Mathematics
Chalmers University of Technology
and Göteborg University

TMA371 PARTIAL DIFFERENTIAL EQUATIONS 1997
ASSIGNMENT 1A

1. Purpose

The purpose of this exercise is

1) to help you better understand the basic differential equation models used in the course. In particular, we study the roles played by the different terms in the differential equations, the signs and sizes of the coefficients, and the different types of boundary conditions;

2) to help you understand what makes the approximate finite element solutions come close to the exact solution. In particular, we attempt to study the effect of the smallness of the meshsize, the degree of the polynomials, and the regularity of the solution. Also, we note the fact that the accuracy drops as we compute derivatives of the solution, i. e., the accuracy is higher in the primitive variable (like the temperature and deflection) than in derived variables (like the heat flux and moments), in general;

3) to help you understand the idea of adaptivity, and the structure of an adaptive algorithm. In particular, we shall see that adaptivity and local mesh refinement can compensate for local non-smoothness of the solution, i. e., by using local mesh refinement it is possible to get the same accuracy as if the solution were smooth. Also, we analyze the *reliability* and *efficiency* of an adaptive algorithm by using test problems, where the solution is known, so that we can compare the estimated error guiding the adaptive algorithm, with the exact error;

4) to let you solve a few "real world" problems or, at least, taste solving such problems, by constructing or completing adequate mathematical models and then simulate the real world phenomena on the computer using an adaptive finite element program.

2. Model understanding—an excursion

2.1. **A general second order equation model.** Consider first the second order differential equation

$$-D(d(x)Du) + c(x)Du + a(x)u = f(x), \quad \text{for } x \in [0,1],$$

together with the boundary conditions

$$dDu + ku = g \quad \text{or} \quad u = g \quad \text{at } x = 0, \tag{1}$$

and

$$-dDu + ku = g \quad \text{or} \quad u = g \quad \text{at } x = 1.$$

This problem may be solved using the MATLAB program `adfem.m`, which you start from MATLAB by typing the command `adfem`. Further details will be given below. See also the `README` file following the `adfem.m` file.

We now (but just temporarily) forget about finite elements, adaptivity and error control by always viewing the computed solution as the true solution of the given problem, if it looks reasonable. For example, we may choose error control in the energy norm and tolerance = 0.01. Recall that the purpose here is to get an understanding of the mathematical model rather than the discretization aspects.

a) Consider first the case $d(x) = 1$, $c(x) = a(x) = 0$, $f(x) = 1$ and take the boundary conditions to be of Dirichlet type and homogenouos (= 0), i. e., take $u = 0$ at both $x = 0$ and $x = 1$. We have seen that this models (one dimensional) stationary heat conduction with heat production f, heat conductivity d, temperature u, and heat flux $-dDu$. More generally, the equation models any *diffusion* type process with u being the concentration of the diffusing substance, d the diffusivity and f the production of the substance. The equation expresses the balance between production and diffusion of u (or rather the substance with concentration u). As you can see, for the given data the solution is a nice and smooth parabola.

b) Now add to the model also production of u by "reaction" proportional to u, by choosing the *absorbtion* coefficient a <u>negative</u>, i. e., take $a(x) \equiv -1$. Moving the $a(x)u$-term to the right hand side in the equation we see that we now have a net production, which is $f - au = 1 + u$, giving somewhat larger u-values than before. Recall that you need not re-run `adfem` to compute the new solution. You may take a short-cut by giving the new non-zero absorbtion coefficient directly at the MATLAB prompt by typing `absorbtion='-1'` followed by the command `solve`.

c) Next change to $a(x) = 1$ corresponding to *absorbtion* proportional to u rather than reaction. Note that the absorbtion has a damping effect resulting in smaller u-values.

d) We now change to a more violent source term by taking $f = 1/x$ (type `force='1/x'` followed by `solve` at the MATLAB prompt). Notice that the solution is still very smooth although the data f is now singular near $x = 0$. This is because there is quite a bit of "diffusion" in our model, i. e., $d(x)$ is not small. We notice, however, that the concentration of u is somewhat larger to the left in the computational domain where the production is most intensive.

