Mech 221 Mathematics Component

Differential Equations

Brian Wetton

www.math.ubc.ca/∼wetton

Lectures 10-13

Outline

Outline

Numerical Approximation of Differential Equations

Consider the first order IVP for $y(t)$:

$$\frac{dy}{dt} = f(t, y)$$

with $y(0) = y_0$ given

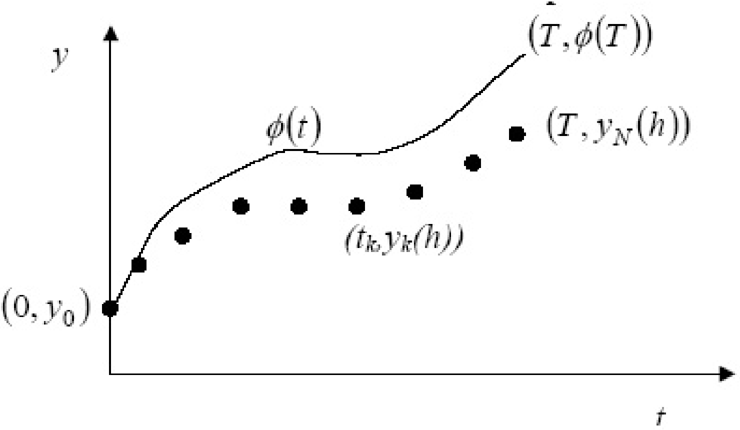
Let the *exact* solution of the problem be $y(t) = \phi(t)$.

Let T be a final time of interest. Divide $[0, T]$ into N equal subdivisions of length $h = T/N$. The ends of the subintervals are $t_k = kh$ for $k = 0 \ldots N$. We want to find approximate solutions y_k at these values of t:

$$y_k \approx \phi(kh)$$

Numerical Approximation of Differential Equations

Diagram



Numerical Approximation of Differential Equations

Notes

- y_0 is given by the initial data.
- We need a method to find the other y_k (the numerical approximation method).
- The method should be efficient (as few computations as possible to get the desired accuracy in the approximate solutions).
- We expect that as $N \to \infty$ ($h \to 0$) the solutions will become more accurate (convergence).

Euler's Method

Derivation

Consider

$$\frac{dy}{dt} = f(t, y)$$

On the grid of points we are tring to compute, we could approximate $\frac{dy}{dt}$ using the forward difference formula:

$$\frac{dy}{dt}(kh) \approx \frac{y_{k+1} - y_k}{h}$$

Thus, the DE reads approximately

$$\frac{y_{k+1} - y_k}{h} \approx f(kh, y_k)$$

Rewriting, multiplying by h and moving all y_k terms to the left gives

$$y_{k+1} = y_k + hf(kh, y_k)$$

Euler's Method-II

Method

$$y_k = y_{k-1} + hf((k-1)h, y_{k-1})$$

Since y_0 is known (initial data) this formula can be used to determine y_1. Then y_1 can be used to determine y_2, etc.

- All y_k for $k > 0$ can be determined iteratively
- This is an efficient process
- This technique is known as Euler's Method

Euler's Method-III

Example 1

Approximate the solution of

$$y' = y$$

with $y(0) = 1$ at $T = 0.1$ using one step of Euler's Method. Compare to the exact solution.

Euler's Method-IV

Example 2

Approximate the solution of

$$y' = \sqrt{y^2 + t^2}$$

with $y(0) = 1$ at $T = 0.2$ using 2 steps of Euler's Method with step size $h = 0.1$.

Euler's Method-V

Accuracy Study-A

Consider

$$y' = y$$

with $y(0) = 1$. Do one step of Euler's Method for different values of h and compare to the exact solution. Is there a pattern to the one-step (local) errors?

Euler's Method-VI

Accuracy Study-A (cont.)

Euler's Method-VII

Accuracy Study-B

Consider

$$y' = y$$

with $y(0) = 1$. Compute using Euler's Method to $T = 1$ using different values of h and compare to the exact solution. Is there a pattern to the fixed-time (global) errors?

Euler's Method-VIII

Accuracy Study-B (cont.)

