MTH3011 PARTIAL DIFFERENTIAL EQUATIONS

Computer laboratory sheet 1 – MATLAB version

Objectives of the session

This computer laboratory session aims to compare some different methods for the numerical solution of an initial-value problem for an ordinary differential equation.

The main objectives of the session are:

- to illustrate three important types of 'time-stepping' numerical methods for determining an approximate solution of an ordinary differential equation (prior to applying similar methods to partial differential equations later in the unit);

- to compare the accuracy of the three numerical methods by solving an ordinary differential equation with a known exact solution and calculating the errors;

- to confirm how the accuracy of each method depends on the size of the chosen time step Δt.

- to provide reinforcement on some of the common procedures (including numerical calculations using formulas, plotting and printing) that will be used for all of the numerical techniques in later sessions.

Specifically, we will compare the backward-difference and centred-difference methods with Euler's method for the approximate solution of the ordinary differential equation

$$\frac{du}{dt} = 2\exp(-t^2) - 2t\,u \quad \text{for } t > 0$$

with the initial condition $u(0) = 0$. These approximate numerical solutions can also be compared with the exact solution for this problem, which can be determined using standard methods from first/second year as

$$u(t) = 2t\exp(-t^2).$$

Euler's method

Euler's method is the simplest method that can be used to determine an approximate solution u_k for the exact solution $u(t_k)$ of the differential equation above at t_k with

$$u_{k+1} = u_k + \Delta t\,[2\exp(-t_k^2) - 2t_k u_k] = u_k[1 - 2(\Delta t)t_k] + 2\Delta t\exp(-t_k^2) \quad \text{for} \quad k = 0,1,2,3,\ldots,$$

with $u_0 = u(0) = 0$ from the initial condition. This method is also known as the ***forward-difference method***.

The backward-difference method

In the ***backward-difference method*** it is assumed that the derivative at t_{k+1} can be approximated in terms of the preceding (earlier) values of the approximate solution u_k, so that

$$\frac{du}{dt}(t_{k+1}) \cong \frac{u_{k+1} - u_k}{\Delta t}.$$

(Recall that for Euler's method the value of du/dt at t_k was estimated using the same expression.)

The differential equation is then approximated by the '*difference equation*'

$$\frac{u_{k+1}-u_k}{\Delta t} = 2\exp(-t_{k+1}^2) - 2t_{k+1}u_{k+1}.$$

We can use this to calculate a sequence of approximate values for u_k by solving the equation

$$u_{k+1}[1+(\Delta t)2t_{k+1}] = u_k + 2\Delta t \exp(-t_{k+1}^2) \quad \text{for} \quad k = 0,1,2,3,...$$

for u_{k+1} in terms of u_k. This gives the '*recurrence relation*'

$$u_{k+1} = \frac{u_k + 2\Delta t \exp(-t_{k+1}^2)}{[1+(\Delta t)2t_{k+1}]} \quad \text{for} \quad k = 0,1,2,3,...$$

with $u_0 = 0$ from the initial condition.

The centred-difference method

In the **centred-difference method** it is recognised that

$$\frac{u_{k+1}-u_k}{\Delta t}$$

is a better approximation to du/dt at the midpoint $t_{k+1/2} = (t_k + \frac{1}{2}\Delta t)$ between t_k and t_{k+1}, so the differential equation is approximated by the '*difference equation*'

$$\frac{u_{k+1}-u_k}{\Delta t} = 2\exp(-t_{k+1/2}^2) - 2t_{k+1/2}[\tfrac{1}{2}(u_k + u_{k+1})]$$

where the average $\frac{1}{2}(u_k + u_{k+1})$ is used as an estimate for the value of $u(t_{k+1/2})$. Starting from the given initial condition $u_0 = 0$, the approximate values u_k of $u(t_k)$ are obtained by solving the equation

$$u_{k+1}[1+(\Delta t)t_{k+1/2}] = u_k[1-(\Delta t)t_{k+1/2}] + 2\Delta t \exp(-t_{k+1/2}^2) \quad \text{for} \quad k = 0,1,2,3,...$$

for u_{k+1} in terms of u_k, which gives the '*recurrence relation*'

$$u_{k+1} = \frac{u_k[1-(\Delta t)t_{k+1/2}] + 2\Delta t \exp(-t_{k+1/2}^2)}{[1+(\Delta t)t_{k+1/2}]} \quad \text{for} \quad k = 0,1,2,3,...$$

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 onto the computer using your usual username and Authcate password. Create a directory called MTH3011 within your private disk storage area and then go into that directory.

2. Log in to the my.monash portal using your favourite browser, start Moodle and then go to the 'Teaching materials/Computer laboratory 1 sheets/MATLAB version' sub-section for this unit.

3. Click on the link for 'matlabsheet1.zip' and then click 'Open' to reveal the file labsheet1.m in Winzip. Double-click on this to open the file in MATLAB and then *immediately save the file* as labsheet1.m in your MTH3011 directory.

Store the time values and the exact solution for the given ODE

4. Lines 6-8 of the template file use the `linspace` command to create a vector `t` containing 16 equally-spaced values of t over $[0,3]$. Copy and paste these three statements into the Command Window and run them. Confirm that the value of `dt` is equal to the time step $\Delta t = 0.2$.

5. At line 10 the `format` command is used to change the displayed format of all variables to 'long', so that they are displayed to more than four decimal places. (This will help us compare the various approximate values later.)

6. Modify line 14 to remove the `%` sign and then add an expression on the right-hand side that stores a vector `u_exact` for the given exact solution at each of the times in the vector `t`. (To perform this in a single line, as a vector operation, use component-wise multiplication, of the form 't.*t', to determine t^2 in the formula.)