e) Presence of diffusion has a *smoothing* effect on the solution. To see this more clearly, consider the case of a variable diffusivity $d(x) = 0.01 + x^4$ (type `diffusion='0.01+x^4'`) and put $f(x) = \sin(40 * x)$. Near $x = 0$ where the diffusivity is small, the rapid variation in production intensity is reflected in a similar variation in the solution u, whereas near $x = 1$ where the diffusivity is "large" the variation in concentration is smaller, i. e., the solution u is "smoother".

f) Now we change to uniformly small diffusion by choosing $d(x) \equiv 0.02$ and again take $f(x) = 1/x$. Note that the solution is now much more sensitive to the singularity in the data $f(x)$. What may be more surprising is that you will see singularities (or rather regions of large local variation) in the solution also if you choose very nice data like $f(x) \equiv 1$. Try this, and try also to explain why it is natural that the solution has this form by comparing with the solution of the equation when $d = 0$.

g) Now its time to include *convection* in the model by choosing $c(x) = 1$ (type `convection='1'`) corresponding to u (or the substance with concentration u) also beeing transported with a "wind" blowing to the right. Also, change to $a(x) = 0$ for simplicity. Again the solution has a "singular" boundary layer, but in this case only near $x = 1$. Note again that both the convection coefficient c, the absorbtion coefficient a and the source term f are smooth and not large. The layer near $x = 1$

is here called an outflow boundary layer and is caused by the somewhat unflexible and "incompatible" boundary condition $u = 0$ at $x = 1$. Motivate why the solution should have this behaviour by comparison with the corresponding solution when $d = 0$ and with a given boundary value only at $x = 0$.

h) Try changing boundary condition at $x = 1$ to the more flexible one $Du = 0$ (which may also be more realistic in many cases, see the pollution problem below) by putting `bctype(2)=1`. What happens now to the boundary layer?

i) Finally, consider problem (1) on the interval $[-1, 1]$ with $d = 0.02$, $c(x) = 1$ for $x < 0$, $c(x) = -1$ for $x > 0$ (type `convection='-x/(abs(x)+eps)'`), $a = 0$, and boundary conditions $u(-1) = 2$, $u(1) = 1$. Try to guess what the solution will look like. Check your guess using `adfem`.

2.2. **A fourth order equation—the beam model.** Consider now the fourth order differential equation

$$D^2(a(x)D^2u) = f(x) \quad \text{for } x \in [0, 1] \tag{2}$$

with boundary conditions

$$u = u0 \quad \underline{\text{and}} \quad Du = u1 \quad \text{(clamped)}$$

or

$$aD^2u = 0 \quad \underline{\text{and}} \quad D(aD^2u) = 0 \quad \text{(free)}$$

at $x = 0$ and $x = 1$, respectively. This models a beam under the load $f(x)$ with $u(x)$ being the deflection due to the load. The coefficient a corresponds to the "stiffness" of the beam, aD^2u is the *bending moment* and $D(aD^2u)$ is the *shear force*. At the boundary points the beam may be "clamped" or "free". Note that we now give two conditions at each boundary point because we have a fourth order equation.

Problem (2) may be solved using the MATLAB program `beam.m`, which you start from MATLAB with the command `beam`.

a) Take first clamped boundary conditions, $f(x) \equiv 1$ and compare the two cases $a(x) \equiv 1$ and $a(x) = 0.01 + (x - 0.5)^2$, where the latter coefficient corresponds to a beam which is thinner or otherwise "softer" in its mid-section than towards its ends.

b) Try some other combinations of a, f and boundary conditions of your own choice. Does the solution look right?

c) For the "experts": Modify the code so that also problems with boundary conditions of the form

$$u = u0 \quad \underline{\text{and}} \quad aD^2u = 0 \quad \text{("freely supported")}$$

can be solved.

