MTH3011 PARTIAL DIFFERENTIAL EQUATIONS

Computer laboratory sheet 2 – MATLAB version

Objectives of the session

This computer laboratory session involves using the forward-difference (Euler) method for the numerical solution of a partial differential equation. It is assumed that you are familiar with the material introduced in Computer laboratory sheet 1 on ordinary differential equations.

The main objectives of the session are:

- to illustrate the basis of a simple 'time-stepping' numerical method for determining an approximate solution of the 'heat equation';
- to illustrate how the time and second-order spatial derivatives are approximated when seeking a numerical solution of a partial differential equation;
- to calculate some numerical solutions at a grid of time and spatial points;
- to compare the numerical solution with a known exact solution (and calculate the errors);
- to illustrate some of the common procedures (including numerical calculations using formulas, plotting and printing two-dimensional solutions) that will be used for all of the numerical techniques in later sessions.

Specifically, we will use the forward-difference method in time and centred-difference method in space to determine an approximate solution of the 'heat equation':

$$\frac{\partial u}{\partial t} = K\frac{\partial^2 u}{\partial x^2} \quad \text{over} \quad 0<x<1 \quad \text{for} \quad t>0, \text{ where } K>0,$$

using the initial condition $u(x,0)=\sin(\pi x)$ and boundary conditions $u(0,t)=0$ and $u(1,t)=0$. This approximate numerical solution will also be compared with the exact solution for this problem, which can be determined using the 'separation of variables' method to be

$$u(x,t)=\sin(\pi x)\exp(-K\pi^2 t).$$

Forward-difference method

As on Computer laboratory sheet 1, the forward-difference method in time for the partial differential equation above is based on approximating the time derivative on the left-hand side by a formula that involves the approximate values of u at $t_k = k\Delta t$ and $t_{k+1} = (k+1)\Delta t$, similar to that used for Euler's method for the solution of an ordinary differential equation. This gives that

$$\frac{\partial u}{\partial t}(x_i,t_k) \cong \frac{u_i^{k+1}-u_i^k}{\Delta t}.$$

Similarly, the spatial derivative on the right-hand side of the differential equation is approximated by the centred spatial difference formula

$$\frac{\partial^2 u}{\partial x^2}(x_i,t_k) \cong \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{(\Delta x)^2}.$$

Combining these two approximations then gives that an approximate solution u_i^k for the exact solution $u(x_i,t_k)$, at $x_i = i\Delta x$ and $t_k = k\Delta t$, will satisfy the '*difference equation*'

$$\frac{u_i^{k+1}-u_i^k}{\Delta t} = K\left(\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{(\Delta x)^2}\right).$$

This equation can then be rearranged into the '*recurrence relation*' that gives the u_i^{k+1} values at each point x_i in terms of some of the approximate values at t_k, using

$$u_i^{k+1} = u_i^k + K\,\Delta t\left(\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{(\Delta x)^2}\right) \quad \text{for} \quad i = 1,2,3,\ldots,(n_x-1) \quad \text{and} \quad k = 0,1,2,3,\ldots,$$

starting from the given initial condition $u_i^0 = u(x_i,0) = \sin(\pi x_i)$ and using the boundary conditions $u_0^k = 0$ and $u_{nx}^k = 0$.

Another convenient way to represent this process is to recognise that all of the u_i^{k+1} values at the next time step t_{k+1} are obtained through a *linear* operation on all of the (known) u_i^k values. Based on the recurrence relation above, this process can be written in the form of a linear operation

$$u_i^{k+1} = su_{i-1}^k + (1-2s)u_i^k + su_{i+1}^k \quad \text{for} \quad i = 1,2,3,\ldots,(n_x-1) \quad \text{and} \quad k = 0,1,2,3,\ldots,$$

in terms of the fixed parameter $s = K\,\Delta t/(\Delta x)^2$. As noted above, the remaining two values u_0^{k+1} and u_{nx}^{k+1} are determined by the specified boundary conditions.

In this class, this problem will be solved when $K = 0.1$ using $\Delta x = 0.2$ with a variety of values of Δt.

Class exercise ***(to be completed during the laboratory session for assessment)***

This version of this sheet describes how the numerical calculations and plotting can be performed using the MATLAB package. *It is assumed that students using this version of the sheet are already familiar with the key commands in MATLAB so the basics of using the package will not be covered – all other students are advised to use the Excel version.*

Download the sample template file for this week

1. Log in to the my.monash portal using your favourite browser, start Moodle and then go to the 'Teaching materials/Computer laboratory 2 sheets/MATLAB version' sub-section for this unit.
2. Click on the link for 'matlabsheet2.zip' and then click 'Open' to reveal the file `labsheet2.m` in Winzip. Double-click on this to open the file in MATLAB and then ***immediately save the file*** as `labsheet2.m` in your `MTH3011` directory.