Use Euler's method to determine an approximate solution of the given ODE at the same times

7. The vector `u_euler` will be used to store the values of the approximate solution based on using Euler's method at each of the times in the vector `t`. At line 18, remove the `%` sign and set the first component of `u_euler` equal to the initial value $u(0) = 0$ for this problem.

8. At lines 19-21, a `for` loop of the form `for k=1:nt` is used to calculate the other 15 components of `u_euler` when `nt=15`. Modify line 20 so that it calculates each `u_euler(k+1)` value in terms of its value `u_euler(k)` at the previous time step. Note that the expression should also include the value of `dt` that was stored earlier, at line 8 of the template file.

9. At line 22 the vector `error_euler` that contains the values of the errors (ie 'approximate-exact') at each time step is stored, based on the values of `u_euler`. Copy and run the commands in lines 14-22 and check that `error_euler(16)=-0.000631`.... What do you notice about these errors? At which time are they largest?

Use the backward-difference and centred-difference methods to perform similar calculations

10. At line 28, store the vector `u_back` that contains approximate values based on using the formula for the *backward-difference method* for the differential equation above. Store also a vector `error_back` which contains the errors in these values.

11. At line 36, store the vector `u_cent` that contains approximate values based on using the formula for the *centred-difference method* for the differential equation above. Store also a vector `error_cent` which contains the errors in these values.

12. Save your edited m-file as `labsheet1.m` in your `MTH3011` directory then type `labsheet1` 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 40 and then display and inspect the stored values in all of the vectors above to ensure that they seem to be sensible.

13. Which of the three numerical methods do you believe to be the most accurate? Why might that have been anticipated even if we had not been able to calculate the actual errors from knowing the exact solution to the differential equation?

Plot the results and errors for each of the methods above

14. The `plot` command at line 45 will graph the exact solution `u_exact` for $u(t)$ by joining the values with line segments and marking the position of each data point with the symbol 'o'. Modify line 45 to plot the three different approximate solutions `u_euler`, `u_back` and `u_cent` on the same graph, using different symbols for the data points in each case.

15. The `plot` command at line 54 (for figure 2) graphs the *errors* in the approximate solution `error_euler` for $u(t)$ by joining the values with line segments and marking the position of each data point with the symbol '+'. Modify line 54 to also plot the *errors* for the other two approximate solutions `u_back` and `u_cent` on the same graph, using different symbols for the data points in each case. Comment on your results in relation to your expectations at step 13 above.

16. Save your edited m-file as `labsheet1.m` in your `MTH3011` directory then type `labsheet1` in the Command Window to run the commands in the 'm-file'. Type a space when the first `pause` statement is reached at line 40, and then type `Ctrl-C` to stop the file at the second `pause` statement in line 59. Repeat this until both of your modified plots at steps 14 and 15 above are working correctly.

Save the values of the approximate solution to a file

17. Save the values of the exact solution, all of the calculated values and each of the errors as columns in an Excel file called `labsheet1.xls`. Modify that file to include a heading that identifies each column then show the values of t to one decimal place and all of the other values to six decimal places.

Compare the approximate solutions for another value of Δt

18. In lines 68-70, the `linspace` command is used to create a vector `t2` containing 31 equally-spaced values of t over $[0,3]$, with spacing '$\Delta t = 0.1$', with time step `dt2=t2(2)-t2(1)`. Copy and paste these statements into the Command Window and confirm that the value of `dt` is equals $\Delta t = 0.1$.

19. Repeat all of the calculations above for this smaller time-step, for example storing the results in the vectors `u_euler2`, `u_back2`, `u_cent2` and `u_exact2` and the calculated errors in the vectors `error_euler2`, `error_back2` and `error_cent2`.

20. Type `clear` to clear the values of all variables, then run the 'm-file' `labsheet1` in the Command Window from the start up to the third `pause` statement at line 100. (You can do this by typing a space twice to get past the `pause` statements at lines 40 and 59, followed by `Ctrl-C` to stop the file at the third `pause` statement at line 100.) Display and inspect the values of `u_euler2`, `u_back2`, `u_cent2` and `u_exact2` to ensure that they seem to be sensible. (If they seem too large, correct any errors in your calculations at lines 80, 88 and 97 of the template file.)

Compare the errors obtained for different values of Δt

21. Choose some typical values of t, for example $t = 0, 1, 2, 3$, and compare the size of the errors obtained using Euler's method for '$\Delta t = 0.1$' with those obtained above for '$\Delta t = 0.2$'. By roughly what factor do they differ? Repeat this for the backward difference and centred-difference methods.

22. Once your values seem sensible, modify the plot commands at lines 105 and 114 (so they are similar to those used for '$\Delta t = 0.2$' at steps 14 and 15 above), save and run the 'm-file' and then check the graphs by proceeding to the *fourth* pause statement at line 119. Type `Ctrl-C` to stop the file at that point.

23. At line 124 the scaled errors (divided by Δt) for the approximate values based on Euler's method are plotted for Euler's method when '$\Delta t = 0.2$'. Modify this plot to include the scaled errors for '$\Delta t = 0.1$' as well. Modify the similar plots for the other two methods at lines 131 and 138 so they include '$\Delta t = 0.2$' as well. Before running the 'm-file', what you would expect these graphs to look like?

Play it again Sam…

24. Type `clear` to clear the values of all variables, then run the complete 'm-file' `labsheet1` in the Command Window from start to finish, typing a space each time it stops at a `pause` statement. What do you conclude from the results shown on all seven graphs?

MAP/map
28/4/16