3. How do we make the error small?

a) Consider the beam problem (2) with $a = 1$ and $f = 1$ with the beam clamped at $x = 0$ and free at $x = 1$, say. What does theory predict about the size of the error as the meshsize is decreased, i. e., by a factor of two by cutting each element into two of half the size? What does theory predict about the size of the error in the first and second derivative, respectively, as the meshsize is decreased by a factor of two? Check the theory by deriving the exact solution formulas for u, Du and D^2u for comparison, and taking first $h = 0.2$ and then $h = 0.1$. Note that `beam` computes both the error `err` and the first and second derivative errors `Derr` and `DDerr`.

b) Then consider the case $f = 1/x$. In this case we expect that D^3u has the asymptotic behaviour of $\log(x)$ near $x = 0$. We recall that $\log(x)$ is "almost bounded" (tends to $-\infty$ very slowly as x tends to zero). That is, u has (almost) three bounded derivatives. What does theory predict in this case about the size of the error (in the maximum norm) as the meshsize is decreased by a factor of two, considering uniform meshes? Check the theory in this case too.

c) Consider the case $f = 1/\sqrt{x}$ corresponding to u having "$3\frac{1}{2}$ bounded derivatives". Again try decreasing the meshsize by a factor of two and check if a rate of convergence corresponding to $3\frac{1}{2}$ derivatives can be observed.

d) What does theory say about the size of the error and its derivative for the corresponding second order problem (1) as the meshsize is decreased by a factor of two? Consider the cases $d = 1$, $c = a = 0$, $f = 1$ and $f(x) = 1/x$, respectively, and using polynomials of degree one and two, respectively, on uniform meshes.

4. Quantitative error analysis

We now turn to the more interesting study of adaptivity and quantitative error control.

a) Note first that one of the ideas of adaptivity is that local singularities in the solution like the ones caused by the data $f = 1/x$ above, can be compensated for by local mesh refinements, so that the rate of convergence in terms of the number of unknowns is the same as for a problem with smooth solution. Check whether or not this is the case by measuring the error and/or the derivative error as a function of the number of unknowns in case of the model problem (1) with $d = 1$, $c = a = 0$, and $f = 1$ and $f = 1/x$, respectively, and homogeneous Dirichlet type boundary conditions. Note that in this case you cannot specify the number of unknowns. Instead use two different tolerances and check the resulting number of unknowns on the final mesh (e. g., by checking `length(u)`). Assume that error $= CN^k$, where C and k are unknown constants and N the number of unknowns, and determine k from the two measurements. The question is whether or not k is (roughly) the same for $f = 1/x$ as for $f = 1$?

b) Check the estimated error versus the true error for a number of example problems of your own choice for which you know the exact solution. Create such problems by inserting the solution (of your choice) into the equation and then use the resulting function as `force`. Recall that, ideally, the ratio between the estimated error (called `error` in `adfem`) and the true error (`exacterror` in `adfem`) should not be less than one (for reliability) and not much larger than one (for efficiency). What one could hope for is a ratio between one and 2–3, say, on the final mesh, at least if the tolerance is set fine enough. Note, however, that in the present `adfem` code the "true error" is estimated using a rather simple quadrature rule which may sometimes result in under-estimation of the true error, and consequently a somewhat larger ratio. On other occations, the residual and the stability constant may be under-estimated by the quadrature resulting in too small an estimated error and possible under-estimation of the true error.

5. Selected applications

Choose one of the following problems:

5.1. **Värmeledning genom en vägg.** En vägg består av tre skikt: ytterst 3 cm trä, värmekonduktivitet 0.14 W/(m K), i mitten 25 cm isoleringsmaterial, värmekonduktivitet 0.024 W/(m K). Uppgiften är att bestämma värmeflödet genom väggen då innertemperaturen är $T_{\text{in}} = 21^\circ$ C och yttertemperaturen $T_{\text{out}} = -10^\circ$ C. Bestäm också en effektiv värmeöverföringskoefficient k sådan att värmeflödet genom väggen kan skrivas $k(T_{\text{in}} - T_{\text{out}})$.

Tips: a) Härled randvärdesproblemet

$$-(d(x)u'(x))' = 0, \quad -1 < x < 1; \quad u(-1) = g_0, \; u(1) = g_1,$$

där samtliga storheter är dimensionslösa.

b) Antag att $d(x) = \epsilon(1 + (\beta x)^2)$ och lös problemet för hand. (Detta görs för att vi ska få ett icke-trivialt testfall för den numeriska beräkningen.)

c) Lös randvärdesproblemet med `adfem`. För att bestämma värmeflödet $q(x) = -d(x)u'(x)$ gör vi så här: Nodvärdena av den beräknade temperaturen finns i vektorn `u`, nätstorleken finns i vektorn `h`. Om vi bildar differensen mellan konsekutiva u-värden med programmet `diff` (`>> help diff`) och dividerar med h-värdena, så får vi temperatur-derivatan u'. Den är en styckvis konstant funktion och fås alltså i vektorn `diff(u)./h`. Värdena av värmeledningskoefficienten i mittpunkterna på elementen finns i vektorn `d`. Värmeflödet ges alltså av `q=-d.*diff(u)./h`. Testa först med ett specialfall där exakta lösningen är känd, dvs välj $d(x)$ så att $u(x)$ kan beräknas symboliskt som t ex i (b).

