# ADAPTIVE APPROACH IN NONLINEAR CURVE DESIGN PROBLEM

Simo Virtanen

Rakenteiden Mekaniikka, Vol. 30
Nro 1, 1997, s. 14-24

## ABSTRACT

In recent years considerable interest has been shown in the development of feedback procedures (adaptive strategies) by which a required accuracy of the finite element solution can be most economically reached. There are many studies where is tried to find good error indicators to show where to refine. But in nonlinear problems the question when to refine is as important as the question where.
In this work p-hierarchical elements are used in the nonlinear curve design problem and different strategies how to refine during the iteration process is compared. First method is to use an iteration stopping criteria with a variable tolerance. At the beginning it is quite rude and when the calculation proceeds the criteria is tightened causing more iteration cycles with more degrees of freedom. The other method is to choose some fixed number of iteration cycles after which the refinement is made. These strategies are compared numerically using two different error measures.

## INTRODUCTION

The important work of Babuska and his collaborators on adaptive finite element methods has generated considerable interest in this subject in recent years. An overview of this work is available in references [1] and [2] and more details can be found in references therein.

Finite element computer programs are self-adaptive when they have a local error indicator capability and a capability to assign automatically additional degrees of freedom to those regions where the indicated error is the greatest. Self-adaptive finite element computer programs can be designed to utilize the results of an available solution in obtaining succeeding solutions. These programs can be based on h- or p-convergent approximations or on combinations of both, hp-version. The objective of an adaptive

method is to analyse the potentially most important contributions of higher order terms and to incorporate them to the solution in a sequential fashion until the desired level of accuracy has been reached.

In order to investigate p-convergence, a finite element computer program having the capability to vary the polynomial order over the finite elements either selectively or uniformly from three to eleven has been coded. Based on this computer program, studies have been carried out concerning the behaviour of adaptive procedures in a nonlinear curve creation problem. In this paper is discussed one important question in the adaptive procedure: when to increase the order of polynomial approximation.

## FINITE ELEMENT DISCRETISATION OF THE PROBLEM

The curve creation problem considered here is the same as in reference [3]. The curve is described parametrically. Thus the FEM representation for the curve in $x,y$-plane is

$$x = \mathbf{N}\hat{\mathbf{a}} = \sum_i N_i \hat{a}_i \tag{1}$$

$$y = \mathbf{N}\hat{\mathbf{b}} = \sum_i N_i \hat{b}_i \tag{2}$$

where $\mathbf{N}$ is the shape function matrix and $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ are lists of nodal parameters. Shape functions $N_i$ are in each element functions of parameter $\xi, \xi \in [-1,1]$ and the derivatives in the nodal parameter lists are with respect to parameter $\xi$. The functional under minimization is the length of the curve

$$L = \int_q^r \sqrt{1+(y_{,x})^2}\, dx \tag{3}$$

The constraint is the area enclosed by the curve $y = y(x)$ and the x-axis in the interval $[q,r]$

$$\int_q^r y\, dx = A \tag{4}$$

When the equations (1) and (2) are substituted into equation (4) we get linear constraint equation

$$\mathbf{G}\hat{\mathbf{b}} = A \tag{5}$$

where the element contributions are

$$G_i^e = \int_{-1}^{1} N_i \, x_{,\xi} \, d\xi \tag{6}$$

When equation (2) is substituted into equation (3) and the integration is thought to have been carried out, the length $L$ is controlled by $\hat{\mathbf{b}}$, which now contains the only unknowns. We now have to find $\hat{\mathbf{b}}$ which minimizes $L$ and satisfies equation (5) plus possible pointwise constraints. In using the Lagrangian procedure we first write a modified function

$$L^* = L(\hat{\mathbf{b}}) + \lambda(\mathbf{G}\hat{\mathbf{b}} - A) \tag{7}$$

Demanding $L^*$ to be stationary w.r.t. $\hat{\mathbf{b}}$ and $\lambda$ the equations obtained are