Euler's Method-IX

Accuracy Summary

- The local error (error after one step) is $O(h^2)$, that is, bounded by a constant times h^2.
- The global error (error at a fixed time T) is $O(h)$, that is, bounded by a constant times h.
- Can prove these results are true when computing smooth solutions.
- Thus, Euler's Method converges (gets closer and closer to the exact solution as $h \to 0$).

Euler's Method for a System

There is nothing special about scalar equations for the use of Euler's method, or the higher order methods we will consider in a couple of lectures. If $\underline{y}$ had n components and $\underline{f}(t, \underline{y})$ was a vector function with n components, and the vector $\underline{y}(0) = \underline{y}_0$ were given, solutions of

$$\underline{y}' = \underline{f}(t, \underline{y})$$

can be approximated by

$$\underline{y}_k = \underline{y}_{k-1} + h\underline{f}((k-1)h, \underline{y}_{k-1})$$

Start with $\underline{y}_0 = \underline{y}(0)$, the resulting numerical approximation is $\underline{y}_k \approx \underline{\phi}(kh)$.

Euler's Method for a System-II

Example

Consider the DE system

$$\begin{aligned} \frac{dx}{dt} &= x + y + t \\ \frac{dy}{dt} &= x - y \end{aligned}$$

with initial data $x(0) = 1$, $y(0) = 0$. Use Euler's method with 2 steps of size $h = 0.1$ to approximate the solution at t=0.2 (both $x(0.2)$ and $y(0.2)$.

Euler's Method for a System-III

Example (cont.)

Outline

Problems with Numerical Methods

An Example

Consider the IVP:

$$y' = y^2$$

with $y(0) = 1$. In an earlier lecture we found the solution

$$y(t) = \frac{1}{1-t}$$

Note that the solution does not exist past $t = 1$.

Problems with Numerical Methods-II

An Example (cont.)

Approximate

$$y' = y^2$$

with $y(0) = 1$ using 2 steps of Euler's Method with $h = 1$.

Problems with Numerical Methods-III

An Example (cont.)

Approximations of the solution $y(2)$ with decreasing h are shown below:

h	$y(2)$ approx.
1	6
1/2	24.5
1/4	3202

The approximations of $y(2)$ exists for all h and $\rightarrow \infty$ as $h \rightarrow 0$, but that is the only indication that the solution does not exist at $t = 2$.

Problems with Numerical Methods

Summary

- The numerical method never gives the exact answer.
- In all practical cases, you do not know the exact solution, so the numerical solution must be used to try and assess its accuracy.
- As the last example shows, this can be difficult.

Problems with Euler's Method

An Example

Consider

$$y' = -100y$$

with $y(0) = 1$. The exact $y(1) = e^{-100}$, very small. Use the trick from the last lecture to quickly compute approximations of $y(1)$ using

1. $h = 0.1$
2. $h = 0.01$
3. $h = 0.001$

Problems with Euler's Method

An Example (cont.)

Problems with Euler's Method

Summary

- Euler's Method does not accurately compute stiff problems (problems with large decay rates) unless the time step is very small (inefficient).
- Euler's Method is not very accurate (inefficient).
- Euler's Method is only for first order equations

Higher Order Differential Equations

Example 1

Consider the second order, homogeneous, linear differential equation for $y(t)$:

$$y'' + a(t)y' + b(t)y = 0$$

with the functions $a(t)$, $b(t)$, and the Initial Values $y(0)$ and $y'(0)$ given. We'll look at second order problems in a few lectures.

Q: What would we do if we wanted to solve this problem numerically?

A: Convert to a two-component first order system, then apply Euler's Method to the system

Higher Order Differential Equations-II

Example 1 (cont.)

Convert

$$y'' + a(t)y' + b(t)y = 0$$

to a first order system.

Higher Order Differential Equations-III

Example 2

Convert the third order equation for $x(s)$ below into a first order system:

$$\frac{d^3x}{ds^3} + e^s \frac{dx}{ds} + x^2 = 0$$

Do not try to solve the system.

