GRID REFINEMENT ANALYSIS FOR VECTOR FUNCTIONS*

N. B. Petrovskaya
Keldysh Institute for Applied Mathematics, Moscow, Russia
e-mail: PetrovskayaN@yandex.ru

Рассмотрен процесс адаптации вычислительной сетки к решению системы обыкновенных дифференциальных уравнений в случае, когда для адаптации используется погрешность интерполяционного полинома. Задача адаптации в этом случае представляет особый интерес, поскольку для каждой компоненты решения необходимо выбрать скалярную функцию ошибки на основе знания погрешности. Описан подход, позволяющий анализировать стандартный алгоритм сгущения сетки. Этот подход применен в задаче адаптации к векторному решению, что дало возможность понять, как результаты адаптации зависят от выбора скалярной функции ошибки. Показано, что выбор такой функции должен быть согласован с механизмом сгущения сетки.

Introduction

Advantages of adaptive grids are well known in numerical solution of complex problems in science and engineering (e.g. see [1, 2] and the references therein). Basically, the algorithm of the solution adaptive grid generation includes two main steps. First, an error estimator adequate to the given problem is determined. After the error estimates are calculated for a given approximate solution, a grid adaptation strategy is elaborated to minimize the error over the grid. A local grid refinement is a widespread approach to adapt the grid to the solution [3 – 5]. Cutting a grid element to decrease its size results in the smaller error on the new grid. This adaptation strategy implies grid subdivision to be made only in the regions where the approximate solution is not accurate enough. For a scalar solution, a refinement criterion applied to the error estimate indicates correctly the regions where the insertion of new nodes is required. The situation is worse, however, when a vector solution is considered. The implementation of the adaptation criterion generally requires that the vector error estimate arising from the consideration of the vector solution should be reduced to a scalar estimate. One way to achieve this goal is to consider a scalar function for the adaptation. Below we refer to the scalar function used to adapt the grid to the vector solution as a key function. Usually the choice of the key function serves the needs of a particular problem under consideration. In practical calculations, one often extracts a solution component which is dominant in the underlying physical process and assigns it as a key function [5 – 7]. Another idea is to calculate

*The work was supported by The Boeing Company under contract 104W and Russian Foundation for Basic Research (grant N 03-01-00063).

the norm of the vector error estimate that may be reduced to calculation of the norm of the vector derivative [8, 9].

The main question arising in consideration of the key function is to what extent the chosen scalar function is good to provide for each scalar component of the solution the convergence rate compatible with that which would be obtained on the adaptive grids if each solution component were considered independently. Despite the importance of this issue, very little on the subject can be found in the literature [10]. Thus, the aim of our paper is to begin a discussion on the choice of scalar key function for adaptation to a vector solution.

The first results of our study presented in the paper concern interpolation error estimators. The interpolation error estimators based of the knowledge of the Taylor series expansion for the approximate solution have been successfully used many times for adaptive grid generation (see [1] and the references cited therein). The advantage of these estimators is that they can be computed directly from the approximate solution. Since the interpolation error estimates do not depend on the specific numerical method used in solution of the problem they are attractive for a wide range of applications [6, 7, 11, 12]. We present an efficient technique developed to analyze the refinement procedure based on interpolation error estimators. This technique is implemented to study the grid refinement for the vector solution to a system of ordinary differential equations (ODE). The results of our work demonstrate that even when a scalar function chosen for adaptation is reliable in the sense that it captures the desired features of the vector solution, the refinement criterion applied to such a function may lead to the grid refinement in a wrong region. In particular, we show that the choice of the key function which is not consistent with the adaptation criterion may dramatically slow down the convergence.

1. The problem statement

We consider a boundary-value problem for a system of ODEs in the general form:

$$\mathbb{D}\mathbf{x}(t) = \mathbf{g}(t), \tag{1}$$

where $\mathbb{D}$ denotes a differential operator which includes boundary conditions. The K-vector solution $\mathbf{x}(t)$=$(x_1(t), x_2(t), ..., x_K(t))$ to the problem (1) is defined on the closed interval $\Omega = [0, 1]$. The function $\mathbf{x}(t)$ is assumed to belong to the appropriate class of smooth functions.

The boundary-value problem (1) is solved numerically on a sequence of solution adaptive grids. Let N be the number of grid elements. After introducing the element partition of the region $\Omega = \bigcup_{i=1}^{N} e_i$, $e_i = [t_0^i, t_1^i], 1 \le i \le N$, and a mesh stepsize $h_i = t_1^i - t_0^i$, the grid refinement procedure is used for grid adaptation to the solution.

The adaptation approach considered in our work is based on interpolation error estimates. For a scalar function $x(t)$, interpolation error estimates are derived from a Taylor series expansion. If the piecewise linear representation is used for the approximate solution x_h, the interpolation error is dominated by the quadratic term in the Taylor series. The value

$$E_i = \ddot{x}_i h_i^2 \tag{2}$$

calculated at the midpoint t_i of the grid element e_i can be considered as a local error estimator on the element.

