Technische Universität Chemnitz

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Gerd Kunert

A note on the energy norm for a singularly perturbed model problem

Preprint SFB393/01-02

Abstract

A singularly perturbed reaction–diffusion model problem is considered, and the choice of an appropriate norm is discussed. Particular emphasis is given to the *energy norm*. Certain prejudices against this norm are investigated and disproved. Moreover, an adaptive finite element algorithm is presented which exhibits an optimal error decrease in the energy norm in some simple numerical experiments. This underlines the suitability of the energy norm.

Keywords: singularly perturbed problem, reaction diffusion equation, adaptive algorithm, error estimator

AMS: 35B25, 65N50, 65N30, 65N15

Preprint-Reihe des Chemnitzer SFB 393

SFB393/01-02 **January 2001**

Contents

Gerd Kunert
TU Chemnitz
Fakultät für Mathematik
09107 Chemnitz
Germany

`http://www.tu-chemnitz.de/~gku`
`http://www.tu-chemnitz.de/sfb393/`

1 Motivation and model problem

Singularly perturbed problems have attracted much interest from both scientists and engineers. They include, for example, diffusion–convection–reaction problems (with a singular parameter) or plate and shell problems (where the singular character is introduced via the geometry). A comprehensive overview of singularly perturbed problems as well as their theoretical and numerical treatment can be found in the textbooks by [MNP01, RST96, MOS96, Mor96] and the citations contained therein.

In this note we consider a singularly perturbed reaction–diffusion problem which serves as a comparatively simple model problem. The most common norms to measure the discretization error of some numerical approximation are the (discrete) L_∞ or maximum norm and the energy norm. Beside those, other specialized norms have been proposed as well. The actual choice of the norm depends, for example, on the numerical method (e.g. finite element/finite different scheme) and the tools to analyse it.

Here we will promote the *energy norm*. More precisely, our aim is twofold:

- *Occasionally it is claimed that the energy norm is unsuitable. The corresponding argument is investigated and disproved.*
- *For our second and more important argument we apply an energy norm error estimator within an adaptive algorithm. This algorithm produces optimal meshes which underlines the suitability of the energy norm.*

For clarity we may stress that we do *not* claim or suggest that other norms are inappropriate; we simply examine the energy norm and conclude its usefulness. In this way we hope to stimulate the discussion about several norms and their fields of application.

In order to eliminate unwanted side effects we examine the following simple model problem of a singularly perturbed reaction–diffusion equation:

$$\left.\begin{array}{rcll} -\varepsilon\Delta u + u &=& f & \text{in } \Omega \\ u &=& g & \text{on } \Gamma_D \equiv \partial\Omega \end{array}\right\} \qquad (1)$$

with a small perturbation parameter $0 < \varepsilon \ll 1$. The corresponding *energy norm* (for some domain ω) then is

$$|||v|||_\omega := \left(\varepsilon\|\nabla v\|^2_{L_2(\omega)} + \|v\|^2_{L_2(\omega)}\right)^{1/2} \qquad , \qquad (2)$$

and it is problem dependent via ε. The singular character of problem (1) usually gives rise to boundary layers [RST96, MOS96]. The smaller ε is the more distinguished these layers will be. Hence standard techniques to solve (1) become very vulnerable for small ε.

Robust numerical methods for (1) and for more general diffusion–convection–reaction problems have been proposed by many authors; the textbooks [RST96] and [MOS96] supply an overview of many such techniques and may serve as a guide. It has turned out that two main approaches are particularly successful:

- *operator fitted* methods (e.g. adapted finite element/finite difference methods on standard meshes),
- *mesh fitted* methods (e.g. standard finite element/finite difference methods on adapted meshes).

For example, uniformly convergent methods and stabilization techniques are discussed in [RST96]. In [MOS96], operator fitted methods and mesh fitted methods are investigated, mostly using the L_∞ norm. In the concept of adaptive algorithms, *error estimators* play a central role. Robust estimators for (1) have been developed within the last decade, e.g. in [Ang95, Ver98, AB99, Kun00, Kun01].

2 The energy norm – Pro and Contra

Our main focus is on the choice of the norm in which to measure the discretization error. In particular we investigate the energy norm and conclude that it is both well–suited and appropriate (despite occasional claims to the contrary). Additionally an important argument is given in favour of the energy norm, and some prejudices are dismissed.

As already mentioned the maximum norm and the energy norm are among the most widely used norms; thus we start with a brief recollection. The L_∞ norm is applied e.g. throughout [MOS96] and partly in [RST96]. As a rough guideline, this maximum norm is particularly successful for analysing finite difference methods. The energy norm is employed e.g. in [Ver98, AB99, Kun00, Kun01] and also partly in [RST96]. This energy norm seems to be better suited to deal with finite element techniques.

After this brief overview let us now examine the energy norm more closely, and in particular the arguments for and against it. To start with, the energy norm is problem dependent, which indeed could be seen as a minor disadvantage. However, in our opinion this slight drawback is made up for, or even outweighed, by the fact that the energy norm is the natural norm associated with the variational formulation of problem (1).