Higher Order Differential Equations-IV

Summary of Conversion to First Order System

- If the equation has derivatives of order n, convert it into a system with n components.
- The n components are the original function and its derivatives up to order $n-1$.
- The derivatives of the first $n-1$ components are matched simply to the values they should have.
- The derivative of the last component is determined by the original equation.
- Systems of higher order equations can also be turned into (larger) first order systems in the same way.

Higher Order Differential Equations-V

Example 3

Consider the differential equation for $x(t)$:

$$\ddot{x} + x^2 = t$$

with $x(0) = 1$ and $\dot{x} = 2$.

1. Write the equation as first order system.
2. Apply two steps of Euler's Method with step $h = 0.1$ to estimate $x(0.2)$ and $\dot{x}(0.2)$.

Higher Order Differential Equations-VI

Example 3 (cont.)

Outline

Richardson Extrapolation of Euler's Method

Formula

Suppose we compute approximations of $y(T)$ with Euler's Method with decreasing step sizes h. Denote the values by Y_h. We know

$$Y_h \approx y(T) + Ch$$

for some constant C (first order convergence).

Going back to the formula from a few lectures ago, or reproducing the algebra, we can see that

$$2Y_h - Y_{2h}$$

gives a better approximation of $y(T)$.

Richardson Extrapolation of Euler's Method-II

Example

Apply Richardson Extrapolation to the Euler Method calculation of $y(1) = e^1$ as the solution of

$$y' = y \quad \text{with } y(0) = 1$$

h	Euler	(error)
1/2	2.25	(0.4683)
1/4	2.441406	(0.2769)
1/8	2.565785	(0.1525)
1/16	2.637928	(0.0804)
1/32	2.676990	(0.0413)

Unfortunately, Richardson extrapolation is not very efficient.

Other Second Order Methods

Improved Euler Method

Improved Euler Method:

1. Start with t_k and $y_k \approx \phi(t_k)$
2. Compute the stages:

$$\begin{aligned} K_1 &= f(t_k, y_k) \\ K_2 &= f(t_k + h, y_k + hK_1) \end{aligned}$$

3. Compute y_{k+1}:

$$y_{k+1} = y_k + \frac{h}{2}(K_1 + K_2)$$

IE has local errors $O(h^3)$ and global errors $O(h^2)$.

Other Second Order Methods-II

Improved Euler Method Example

Use one step of Improved Euler to approximate $y(0.2)$ where $y(t)$ solves

$$y' = 3 - t + y$$

with $y(0) = 1$.

Other Second Order Methods-III

Modified Euler Method

Improved Euler Method:

1. Start with t_k and $y_k \approx \phi(t_k)$
2. Compute the stages:

$$
\begin{aligned}
K_1 &= f(t_k, y_k) \\
K_2 &= f(t_k + \frac{h}{2}, y_k + \frac{h}{2} K_1)
\end{aligned}
$$

3. Compute y_{k+1}:

$$y_{k+1} = y_k + hK_2$$

ME also has local errors $O(h^3)$ and global errors $O(h^2)$. ME and IE have similar performance.

Other Second Order Methods-IV

Modified Euler Method Example

Use one step of Modified Euler to approximate $y(0.2)$ where $y(t)$ solves

$$y' = 3 - t + y$$

with $y(0) = 1$.

Other Second Order Methods-V

Global Accuracy Test

Apply original Euler, IE, and ME to

$$y' = y \quad \text{with } y(0) = 1$$

h	Euler error	IE/ME error
1/2	0.4683	0.0777
1/4	0.2769	0.0234
1/8	0.1525	0.0064
1/16	0.0804	0.0017

Example consistent with second order methods having global errors $O(h^2)$.

Other Second Order Methods-VI

Accuracy Comparison to Euler's Method

Consider applying Euler's method to a problem, and one of the second order methods. An accuracy of ϵ is required out to time T. Errors from Euler are given approximately by

$$C_1 h = C_1 T/N$$

To make the error size ϵ we need to take

$$N_1 = C_1 T/\epsilon$$

(or more) steps. Each step costs one function evaluation.