For the vector solution $\mathbf{x}(t)$, one should modify the interpolation error estimate to deal with a scalar function instead of a vector derivative, since the adaptation strategy usually requires

a scalar quantity to be minimized over the grid. Now the error estimate on the element reads

$$E_i = f_i h_i^2, \tag{3}$$

where the scalar function $f(t)$ is chosen to satisfy the requirements of the particular problem under consideration. A reasonable choice of the scalar key function for the adaptation is to use the norm of the derivative $\ddot{\mathbf{x}}(t)$. The consideration of the Euclidian norm yields the function

$$f(t) = \sqrt{\sum_{k=1}^{K} \ddot{x}_k(t)^2}. \tag{4}$$

The key function (4) allows us to interpret the value E_i in (3) as the Euclidian norm

$$E_i = ||\mathbf{E}_i|| = \sqrt{\sum_{k=1}^{K} E_{ik}^2}$$

of the vector error $\mathbf{E}_i = (E_{i_1}, E_{i_2}, ..., E_{i_K})$ on element e_i, the vector components $E_{i_k} = \ddot{x}_k(t_i)h_i^2$, $k = 1, ..., K$ being considered as local error estimators for each component of the vector solution $\mathbf{x}(t)$. Thus, the approach (4) imitates, in a certain sense, the adaptation procedure for each solution component.

After local error estimates (3) are calculated in each grid element, the refinement strategy is defined by the value $E_{\max}$:

$$E_{\max} = \max_{1 \le i \le N} \{E_i\}. \tag{5}$$

The elements e_j in which error estimates satisfy the condition

$$E_j > \tau * E_{\max}, \tag{6}$$

where τ is a specified tolerance, $0 < \tau < 1$, are refined by cutting them into halves. The approximate solution is then calculated over the new grid and the procedure (3), (6) is repeated to meet the convergence with the prescribed accuracy.

For a scalar solution, the criterion (6) applied to the error estimator (2) correctly indicates the regions where the refinement is required. It is not true, however, for a vector interpolation error estimate reduced to a scalar key function. Below we analyze the refinement procedure (3), (6) and show that the refinement criterion (6) may not properly work even if the choice (4) is reasonable.

2. The analysis of the field $\varepsilon_u(t, N)$

In this section we introduce a parametric family $\mathcal{E}_u$ of continuous error functions on uniform grids. We will denote these functions as $\varepsilon_u(t, N)$. Grid adaptation based on the algorithm (3), (6) results in a nonuniform grid. However, starting with a uniform initial grid (that is a common choice for many computational problems), the refinement procedure (3), (6) is fully determined by the error on the uniform grids. The concept of the continuous error functions appeared to be useful in the analysis of the behaviour of the error (3) over the nonuniform grid.

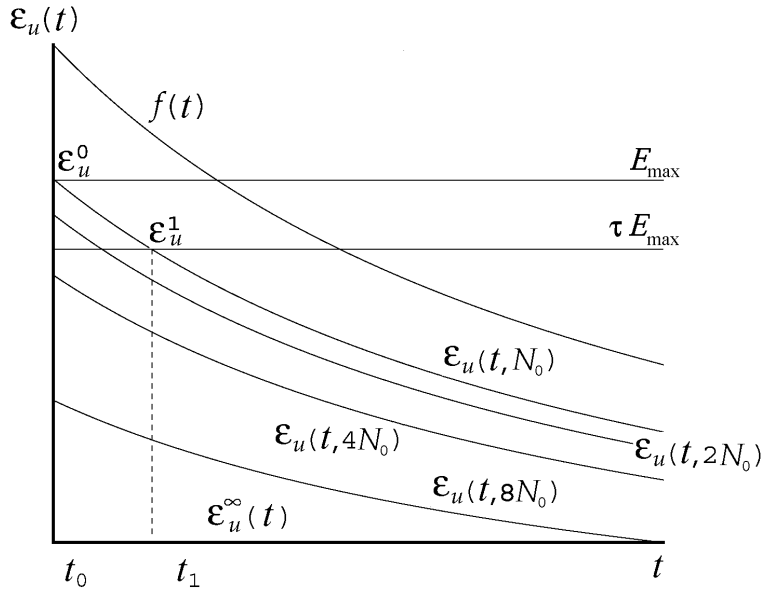

Fig. 1. Representation of the interpolation error estimate in the $(t, \varepsilon_u(t))$-plane.

Let N be the number of grid elements and $h_u = h_u(N) = 1/N$ be a mesh stepsize on a uniform grid $G_u(N)$. For the given key function $f(t)$, the continuous function $\varepsilon_u(t, N) \subset \mathcal{E}_u$ is defined as

$$\varepsilon_u(t, N) = \alpha(N) f(t), \tag{7}$$

where $\alpha(N) = h_u^2$.