The main argument invoked against the energy norm is that it allegedly cannot distinguish between layer functions and the zero function. Consider, for example, a typical layer function $v(x) := e^{-x/\sqrt{\varepsilon}}$ in $\Omega = (0, 1)$, as in [MOS96, pages 12f]. Then

$$|||v - 0|||_\Omega = \mathcal{O}(\varepsilon^{1/4}) \qquad \text{and} \qquad \|v - 0\|_{L_\infty} = 1 \qquad .$$

Truly, the energy norm of the difference of this layer function v and the zero function vanishes *in the limiting case* $\varepsilon \to 0$. We claim, however, that no further conclusions can be drawn. Firstly, any actual numerical simulation is performed for some fixed value of ε. Then the energy norm $|||v - 0|||$ is small indeed but only on an *absolute scale*. Nevertheless $|||v - 0|||$ can large in *relative terms*, e.g. compared with the discretization error (cf. also example 2 below). Hence any conclusions at this stage seem to be premature.

Secondly, the layer function v and the zero function cannot be compared as they are since they obey different boundary conditions.

Our main argument in favour of the energy norm states that an adaptive finite element algorithm can decrease the error (in the energy norm) with an optimal rate. Such an adaptive algorithm naturally incorporates an energy norm error estimator. This estimator should also be robust with respect to the small perturbation parameter ε, which has been achieved recently in [Ver98, AB99, Kun00, Kun01].

Surprisingly, extensive literature search has revealed that apparently none of these estimators has been applied yet in adaptive algorithms (at least no results have been published). Only [Pap98, PV00] carried out some investigations which are close to our intentions. Their examinations are for two–dimensional domains, and the problems there are more general than (1). The numerical examples show that boundary and interior layers can be resolved appropriately by an adaptive algorithm, even if the results are not optimal yet. This mainly seems to be due to the more complicated problem where several theoretical questions are unsolved.

In the next section we want to bridge the gap between the analytically known energy norm error estimator and numerical reality. Hence on two examples we demonstrate the potential of the energy norm to control an adaptive finite element algorithm towards useful or even optimal meshes. The criterion to judge an adaptive algorithm is the error decrease (in the chosen norm) with respect to the number of unknowns N. Applying for example linear finite elements and the energy norm, the asymptotically optimal error decrease is $\mathcal{O}(N^{-1/d})$ for $\Omega \subset \mathbb{R}^d$.

3 The adaptive algorithm and numerical experiments

In order to concentrate on the role of the energy norm and to eliminate unwanted influences we retreat to the comparatively simple model problem which is a special case of (1). Our actual example enjoys several favourite properties, namely

- it is a one–dimensional problem,
- the analytical solution u and thus the discretization error $u - u_h$ are known,
- the computational implementation is easily accomplished, e.g. in `MATLAB`.

The classical formulation of our model problem is a special case of (1) and reads

$$\begin{aligned} -\varepsilon\, u'' + u &= f \qquad \text{in } \Omega = (0,1) \qquad , \\ u(0) &= 1, \qquad u(1) = 0 \qquad , \end{aligned}$$

with $f \in \mathbb{P}^2(\Omega)$. The discretization with linear finite elements utilizes a mesh with nodal points $\{x_i\}_{i=0}^N$. The finite element solution is denoted by u_h. The adaptive algorithm consists of the steps *Solve system of equations – Estimate error – Refine mesh*. The last two ingredients are described now.

Error estimation: The energy norm error estimator is the one–dimensional counterpart of [Ver98]. For a proper description, define the following data.

Mesh size	$h_i := x_i - x_{i-1}$	$i = 1 \dots N$
Finite element	$T_i := (x_{i-1}, x_i)$	$i = 1 \dots N$
Macro element	$\omega_i := (x_{i-2}, x_{i+1})$	$i = 2 \dots N-1$
	$\omega_0 := (x_0, x_2) \qquad \omega_N := (x_{N-2}, x_N)$	
Element residual	$R_i := f - (-\varepsilon\, u_h'' + u_h)$	$i = 1 \dots N$
Jump residual	$J_i := \varepsilon \cdot [u_h'(x_i + 0) - u_h'(x_i - 0)]$	$i = 1 \dots N-1$
	$J_0 = J_N := 0$	
Scaling factor	$\alpha_i := \min\{1\,,\, \varepsilon^{-1/2} h_i\}$	$i = 1 \dots N$
Local error estimator	$\eta_i^2 := \alpha_i^2\, \|R_i\|_{L_2(T_i)}^2 + \varepsilon^{-1/2}\alpha_i \cdot (J_{i-1}^2 + J_i^2)$	$i = 1 \dots N$

Note that x_i and J_i are node related data whereas h_i, T_i, R_i, α_i, η_i are element related data. Verfürth [Ver98] has proven that the energy norm of the error is bounded locally from below and globally from above:

$$\begin{aligned} \eta_i &\le c_L\, |||u - u_h|||_{\omega_i} \qquad \forall\, i = 1 \dots N \\ |||u - u_h|||_{\Omega} &\le c_U \left(\sum_{i=1}^{N} \eta_i^2 \right)^{1/2} \qquad . \end{aligned}$$

The constants c_L, c_U are independent of ε, i.e. the error estimation is robust.