Familiarise yourself with the grid of space and time values at which the approximate solution will be found

3. The value $n_x = 5$ will be used initially for this problem, so the `linspace` command (and the transpose operator `'`) is used in lines 6-8 to create a ***column*** vector `x` containing 6 equally-spaced values of x over $[0,1]$, with a spatial step `dx` of $\Delta x = 0.2$.
4. The value $n_t = 25$ will be used initially for this problem, so the `linspace` command is used in lines 10-12 to create a ***row*** vector `t` containing 26 equally-spaced values of t over $[0,2.5]$, with a time step `dt` of $\Delta t = 0.1$.
5. Notice also that $s = K\,\Delta t/(\Delta x)^2$ for this problem is stored at line 17 of the template file because, as noted above, that value simplifies some of the formulae to be used.

Set up the initial and boundary conditions for the solution of the PDE

6. In line 21 of the template file, remove the `%` sign and then modify this line so that it stores the initial values `u_approx(i,1)` $= \sin(\pi x_{i-1})$ for $i = 1,\ldots,(n_x+1)$.

7. In lines 23 and 24 of the template file, remove the `%` sign and then modify those lines so that they store the boundary values `u_approx(1,k)` $= 0$ for $k = 1, \ldots, (n_t + 1)$.

8. Save these additional commands in your file `labsheet2.m`.

Use the forward-difference formula to generate the approximate solutions at every time step

9. At line 30 of the template file, remove the `%` sign and then modify this line so that at each appropriate value of `i` and `k` it stores the approximate values of `u_approx(i,k+1)` based on the recurrence relation given on page 2 of this sheet.

Graph the approximate solution as a surface plot

10. Modify the `mesh` command at line 39 so that it plots all of the values of `u_approx` as a surface plot over the entire (x,t) domain. Note that some initial labelling of the plot has been done in the template file, but you can modify or enhance it if you wish.

11. Save your edited m-file as `labsheet2.m` in your `MTH3011` directory then type `labsheet2` in the Command Window to run the commands in the file. Type `Ctrl-C` to stop the file at the first `pause` statement in line 49 and then inspect the plotted values of `u_approx` to see whether they seem to be sensible (bearing in mind the properties of the given exact solution).

12. Once your results look reasonable, proceed to the next step below – otherwise check your file to make sure that steps 6, 7 and 9 have been implemented correctly. Note that the values of `u_approx` are stored in an Excel file at lines 47-48 in case you want to inspect them in that manner instead.

Calculate the exact solution at the same times and positions, and then graph those values

13. Modify line 53 of the template file so that it calculates the exact solution at all of the points in the domain, and then modify line 56 (as in step 10 above) so that it is plotted as a separate figure.

14. Save these additional commands in your file `labsheet2.m` and then run the additional commands by typing `labsheet2` in the Command Window. Type a space when the first `pause` statement is reached at line 49, and then type `Ctrl-C` to stop the file at the second `pause` statement in line 64.

Calculate and graph the errors in the approximate solution

15. Modify line 68 of the template file so that it uses the exact solution to identify the errors ('approximate-exact') in the approximate solution `u_approx` at all of the points in the domain. If everything else is correct, this should be able to be done by a single command which subtracts two matrices.

16. Modify line 71 (in a similar manner to above) so that the errors plotted as a separate figure.

17. Save these additional commands in your file `labsheet2.m` and again run the additional commands by typing `labsheet2` in the Command Window. Type a space each time the first two `pause` statements at lines 49 and 64 are reached.

18. Once your results look reasonable, comment on the distribution of the errors across the domain and then proceed to the next step below – otherwise type `Ctrl-C` to stop the file and then make sure that steps 13-16 have been implemented correctly.

Repeat the calculations for a larger value of the time step $\Delta t = 0.2$

19. At lines 80-82 of the template file, a new row vector `t2` is stored containing 26 equally-spaced values of t over $[0,5]$, with a time step `dt2` of $\Delta t = 0.2$

20. At line 86, define the new scalar variable `s2` that corresponds to $s = K\,\Delta t / (\Delta x)^2$ for the same spatial grid `x` as previously but uses this new value $\Delta t = 0.2$.

21. In a similar manner to steps 6-10 above, store the initial and boundary conditions for this case into the matrix `u_approx2` (lines 90-92), add the recurrence relation (by writing line 98 in terms of `s2`) to calculate the approximate solution over the domain and then display those results as a surface plot (by modifying line 107).

22. Save these additional commands in your file `labsheet2.m` and then run the additional commands by typing `labsheet2` in the Command Window. Type a space three times to get to the fourth `pause` statement at line 112.

23. Once you are satisfied that your calculations are correct, compare them to those obtained earlier for $\Delta t = 0.1$ and then proceed to the next step below.

Repeat the calculations for an even larger value of the time step $\Delta t = 0.5$

24. Copy your modified commands from lines 80-112 and paste them into the file just before end, at line 116 of the template.

25. Use `nt3=30` in a new row vector `t3` which contains 31 equally-spaced values of t over $[0,15]$, with a time step `dt3` of $\Delta t = 0.5$. Also calculate the scalar variable `s3` that corresponds to the value of $s = K\Delta t/(\Delta x)^2$ for this case.

26. Also modify the initial and boundary conditions appropriately to store them into the matrix `u_approx3` for this number of time steps `nt3=30`. Modify the recurrence relation (in terms of `s3`) to calculate the approximate solution over the domain and then display those results as a surface plot in the same manner as previously.

27. Comment upon the features of the numerical solutions in this case. Are they what you expected?

MAP/map
1/3/16