KELDYSH INSTITUTE of APPLIED MATHEMATICS
RUSSIAN ACADEMY of SCIENCES

N.B.Petrovskaya

Minimization of Interpolation Error Estimates for Vector Functions.

Moscow

N.B.Petrovskaya

Minimization of Interpolation Error Estimates for Vector Functions.

Abstract

For interpolation error estimates, the adaptation problem for a vector function is different from that for a scalar solution.Since solution components have different error estimates, the problem appears how to generate a grid which is optimal for each scalar solution function. In our work, we address the issue of the efficient adaptation strategy in case that the grid is adapted to a vector solution.

Introduction. [1] In the present work we address the issues arising when grid adaptation to the solution is considered for a system of equations. Dealing with interpolation error estimates, which are common in many practical applications, the adaptation problem for a vector function is different from that for a scalar solution. Ideally, each solution component requires its own grid to provide the optimal convergence rate. Thus, a vector interpolation error estimate should be computed at each solution-adaptation iteration. Since the computation of the vector error estimate renders the adaptation procedure inefficient, the problem appears how to generate a grid which is optimal for each scalar solution function, provided the computational cost of the algorithm is relatively low.

One approach to resolve the issue of efficiency is to consider a scalar function as a key function for the adaptation to the vector solution. In practical calculations, the simplest strategy is to assign a solution component as a key function. In aerodynamic applications, the physics of the problem dictates the choice of the pressure or a scalar function of the velocity field for the adaptation purposes (*e.g.*, see [1]). Another example is given by a plastic deformation scalar field, which is often used to generate solution adaptive grids in the problems of elastoplasticity [4]. In spite of the convenience of this approach, grids obtained as a result of the adaptation to a single component of the solution do not always provide the optimal convergence for the other components. Thus, another idea is to use the norm of the vector error estimate as a key function. This approach imitates, in a certain sense, the adaptation procedure for each solution component. However, the key function based on the norm of the vector derivative may result in a uniform refinement throughout a whole domain of interest, whereas each component of the solution, being considered separately, requires only a refinement in a local region [13]. That gives rise to the issue of the choice of the key function for the adaptation.

Another possible statement of the adaptation problem for vector functions concerns an adaptation strategy. Instead of seeking a key function which serves the needs of the adaptation algorithm, one can try to modify the adaptation procedure to improve the results for the adaptation to an arbitrary scalar function. Since a proper adaptation strategy implies that minimization of the error is ensured for a given scalar field, there are no particular requirements to the key function in this approach, except for it should be easily computable and reliable in the sense that it captures the

[1]This work was supported by The Boeing Company under contract No 104Y and Russian Foundation for Basic Research, grant No 03-01-00063.

desired features of the vector solution. Apparently, the norm of the vector function satisfies both requirements. In our work, we address the issue of the efficient adaptation strategy based on interpolation error estimates. The earlier study has been carried out in [13], where the attempt has been made to correct the refinement algorithm in case that the standard refinement procedure generates a quasiunifom grid. In the present work, a general algorithm is developed to minimize the interpolation error estimate for a system of ordinary differential equations.

1. The statement of the adaptation problem for vector functions. We consider a smooth vector function $\mathbf{x}(t) = \{x_1(t), x_2(t),...,x_K(t)\}$ ($x_k(t) \in C^\infty$, $\forall k = 1, ..., K$) on closed interval $\Omega = [0, 1]$. Function $\mathbf{x}(t)$ is defined as a solution to the general boundary problem

$$\mathbb{D}\mathbf{x}(\mathbf{t}) = \mathbf{g}(\mathbf{t}); \qquad \mathbf{x}(0) = \mathbf{x_0},\ \mathbf{x}(1) = \mathbf{x_1}, \tag{1}$$

where $\mathbb{D}$ denotes a differential operator.

The system (1) is solved numerically on a sequence of solution adaptive grids. Let us introduce the element partition of the region $\Omega = \bigcup_{i=1}^{N} e_i$, $e_i = [t_0^i, t_1^i]$, and the mesh stepsize $H_i = t_1^i - t_0^i$, $1 \le i \le N$, where N is the number of grid elements. We define the grid $\mathbf{H}$ as a set $\{H_i\}$, $1 \le i \le N$. The grid $\mathbf{H}$ is considered to be the optimal one, if a discrete function called "the error estimate" is minimized over the grid. The definition of the error estimate will be discussed later.

The adaptation problem for the vector function $\mathbf{x}(t)$ may be stated as follows. Suppose that we are able to generate the optimal grid $\mathbf{H}_k$ for each solution component $x_k(t)$. After the errors are calculated over the grids $\{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_K\}$, the optimal grid for the system can be considered as a node distribution which holds the selected criterion of the vector error minimization on any grid element e_i. Such a formulation implies that the adaptation procedure comprises the two basic steps:

- **Problem *I*.**

  A. Grid generation for scalar functions:

  - The choice of the error estimate which guides the adaptation procedure for each solution component;
  - The choice of the adaptation strategy for each solution component;

B. Transformation of a "multigrid" system obtained at the step A to a single grid;

In practice, a usual way to tackle the problem I is to eliminate step A from the consideration, as its implementation makes the adaptation process inefficient, and to deal with the following problem:

- **Problem II.**

  A. The choice of a scalar key function for the adaptation;

  B. The choice of the error estimate and the adaptation strategy for the key function;

Solving the problem II allows us to avoid generation of a grid set $\{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_K\}$. However, it may happen that being implemented instead of solving the full problem I, the adaptation strategy II results in a grid which is not optimal for a particular problem. Below we demonstrate the difference between the results of the adaptation II and that obtained with the algorithm I.

We begin with the discussion of the standard refinement procedure. For our further purposes we consider interpolation error estimates based on the second derivative of the solution. For scalar function $x(t)$, error estimate E_i on element e_i is defined as

$$E_i = \ddot{x}(t_i)H_i^2, \tag{2}$$

where t_i is the midpoint of the element. After calculating local error indicators E_i, $1 \leq i \leq N$ over the grid, those elements e_j are refined, which error estimates satisfy the condition

$$E_j > \tau * E_{max}, \tag{3}$$

where $E_{max} = \max\limits_{1 \leq i \leq N} \{E_i\}$, and τ is a specified tolerance.

For the adaptation process based on the algorithm II, the common choice of the key function $F(t)$ is to define it as a solution to the equation

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

Then the local error indicator E_i on element e_i is calculated as

$$E_i = \ddot{F}(t_i)H_i^2 = ||\ddot{\mathbf{x}}(t_i)||_{L^2} H_i^2. \tag{5}$$

At the first glance, the approach (4), (5) imitates the algorithm I, since the choice of the key function (4) affords to interpret the value E_i as the Euclidian norm of the error vector $\mathbf{E}_i = (E_{i1}, E_{i2}, ..., E_{iK})$ on element e_i, $E_i = ||\mathbf{E}_i|| = \sqrt{\sum_{k=1}^{K} E_{ik}^2}$. The vector components $E_{ik} = \ddot{x}_k(t_i)H_k^2$, $k = 1, ..., K$, calculated at the midpoint t_i of element e_i may be considered as local error indicators for each component of vector solution $\mathbf{x}(t)$. However, the approach (5) is not always correct. It is not difficult to see that for any vector function $\mathbf{x}(t)$, which scalar components satisfy

$$\ddot{x}_k(t) = \sqrt{C - \sum_{l \neq k}^{K} \ddot{x}_l(t)^2}, \qquad C = const, \tag{6}$$

the initial grid is refined uniformly due to the function $F(t) = const$ in the error estimate (5). Meanwhile, if we consider each scalar function $x_k(t)$, $k = 1, ..., K$, the grid refinement may be only required in a local region. Do we really "double toil and trouble" at every refinement step, or one should admit a uniform grid to be the optimal grid for vector function (6)? To answer these questions, we below discuss the adaptation procedure based on the solution of the full problem I.