The parametric family of the functions $\varepsilon_u(t, N)$ generated by the function $f(t)$ is shown in fig. 1 for a monotone function $f(t)$. The field $\mathcal{E}_u$ is discrete due to the integer parameter N in (7). The region, where the curves $\varepsilon_u(t, N)$ are located, has the function $f(t)$ as its upper boundary. It follows from the definition (7) that for any two curves $\varepsilon_u(t, N_1)$ and $\varepsilon_u(t, N_2)$ the relation holds

$$\varepsilon_u(t, N_2) = \gamma \varepsilon_u(t, N_1), \tag{8}$$

where $\gamma = (N_1/N_2)^2$. Since $\gamma \to 0$ when the number of grid elements $N_2 \to \infty$, the curve $\varepsilon_u^\infty(t)$, which coincides with the t-axis and represents an equidistributed error on a uniform grid, is the lower boundary for the field $\varepsilon_u(t, N)$.

Suppose that the function $f(t)$ is calculated from the exact solution. Evidently, in this case the discrete error (3) over the uniform grid $G_u(N)$ is

$$E_i \equiv \varepsilon_{u_i}, \quad 1 \le i \le N, \tag{9}$$

where ε_{u_i} is the value of the function $\varepsilon_u(t, N)$ calculated at the midpoint t_i of the grid element. In the $(t, \varepsilon_u(t))$-plane, the error $E_i, 1 \le i \le N$ considered over the uniform grid is represented by a set of points distributed over the curve $\varepsilon_u(t, N)$.

Now the refinement procedure can be interpreted in terms of transition between different curves $\varepsilon_u(t, N)$ in the $(t, \varepsilon_u(t))$-plane (see fig. 1). Let N_0 be the number of grid elements on the uniform initial grid $G(N_0)$ which produces the error $\varepsilon_u(t, N_0)$. According to the algorithm (5), (6), the error should be reduced at the part $[\varepsilon_u^0, \varepsilon_u^1]$ of the error curve $\varepsilon_u(t, N_0)$. The interval $[t_0, t_1]$ is subject to the refinement, as the segment $[\varepsilon_u^0, \varepsilon_u^1]$ of the error curve corresponding to the interval $[t_0, t_1]$ lies inside the band $[\tau * E_{\max}, E_{\max}]$. Since each grid element belonging to

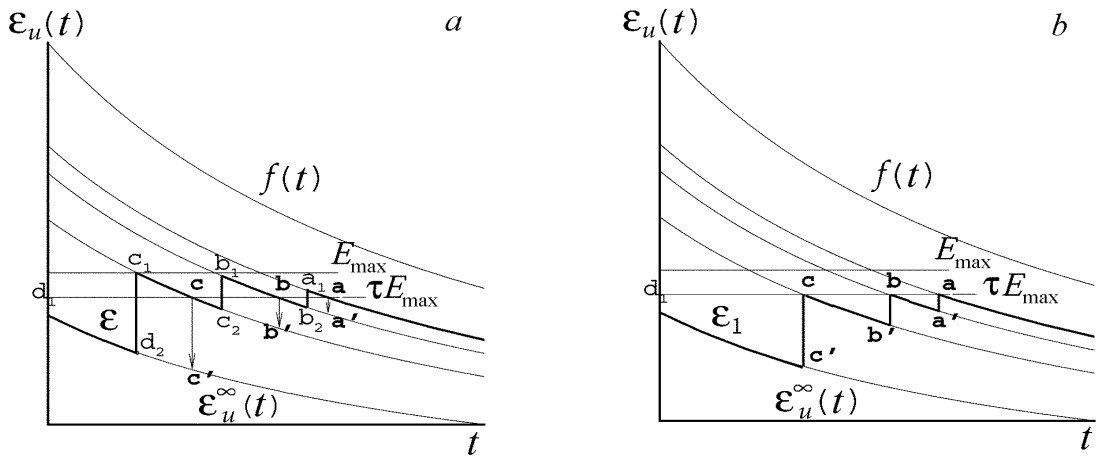

Fig. 2. The refinement procedure in the $(t, \varepsilon_u(t))$-plane. The interpolation error curve (a) before and (b) after the refinement.

the interval $[t_0, t_1]$ is cut into halves, the refinement of $[t_0, t_1]$ can be considered as generation of the uniform grid $G_u(2N_0)$ with the grid stepsize $h_u = 1/(2N_0)$ over the interval. Thus, the refinement (5), (6) results in a jump from the curve $\varepsilon_u(t, N_0)$ in the $(t, \varepsilon_u(t))$-plane to the curve $\varepsilon_u(t, 2N_0)$ which is related to the error on the uniform grid with the doubled number of grid elements [1]. The next steps of the refinement procedure involve the curves $\varepsilon_u(t, 4N_0)$, $\varepsilon_u(t, 8N_0)$, etc., a nonuniform grid being generated as a result.