$$\frac{\partial L}{\partial \hat{\mathbf{b}}} + \mathbf{G}^{\mathrm{T}}\lambda = \mathbf{0} \tag{8}$$

$$\mathbf{G}\hat{\mathbf{b}} - A = 0 \tag{9}$$

From equation (3) we see that the typical element of the vector $\frac{\partial L}{\partial \hat{\mathbf{b}}}$ is (before going into parametric form)

$$\frac{\partial L}{\partial \hat{\mathbf{b}}} = \int_q^r \frac{1}{\sqrt{1 + y_{,x}^2}} y_{,x} \frac{\partial y_{,x}}{\partial \hat{b}_i} dx \tag{10}$$

From equations (2) and (10) we see that if the square-root term would be nonexistent, then the right hand side of (10) would be linear in $\hat{b}_j$. Because other terms in equations (8) and (9) are linear in $\hat{b}_j$, we can try to find the solution iteratively by approximating during the $q^{\text{th}}$ iteration cycle the denominator by the values of the previous cycle. Hence we obtain for the $q^{\text{th}}$ iteration cycle

$$\frac{\partial L^{(q)}}{\partial \hat{b}_i} = \sum_j K_{ij}^{(q)} \hat{b}_j^{(q)} \tag{11}$$

where the typical element contribution is

$$K_{ij}^{e(q)} = \int_{-1}^{1} \frac{1}{\sqrt{1+(y_{,x}^{(q-1)})^2}} N_{i,x} N_{j,x} x_{,\xi} d\xi \tag{12}$$

Hence, during iteration cycle $q$ we have to solve a linear system of equations (vide equations (8) and (9))

$$\begin{bmatrix} \mathbf{K}^{(q)} & \mathbf{G}^{\mathrm{T}} \\ \mathbf{G} & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}}^{(q)} \\ \lambda^{(q)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ A \end{bmatrix} \tag{13}$$

However, before solution the given values for elements of $\hat{\mathbf{b}}$ must be inserted. Convergence of the iteration is followed by calculating after each cycle the current value for the length $L$. Iteration is stopped when the improvement (relative shortening) is under a preset limit.

## HIERARCHICAL ELEMENTS

To achieve continuity of the function itself and its first derivative, the element type used here is the one-dimensional Hermitian isoparametric element with two nodes. The nodal parameter vector for a typical element e is

$$\hat{\mathbf{a}}^e = \left[\hat{x}_1 \quad (\hat{x}_{,\xi})_1 \quad \hat{x}_2 \quad (\hat{x}_{,\xi})_2\right]^{\mathrm{T}} \tag{14}$$

and an analogous expression for $\hat{\mathbf{b}}^e$. The interpolating functions are the first order Hermitian polynomials $H_{ij}^1 \; (i = 0,1; j = 1,2)$.

Hierarchical finite elements may be defined as those in which successive refinements are additive in the manner of additional terms in a typical Fourier series. The hierarchical, additional degrees of freedom are introduced in using Legendre-type surplus functions $S_i$, which have zero values and slopes at nodes, as shape functions $N_{i+5}$, i.e. the same

kind of functions as Delpak and Peshkam used in reference [4]. Computational experience has shown that basis functions based on Legendre polynomials have good properties from the point of view of numerical stability.

The surplus function of the $i^{th}$ order is defined as

$$N_{i+5} = S_i = \alpha(i)\frac{d^i}{d\xi^i}(\xi^2 - 1)^{i+2}, \quad i \geq 0 \tag{15}$$

where $\alpha(i) = (i!\,2^i)^{-1}$ is a normalisation factor to reduce the values of the surplus functions so that $|S_i(\xi)| \leq 1, \quad \xi \in [-1,1]$.

## ERROR MEASURES

The selection of an error measure depends on the goal of the computation, and on which response quantities need to be accurately determined. This is particularly important since the smallness of one measure does not guarantee the smallness of the errors in other response quantities. Here the accuracy of the solution is estimated by two different error measures.