2. The solution of full adaptation problem. In our further consideration we assume that for scalar functions the interpolation error estimate based on the second derivative provides the best adaptation to the solution. In other words, the interpolation error (2) is assumed to be the same as the solution error. Also, the refinement procedure (5), (3) is assumed to be the best adaptation technique in solution of the problem I.

Let optimal node distributions $\{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_K\}$ be generated as a result of step A in problem I. The interpolation error is $\mathbf{E}_k = (E_{1k}, E_{2k}, ..., E_{Nk})$ for the scalar function $x_k(t)$ over grid $\mathbf{H}_k$. After the error vector $\mathbf{E}_i = (E_{i1}, E_{i2}, ..., E_{iK})$ is formed on element e_i, the optimal grid for the system (1) may be defined as the node distribution $\mathbf{H}$ which provides the error $E_i = ||\mathbf{E}_i||$ on each element e_i, $1 \leq i \leq N$ [2].

We consider the Euclidian norm of the error vector $\mathbf{E}$ on element e_i, $||\mathbf{E}_i||_{L^2} = \sqrt{\sum_k E_{ik}^2}$, $k = 1, ...K$. After the norm of the error vector is defined,

[2]It is possible to give another formal definition for function $\mathbf{H}$, as another criterion of error minimization can be selected, according to the general formulation of problem I.

the problem is stated as follows:
For each grid element e_i, $1 \le i \le N$, find error estimate E_i in the form

$$E_i = \ddot{\mathcal{F}}_i H_i^2 \tag{7}$$

which ensures the error $E_i = ||\mathbf{E}_i||_{L^2}$ on the element e_i, provided the condition

$$\sum_{i=1}^{N} H_i = 1, \quad 1 \le i \le N. \tag{8}$$

holds. The choice of the error estimate in form (7) is dictated by the requirement that the error estimate on grid $\mathbf{H}$ should be consistent with that used to generate grids $\{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_K\}$. The condition (8) is necessary for the formulation of the problem, as it guarantees that the partition $\mathbf{H}$ covers the whole domain Ω.

The solution to the problem (7), (8) is a pair of discrete functions $(\mathcal{F}_i, H_i)$ defined at the midpoint of each grid element e_i. For the sake of simplicity, below we consider the two - component vector function $\mathbf{x}(t) = \{x_1(t), x_2(t)\}$. Let H_{ik}, $k = 1, 2$ be a mesh step size on element e_i considered on the grid $\mathbf{H}_k$, $k = 1, 2$. A demonstrative way to solve the problem (7) is to consider it in the $(\ddot{x}_{i1}, \ddot{x}_{i2})$ plane (see Fig.1). The error vector $\mathbf{E}_i$ on the element e_i is represented in the $(\ddot{x}_{i1}, \ddot{x}_{i2})$ basis as (index i is omitted)

$$\mathbf{E} = (E_1, E_2) = (\alpha_1 \ddot{x}_1, \alpha_2 \ddot{x}_2), \tag{9}$$

where basis coefficients $\alpha_k = H_k^2$, $k = 1, 2$. It can be seen from the figure, that the condition $||\bar{\mathbf{E}}|| = ||\mathbf{E}||$ holds for any vector $\bar{\mathbf{E}} = (\bar{E}_1, \bar{E}_2) = (\bar{\alpha}_1 \ddot{x}_1, \bar{\alpha}_2 \ddot{x}_2)$ which endpoint belongs to the circumference C_E. Thus, the problem may be formulated as follows:
Find the angle ϕ, such that the initial vector $\mathbf{E}$ should be rotated through ϕ in order to obtain the new vector

$$\bar{\mathbf{E}} = (\bar{\alpha}_1 \ddot{x}_1, \bar{\alpha}_2 \ddot{x}_2), \tag{10}$$

where

$$\bar{\alpha}_1 = \bar{\alpha}_2 \equiv H^2. \tag{11}$$

The rotation transformation for the vector $\mathbf{E}$ yields

$$\begin{cases} \bar{E}_1 = E_1 \cos\phi - E_2 \sin\phi \\ \\ \bar{E}_2 = E_1 \sin\phi + E_2 \cos\phi. \end{cases} \tag{12}$$

After substitution (9), (10), and (11) into the system (12) the unknown values ϕ and H can be easily found as

$$\tan(\phi) = \frac{E_1\ddot{x}_2 - E_2\ddot{x}_1}{E_1\ddot{x}_1 + E_2\ddot{x}_2}, \quad H^2 = \frac{||\mathbf{E}||}{\sqrt{\ddot{x}_1^2 + \ddot{x}_2^2}}. \tag{13}$$

It follows immediately from (13) that the adaptation to the key function $\mathcal{F}$ defined on element e_i as

$$\ddot{\mathcal{F}}_i = \sqrt{\ddot{x}_{1i}^2 + \ddot{x}_{2i}^2} \tag{14}$$

ensures the error E_i=$||\mathbf{E}||$ on grid $\mathbf{H}$.

The obtained error indicator

$$E_i = \sqrt{\ddot{x}_{1i}^2 + \ddot{x}_{2i}^2} H_i^2 = \ddot{\mathcal{F}}_i H_i^2 \tag{15}$$

seems to be identical with that in (5). However, there is a considerable difference between the two estimates. Namely, $\mathcal{F}_i \neq ||\ddot{\mathbf{x}}_i||_{L^2}$, since the midpoints of element e_i have different coordinates on grids $\mathbf{H}_1$ and $\mathbf{H}_2$ where the derivatives $\ddot{x}_k(t_{i_k})$, $k = 1, 2$ are calculated. Is the choice of the function $\mathcal{F}$ better than the adaptation to the key function F? In the next section, the analysis of the node distribution (13) will be made to compare the results of the adaptation for the two functions.

3. The analysis of distribution H. The new grid $\mathbf{H}$ generated as a result of (13) can be considered on each grid element e_i as a weighted sum of functions $\mathbf{H}_1$ and $\mathbf{H}_2$

$$H_i^4 = \rho_1 H_{1i}^4 + \rho_2 H_{2i}^4, \tag{16}$$

where the weight coefficients $\rho_k = \dfrac{\ddot{x}_{ki}^2}{\ddot{x}_{1i}^2 + \ddot{x}_{2i}^2}$, $k = 1, 2$. Since the condition $\sum\limits_{k=1}^{2} \rho_k = 1$ holds for the weight coefficients in (16) on each element e_i, $i = 1, ..., N$, the value H_i is bounded by

$$\min\{H_{1i}, H_{2i}\} \leq H_i \leq \max\{H_{1i}, H_{2i}\}$$

on element e_i. Suppose that $\max\left\{H_1^{i1}, H_2^{i1}\right\} = H_1^{i1}, \forall i1 : 1 \leq i1 \leq N_1$, and $\max\left\{H_1^{i2}, H_2^{i2}\right\} = H_2^{i2}, \forall i2 : 1 \leq i2 \leq N_2$, where N_1+$N_2 = N$. Accumulating the contributions due to the elements which belong to the discrete sets $\mathcal{N}_1$ and $\mathcal{N}_2$, we obtain

$$\sum_{i1=1}^{N_1} H_2^{i1} + \sum_{i2=1}^{N_2} H_1^{i2} \leq \sum_{i}^{N} H_i \leq \sum_{i1=1}^{N_1} H_1^{i1} + \sum_{i2=1}^{N_2} H_2^{i2}.$$