The refinement process (5), (6) is fully controlled by the function $\varepsilon_u(t, N_0)$. Suppose that the curve $\mathcal{E}$: $d_1d_2 - c_1c_2 - b_1b_2 - a_1a$ which represents the error on the nonuniform grid $G(N)$ is obtained after several refinement steps (see fig. 2, a where the error curve $\mathcal{E}$ is shown in bold). At the current step of the refinement the error is reduced at the parts c_1c, b_1b, and a_1a of the curve $\mathcal{E}$. A jump from a higher level of the error to a lower one is shown schematically by arrows. The curve $\mathcal{E}_1$: $d_1c' - cb' - ba' - a$ shown in fig. 2, b represents a new error distribution after the refinement step. Let $m \equiv m_k$ be the number of refinements made on the interval corresponding to the segment s_k of the error curve $\mathcal{E}_1$. Since the relation (8) holds, the error $\varepsilon_u^k(t, N_k)$ at the segment s_k reads

$$\varepsilon_u^k(t, N_k) = \gamma_k \varepsilon_u(t, N_0) = \left(\frac{N_0}{N_k}\right)^2 \varepsilon_u(t, N_0) = \frac{1}{2^{2m}} \frac{1}{N_0^2} f(t), \tag{10}$$

where $N_k = 2^m N_0$ is the number of grid elements at the interval. Provided the initial grid is uniform, the error $\mathcal{E}_1$ is only defined by the function $f(t)$ and the number N_0 of grid elements on the uniform initial grid. Below we discuss the two factors controlling the value of the interpolation error on adaptive grids in more detail.

3. The choice of the function $f(t)$

It follows from the error estimate (3) that, while solving a system of equations, one may expect a uniform grid obtained as a result of the refinement. Actually, the condition

$$E_i = \text{const} \equiv C_1, \ \forall i = 1, 2, ..., N$$

[1] For the sake of clarity, the distance between the curves is not properly scaled in the figure.

is sufficient for the uniform refinement of the whole domain Ω, provided the algorithm (3), (6) is applied. Let, for instance, a system of two equations be considered. Evidently, that any vector function $\mathbf{x}(t) = \{x_1(t),\, x_2(t)\}$, which holds the condition

$$\ddot{x}_2(t) = \sqrt{C - (\ddot{x}_1(t))^2}, \tag{11}$$

keeps the grid uniform during the adaptation process (3), (6), provided the uniform initial grid is generated.

It is interesting to compare the refinement for the vector solution (11) with the procedure (2), (6) made for each solution component independently. Consider an example of the solution $(x_1(t), x_2(t))$ such that the solution derivatives are $\ddot{x}_1(t) = \dfrac{1}{1+t}$, $\ddot{x}_2(t) = \dfrac{\sqrt{2t+t^2}}{1+t}$. For the above solution we have $C = 1$, and $E_i = h_u^2$, $\forall i = 1, ..., N$. Meanwhile, each component of the solution, considered separately, only requires the refinement in a local region. Since the derivative $x_1^{(3)}(t) = \dfrac{-1}{(1+t)^2} < 0$, $\forall t \in [0,1]$, is a monotone function, the maximum of the function $\ddot{x}_1(t)$ is $\max\limits_{t\in[0,1]} \ddot{x}_1(t) = \ddot{x}_1(t_0 = 0) = 1$. For the second solution component, $x_2^{(3)}(t) = \dfrac{1}{(1+t)^{3/2}} > 0, \forall t \in [0,1]$; therefore, $\max\limits_{t\in[0,1]} \ddot{x}_2(t) = \ddot{x}_2(t_0 = 1) = \sqrt{3}/2$. Thus, according to the algorithm (2), (6), a uniform initial grid should be refined near the left and the right endpoints of the interval $[0,1]$, respectively.

Let us examine the refinement procedure for the sloping function $f(t) \approx$ const which generates the initial error $\varepsilon_u(t, N_0)$ in the $(t, \varepsilon_u(t, N)$-plane (see curve A in fig. 3). The error after the refinement is presented by the curve $\mathcal{E}$: $bb' - a'A$ in the figure. It can be seen from the figure that the closer the function $f(t)$ is to a constant value, the wider is the region $[t_0, t_1]$ to be uniformly refined. In the limiting case that the line $\tau * E_{\max}$ is an asymptote for the function $f(t)$, the number of grid elements is doubled over the whole domain Ω, even if the vector $\mathbf{x}(t)$=$(x_1(t), x_2(t))$ is not the solution to the equation (11).

The algorithm (3), (6) provides the convergence of the interpolation error (3) to the final curve $\varepsilon_f(t)$ in the L^∞-norm. At each refinement step the maximum value $E_{\max}$ of the current error is reduced by multiplying it by the factor 1/4. However, the requirement of the error reduction in the L^∞-norm can be achieved without transition to the error curve corresponding to the grid with the doubled number of the nodes.

Implying the convergence in the L^∞-norm, the obvious way to insert the fewer number of new nodes into the grid is to use the curve $\varepsilon_u(t, N_1)$ (the curve C in fig. 3) instead of the curve $\varepsilon_u(t, 2N_0)$ (the curve B in the figure). Now the distribution of the error after the refinement step is presented by the curve $\mathcal{E}_{\text{corr}}$: $cc' - a'A$. It can be seen from the figure that the requirement

$$E_{\max}^{\text{new}} \le \tau E_{\max}$$

holds for the maximum $E_{\max}^{\text{new}}$ on the new error curve. The error (3) is reduced in the L^∞-norm, while the number of elements on the new grid $N_1 < 2N_0$.