Mesh refinement: Here it suffices to choose a simple strategy. Start with an equidistributed mesh of 10 elements. Once the error estimators are computed, an element T_i is bisected iff

$$\eta_i \ge \gamma \cdot \max_{k=1 \dots N} \eta_k \qquad ,$$

where the refinement parameter is set to $\gamma := 0.1$. More sophisticated refinement strategies are possible of course.

Example 1: Single boundary layer

This experiment features a single boundary layer. Thus it is best suited to illustrate the principal behaviour of the adaptive algorithm whilst eliminating perturbing influences at the same time. With the choice $f \equiv 0$ of the right hand side one obtains a sharp boundary layer of the type $e^{-x/\sqrt{\varepsilon}}$ at $x = 0$, see Figure 1.

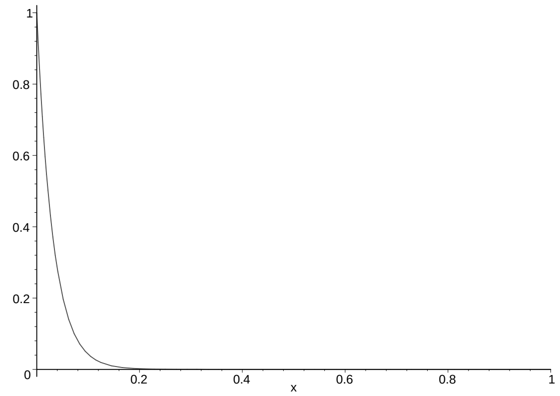

Figure 1: Analytic solution u for example 1, $\varepsilon = 10^{-3}$

In Figure 2 we present the results of our adaptive algorithm for different values of ε. The figure depicts the error decrease in the energy norm with respect to the number of unknowns N. It turns out that in all cases the convergence rate is very close to the optimal rate of $\mathcal{O}(N^{-1})$ as soon as the boundary layer is resolved. As desired, the convergence rate is independent of ε. Hence the chosen algorithm is optimal (with respect to the energy error decrease), i.e. optimal meshes are obtained.

For completeness we also present the corresponding error decrease in the (nodal) L_∞ norm in Figure 3. Again, as soon as the boundary layer is resolved, the maximum norm of the error drops as well. The convergence rate here is approximately $\mathcal{O}(N^{-1.4})$ which is sub–optimal. (The nodal error $\|u - Iu\|_{L_\infty}$ of the linear interpolant Iu is at most $\mathcal{O}(N^{-2}\ln^2 N)$, see [RST96, Section 2.4.2]. Analytical investigations of the approximation error $\|u - u_h\|_{L_\infty}$ on Shishkin meshes result in the bound $\mathcal{O}(N^{-1}\ln N)$, cf. [MOS96, RST96]. Numerical experiments however give a convergence rate of approximately $\mathcal{O}(N^{-2.0})$ on Shishkin like meshes for our problem.) Note that little conclusions can be drawn from this information since the underlying adaptive algorithm has been designed to be optimal for the energy norm.