Taking into account that both discrete functions $\mathbf{H}_1$ and $\mathbf{H}_2$ hold the condition (8), the following estimate appears

$$1 - \sum_{i2=1}^{N_2}(H_2^{i2} - H_1^{i2}) \le \sum_{i}^{N} H^i \le 1 + \sum_{i2=1}^{N_2}(H_2^{i2} - H_1^{i2}). \tag{17}$$

Since $H_2^{i2} > H_1^{i2}$ on any element $e_{i2} \subset \mathcal{N}_2$, the sum $\sum\limits_{i2=1}^{N_2}(H_2^{i2} - H_1^{i2}) > 0$, except for identical grids $H^{1i} \equiv H^{2i}, \forall i : 1 \le i \le N$. Hence, in the worst case we have $\sum\limits_{i}^{N} H_i < 1$, and the partition (13) does not cover the whole interval $[0, 1]$, so that the distribution $\mathbf{H}$ should be scaled by the factor $\alpha = (\sum\limits_{i=1}^{N} H_i)^{-1}$. The new node distribution $\tilde{\mathbf{H}}$ is given by

$$\tilde{H}_i = \alpha H_i, \quad 1 \le i \le N. \tag{18}$$

For $\alpha > 1$ the similarity transformation (18) stretches the grid that results in increasing the error.

Let us compare the error components for functions $\mathcal{F}$ and F for the solution (6). It follows from (16) that the error $\tilde{E}_{ki}$ for each solution component reads

$$\tilde{E}_{ki}^2 = \ddot{x}_{ki}^2 \tilde{H}_i^4 = \alpha \ddot{x}_{ki}^2 (\rho_{1i} H_{1i}^4 + \rho_{2i} H_{2i}^4) \tag{19}$$

on the grid $\tilde{\mathbf{H}}$ generated as a result of the adaptation to $\mathcal{F}$. Substituting the weight coefficients ρ_k into (19), we obtain

$$\tilde{E}_{ki}^2 = \alpha \tilde{\rho}_{ki} (E_{1i}^2 + E_{2i}^2), \quad k = 1, 2, \tag{20}$$

where E_{ki}, $k = 1, 2$, is the error on the optimal grid $\mathbf{H}_k$, $k = 1, 2$, and $\tilde{\rho}_{ki} = \dfrac{\ddot{x}_{ki}^2}{\ddot{x}_{1i}^2 + \ddot{x}_{2i}^2}$. Note that $\tilde{\rho}_{ki} \ne \rho_{ki}$, since the value of the derivative $\ddot{x}_{ki}^2$ is now defined on the grid $\tilde{\mathbf{H}}$. The error norm is now given by

$$||\tilde{\mathbf{E}}_i||^2 = \tilde{E}_i^2 = \tilde{E}_{1i}^2 + \tilde{E}_{2i}^2 = \beta (E_{1i}^2 + E_{2i}^2) = \beta_i ||\mathbf{E}_i||^2, \; 1 \le i \le N, \tag{21}$$

where the discrete function β_i is defined on element e_i as follows

$$\beta = \alpha(\tilde{\rho}_{1i} + \tilde{\rho}_{2i}).$$

For the function $F = const$, a uniform grid $\mathbf{H}_u$ is generated. The error components on the grid $\mathbf{H}_u$ are $(E_{ki}^u)^2 = \ddot{x}_{ki}^2 H_{ui}^4$, $k = 1, 2$. Evidently,

$$||\mathbf{E}_i^u||^2 = {E_{1i}^u}^2 + {E_{2i}^u}^2 > {E_{1i}}^2 + {E_{2i}}^2 = ||\mathbf{E}_i||, \tag{22}$$

since for each solution component $x_k(t)$ the error E^u_{ki} on the uniform grid $\mathbf{H}_u$ is assumed to be greater than the error E_{ki} on the optimal grid $\mathbf{H}_k$. Thus, according to (21), the relation between the errors $||\mathbf{E}^u_i||$ and $||\tilde{\mathbf{E}}_i||$ depends on the value β_i. For those problems where $||\beta||_{L^\infty} \approx 1$, the adaptation to function F is inefficient, as the estimate $||\mathbf{E^u}|| > ||\tilde{\mathbf{E}}||$ holds.

The following example considered earlier in [13] illustrates the solution of the full adaptation problem. Let the second derivatives of the solution $\mathbf{x}(t)$ be given by $\ddot{x}_1(t) = \dfrac{1}{1+t}$, and $\ddot{x}_2(t) = \dfrac{\sqrt{2t+t^2}}{1+t}$. The derivatives $x^{(3)}_1(t) = \dfrac{-1}{(1+t)^2} < 0$, and $x^{(3)}_2(t) = \dfrac{1}{(1+t)^{3/2}} > 0$ are monotone functions over the domain $[0,1]$, and $\max\limits_{t\in[0,1]} \ddot{x}(t) = \ddot{x}(t_0 = 0) = 1$, $\max\limits_{t\in[0,1]} \ddot{y}(t) = \ddot{y}(t_0 = 1) = \sqrt{3}/2$ (see Fig.2a). Hence, grids $\mathbf{H}_1$ and $\mathbf{H}_2$, which are optimal for the problem, will be refined about the left and the right endpoints of the interval $[0,1]$, respectively.

Let us start with uniform initial grids $\mathbf{H}^0_1 : H^0_{1i} = 1/N_0 \equiv H_0$ and $\mathbf{H}^0_2 : H^0_{2i} = 1/N_0 \equiv H_0$, where N_0 is the number of grid elements. After a few first steps of the refinement procedure (2), (3) the number of grid elements is increased to N, and the grids $\mathbf{H}_1$ and $\mathbf{H}_2$ are transformed as follows

$$\mathbf{H}_1 = \begin{cases} H_0/2, & i < N_1, \\ \\ H_0, & N_1 < i < N, \end{cases} \qquad \mathbf{H}_2 = \begin{cases} H_0, & i < N_2, \\ \\ H_0/2, & N_2 < i < N. \end{cases} \tag{23}$$

The distribution $\tilde{\mathbf{H}}$ generated with (13), (18) along with the uniform grid $\mathbf{H}_u$ is shown in Fig.2b. It can be seen from the figure that the grid $\tilde{\mathbf{H}}$ is in better agreement with our intuitive idea of the adaptation that the grid $\mathbf{H}_u$ appearing as a result of the adaptation to the function F. The distribution $\tilde{\mathbf{H}}$ keeps the grid refined in the regions where at least one component of the solution needs a fine mesh. Let us note that, unlike the function $\ddot{F}(t) \equiv 1$, the function $\ddot{\mathcal{F}}$ is not constant over the domain. For instance, $\ddot{\mathcal{F}}_i$ is defined as

$$\ddot{\mathcal{F}}_i = \sqrt{\frac{1}{1+t_i} + \frac{\sqrt{4t_i + 4t_i^2}}{1+2t_i}}$$

on any element e_i, $1 \le i \le N1$, where t_i is the midpoint of the element e_i on grid $\mathbf{H}_1$. However, the knowledge of the function β is required to make a conclusion about the advantages of the distribution $\tilde{\mathbf{H}}$ in solution of problem (1).