The transition to the curve $\mathcal{E}_{\text{corr}}$ may be reached as follows. The relation (8) written for the curves $\varepsilon_u(t, N_0)$ and $\varepsilon_u(t, N_1)$ yields at the maximum point

$$E_{\max} = (N_1/N_0)^2 \tau E_{\max}. \tag{12}$$

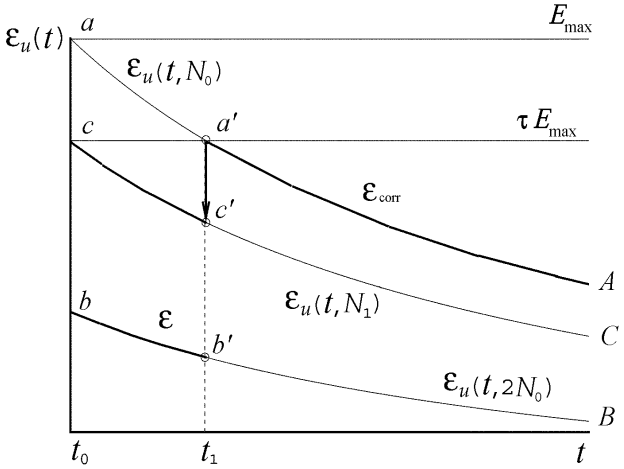

Fig. 3. The correction of the refinement algorithm for a sloping function $f(t)$. Representation of the correction algorithm as the transition between the curves in the $(t, \varepsilon_u(t))$-plane.

It follows from (12) that the inequality $N_1 < 2N_0$ holds only for the refinement parameter $1/4 < \tau < 1$. For the value $\tau < 1/4$, the curve $\varepsilon_u(t, N_1)$ lies under the curve $\varepsilon_u(t, 2N_0)$ that corresponds to the larger number of the nodes inserted into the grid.

Let the parameter $\tau > 1/4$ be defined for the refinement procedure. The number of nodes corresponding to the curve $\mathcal{E}_1$ is obtained from (12) as

$$N_1 = [(N_0/\sqrt{\tau})], \tag{13}$$

where the brackets denote the operation of taking the integer part of the fraction. The number n_1 of new nodes to be inserted into the segment $[t_i, t_{i+1}]$ marked for the refinement is calculated as

$$n_1 = [N_1 h_i] \tag{14}$$

where $h_i = t_{i+1} - t_i$. The new n_1 nodes are equidistributed over the segment $[t_i, t_{i+1}]$.

The consideration above leads us to the conclusion that, staying within the framework of the refinement algorithm (3), (6), it is impossible to reduce the number of the nodes inserted into the grid. The transition to the curve $\varepsilon_u(t, N_1)$ requires the redistribution of grid nodes. This appears from the fact that for the procedure (3), (6) the number of new nodes added to the grid at each refinement step depends on the number of nodes on the initial grid. It follows from the estimate (10) that every initial grid produces its own parametric family $\{\varepsilon_u(t, N)\}$. Hence, the refinement (5), (6) does not allow the transition between any two curves $\varepsilon_u(t, N_1)$ and $\varepsilon_u(t, N_2)$ generated starting from different initial grids.

There are some formal ways to avoid the situation that the number of the nodes is unnecessarily doubled over the grid at each refinement step. For instance, if the norm (4) yields a sloping function, it is possible to consider another norm of the vector derivative in order to define the function $f(t)$. Another idea is to modify the interpolation error estimate by including higher derivatives into consideration, provided the higher order polynomial presentation of the approximate solution is used. However, such modifications do not ensure that a more efficient refinement procedure will be provided.

4. The choice of the initial grid

The issue of the initial grid arises from the discrete nature of the error estimate (3). Since the function $f(t)$ is discretized to calculate (3), the initial error curve $\varepsilon_u(t, N_0)$ governing the refinement process should be considered as a discrete set of points. Since the position of the maximum defined over this discrete set may not coincide with the maximum point on the continuous error curve, the refinement criterion (6) applied to the discrete error curve may indicate the regions where the refinement is not actually needed.

An evident example associated with the estimate (10) is that the uniform initial grid $G(N_0)$ may not be fine enough as to resolve the particular features of the function $f(t)$ on the initial curve $\varepsilon_u(t, N_0)$. The worst situation is that a singularity is located near the endpoint opposite to that where the maximum of $f(t)$ is located. If the function $f(t)$ on the nodes of the coarse grid $G(N_0)$ does not bring any information about the location of the singularity, the procedure (3), (6) will generate many odd refinement steps. In a situation like this, the generation of the fine initial grid which indicates the location of the singularity may appear to be a more effective approach than the excessive refinement of the coarse grid.