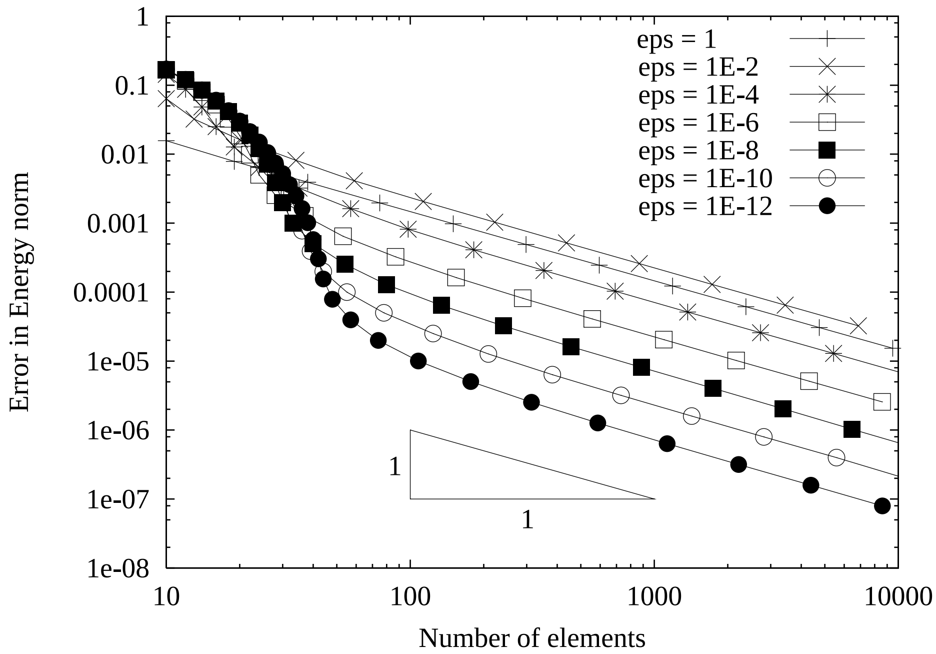

Figure 2: Example 1: Error decrease in the energy norm

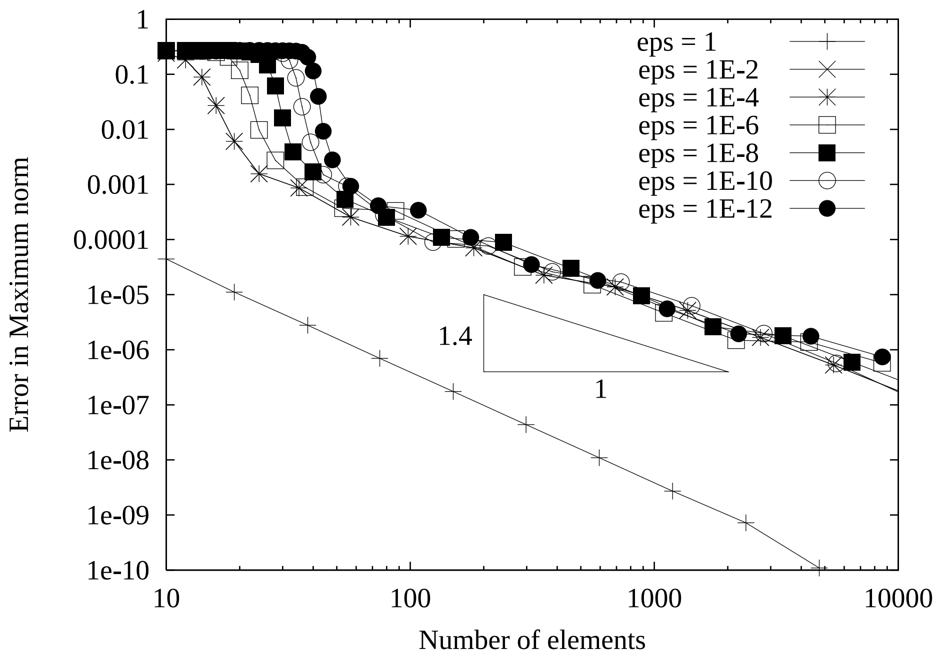

Figure 3: Example 1: Error decrease in the nodal maximum norm

Example 2: Boundary layer + quadratic function

In this experiment the same boundary layer as before is superposed on a quadratic function. In the context of asymptotic expansions, the layer function can be seen as the inner expansion whereas the quadratic function represents (exemplarily) the outer expansion. Here we choose $f = 2\varepsilon + x(1-x)$ which yields the analytic solution $u = e^{-x/\sqrt{\varepsilon}} + x(1-x)$ for small $\varepsilon \ll 1$, see Figure 4.

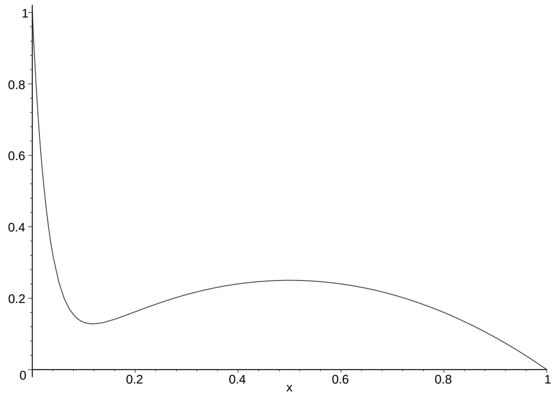

Figure 4: Analytic solution u for example 2, $\varepsilon = 10^{-3}$

In Figure 5 the error decrease in the energy norm is depicted. Again a convergence rate is observed which is close to the optimal rate of $\mathcal{O}(N^{-1})$. It is remarkable that this optimal rate is achieved although the energy norm of the layer function is $\mathcal{O}(\varepsilon^{1/4})$, and thus much smaller than the energy norm of the quadratic superposition (which is $\mathcal{O}(1)$).

For completeness the results for the discrete maximum norm are given as well, cf. Figure 6. Similar conclusions can be drawn as for example 1.