4. The minimization of the interpolation error. The analysis of the distribution $\mathbf{H}$ does not give us any practical algorithm suitable for the

purposes of the adaptation. Evidently, the grid generation for each solution component is not consistent with the basic idea of the adaptation which is to increase the overall efficiency of computations. Nevertheless, the study of the full adaptation problem I is useful, as it allows us to conclude that solving the problem II instead of the problem I is reasonable, provided a correct adaptation strategy is used.

On the surface, this conclusion is in contradiction with the results obtained above, as the analysis of the full adaptation problem demonstrates that the consideration of a scalar key function instead of a separate treatment of each solution component may results in the inefficient adaptation. However, the generation of a uniform grid for the solution (6) is not a consequence of the wrong choice of the error estimate based on the key function (4). In our opinion, it happens due to the solution of the basic adaptation problem:

$$\textit{The given number } N \textit{ of grid elements, minimize the error.} \qquad (24)$$

is displaced with solving another, "auxiliary" problem.

The consideration of the auxiliary adaptation problem is a common approach to generate adaptive meshes. Generally, the auxiliary task imitates the problem (24). It may be formulated as error reduction to a given value at each step of the adaptation procedure, or error equidistribution, etc. Solving the auxiliary problem is attractive for the practical applications, as usually such an algorithm meets the requirements to the design of industrial codes [3]. In very many cases the auxiliary task does help us to achieve our main goal (24). However, as we see from the above analysis, the auxiliary problem does not always guarantee the expected efficiency. If solving the auxiliary problem does not bring the proper results, it is reasonable to try to find the solution of the basic problem (24) in order to make a conclusion about the efficiency of the auxiliary approach.

Let us note that there are no difficulties with the choice of a scalar key function for the problem (24), as we now seek a function which takes into account the properties of the solution rather than a function which satisfies the artificial needs of the auxiliary problem. Obviously, the norm of the vector solution (with weight coefficients, if one is interested in some particular features of the solution) is an appropriate option in this case.

The problem (24) does not include any explicit suggestions about the form of the error estimate. Its formulation reads that a quantity loosely called "the error" is minimized on the optimal grid, the true error, i.e. the difference between the exact and approximate solution, being bounded by this quantity. The error estimates appearing in the adaptation problem (24)

may be derived from the interpolation theory [5, 14], or they may be developed as a result of a posteriori error analysis in weighted residual methods [17]. The considered error estimators may have a very particular form, as they may be required to capture both the geometry and solution behavior, [2]. However, the adaptation strategy used to solve the problem (24) is often viewed as being independent of the choice of the error estimate. For instance, the standard refinement procedure (3) (with some slight variations) is implemented by a number of authors (*e.g.* see [4, 9, 17]), in spite of the fact that the essentially different error estimators are exploited in their work. In our opinion, one should adjust the solution of the problem (24) to the form of the error estimate. Below we demonstrate how the definition of the error may be taken into account for the choice of the adaptation technique in the simplest case that the interpolation error estimate based on the second derivative is minimized.

Interpolation error estimates are derived from a Taylor series expansion of function $f(t)$ about point t_0

$$f(t_0+dt) = f(t_0) + \frac{df}{dt}dt + \frac{1}{2}\frac{d^2f}{dt^2}dt^2 + ... + \frac{1}{k!}\frac{d^{(k)}f}{dt^k}dt^k + R_{k+1},$$

$$R_{k+1} = \frac{1}{(k+1)!}\frac{d^{(k+1)}f(\xi)}{dt^{k+1}}dt^{k+1},\ \xi = t_0 + \theta dt,\ 0<\theta<1.$$

If the function f is represented as a polynomial of degree k, the remained term R_{k+1} is precisely the error. A computable error estimate is then based on bounding the unknown derivative $f^{(k+1)}(\xi)$ by a computable quantity, e.g. the value $f^{(k+1)}(t_0)$.

Let us now write the Taylor series expansion as a sum of the differentials

$$\begin{gathered} f(t_0+dt) = f(t_0) + df + \frac{1}{2}d^2f + ... + \frac{1}{k!}d^kf + R_{k+1}, \\ R_{k+1} = \frac{1}{(k+1)!}d^{k+1}f(\xi),\ \xi = t_0 + \theta dt,\ 0<\theta<1. \end{gathered} \tag{25}$$

It follows from (25), that the interpolation error estimate can be treated as the $(k+1)$-th differential of function $f(t)$. Below we consider the case of a piecewise linear approximation to the solution. The approximate solution is represented as $f(t_i+H_i) \approx f(t_i) + E_i^1$ on element e_i, where

$$E_i^1 = \dot{f}_i H_i \approx (df)_i, \tag{26}$$

and the leading error term is estimated as

$$E_i^2 = \ddot{f}_i H_i^2 \approx (d^2f)_i. \tag{27}$$

Thus, the problem of error minimization in case of the error estimate (27) is stated as

The given number N of grid elements, generate a grid where $(d^2 f)_i = 0$ on any grid element e_i, $1 \le i \le N$.

The solution of this problem can be obtained from the identity

$$d^{k+1} f = d(d^k f). \tag{28}$$

Since $d(C) = 0$, for $C = const$, we suggest that equidistribution of the discrete function $(d^k f)_i$ over the grid results in the minimization of the function $(d^{k+1} f)_i$, $1 \le i \le N$. Hence, for $k = 1$ our goal is to generate a grid, where $(df)_i = const$ on any grid element e_i, $1 \le i \le N$. In other words, we want to obtain the equidistributed error (26) in order to minimize the error (27).

Let us note that, according to (28), any adaptation procedure (e.g. node redistribution or grid subdivision) equidistributing the error of the k-th order reduces the error of the $(k+1)$-th order at each solution – adaptation iteration. The error indicator in the AHA-2D grid generator [11] gives us an illustrative example. The value E_e of the error indicator on edge e reads

$$E_e = \left(w_1 \left| \frac{\partial u}{\partial \mathbf{e}} \right| + w_2 \left| \frac{\partial^2 u}{\partial \mathbf{e^2}} \right|^{1/2} \right) = w_1 |(\nabla u, \mathbf{e})| + w_2 |(\mathbf{e}^T H \mathbf{e})|^{1/2}, \tag{29}$$

where $\mathbf{e} = \mathbf{r}_i - \mathbf{r}_j$, H, w_1 and w_2 are edge vector, Hessian and weight coefficients, respectively. Equidistribution of the gradient term based on edge and macrocell subdivision results in the minimization of the second order error, while equidistribution of the Hessian term reduces the high order error components at each step of the refinement/derefinement procedure implemented in the AHA-2D code. A similar edge based error indicator is exploited in BIT-2D grid generator [16]. A particular feature of the BIT-2D technology is that the grid edges are aligned to the contour levels of the solution to efficiently minimize the error.

Another example has been reported in [12], where a linear finite volume scheme has been used for solving 1D convection-diffusion equation on adaptive grids. The comparison between grid subdivision and error equidistribution has been made in [12] for the interpolation error estimate based on the second derivative. Evidently, equidistribution of the error (27) minimizes the value of $E_i^3 = x^{(3)}(t_i) H_i^3$. Comparing the convergence results for both adaptation algorithms, only a slight difference in the convergence rate has been observed. Based on the analysis carried out in [13], one may suggest that in the considered case grid subdivision performs equidistribution of the

function $\mathcal{E}(t, N)$, which represents the error (27), rather than error reduction at each adaptation step. If so, exploiting the first differential (26) as an error indicator in the grid subdivision procedure would yield the better convergence rate, provided the error (27) is dominant in the approximate solution.