Meanwhile, a fine grid is not always the optimal choice to start the refinement (3), (6). The following numerical example illustrates the impact of the initial grid on the adaptation to the vector solution. We consider the boundary-value problem for the model system of ODE with Dirichlet boundary conditions on the region $\Omega = [0, 1]$

$$\begin{cases} \dfrac{d^2x_1(t)}{dt^2} + \mu x_1(t) + \gamma x_2(t) = 0, \\ \dfrac{dx_2}{dt} + \delta x_2(t) = 0,\ t \in (0, 1), \\ x_1(0) = 0,\ x_1(1) = 1,\ x_2(0) = 1. \end{cases} \tag{15}$$

Since in our numerical calculations we are interested in the study of the refinement procedure rather than the physical aspects of the problem, the system parameters are chosen $\gamma = 0.0$, $\mu = -100.0$, $\delta = 20.0$. The choice $\gamma = 0.0$ makes the system decoupled giving us the opportunity to compare the results of the adaptation to the vector solution with that obtained for each equation considered independently.

The analytical solution to the decoupled boundary problem (15) is

$$x_1(t) = A\left(e^{\alpha_1 t} - e^{\alpha_2 t}\right), \quad x_2(t) = e^{-\delta t}, \tag{16}$$

where $\alpha_{1,2} = \pm\sqrt{-\mu}$, $\quad A = \dfrac{1}{2sh\sqrt{-\mu}}$. The solution $\mathbf{x}(t)$ is shown in fig. $4, a$. Since the regions, where either solution component has the steep gradient, are located near the opposite endpoints of the interval, one may expect two nonoverlapping regions to be marked for the refinement. Suppose that each equation in the system (15) is solved independently on a sequence of adaptive grids, and N_k, $k = 1, 2$ is the number of grid elements required to obtain the solution error for the scalar solution component $x_k(t)$ with the given accuracy. Let now N be the number of grid elements required to obtain the same value of the error on the grid adapted to the vector solution. In the latter case, the solution error is calculated, for instance, as the appropriate norm of the vector solution error. For the decoupled boundary problem (15), the most favorable choice of the function $f(t)$ would be that providing the estimate

$$N \approx N_1 + N_2. \tag{17}$$

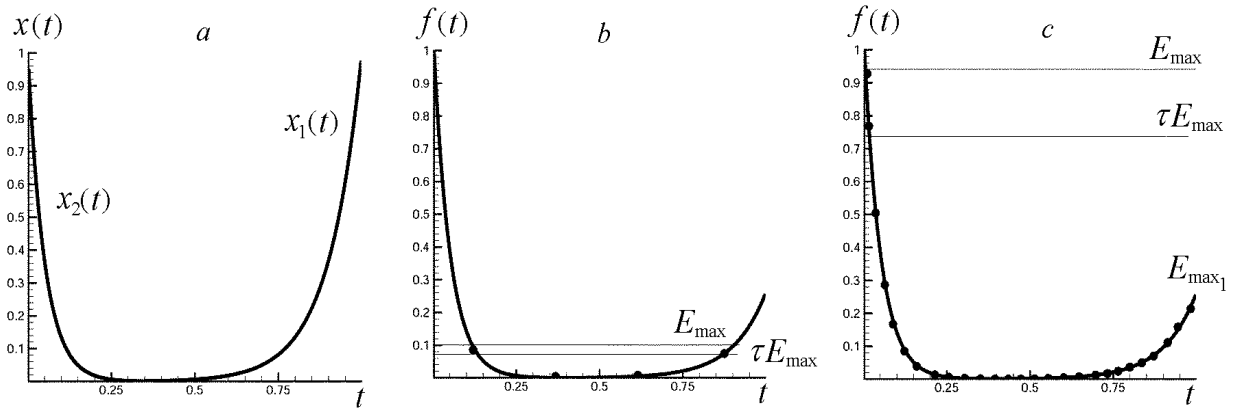

Fig. 4. The components $x_1(t)$ and $x_2(t)$ of the vector solution to the problem (15) (a). The refinement scenario for the problem (15) on (b) the coarse and (c) fine initial mesh.

The estimate (17) corresponds to the simultaneous refinement of both solution components in the nonoverlapping regions.

Consider the key function $f(t)$ defined as the norm (4) of the vector derivative. Taking into account the vector solution (16), the function $f(t)$ is calculated as

$$f(t) = \sqrt{\ddot{x}_1(t)^2 + \ddot{x}_2(t)^2} = \sqrt{\mu^2 x_1(t)^2 + \delta^4 x_2(t)^2}. \tag{18}$$

It follows from (16), (18) that $f(t) \approx x_1(t)$, and $f(t) \approx x_2(t)$ near the right and the left interval endpoints, respectively. Thus, the continuous function $f(t)$ seems to provide the refinement scenario (17) as it captures the regions of steep gradient for both solution components. However, the results of the adaptation depend strongly on which initial grid is generated.

Our first numerical test is to study the convergence on adaptive grids, provided the initial coarse grid is considered. The discrete function $f(t_i)$ over the uniform coarse grid is shown in fig. $4, b$. It can be seen from the figure that for each solution component the whole region of its steep gradient lies inside a single grid element. The refinement criterion (6) applied to the initial error curve captures any region where the refinement is required for either solution component.