The first error measure $e_{r1}$ is the relative error in the length of the curve

$$e_{r1} = \frac{|L - L_n|}{L} \tag{16}$$

This is chosen as an error measure because the length of the curve is that quantity which is minimized and also the change of its value is used as a criteria to stop the iteration process.

The second error measure $e_{r2}$ is the distance $d(u,\tilde{u})$ of $\tilde{u}$ from $u$ defined with the aid of the $L_2$-norm, where $u$ and $\tilde{u}$ are the exact and approximate solutions respectively

$$e_{r2} = d(u,\tilde{u}) = \|u - \tilde{u}\|_{L_2} = \sqrt{\int (u - \tilde{u})^2\, dx} \tag{17}$$

## ADAPTIVE PROCEDURE

In this section we present the adaptive procedure used and concentrate on the question when to refine. Because for nonlinear problems only a few results have been obtained using hierarchical refinement, the author has find just two references where this question has been treated. Rank and Werner [5] have made refinement after every iteration step and in [6] Oden et al. just generate a new mesh 'after each of a specified number of time steps' but do not mention what this specified number is.

It seems natural that when solving the equation (13) at the beginning when the mesh is coarse, it is not necessary to iterate the solution with the same accuracy as later when the mesh is refined. It is so because at the beginning we only want to know where to introduce more degrees of freedom. The numerical examples show that the error indicators remain stable after only one iteration cycle. But perhaps it is not the best way to make refinement after every iteration cycle either. That is because the introduction of the additional hierarchical degrees of freedom can cause disturbance at the iteration process. So we have to find a compromise between refinement after every iteration cycle and refinement after the iteration cycle at which a stopping criteria is satisfied. There are two ways to do this. The first way is to choose some fixed number of iteration cycles after which the refinement is made. The second way is to use an iteration stopping criteria with a variable tolerance. At the beginning it is quite rude and then when the calculation proceeds the criteria is tightened causing more iteration cycles when we have a finer mesh.

The numerical example shows that this latter method is better because the refinement is most efficient when it is made at every second or third iteration cycle and with the changing stopping criteria this is easiest achieved. Like said before, the error indicators remain stable already after one iteration cycle, so it can be said where to refine. But two or three iteration cycles are needed to stabilize the $\hat{\mathbf{b}}$-vector in equation (13) so that it can be used as a starting vector in the iteration process after refinement.

## EXAMPLE

The example is a test problem for which an analytical solution exists. It is a modification of the classical Dido's problem, 'among curves of given length, find the one that encloses largest area together with the straight line connecting the endpoints'. The solution of this is simply a circular arc. According to the reciprocity law of isoperimetric variational problems the solution is the same as in our problem, 'find the shortest curve enclosing a given area'.

The constraint is the area of a half circle. A mesh of two elements and three nodes is used. As boundary conditions only the function value is fixed to zero at the end nodes. Also the infinite slope is given as input data at the end nodes.

First is used the strategy to refine after a fixed number of iteration cycles. The refinement is always uniform in this example. In figures 1 and 2 are the errors $e_{r1}$ and $e_{r2}$ as a function of the number of the iteration cycle using refinement after every cycle, every $2^{nd}$, $3^{rd}$, $4^{th}$ or every $5^{th}$ cycle. The behaviour of both errors is alike. In every curve is clearly distinguished that part where refinements are occurred and that part where the maximum $p$ is reached and the curve becomes smooth.

