each element t of triangulation $\mathcal{T}$ is calculated. Element t is denoted by four vertices which define unique bilinear function $\phi(x,y)$ of two variables of the following form.

$$\phi(x,y) = a + bx + cy + dxy \tag{59}$$

It is also possible to calculate gradient of this function of the form as mentioned below.

$$(b+dy, c+dx) \tag{60}$$

It is also necessary to calculate the Gradient superapproximation presented in section 3. Two bilinear functions ϕ_x and ϕ_y that define gradient in the following form.

$$(\phi_x, \phi_y) \tag{61}$$

Now it is possible to calculate a posteriori error estimation on element t using the following formula.

$$\tau_t = \frac{1}{|t|}\int_t \left(\frac{\partial \Pi_t[\phi]}{\partial x} - \Pi_t[\phi_x]\right)^2 + \left(\frac{\partial \Pi_t[\phi]}{\partial y} - \Pi_t[\phi_y]\right)^2 \mathrm{d}x\mathrm{d}y \tag{62}$$

A posteriori error estimation is the volume between two curves (for derivative by x and also by y) which is normalized to the element area ($|t|$).

This formula is not applicable directly for the wave equation because it is suitable only for static boundary value problems (like Laplace's equation). The wave equation includes time domain and it is necessary to reflect that.

$\tau_t(t)$ is calculated for each time step and maximum value is chosen. This is considered as an a posteriori error estimation for element t. Note that this value τ_t does not reflect real error of the solution but it provides the capability to find an element with the biggest error.

4.2 Using a Posteriori Error Estimation

It is possible to define a method which attempts to iteratively adapt triangulation to obtain a more precise solution. The idea behind this method is to start with very rough triangulation, which produces quite big error of the solution. A posteriori error estimation is calculated and elements with big error are identified. These elements are then replaced by smoother triangulation, thus more precise solution is obtained (Ainsworth and Oden, 1997), (Babuška and Rheinboldt, 1978) and (Verfürth, 1994) .

The advantage is that it is not necessary to update triangulation uniformly, which means that less equations are solved and the system can converge faster.

The following algorithm is used:

1. initialize a rough triangulation,
2. solve the wave equation,
3. calculate a posteriori error estimation,
4. find element with big error and update the triangulation,
5. if error is still big enough, continue with point 2.

4.3 Triangulation Update

When a posteriori error estimation is calculated and the elements with big error are identified, triangulation update is made. Because of using a specific triangulation with the orthogonal edges only, it is necessary to split an element (vertically or horizontally) and all the elements of the same horizontal (eventualy vertical) position. The scheme is shown on the following Figure 7.

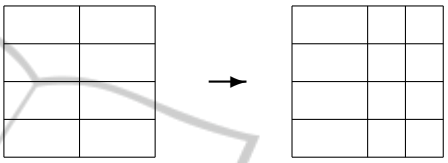

Figure 7: Vertical triangulation update.

5 RESULTS

A posteriori error estimation is used to wave equation in the following form

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial^2 u}{\partial t^2} \tag{63}$$

with the following domain Ω and Dirichlet boundary values on Ω

$$\begin{aligned}
\Omega &= \langle 0,1\rangle \times \langle 0,1\rangle \times \langle 0,1\rangle && (64)\\
u(0,y,t) &= u(1,y,t) && (65)\\
u(1,y,t) &= \sin(\pi y)(\cos(\pi t) + \sin(\pi t)) && (66)\\
u(x,0,t) &= u(x,1,t) && (67)\\
u(x,1,t) &= \sin(\pi x)(\cos(\pi t) + \sin(\pi t)) && (68)
\end{aligned}$$

and following Cauchy initial values.

$$\begin{aligned}
u(x,y,0) &= \sin(\pi x) + \sin(\pi y) && (69)\\
\frac{\partial u(x,y,0)}{\partial t} &= \pi(\sin(\pi x) + \sin(\pi y)) && (70)
\end{aligned}$$

It is well known that this equation has the analytical solution in the following form.

$$u(x,y,t) = \sin(\pi x)(\cos(\pi t) + \sin(\pi t)) + \sin(\pi y)(\cos(\pi t) + \sin(\pi t)) \tag{71}$$

The Finite difference method is used to create a set of ordinary differential equations that were solved using the Modern Taylor Series Method. The system is solved for time interval $t \in \langle 0,1\rangle$.

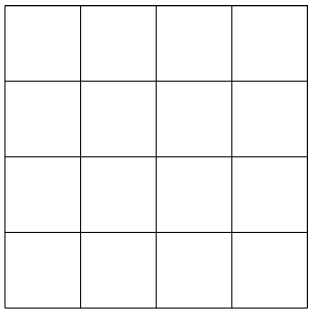

Figure 8: The iterative solution of the wave equation – the initial triangulation.

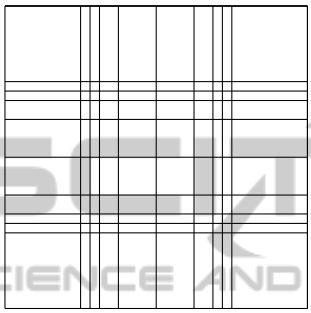

Figure 9: The iterative solution of the wave equation – fifth iteration.