To obtain the numerical solution to the problem, we use a standard finite element method with a piecewise linear presentation of the approximate solution on adaptive grids. The error estimate is computed directly from the exact solution in order to eliminate the influence of approximate derivatives. The convergence history on adaptive grids is presented in fig. $5, a$, where the results of the adaptation to the vector solution are compared with the solution error obtained on adaptive grids generated for each solution component considered separately. The curve I represents the solution error obtained when the key function (18) is employed for the adaptation. The curves II' and II'' present the solution error calculated for the scalar function $x_k(t)$, $k = 1, 2$, on grids generated as result of the adaptation to the solution component $x_k(t)$, $k = 1, 2$.

The error for the vector solution on element e_i is defined as to be consistent with the choice of the key function. Namely, we calculate the norm of the vector error

$$\mathrm{err}_i = \sqrt{\mathrm{err}_{i_1}^2 + \mathrm{err}_{i_2}^2}. \tag{19}$$

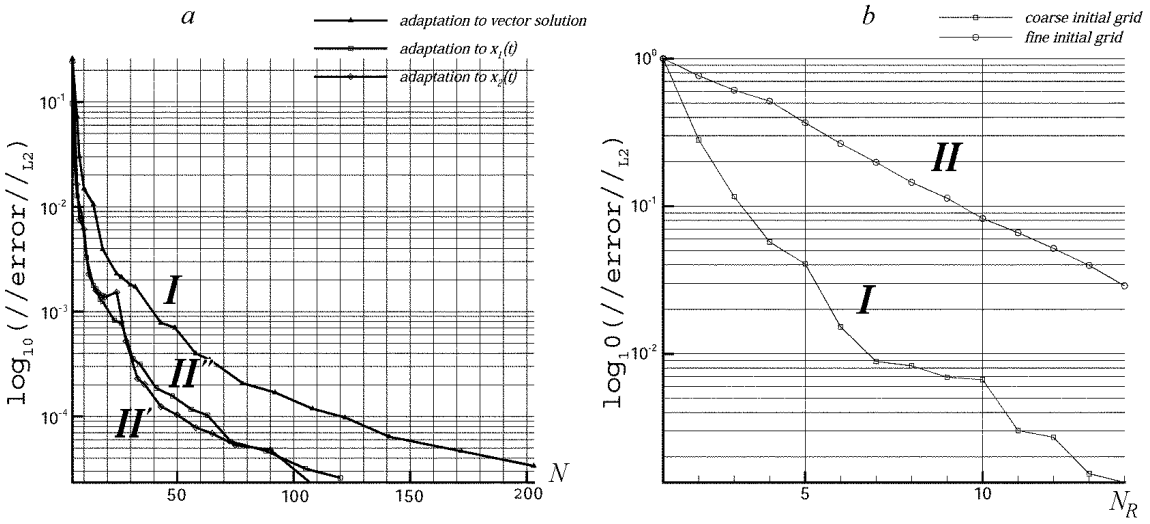

Fig. 5. The convergence history on adaptive meshes for the problem (15). (a) The curve I presents the L^2-norm of the solution error calculated on grids adapted to the vector solution. The curves II' and II'' present the L^2-norm of the solution error calculated for the solution component $x_k(t)$, $k = 1, 2$ on grids generated as a result of the adaptation to the scalar function $x_1(t)$ and $x_2(t)$, respectively. N is the number of grid cells. (b) The relative solution error is calculated on adaptive grids obtained with the error estimate (3), (18), beginning with the coarse (curve I) and the fine (curve II) meshes. N_R is the number of refinements.

Note that the error components err_{i_k} are different from those calculated to plot the curves II' and II''. The scalar error err_{i_k}, $k = 1, 2$, in (19) is calculated for the k-th solution component over the grid obtained as a result of the adaptation to the vector solution.

The error measured in the L^2-norm is shown in the semilogarithmic scale. In all cases considered in the test the uniform initial grid contains $N_0 = 5$ nodes. It can be seen from the figure that, being in a good agreement with the estimate (17), the convergence plots confirm the refinement scenario discussed above. The situation is worse, however, when the initial fine grid is generated. In this case, the region of the steep gradient of the function $x_2(t)$ is already resolved on the initial grid (see fig. $4, c$) that makes the refinement criterion (6) less effective. Now the criterion (6) is not sensitive to the behaviour of the solution component $x_1(t)$.

First steps of the procedure (5), (6) do not capture the region where the refinement is required to resolve the function $x_1(t)$. New nodes are only inserted near the left endpoint of the interval. At each refinement step, the position of the maximum is shifted along the curve $\varepsilon_u(t, N_0)$ until the error $E_{\max}$ is reduced to the value $E_{\max} \approx E_{\max_1}$. The local maximum $E_{\max_1}$ corresponds to the maximum on the error curve generated for the solution component $x_1(t)$. Only after the error $E_{\max_1}$ is reached, both regions where the refinement is required lie inside the band $[\tau * E_{\max}, E_{\max}]$ that ensures the simultaneous refinement for both scalar functions $x_1(t)$ and $x_2(t)$.