5. Equidistribution of the interpolation error for a monotone function. In this section we address the issue of error equidistribution in case that the error (26) is considered. A straightforward way to achieve $E_i^1 = const$ over the grid is to derive node distribution $\mathbf{H}$ from the definition of E_i^1. Under the statement of the problem we obtain

$$H_i = \frac{\epsilon}{\dot{f}_i}, \tag{30}$$

where the constant ϵ is defined to satisfy the condition (8).

Although the algorithm (30) performs easily for the given number N of grid elements, the adaptation procedure based on node relocation technique (30) is inefficient. Actually, suppose that the number of grid nodes is increased from N to N_1 at the next step of the adaptation. To redistribute the nodes over the new grid the values of the derivative $\dot{f}_i$ are required at new $N_1 - N$ points. In order to choose these points one has to solve the equation on the uniform grid with N_1 nodes. Involving the calculations over the uniform grid into the adaptation procedure renders the algorithm (30) useless.

The effective approach to solve the problem of error minimization has been developed in [6]. For the algorithm suggested in [6], the idea of grading functions (described also in [7]) is essential. Following [7], we briefly recall the basic concepts. Let a uniform grid be generated in a reference domain $0 < \xi < 1$. The grading function $x(\xi)$ is a mapping between the domain ξ and the physical domain $a < t < b$ that transforms grid $\xi_i = i/N$ into the desired grid $\{t_i\}$. It has been proven in [6] that the grading function in the form

$$\xi(t) = \frac{\int_a^t (\ddot{f})^{2/5} dt}{\int_a^b (\ddot{f})^{2/5} dt} \tag{31}$$

minimizes the L^2 norm of the error $e = f - f_I$ with respect to variations in the mesh coordinates, where f_I is the interpolant of the function f. The grid mapping (31) can be also applied to optimize the mesh in solution of boundary problems. The designed technique, however, is laborious, since the

nonlinear equation (31) must be solved at any mesh point in order to obtain the coordinate t_i. Also, the class of considered problems is restricted, as the solution $f(t)$ is required to be a monotone function.

We suggest another approach to minimize the error (27) by means of equidistributing the error (26) over the grid. The developed algorithm takes an advantage of the relationship between the first differential of the function f and its increment:

$$\Delta f \equiv f(t+dt) - f(t) = df + \circ(dt). \tag{32}$$

The identity (32) allows us to equidistribute the value of Δf instead of equidistributing the error (26).

First, we consider a function $f(t)$ monotone over the domain $0 \le t \le 1$. The idea now is to generate a uniform grid $\{f_i\}$ in the reference domain f_r which is identified with the range of values of the monotone function $f(t)$ and then to map the partition $\{f_i\}$ into the physical domain $0 < t < 1$. The algorithm for the increasing function $f(t)$ may be written as follows

Algorithm I.

1. Generation of a uniform grid in the reference domain:

   1.1. Calculate the difference $f(1) - f(0)$ between the values of the function at the segment endpoints.

   1.2. For the given number N of grid elements, calculate the value $\Delta f = \dfrac{f(1) - f(0)}{N}$.

   1.3. Equidistribute the nodes over the reference domain $[f(0), f(1)]$ as $f_i = f(0) + i * \Delta f$, $1 \le i \le N$.

2. Mapping the uniform grid to a mesh in the physical domain:

   For $1 \le i \le N$ Do:

   2.1. Find the point $P_i = (t_i, f_i)$ of intersection between the solution curve $f(t)$ and the line $f_i = const$ as the solution to the equation

$$f(t_i) = f_i. \tag{33}$$

   2.2. Assign the abscissa t_i of the point P_i as a new grid node.

   EndDo

The simplest example of the mapping function $t(f)$ is given by a linear function $f_l(t) = at+b$. Obviously, a uniform grid is generated for the function

$f_l(t)$ over the domain $[0,1]$ as a result of solving the equation

$$at_i + b = \frac{ai}{N} + b.$$

For a decreasing function $f(t)$, a slight modification of the algorithm **I** should be made to take into account node ordering in the reference domain. The function $f_c(t) = e^{-t}$ is frequently used for modeling the behavior of complicated physical and biological systems, [15]. For the function $f_c(t)$ the value Δf is calculated as $\Delta f = \dfrac{(1-e^{-1})i}{N}$. A fine mesh $t_{N-i} = \ln\left(\dfrac{N}{(1-e^{-1})i+N}\right)$ is generated in the region of the steep gradient about the point $t = 0$. Figure 3a illustrates the algorithm of error minimization for the function $f_c(t)$. The distribution $H_i = t_{i+1} - t_i$ over the domain $[0,1]$ is shown in Fig.3b.

The boundary problem

$$\dot{f}(t) = \mu\ddot{f}(t), \quad f(0) = 0,\ f(1) = 1 \tag{34}$$

is often considered to be a good model for study of convection - dominated problems (*e.g.*, see [18]). The exact solution to the problem is given by

$$f(t) = \sinh(t/\mu)/\sinh(1/\mu).$$

The function $f(t)$ is shown in Fig.4a for the value $\mu = 0.5 \cdot 10^{-2}$. The boundary layer of the width $|r_b|$ presents near the point $t = 1.0$, while the solution varies slowly in the far field region r_f.

We use the linear finite element discretization of the problem (34) to find the approximate solution $\bar{f}$ on adaptive grids. Fig.4b illustrates the algorithm of error minimization applied to the problem (34). For the distribution **H** shown in Fig.4b, the coordinates t_i of grid nodes are computed directly from

$$t_i = \mu\ln\left(\frac{N + (e^{1/\mu} - 1)i}{N}\right).$$

It can be seen from the figure that equidistribution of the error *df* results in a very fine mesh in the boundary layer region r_b. Meanwhile, the whole far field region r_f corresponds to a single grid cell C_{r_f}. For instance, the size of the far field cell $C_{r_f} = 0.98$ on the grid with the number of elements $N = 50$.

For better understanding of the results of the adaptation let us consider the solution error $e_i = |f_i - \bar{f}_i|$ computed elementwise on a sequence of grids

with the equidistributed value of df. The plots of the error function are shown in Fig.4c. The maximum of the solution error keeps in the cell C_{r_f}, whereas the boundary layer is resolved well. The obtained error distribution over the domain demonstrates the difference between the interpolation error and the solution error. Although the condition $d^2 f = 0$ holds for the interpolation error used as an error estimate, the solution error is reduced slowly in the far field region. The reason is that the error df is the leading term in the error estimate in the far field, where the function $f(t) \approx const$ (see Fig.4a). The solution error on the element e_{r_f} can be estimated as

$$E_{r_f} \approx df = const = 1/N,$$

where N is the number of grid elements. This estimate is confirmed by the results in Fig.4c. For instance, the value $df = 0.571 \cdot 10^{-2}$ is in a good comparison with the solution error $E_{r_f} = 0.509 \cdot 10^{-2}$ on the grid with $N = 175$. Hence, in the far field region the adaptation procedure should reduce the error df rather than equidistribute it.

Keeping in the mind the behavior of the error in the far field, in our further consideration we compute the norm of the error only in the region r_b, since our main goal is to study the resolution of the boundary layer. We compare the results of the adaptation on grids refined with the standard procedure (2), (3) with those obtained with the algorithm **I**. For the standard refinement procedure the width $|r_b|$ of the boundary layer is estimated as $|r_b| \sim O(\sqrt{\mu})$. We consider $|r_b| = 1 - |C_{r_f}|$ on redistributed grids, where $|C_{r_f}|$ is the size of the far field cell.