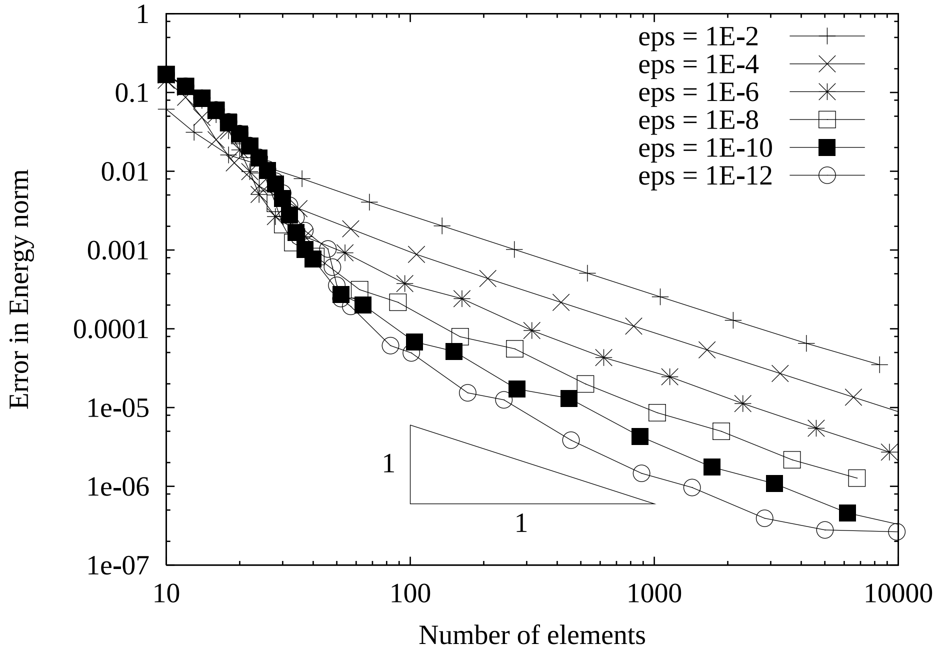

Figure 5: Example 2: Error decrease in the energy norm

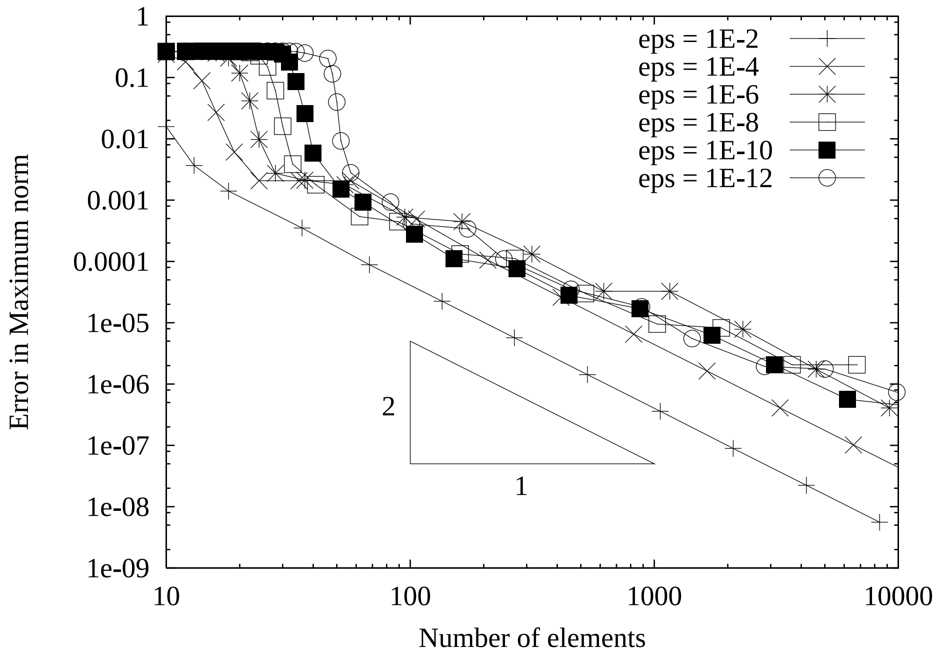

Figure 6: Example 2: Error decrease in the nodal maximum norm

4 Conclusions

We have seen that the choice of an appropriate norm for a singularly perturbed reaction–diffusion problem is, to some extend, a philosophical question. The answer depends on several aspects, for example the approximation method and the choice of the tools to analyse it.

We have shown that there are good reasons to opt for the energy norm. Firstly certain prejudices against this norm have analysed and disproved. Secondly, a convincing adaptive algorithm for a simple model problem encourages to employ the energy norm. Simultaneously we hope to stimulate the discussion about the different choices of the norm.

A Program listing

The following `MATLAB` program implements the aforementioned adaptive algorithm for example 1 and displays several interesting pieces of information as well as figures. The code is also available at