Let us compare the adaptation to the vector solution, starting with the coarse and fine grids. The function (18) is employed as a key function for the adaptation. The curves I and II in fig. $5, b$ present the relative convergence rate for the coarse (the number of nodes $N_c = 5$) and fine ($N_f = 50$) uniform initial grid, respectively. The L^2-norm of the solution error (19) scaled by one is shown versus the number N_R of refinement cycles. The convergence plots demonstrate the slower convergence rate when the fine initial grid is generated for the grid

adaptation. It follows from the previous consideration that the refinement of the fine initial grid reduces only the component err_{i_2} of the solution error (19) over the grid, while the solution adaptation initiated from the coarse grid impacts on both of the components of the error.

The choice of the initial grid may be considered as the problem of the initial discretization of the function $f(t)$. Thus, the above difficulties associated with the choice of the initial grid indicate that the key function (18) is not consistent with the refinement criterion used in the problem.

Conclusions

In our paper we have presented a new technique which appeared to be convenient for the analysis of grid adaptation based on interpolation error estimators. We have implemented the developed approach to study the grid refinement procedure for vector functions. The analysis made in the work has shown that for vector solutions the results of the adaptation depend strongly on the choice of the scalar key function $f(t)$ and the initial grid. Comparing these two factors, the problem of the key function is more important, as the choice of the initial grid may be considered in terms of the initial discretization of the function $f(t)$.

Solving the problem of adaptive grid generation, the issues of the error estimate and the adaptation strategy are generally viewed as being independent of each other. Meanwhile, it has been demonstrated in our paper that the form of the error estimate should be consistent with the adaptation criterion used in the problem. Dealing with the interpolation error estimates, this requirement is especially important when an error estimate of the vector solution is elaborated. The results of our work show that the refinement criterion applied to a scalar function chosen for the adaptation may indicate wrong regions where grid enrichment is needed. In the latter case the convergence rate obtained on adaptive grids is close to that on uniform grids, even though the desired features of the solution to a given problem are taken into account in the choice of the key function involved in the estimate.

The problem of the key function requires a further careful study and should be solved taking into account the adaptation strategy used to generate the grid. One way to avoid the quasiuniform grid as a result of the refinement is to adjust the adaptation strategy to the error estimator used in the problem. Instead of seeking a key function which serves the needs of the considered refinement algorithm, one can try to render the refinement procedure more efficient (e.g. combining refinement with node moving) for a given scalar field. For vector functions, this approach to the adaptation may appear to be effective in the case when the standard refinement criterion does not work for the chosen key function.

References

[1] Carey G.F. Computational Grids: generation, adaptation, and solution strategies. Washington: Taylor & Francis, 1997. 496 p.

[2] Gil'manov A.N. Methods of adaptive meshes in gas dynamic problems. M.: Nauka, 2000. 248 p. (in Russian).

[3] Berger M.J. and Colella P. Local adaptive mesh refinement for shock hydrodynamics // J. Comput. Phys. 1989. N 82. P. 64–84.

[4] Lebedev A.S., Liseikin V.D., Hakimzyanov G.S. Approaches to adaptive grid generation // Comput. Technologies. 2002. Vol. 7, N 3. P. 29–43 (in Russian).

[5] Bieterman M.B., Bussoletti J.E., Hilmes C.L. et al. An adaptive grid method for analysis of 3D aircraft configurations // Comput. Methods Appl. Mech. Eng. 1992. N 101. P. 225–249.

[6] Borges L., Feijóo R., Zouain N. A directional error estimator for adaptive limit analysis // Mech. Research Comm. 1999. N 26. P. 555–563.

[7] Martynov A.A., Medvedev S.Yu. A robust method of anisotropic grid generation // Workshop "Grid Generation: Theory and Applications". M., 2002.

[8] Ivanenko S.A, Azarenok B.N. Application of adaptive grids in numerical analysis of time-dependent problems in gas dynamics // Intern. J. for Numer. Meth. in Fluids. 2002. Vol. 39, N 1. P. 1–22.

[9] Pereyra V., Sewell E.G. Mesh selection for discrete solution of Boundary problems in ordinary differential equations // Numer. Math. 1975. Vol. 23. P. 261–268.

[10] Borouchaki H., Castro-Diaz M.J., George P.L. et al. Anisotropic adaptive mesh generation in two dimensions for CFD // 5th Intern. Conf. On Numerical Grid Generation in Computational Field Simulations. Mississippi State Univ., 1996. Vol. 3. P. 197–206.

[11] Sorokin A.M., Vladimirova N.V. Anisotropic adaptation of unstructured volume grids to numerical solutions of three dimensional gas dynamics problems // Workshop "Grid Generation: Theory and Applications". M., 2002.

[12] Buscaglia G.S., Dari E. Anisotropic mesh optimization and its application in adaptivity // Intern. J. Numer. Methods Eng. 1997. N 40. P. 4119–4136.

Received for publication February 16, 2004