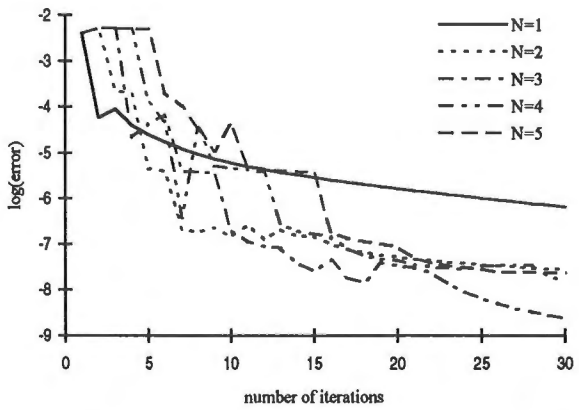


Fig. 1 The error $e_{r1}$ as a function of iteration cycle using refinement after 1,2,3,4 or 5 cycles.

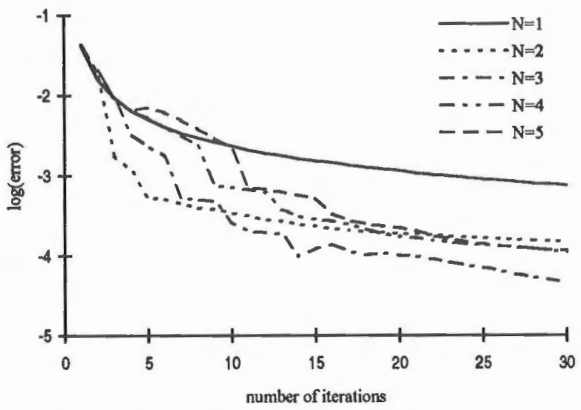


Fig. 2 The error $e_{r2}$ as a function of iteration cycle using refinement after 1,2,3,4 or 5 cycles.

Refinement after every cycle is clearly the worst alternative. Which one is best depends on the accuracy wanted. When a very small error is a target, refinement in every third cycle is the best choice, but if there is no need for small errors but an error $e_{r1}$ in the interval 0.01%...0.0001% or $e_{r2}$ between 1%...0.1% is satisfactory then refinement at every second cycle is most suitable although there are no great differences between these strategies. When refinement is done at longer intervals, every $4^{th}$ or $5^{th}$ cycle, the result gets worse. The situation is the same even if the computation time is considered.

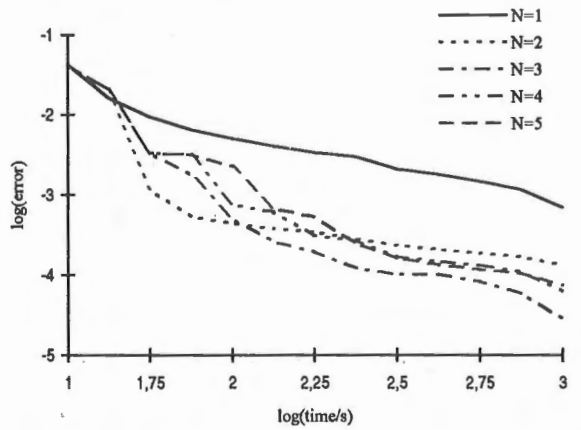


Fig. 3 The error $e_{r2}$ as a function of time using refinement after 1,2,3,4 or 5 cycles.

In figure 3 is the error $e_{r2}$ as a function of the time in log-log scale, and as can be seen the nature of the curves does not change.

The second strategy is to use an iteration stopping criteria in which the tolerance $\Delta$ is variable. The idea is to start with a big value of $\Delta$ and iterate until this criteria is fulfilled i.e. the relative change of the length of the curve is smaller than $\Delta$. Then the refinement is done according to the error indicators. Also a new smaller value of $\Delta$ is chosen and a new iteration is made with these new degrees of freedom. And so on every time when iteration stops the refinement is made and a smaller value of $\Delta$ is chosen. The difficulty is how to choose the value of $\Delta$. Four different cases were calculated. In every case the method is the same. First the starting value $\Delta_0$ is chosen and then an equation

$$\Delta_i = \gamma \Delta_{i-1} \tag{18}$$

is used, where $\gamma$ is some fixed number. In the following table 1 are the values of $\Delta_0$ and $\gamma$ for every case.