5.2. **A convection-diffusion problem.** Consider a point-source of pollution located at $x = 0$ (in a river, for instance) with diffusion coefficient 0.05. Compare the profiles of the concentration of the pollutant at "wind/water velocity" 0.05 and 1, respectively, (in the "downstream" direction of the positive x-axis). Use boundary conditions $u = 0$ in the upwind/upstream direction, at $x = -2$ say, and boundary condition $Du = 0$ in the downwind/downstream direction, at $x = 5$, say.

5.3. **Moment in a spring-board.** Consider a spring-board (trampoline) of length 2.0 with linearly varying thickness between 0.1 at the clamped end (with $u = Du = 0$) to 0.05 at the free end, and assume that the stiffness is proportial to the thickness with constant 10^{-4}. Let the spring-board be loaded by a person with weight 75 at the free tip. Compute the deflection and the moment of the spring-board as functions of x. To obtain the moment aD^2u from the MATLAB deflection vector u, type

```
mom=a.*diff([0 diff(u)])/h/h;
plot(x(1:length(mom))+h*ones(size(mom))/2,mom)
```

5.4. **Reaktion-diffusion i porös katalysator-pellet.** Ett ämne i gasform diffunderar in i en porös katalysator-pellet och omsätts vid isoterma och stationära betingelser. Vi får reaktions-diffusions ekvationen,

$$-D\Delta c = -kc \quad \text{i } V; \qquad c = c_f \quad \text{på } S, \tag{3}$$

med koncentrationen c [mol/m^3] i katalysatorn, den konstanta koncentrationen c_f i omgivningen, diffusionskonstanten D [m^2/s], reaktionshastighetskonstanten k [s^{-1}], storleken L [m], och där $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$.

a) Inför dimensionslösa variabler $(\hat{x}, \hat{y}, \hat{z}) = (x/L, y/L, z/L)$, $\hat{c} = c/c_f$ så att (3) övergår i

$$-\hat{\Delta}\hat{c} = -\phi^2 \hat{c} \quad \text{i } \hat{V}; \qquad \hat{c} = 1 \quad \text{på } \hat{S}, \tag{4}$$

där $\hat{\Delta} = \frac{\partial^2}{\partial \hat{x}^2} + \frac{\partial^2}{\partial \hat{y}^2} + \frac{\partial^2}{\partial \hat{z}^2}$ och ϕ kallas Thiele-modulen. Uttryck ϕ i de givna storheterna.

b) För olika värden på ϕ vill vi undersöka dels hur långt in i katalysatorn reaktionen når, dels beräkna kvoten mellan den uppnådda reaktionshastigheten och den ideala reaktionshastighet man skulle få utan hindrande diffusion, dvs om $c = c_f$. Denna kvot kallas effektivitetsfaktorn och ges alltså av

$$\eta = \frac{\int_V kc\,dV}{kc_f V} = \frac{\int_V c\,dV}{c_f V} = \frac{1}{\hat{V}} \int_{\hat{V}} \hat{c}\,d\hat{V}. \tag{5}$$

Visa detta! Visa också att

$$\eta = \frac{1}{\phi^2} \frac{1}{\hat{V}} \int_{\hat{S}} \frac{\partial \hat{c}}{\partial \hat{n}}\,d\hat{S}. \tag{6}$$

Tips: (4) och Gauss sats.