In practical calculations, the exact solution $f(t)$ is not available so that we need to solve the equation

$$f_I(t_i) = f_i, \tag{35}$$

where f_I is an interpolant of the function f. To solve the equation (35), the piecewise linear reconstruction of the solution is used, the nodal values being taken from the exact solution. In the standard refinement (3) the exact second derivatives of the solution are used in the error estimate (2).

the convergence history in the boundary layer region is presented in Fig.4d. The solution adaptive grids are obtained with the grid subdivision and error equidistribution algorithms. The solution error measured in L^2-norm is shown in the semilogarithmic scale. The convergence plots demonstrate the better convergence of the error minimization algorithm on coarse grids with $N < 30$. On finer grids, both adaptation procedures resolve the boundary layer with the same accuracy.

The nodal distribution over the interval $(0,1)$ on refined (G^S) and redistributed (G^R) grids is shown in Table 1 for the number of grid elements $N = 32$. Grids G^R_{exact} and G^R_{approx} are obtained as a result of the solution of the equation (33), and (35), respectively. To generate the grid G^S_{exact} the derivatives are taken from the exact solution, while on the grid G^S_{approx} the error estimate is calculated from the approximate solution.

Table 1.

Nodal distribution for problem (34) on adaptive grids: G^S - standard refinement, G^R - node redistribution (N=32).

Interval	Grid G^S_{exact}	Grid G^S_{approx}	Grid G^R_{exact}	Grid G^R_{approx}
$(0.00, 0.80)$	4	4	0	0
$(0.80, 0.90)$	1	2	0	0
$(0.90, 0.92)$	0	0	0	0
$(0.92, 0.94)$	1	1	0	0
$(0.94, 0.96)$	1	1	0	0
$(0.96, 0.98)$	4	4	0	15
$(0.98, 0.99)$	5	5	4	8
$(0.99, 0.995)$	6	6	7	4
$(0.995, 1.00)$	9	8	20	4

It can be seen from the table that redistribution of the nodes depends strongly on the equation solved to obtain a new grid node, while the refinement procedure is not sensitive to using the approximate derivatives instead of the exact ones. Another advantage of the procedure (2), (3) is that the refined grid contains the larger number of the nodes in the far field, therefore, one may expect the better convergence rate in the far field region. Nevertheless, our hope is that the algorithm of error minimization can be modified to reduce the error in the far field adequately.

6. Equidistribution of the interpolation error for a nonmonotone function. In this section we generalize the algorithm **I** to permit the treatment of arbitrary continuous functions. Dealing with the functions which are not monotone over the domain, the value Δf should be defined more thoroughly. We will use the concept of the total variation (*e.g.*, see [8], [10]) to calculate the increment Δf of the function. Let $0 = t_0 < t_1 .... < t_M = T$ form a partition of the closed interval $[0, T]$. We denote the class of functions of variation $V = \sum_{m=1}^{M} |f(t_m) - f(t_{m-1})|$ over the interval $[0, T]$ as $\mathcal{V}(T, V)$. The

norm of $\mathcal{V}(T,V)$ is defined as

$$\mathcal{V} \equiv ||f||_{\mathcal{V}(T,V)} = \sup_{\substack{M \geq 1 \\ 0 = x_0 > ... > x_M = T}} \sum_{m=1}^{M} |f(t_m) - f(t_{m-1})| \tag{36}$$

The norm $\mathcal{V}$ is called the total variation of function $f(t)$ on $[0,T]$. If $\mathcal{V} < \infty$, the function $f(t)$ is a function of bounded variation.

For continuous functions of bounded variation, the value $\mathcal{V}$ is calculated as

$$\mathcal{V} = \sum_{m=1}^{M} |f(t_m) - f(t_{m-1})|, \tag{37}$$

where t_m, $0 \leq m \leq M$ are the points of extremum. Evidently, monotone functions considered in the previous section correspond to $M = 1$ in (37),

$$\mathcal{V} = |f(T) - f(0)|.$$

In our further consideration, we will always use the definition (37). Let us note, however, that the definition (36) is more general, as it enables to extend the analysis to the function classes that may have discontinuities.

The error minimization algorithm for a continuous function of bounded variation defined on the closed interval $[0,1]$ may be written as follows
Algorithm II.

1. Generation of a uniform grid in the reference domain:

   1.1. Find the points of extremum $P_m^{ext} = (t_m^{ext}, f_m^{ext})$, $1 \leq m \leq M$ *for the function* $f(t)$ *in the domain* $[0,1]$. *Find* $f_{min} = \min\limits_{0 \leq t \leq 1} f(t)$.

   1.2. For the given number M *of extremum points, calculate the total variation (37) of the function* $f(t)$.

   1.3. Define the reference domain f_r *as the closed interval* $[f_{min}, f_{min} + \mathcal{V}]$ *. For the given number* N *of grid elements, calculate the value* $\Delta f = \dfrac{\mathcal{V}}{N}$.

   1.4. Equidistribute the nodes over the domain f_r *as* $f_i = f_{min} + i * \Delta f, 1 \leq i \leq N$.

2. Mapping the uniform grid to a mesh in the physical domain:

   For $1 \leq m < M$ Do:

   2.1. Define $N_1 : t_{N_1} = t_m^{ext}$ *and* $N_2 : t_{N_2} = t_{m+1}^{ext}$.

For $1 \le i \le N$ Do:

If $f_i \in [f_m^{ext}, f_{m+1}^{ext}]$ Then

2.2. Find the point $P_i = (t_i, f_i)$ of intersection between the solution curve $f(t) : t \in [t_{N_1}, t_{N_2}]$ and the line $f_i = const$.

2.3. Assign the abscissa t_i of the point P_i as a new grid node.

EndDo

EndDo

The main advantage of involving the calculation of the total variation into the algorithm **II** is that it allows to control the number N of grid elements in the physical domain, the value Δf of the error being generated as a mesh step size in the reference domain. For the error control, the number N should be defined in the reference domain as the integer part of

$$\eta = \frac{\max\limits_{0 \le t \le 1} f(t) - \min\limits_{0 \le t \le 1} f(t)}{\Delta f},$$

that results in the increased number of the grid elements in the physical domain. This observation enables us to comment the issue of initial grids. Actually, the initial partition of the domain $[f_{min}, f_{min} + \mathcal{V}]$ must contain at least one interior point. For rapidly oscillating functions this requirement leads to generation of the adequate initial grid with the large number of grid elements.

Now we consider the vector solution $x(t)$. After a key function is chosen, the algorithm II can be applied to generate a grid for a system of equations. Let us define the key function $f(t)$ as the L^1-norm of the vector $\mathbf{x}(t)$

$$f(t) = \frac{1}{K} \sum_{k=1}^{K} |x_k(t)| \tag{38}$$

Suppose that the algorithm **I** generates optimal grid $\mathbf{H}_k$ for each scalar function $x(t)$ considered separately. The error on the grid $\mathbf{H}_k$ is calculated as $dx_k = \dfrac{\mathcal{V}_k}{N}$, where $\mathcal{V}_k$ is the total variation of the solution component $x_k(t)$. The norm of the error vector $\mathbf{E}$ on the element e_i is

$$||\mathbf{E}_i|| \equiv E_i = \frac{1}{K} \sum_{k=1}^{K} dx_k = \frac{1}{KN} \sum_{k=1}^{K} \mathcal{V}_k = \frac{\bar{\mathcal{V}}}{N},$$

where the mean total variation $\bar{\mathcal{V}} = \frac{1}{K} \sum\limits_{k=1}^{K} \mathcal{V}_k$.