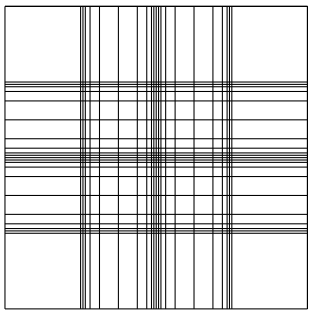

Figure 10: The iterative solution of the wave equation – the final triangulation.

Triangulations shown on Figures 8, 9 and 10 were used.

The following Figures 11, 12 and 13 show the absolute error of the solution for time $t = 1$ for three chosen triangulations.

6 CONCLUSIONS

A combination of the Modern Taylor Series Method and the a posteriori error estimation based on the Gradient super-approximation method is presented. A posteriori error estimation is used for the wave equation solution.

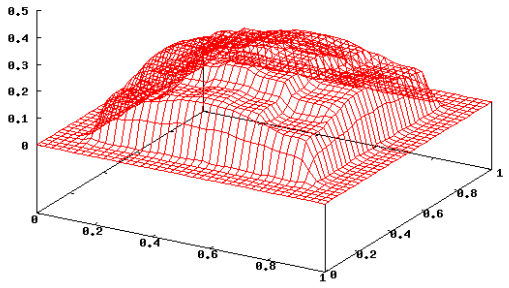

Figure 11: The absolute error of the solution for $t = 1$ – the initial triangulation.

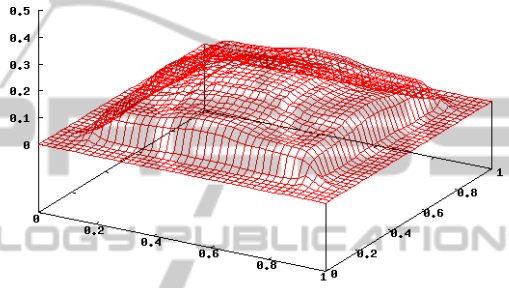

Figure 12: The absolute error of the solution for $t = 1$ – fifth iteration.

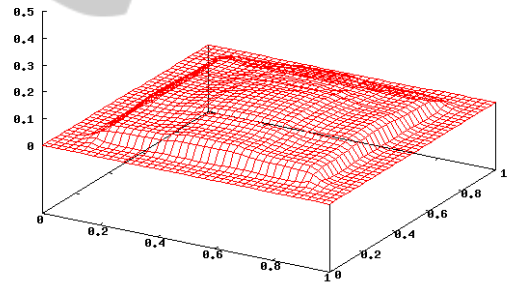

Figure 13: The absolute error of the solution for $t = 1$ – the final triangulation.

The MTSM provides very precise and stable solution of a system of ordinary differential equations and the a posteriori error estimation provides capability to update the triangulation in better way than the uniform update.

The result is the smaller number of ordinary differential equations that needs to be calculated and the solution is therefore calculated faster.

ACKNOWLEDGEMENTS

This paper has been elaborated in the framework of the project New creative teams in priorities of scientific research, reg. no. CZ.1.07/2.3.00/30.0055 (as well as the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/ 02.0070), supported by Operational Programme Education for Competitiveness and co-financed by the European Social Fund and the state budget of the Czech Republic. The paper includes the solution results of the international AKTION research project Number 69p22 and the internal BUT projects FIT-S-12-1 and FIT-S-14-2486.

REFERENCES

Ainsworth, M. and Oden, J. T. (1997). A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1):1–88.

Babuška, I. and Rheinboldt, W. C. (1978). A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615.

Dalík, J. (2008). *Matematika, numerické metody*.

Dalík, J. (2010). Averaging of directional derivatives in vertices of nonobtuse regular triangulations. *Numerische Mathematik*, 116(4):619–644.

Dalík, J. (2012). Approximations of the partial derivatives by averaging. *Central European Journal of Mathematics*, 10(1):44–54.

Dalík, J. and Valenta, V. (2013). Averaging of gradient in the space of linear triangular and bilinear rectangular finite elements. *Central European Journal of Mathematics*, 11(4):597–608.

Filip Kocina, Jiří Kunovský, M. M. G. N. A. S. and Šátek, V. (2014). New trends in taylor series based computations.

Fuchs, G., Šátek, V., Vopěnka, V., Kunovský, J., and Kozek, M. (2013). Application of the modern taylor series method to a multi-torsion chain. *Simulation Modelling Practice and Theory*, 2013(33):89–101.

Gabriela Nečasová, Václav Šátek, J. K. J. C. and Veigend, P. (2015). Taylor series based differential formulas. In *MATHMOD 2015, Vienna*, pages 389–390.

Kunovský, J. (1995). *Modern Taylor Series Method*. Habilitation work, Brno University of Technology, Brno.

Kunovský, J. (2014). High performance computing - tksl software.

R. Barrio, M. Rodríguez, A. A. and Blesa, F. (2011). Breaking the limits: the taylor series method. *Applied mathematics and computation*, 217(20):7940–7954.

R. Barrio, F. B. and Lara, M. (2005). Vsvo formulation of the taylor method for the numerical solution of odes. *Computers & Mathematics with Applications*, 50(1):93–111.

Satek, V., Kunovský, J., Kraus, M., and Kopriva, J. (2009). Automatic method order settings. In *Computer Modelling and Simulation, 2009. UKSIM'09. 11th International Conference on*, pages 117–122. IEEE.

Strikwerda, J. (1989). *Finite Difference Schemes and Partial Differential Equations*. Chapman & Hall, New York, NY.

Verfürth, R. (1994). A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics*, 50(1):67–83.