`http://archiv.tu-chemnitz.de/pub/2001/0006/data/reacdiff.m`

When you run the program in `MATLAB`, you are first prompted for the value of ε (the default value is 10^{-6}). After that only the `Return` key has to be hit repeatedly.

```
% Solve a singularly perturbed reaction diffusion model problem in 1D.
% An adaptive program with the residual error estimator of Verfuerth.
%
%        - eps*u'' + u = 0        in Omega = (0,1)
%        u(0)=1          u(1)=0
%
% Analytical solution: Set eps2 := sqrt(eps)
% u = ( exp(-x/eps2) - exp((x-2)/eps2) ) / (1-exp(-2/eps2))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialize.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n0 = 10;        % number of initial intervals (equidistant mesh)
n = n0;         % present number of intervals
x = 0:1/n:1;    % the initial mesh

gamma = 0.1;    % Refinement occurs if eta_T > gamma*eta_max
                % eta_T := error estimator, eta_max := max_T {eta_T}

eps=input('Please enter epsilon:          ');
if isempty(eps) == 1
        eps = 1e-6;
        fprintf('Use default value for epsilon: %4.1e\n',eps);
end
eps2 = sqrt(eps);
eps32 = eps2^3;

fprintf('Number of initial intervals:   %2d\n',n0);
fprintf('Refinement parameter gamma:   %5.2f\n\n',gamma);

fprintf('                         Upper Bd  Lower Bd\n');
fprintf('Elemente: |||Err|||   Err/Est   Est/Err     C-Err   Konv-rate\n');
fprintf('-----------------------------------------------------------\n');

fid = fopen('result-Enorm','w');
fprintf(fid,'# Energienorm, Epsilon =   %4.1e\n\n',eps);
fclose(fid);
fid = fopen('result-Cnorm','w');
fprintf(fid,'# Maximum Norm, Epsilon =   %4.1e\n\n',eps);
fclose(fid);

while 1    % Adaptive infinite loop
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the Finite Element Solution.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

h = x(2:n+1) - x(1:n);        % mesh sizes

D = [1 , eps*( 1./h(1:n-1) + 1./h(2:n) ) , 1]' + ( [h 0] + [0 h])'/3;
NDu = [-eps*1./h + h/6 , 0]';
NDo = [0 , -eps*1./h + h/6 ]';

A = spdiags([NDu D NDo] , -1:1 , n+1,n+1);        % stiffness matrix
A(1,1) = 1;
A(1,2) = 0;
A(n+1,n) = 0;
A(n+1,n+1) = 1;

b = [1; zeros(n,1) ];    % right hand side
uh = A\b;                % FEM solution

% Exact solution: Numerically stable for small eps
u = ( exp(-x'/eps2) - exp((x'-2)/eps2) ) / (1-exp(-2/eps2));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the error estimator.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% The local error estimator eta_T is for an interval T with two
% nodes E_i.
% The face residual r_E is the canonical choice, i.e. eps*Gradient jump.
%
% eta_T^2 := alpha_T^2 * || r_T ||_T^2 +
%              + eps^{-1/2}*alpha_T * Summe_E_i | r_E |^2
% with alpha_T := min(1,h_T/sqrt(eps))

al = min( 1 , h / sqrt(eps))';

% Element residual:
norm_rT2 = h'/3 .* ( uh(1:n).^2 + uh(2:n+1).^2 + uh(1:n).*uh(2:n+1) );

% "Face" residual = gradient jump * eps. Zero at x=0 and x=1.
rE = eps*[ 0 ;( uh(3:n+1) -uh(2:n)) ./h(2:n)' - ( uh(2:n) -uh(1:n-1))...
./h(1:n-1)' ; 0] ;
```

```
% Error estimator (squared)
Est = al.^2 .* norm_rT2 + eps^(-1/2) * al .* (rE(1:n).^2 + rE(2:n+1).^2);

Est_glob = sqrt(sum(Est));          % global error estimator
max_err = max(abs(uh - u));         % L_{infinity} error
Est_max = max(Est);                 % Maximum local error estimator


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the (analytical) error in the energy norm.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x1 = x(2:n+1)';          % Auxiliary (column) vector x_{i+1}
x0 = x(1:n)';

% Error fct = [ e^{-x/eps2} - e^((x-2)/eps2) ] / (1-e^{-2/eps2}) - c*x-d
% ||| error |||_T gives a formula of about one page length (obtained via
% MAPLE).
% For small eps this formula in unstable (gives NaN). Then one has
% approximately u ~ e^{-x/eps2}, and the error is (approximately and
% very accurately) as below.

c = ( uh(2:n+1) - uh(1:n) ) ./ h';
d = uh(1:n) - c.*x0;

if eps >= 0.001          % analytical formula by MAPLE

H0 = -1/6*(-12.*c.*x0.^2.*d.*exp(2/eps2)-6.*c.*x1.^2.*d.*exp(4/eps2)+6.*
c.*x0.^2.*d.*exp(4/eps2)+4.*c.^2.*x1.^3.*exp(2/eps2)+2.*c.^2.*x0.^3.*exp
(4/eps2)-4.*c.^2.*x0.^3.*exp(2/eps2)+12*c*eps.*exp(-(x0-4)/eps2)-12.*c*e