Let us now calculate the error on the grid **H** obtained as a result of the adaptation to the key function $f(t)$. The value df on the grid **H** generated with the algorithm **II** is given by

$$df = \frac{\mathcal{V}_f}{N},$$

where $\mathcal{V}_f$ is the total variation of the function $f(t)$. It is not difficult to see, that in general case $\mathcal{V}_f \neq \bar{\mathcal{V}}$, therefore, $df_i \neq E_i$.

The value $\mathcal{V}_f > \bar{\mathcal{V}}$ indicates that we are not able to achieve the desired value of the error dx_k for the solution component $x_k(t)$ on grid **H**, the number N of the grid elements being sufficient to resolve the function $x_k(t)$ over a separate grid. In this case the price one should pay for the reduction of the vector error estimate to the scalar one is that the number of grid elements should be increased to resolve each scalar solution component with the desired accuracy dx_k on the grid **H**.

Conclusions.

- The key function problem arising when interpolation error estimates are considered for vector functions can be solved by means of a proper choice of the adaptation strategy. Generally, the adaptation procedure should depend on what error estimate is used.

- For the interpolation error estimates, equidistribution of the k-th order error over the grid results in minimization of the $(k+1)$-th order error. For discretization methods of the k-th order, the value of the k-th derivative is available from the solution. Hence, it is possible to compute the discrete function $f(t_i)$ similar to the k-th differential of the solution. The algorithm of error minimization developed in the work can be then applied to the function $f(t)$ in order to minimize the interpolation error estimate.

- The example of the convection-dominated problem considered in the work demonstrates that a combined error estimate may be required to resolve the solution in various domains. In case of a piecewise linear approximation to the solution it is reasonable to use the error df in the far field region, while the error estimate d^2f is needed to obtain the adequate resolution of the boundary layer. From a practical point of view, the issue of combined error estimates requires a further discussion in terms of both the accuracy and overall efficiency of the code.

- The results obtained in the work show that an additional error may appear in the interpolation error indicator due to reduction of the vector

error estimate to the scalar one. The impact of this error is that a finer grid may be required to achieve the desired accuracy for the vector solution on a single grid in comparison with the case that each solution component is treated on a separate grid.

References

[1] G.A.Ashford, K.G.Powell. *Adaptive Unstructured Triangular Mesh Generation and Flow Solvers for the Navier-Stokes Equations at High Reinolds Number*, AIAA Paper 95-1724, 1995.

[2] M.Berzins. *Mesh Quality - Geometry, Error Estimates or Both*, Engineering with Computers, vol. 15, pp. 236-247. 1999.

[3] M.B.Bieterman, J.E.Bussoletti, C.L.Hilmes, F.T.Johnson, R.G.Melvin, and D.P.Young. *An Adaptive Grid Method for Analysis of 3D Aircraft Configurations*, CMAME, Vol.101, pp. 225-249, 1992.

[4] L.Borges, R.Feijóo, N.Zouain. *A Directional Error Estimator for Adaptive Limit Analysis*, Mechanics Research Communications, Vol.26, No.5, pp.555-563, 1999.

[5] G.S.Buscaglia, E.Dari. *Anisotropic Mesh Optimization and its Application in Adaptivity*, IJNME, Vol.40, pp.4119-4136, 1997.

[6] G.F.Carey, H.Dihn. *Grading Functions and Mesh Redistribution*, SIAM J. Numer.Anal. Vol. 50, No.5, pp. 1028 -1040, 1985.

[7] G.F.Carey. *Computational Grids: Generation, Adaptation, and Solution Strategies*, Taylor & Francis, Washington, 1997.

[8] J.Dieudonné. Foundations of Modern Analysis. Academic Press, New York, 1960.

[9] F.K.Hebeker, R.Rannacher. *An Adaptive Finite Element Method for Unsteady Convection - Dominated Flows with Stiff Source Terms*, SIAM J. Sci.Comput., Vol.21, No.3, pp.799-818.

[10] A.Mukherjea, K.Pothoven. *Real and Functional Analysis. Part A: Real Analysis*, Plenum Press, New York, 1984.

[11] A.A.Martynov, S.Yu.Medvedev. *Anisotropic Grid Refinement Technique and Application to Model Spreading Wake Problem*, GGNS technical report, BOIM, Moscow, 1999.

[12] R.G.Melvin, F.T.Johnson. Personal Communication, The Boeing Company, M/S 67-LF, Seattle, WA, January 2001.

[13] N.B.Petrovskaya. *Error Indicators and Analysis of Refinement Technology for Vector Functions*, Final technical report for contract 104W, Boeing Operations International, Inc., Moscow, 2000.

[14] J. Peiró. *A Finite Element Procedure for the Solution of the Euler Equations on the Unstructured Meshes*, Ph.D.thesis, Department of Civil Engineering, University of Swansea, UK, 1989.

[15] C.E.Shannon, W.Weaver. *The Mathematical Theory of Communication*, The University of Illinois Press, Urbana, 1949.

[16] A.M.Sorokin, N.A.Vladimirova. *The Progress in Constrained Directions Refinement Applied to Navier-Stokes Solutions*, in TsAGI technical report for contract 101AT, Boeing Operations International, Inc., Moscow, June 2000.

[17] A.Süli. *A Posteriori Error Analysis and Adaptivity for Finite Element Approximations of Hyperbolic Problems*, Report No. 97/21, Oxford University Computing Laboratory, Numeric Analysis Group, 1997.

[18] V.Venkatakrishnan, R.G.Melvin, F.T.Johnson, L.B.Wigton, *Adaptation by Grid Refinement in One Dimension*, GGNS technical report, The Boeing Company, M/S 67-LF, Seattle, WA, 1999.

Figure 1: Representation of the full adaptation problem in the $(\ddot{x}_1, \ddot{x}_2)$-plane.