c) Speciell geometri: (i) platta

$$\begin{aligned} -\frac{d^2 c^*}{dr^2} &= -\phi^2 c^*, \quad -1 < r < 1, \\ \hat{c}(-1) &= \hat{c}(1) = 1, \end{aligned} \tag{7}$$

eller på grund av symmetri ($\hat{c}$ är en jämn funktion, dvs $\hat{c}(-r) = \hat{c}(r)$, och för varje jämn funktion gäller att derivatan är udda och därför är derivatan $= 0$ i origo)

$$\begin{aligned} -\frac{d^2 \hat{c}}{dr^2} &= -\phi^2 \hat{c}, \quad 0 < r < 1, \\ \frac{d\hat{c}}{dr}(0) &= 0, \ \hat{c}(1) = 1. \end{aligned} \tag{8}$$

(ii) cirkulär cylinder

$$\begin{aligned} -\frac{1}{r}\frac{d}{dr}\left(r\frac{d\hat{c}}{dr}\right) &= -\phi^2 \hat{c}, \quad 0 < r < 1, \\ \frac{d\hat{c}}{dr}(0) &= 0, \ \hat{c}(1) = 1. \end{aligned} \tag{9}$$

(iii) sfär

$$\begin{aligned} -\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\hat{c}}{dr}\right) &= -\phi^2 \hat{c}, \quad 0 < r < 1, \\ \frac{d\hat{c}}{dr}(0) &= 0, \ \hat{c}(1) = 1. \end{aligned} \tag{10}$$

(Den dimensionlösa variabeln r borde egentligen tecknas $\hat{r}$, men det blir alltför svårläst.)

d) Lös (8) exakt och bestäm och plotta η som funktion av ϕ.

e) Lös (10) exakt och bestäm och plotta η som funktion av ϕ. Tips: sätt $u(r) = r\hat{c}(r)$.

f) Lös (8), (9), (10) numeriskt med programmet `adfem` i MATLAB och bestäm η för några olika värden på ϕ (små och stora). Jämför med exakta lösningar i (d) och (e). Vilken geometri är fördelaktigast? Tips: Funktionsvärdena $\hat{c}(r_i)$ till lösningen finns i vektorn `u`, beräkningsnoderna r_i i vektorn `x` och nätstorleken h_i i vektorn `h`. Kommandot `integrate(u.*x,h)` beräknar integralen $\int_0^1 \hat{c}(r) r\, dr$.

g) Varianter för den ambitiöse teknologen. Gör sammalunda för någon av följande modifikationer. (i) Robins randvillkor:

$$\frac{d\hat{c}}{dr}(0) = 0, \ \frac{d\hat{c}}{dr}(1) + \nu(\hat{c}(1) - 1) = 0. \tag{11}$$

(ii) Tät kärna:

$$\begin{aligned} &-\frac{1}{r^2}\frac{d}{dr}\Big(r^2\frac{d\hat{c}}{dr}\Big) = -\phi^2\hat{c}, \quad \omega < r < 1, \\ &\frac{d\hat{c}}{dr}(\omega) = 0, \ \hat{c}(1) = 1. \end{aligned} \tag{12}$$

(iii) Förgiftning:

$$\begin{aligned} &-\frac{1}{r^2}\frac{d}{dr}\Big(r^2\frac{d\hat{c}}{dr}\Big) = -\phi^2\hat{c}, \quad 0 < r < \omega, \\ &-\frac{1}{r^2}\frac{d}{dr}\Big(r^2\frac{d\hat{c}}{dr}\Big) = 0, \quad \omega < r < 1, \\ &\frac{d\hat{c}}{dr}(0) = 0, \ \hat{c}(1) = 1. \end{aligned} \tag{13}$$

h) Generalisera ovanstående (tät kärna, förgiftning, etc) genom att betrakta den mer realistiska situationen då diffusionskonstanten D och reaktionshastighetskonstanten k inte är konstanta utan varierar med rumskoordinaterna.

6. Hints

To simulate point loads you may use suitable approximate delta functions of your own choice, for example,

$$\delta(x) = \frac{1}{\sqrt{\pi}\epsilon}\exp\Big(-\Big(\frac{x}{\epsilon}\Big)^2\Big),$$

or, in the case of problem 5.3,

$$\delta(x) = \frac{2}{\sqrt{\pi}\epsilon}\exp\Big(-\Big(\frac{x-2}{\epsilon}\Big)^2\Big),$$

with ϵ suitably small positive. Note, however, that you may not take ϵ too small because, if the tolerance is not small enough, then the mesh may be too coarse for the quadrature rule to see the interval where δ is large.