ps.*exp(-(x0-2)/eps2)+6.*d.^2.*x0.*exp(4/eps2)-12.*d.^2.*x0.*exp(2/eps2)
-12.*c*eps.*exp((x0+2)/eps2)+12.*c*eps2.*x0.*exp((x0+2)/eps2)+2.*c.^2.*x
0.^3+6.*d.^2.*x0+3*eps2.*exp(-2*(x1-2)/eps2)+12.*x1.*exp(2/eps2)+3*eps2.
*exp(2.*x0/eps2)-12.*x0.*exp(2/eps2)-3*eps2.*exp(2.*x1/eps2)+12.*d*eps2.
*exp((x0+2)/eps2)-2.*c.^2.*x1.^3.*exp(4/eps2)-12.*d*eps2.*exp(-(x0-2)/ep
s2)+12.*c*eps.*exp(x0/eps2)+12.*d*eps2.*exp(-(x0-4)/eps2)-3*eps2.*exp(-2
*(x0-2)/eps2)+6.*c.*x0.^2.*d-6.*c.*x1.^2.*d+12.*c.*x1.^2.*d.*exp(2/eps2)
-2.*c.^2.*x1.^3-6.*d.^2.*x1-12.*c*eps2.*x1.*exp(-(x1-4)/eps2)+12.*c*eps2
.*x1.*exp(-(x1-2)/eps2)+12.*c*eps2.*x0.*exp(-(x0-4)/eps2)-12.*c*eps2.*x0
.*exp(-(x0-2)/eps2)-12.*c*eps2.*x0.*exp(x0/eps2)+12.*c*eps2.*x1.*exp(x1/
eps2)-12.*c*eps2.*x1.*exp((x1+2)/eps2)-12.*d*eps2.*exp(x0/eps2)+12.*c*ep
s.*exp(-(x1-2)/eps2)+12.*c*eps.*exp((x1+2)/eps2)-12.*d*eps2.*exp((x1+2)/
```

```
eps2)-12.*d*eps2.*exp(-(x1-4)/eps2)-6.*d.^2.*x1.*exp(4/eps2)+12.*d.^2.*x
1.*exp(2/eps2)+12.*d*eps2.*exp(-(x1-2)/eps2)+12.*d*eps2.*exp(x1/eps2)-12
.*c*eps.*exp(-(x1-4)/eps2)-12.*c*eps.*exp(x1/eps2))/(exp(2/eps2)-1).^2;


H1 = 1/2*(-eps.*exp(-2*(x1-2)/eps2)+4.*x1.*exp(2/eps2)*eps2-4*c*eps32.*
exp(-(x1-4)/eps2)+4*c*eps32.*exp(-(x1-2)/eps2)+exp(2.*x1/eps2)*eps+4*c*e
ps32.*exp((x1+2)/eps2)-4*c*eps32.*exp(x1/eps2)+2*c.^2.*x1*eps32.*exp(4/e
ps2)-4*c.^2.*x1*eps32.*exp(2/eps2)+2*c.^2.*x1*eps32+eps.*exp(-2*(x0-2)/e
ps2)-4.*x0.*exp(2/eps2)*eps2+4*c*eps32.*exp(-(x0-4)/eps2)-4*c*eps32.*exp
(-(x0-2)/eps2)-exp(2.*x0/eps2)*eps-4*c*eps32.*exp((x0+2)/eps2)+4*c*eps32
.*exp(x0/eps2)-2*c.^2.*x0*eps32.*exp(4/eps2)+4*c.^2.*x0*eps32.*exp(2/eps
2)-2*c.^2.*x0*eps32)/eps32/(exp(2/eps2)-1).^2;

else        % approximate formula for small epsilon

    % First  Int (err')^2
    H1 = ( exp(-x1/eps2)-exp(-x0/eps2) ) .* ...
        ( ( exp(-x1/eps2)+exp(-x0/eps2) )/(-2)/eps2 - 2*c) ...
        + c.^2.*h';

    % Now Int (err)^2
    H0_a = - eps2/2 * ( exp(-2*x1/eps2)-exp(-2*x0/eps2) );
    H0_b = 2*eps2* exp(-x1/eps2) .* (c.*x1 + c*eps2 + d) - ...
      (2*eps2* exp(-x0/eps2) .* (c.*x0 + c*eps2 + d) );
    H0_c = c.^2/3.*(x1.^3-x0.^3) + c.*d.*(x1.^2-x0.^2) + d.^2 .*h';
    H0 = H0_a + H0_b + H0_c;

end        % end if

% The element error (squared) in the energy norm. Verified with MAPLE.
Err = eps*H1 + H0;

Err_glob = sqrt(sum(Err));        % global error
Err_max = max(Err);               % Maximum local error, squared

% The ratio of the lower error bound. Has to be bounded from above.
low_bd = Est ./ ( Err + [0 ;Err(1:n-1)] + [Err(2:n) ;0] );
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The convergence rate (energy norm) between two successive steps.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
err_new = Err_glob;
n_new = n;

if n > n0
        konv = -log(err_new/err_old) / log(n_new/n_old);
else
        konv = 0;
end;
n_old=n_new;
err_old=err_new;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The result and some nice plots:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subplot(2,2,1); plot(x,[uh  u]);
title('u and u_h')

subplot(2,2,2); plot(x,uh - u);
title('u - u_h');

subplot(2,2,3); plot(x,log10([h 0.1001]));
axis([0 1 log10(h(1))-1 0])
title('logarithmic mesh density');

subplot(2,2,4); plot(x,[sqrt(low_bd); sqrt(low_bd(n))]);
title('local Ratio Lower Bound: Est/Err');

fprintf('  %5d    %7.2e  %7.4f   %7.4f    %8.3e   %6.2f\n',...
n, Err_glob,Err_glob/Est_glob,max(sqrt(low_bd)),max_err,konv);

% Write to some file "result-...".
fid = fopen('result-Enorm','a');
fprintf(fid,'  %5d    %10.4e \n',n, Err_glob);
fclose(fid);
fid = fopen('result-Cnorm','a');
fprintf(fid,'  %5d    %10.4e \n',n, max_err);
fclose(fid);

pause
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Refinement:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loop over all elements:
% When Eta_T > gamma * Eta_max then insert new node.

m = 1;
clear y
for i=1:n
        y(m) = x(i);
        if Est(i) > gamma^2*Est_max        % mesh control via Estimator
%       if Err(i) > gamma^2*Err_max        % mesh control via true error
                y(m+1) = (x(i)+x(i+1))/2;
                m = m+2;
        else
                m = m+1;
        end;
end;
y(m) = 1;

x = y;
n = m-1;

end        % End while
```