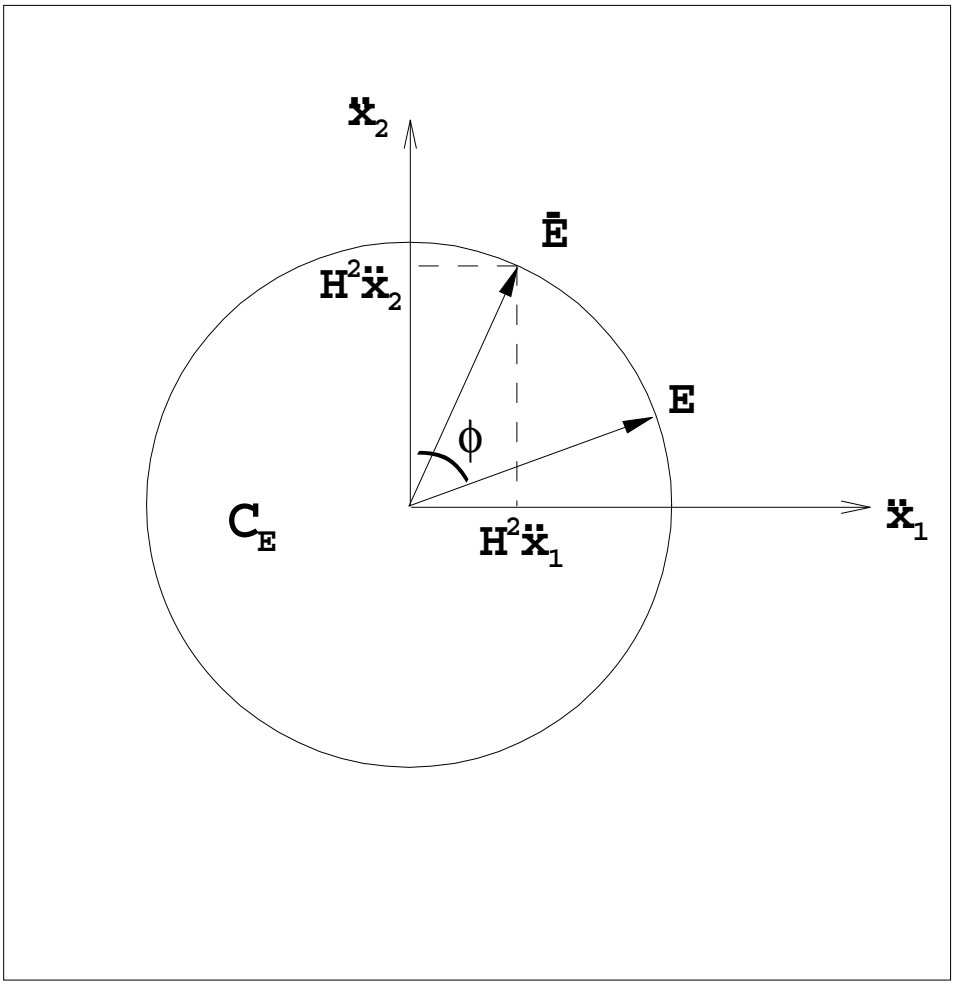

Figure 2: Solution of the full adaptation problem: a) The second derivatives $\ddot{x}_1(t)$ and $\ddot{x}_2(t)$ used in the error estimates on optimal grids. b). The mesh step size distribution on the optimal and uniform grids.

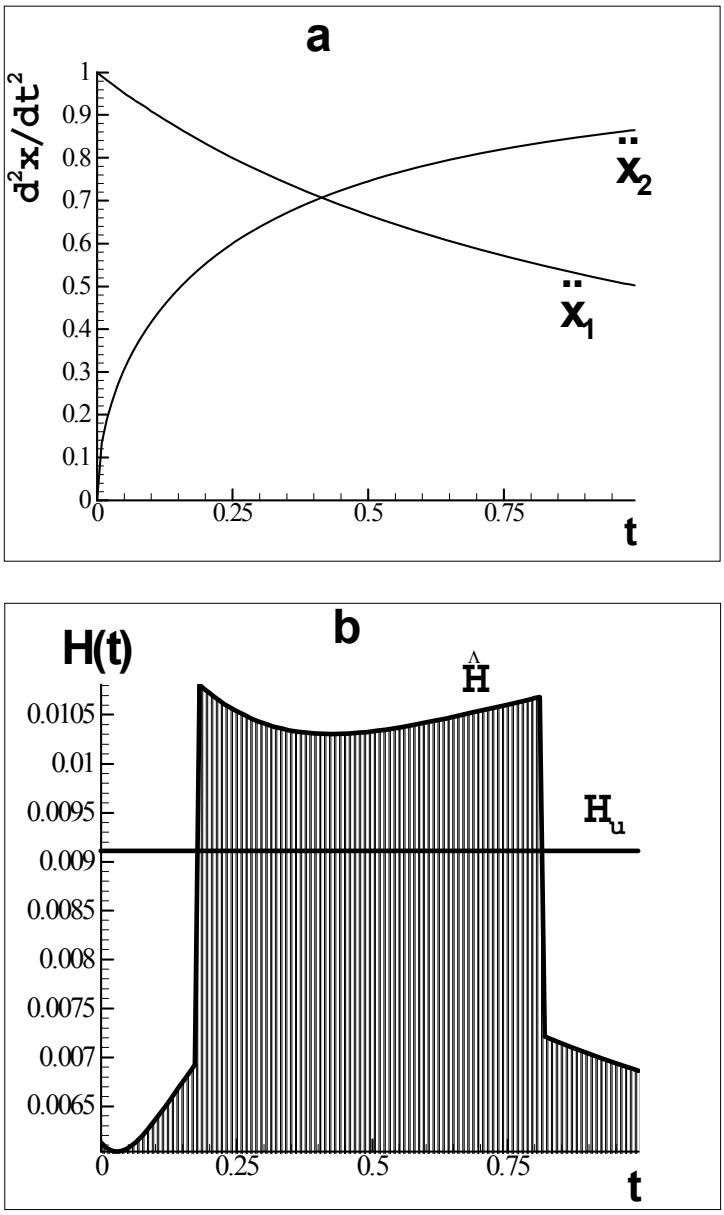

Figure 3: Adaptation to the function e^{-t}: a). The algorithm of error minimization. b). The mesh step size distribution.

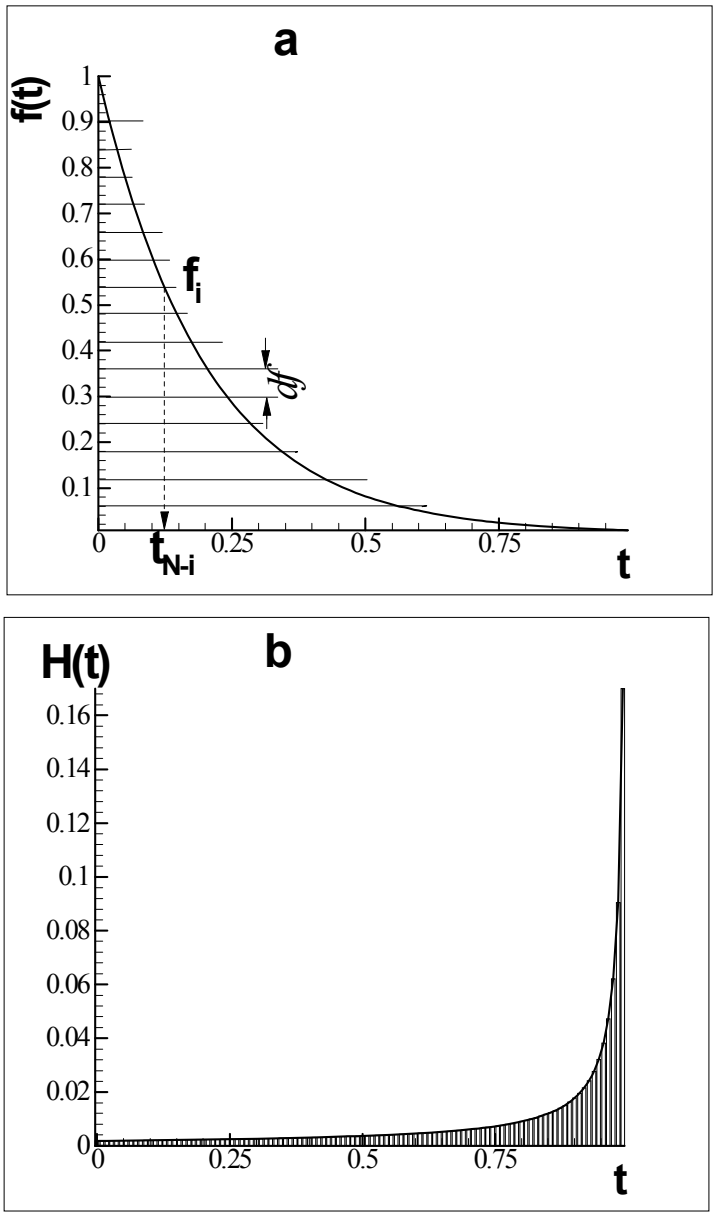

Figure 4: Error minimization in solution of the boundary problem: a). Solution to the problem (34). b). The grid **H** where the error $d^2 f$ is minimized. (I. Close-up of the distribution **H** in the boundary layer region.) c). The distribution of the error on a sequence of the adaptive grids. d). The convergence history on adaptive grids.