Table 1 The values of $\Delta_0$ and $\gamma$ used.

| Case | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\Delta_0$ | 0.1 | 0.1 | 0.2 | 0.1 |
| $\gamma$ | 0.1 | 0.02 | 0.01 | 0.01 |

The selection of these values was based on the experience gained on calculating the results with the first strategy. The results are shown in figures 4 and 5. There are the errors $e_{r1}$ and $e_{r2}$ as a function of the iteration cycle number. In these figures is also drawn the errors using the first strategy and refinement after every three iteration cycles. The differences are not great but perhaps the case C=2 is the best one. As a summary it can be said that with this changing stopping criteria it is easier to get the fastest convergence rate. The word 'easier' means here that this strategy is not very sensitive as to how the values $\Delta$ and $\gamma$ are chosen.

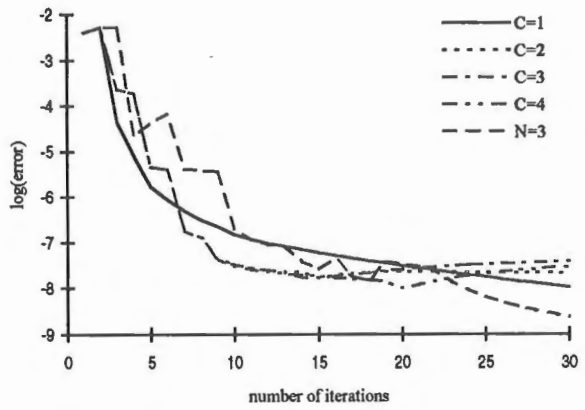


Fig. 4 The error $e_{r1}$ as a function of iteration cycle using changing stopping criteria.

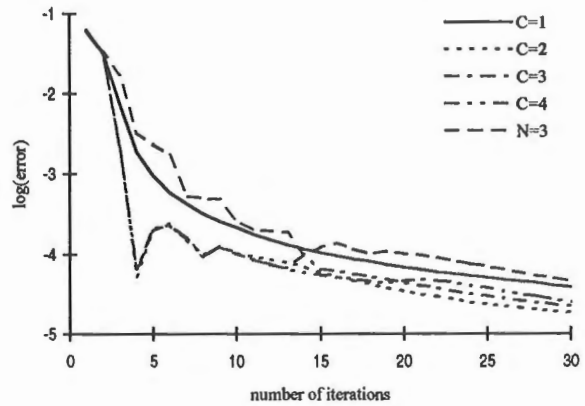


Fig. 5 The error $e_{r2}$ as a function of iteration cycle using changing stopping criteria.

## REFERENCES

[1] *Babuska I., Suri M.,* The p- and h-p versions of the finite element method, an overview. Computer methods in applied mechanics and engineering, Vol. 80, 1990, p. 5-26.

[2] *Noor A.K., Babuska I.,* Quality assessment and control of finite element solutions. Finite elements in analysis and design, Vol 3, 1987, p. 1-26.

[3] *Pramila A. and Virtanen S.*, Surfaces of minimum area by FEM. International journal for numerical methods in engineering, Vol 23, 1986, p. 1669-1677.

[4] *Delpak R. and Peshkam V.*, A study of the influence of hierarchical nodes on the performance of selected parametric elements. International journal for numerical methods in engineering, Vol 22, 1986, p. 153-171.

[5] *Rank E. and Werner H.*, An adaptive finite element approach for the free surface seepage problem. International journal for numerical methods in engineering, Vol 23, 1986, p. 1217-1228.

[6] *Oden J.T., Bass J.M., Huang C.-Y. and Berry C.W.*, Recent results on smart algorithms and adaptive methods for two- and three-dimensional problems in computational fluid mechanics. Computers & structures, Vol 35, 1990, p. 381-396.

Simo Virtanen, lecturer
Tampere University of Technology
P.O.Box 589
FIN-33101 Tampere
Finland

E-mail:virtanen@butler.cc.tut.fi