References

[AB99] M. Ainsworth and I. Babuška. Reliable and robust a posteriori error estimation for singularly perturbed reaction–diffusion problems. *SIAM J. Num. Anal.*, 36(2):331–353, 1999.

[Ang95] L. Angermann. Balanced a-posteriori error estimates for finite volume type discretizations of convection-dominated elliptic problems. *Computing*, 55(4):305–323, 1995.

[Kun00] G. Kunert. Robust a posteriori error estimation for a singularly perturbed reaction–diffusion equation on anisotropic tetrahedral meshes. *submitted to Adv. Comp. Math.*, 2000.

[Kun01] G. Kunert. Robust local problem error estimation for a singularly perturbed problem on anisotropic finite element meshes. Preprint SFB393/01-01, TU Chemnitz, January 2001.
Also http://archiv.tu-chemnitz.de/pub/2001/0001/index.html.

[MNP01] V. Maz'ya, S. Nazarov, and B. Plamenevskij. *Asymptotic theory of elliptic boundary value problems in singularly perturbed domains. Vol. I and II.* Operator Theory: Advances and Applications, 111/112. Birkhäuser, Basel, 2001. Translation from German.

[Mor96] K. W. Morton. *Numerical solution of convection-diffusion problems.* Chapman & Hall, London, 1996.

[MOS96] J. J. H. Miller, E. O'Riordan, and G.I. Shishkin. *Fitted numerical methods for singularly perturbed problems. Error Estimates in the maximum norm for linear problems in one and two dimensions.* World Scientific Publications, Singapore, 1996.

[Pap98] A. Papastavrou. *Adaptive Finite Element Methoden für Konvektions–Diffusionsprobleme.* PhD thesis, Ruhr–Universität Bochum, 1998.

[PV00] A. Papastavrou and R. Verfürth. A posteriori error estimators for stationary convection–diffusion problems: a computational comparison. *Comput. Methods Appl. Mech. Eng.*, 189(2):449–462, 2000.

[RST96] H.-G. Roos, M. Stynes, and L. Tobiska. *Numerical methods for singularly perturbed differential equations. Convection-diffusion and flow problems.* Springer, Berlin, 1996.

[Ver98] R. Verfürth. Robust a posteriori error estimators for singularly perturbed reaction–diffusion equations. *Numer. Math.*, 78:479–493, 1998.

Other titles in the SFB393 series:

00-22 V. Uski, B. Mehlig, R. A. Römer, M. Schreiber. Numerical study of eigenvector statistics for random banded matrices. May 2000.

00-23 R. A. Römer, M. Raikh. Aharonov-Bohm oscillations in the exciton luminescence from a semiconductor nanoring. May 2000.

00-24 R. A. Römer, P. Ziesche. Hellmann-Feynman theorem and fluctuation-correlation analysis of i the Calogero-Sutherland model. May 2000.

00-25 S. Beuchler. A preconditioner for solving the inner problem of the p-version of the FEM. May 2000.

00-26 C. Villagonzalo, R.A. Römer, M. Schreiber, A. MacKinnon. Behavior of the thermopower in amorphous materials at the metal-insulator transition. June 2000.

00-27 C. Mehl, V. Mehrmann, H. Xu. Canonical forms for doubly structured matrices and pencils. June 2000. S. I. Solov'ev. Preconditioned gradient iterative methods for nonlinear eigenvalue problems. June 2000.

00-29 A. Eilmes, R. A. Römer, M. Schreiber. Exponents of the localization lengths in the bipartite Anderson model with off-diagonal disorder. June 2000.

00-30 T. Grund, A. Rösch. Optimal control of a linear elliptic equation with a supremum-norm functional. July 2000.

00-31 M. Bollhöfer. A Robust ILU Based on Monitoring the Growth of the Inverse Factors. July 2000.

00-32 N. Arada, E. Casas, F. Tröltzsch. Error estimates for a semilinear elliptic control problem. July 2000.

00-33 T. Penzl. LYAPACK Users Guide. August 2000.

00-34 B. Heinrich, K. Pietsch. Nitsche type mortaring for some elliptic problem with corner singularities. September 2000.

00-35 P. Benner, R. Byers, H. Faßbender, V. Mehrmann, D. Watkins. Cholesky-like Factorizations of Skew-Symmetric Matrices. September 2000.

00-36 C. Villagonzalo, R. A. Römer, M. Schreiber, A. MacKinnon. Critical Behavior of the Thermoelectric Transport Properties in Amorphous Systems near the Metal-Insulator Transition. September 2000.

00-37 F. Milde, R. A. Römer, M. Schreiber. Metal-insulator transition in anisotropic systems. October 2000.

00-38 T. Stykel. Generalized Lyapunov Equations for Descriptor Systems: Stability and Inertia Theorems. October 2000.

00-39 G. Kunert. Robust a posteriori error estimation for a singularly perturbed reaction-diffusion equation on anisotropic tetrahedral meshes. November 2000.

01-01 G. Kunert. Robust local problem error estimation for a singularly perturbed problem on anisotropic finite element meshes. January 2001.

The complete list of current and former preprints is available via
`http://www.tu-chemnitz.de/sfb393/preprints.html`.