Similarly, a second order method has errors approximately

$$C_2 h^2 = C_2 T^2/N^2$$

To make the error size ϵ we need to take

$$N_2 = T\sqrt{C_2/\epsilon}$$

(or more) steps. Each step costs two function evaluations.

Other Second Order Methods-VII

Accuracy Comparison to Euler's Method (cont.)

The ratio of the amount of work for original Euler to the amount of work for a second order method is given by

$$\frac{N_1}{2N_2} = \frac{C_1}{2\sqrt{C_2}}\frac{1}{\sqrt{\epsilon}}$$

If high accuracy is needed (ϵ small), original Euler will always take more computational work than a second order method.

Other Second Order Methods-VIII

Discussion

- Second order methods have local errors $O(h^3)$ and global errors $O(h^2)$.
- For sufficiently high accuracy requirements, second order methods are always more efficient than the original first order Euler Method.
- For most practical applications and accuracy requirements, fourth order methods are optimal, second order methods are pretty close.
- These second order methods are in a class of methods for approximating ODEs called Runge Kutta methods.
- Runge Kutta methods are found by matching Taylor series expansions in the solutions starting from simple forms that minimize storage and computational cost per step.

Fourth Order Methods

Fourth Order Runge Kutta

Fourth order Runge-Kutta method:

1. Start with t_k and $y_k \approx \phi(t_k)$
2. Compute the stages:

$$\begin{aligned} K_1 &= f(t_k, y_k) \\ K_2 &= f(t_k + h/2, y_k + \frac{h}{2}K_1) \\ K_3 &= f(t_k + h/2, y_k + \frac{h}{2}K_2) \\ K_4 &= f(t_k + h, y_k + hK_4) \end{aligned}$$

3. Compute y_{k+1}:

$$y_{k+1} = y_k + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

RK4 has local errors $O(h^5)$ and global errors $O(h^4)$.

Fourth Order Methods-II

Example

Use one step of RK4 to approximate $y(0.2)$ where $y(t)$ solves

$$y' = 3 - t + y$$

with $y(0) = 1$.

Fourth Order Methods-III

Example (cont.)

Outline

Stiff Problems and Implicit Methods

Example of Stiff ODE

Consider the problem

$$y' = -\lambda y \text{ with } y(0) = 1$$

where λ is large (think vector problem with a large, negative real eigenvalue).

We already know that if λ is large, then h must be very small in order for original Euler (Forward Euler) to have reasonable approximate solutions. "Reasonable" has two definitions:

A: $y_k \to 0$ as $k \to \infty$.

B: $y_k \to 0$ as $k \to \infty$ and $y_k > 0$ for all k.

Stiff Problems and Implicit Methods-II

Stable Time Steps

Determine the restriction on h that makes the problem

$$y' = -\lambda y \text{ with } y(0) = 1$$

reasonable by the two definitions. Use the trick we've used before on these constant coefficient problems to simplify the form of y_k. The resulting restriction on h is called a stability restriction. All the methods we have considered so far have similar time step restrictions.

Stiff Problems and Implicit Methods-III

Stable Time Steps (cont.)

Stiff Problems and Implicit Methods-IV

Backward Euler Method

$$\frac{dy}{dt} = f(t, y)$$

Approximate $\frac{dy}{dt}$ using the backward difference formula:

$$\frac{dy}{dt}(kh) \approx \frac{y_k - y_{k-1}}{h}$$

Thus, the DE reads approximately

$$\frac{y_k - y_{k-1}}{h} \approx f(kh, y_k)$$

Rewriting gives

$$y_k - hf(kh, y_k) = y_{k-1}$$

This is an implicit equation for y_k given y_{k-1}. In general, it is solved iteratively using Newton's method. This method is called the Backward Euler method. It has global accuracy of first order.

Stiff Problems and Implicit Methods-V

Use BE on the Stiff ODE

Determine the restriction on h that makes the problem

$$y' = -\lambda y \quad \text{with } y(0) = 1$$

reasonable by the two definitions. You will see that the BE is stable for any h. Thus, even though BE is more work per time step (implicit solve) you can use much larger time steps for stiff problems.

Stiff Problems and Implicit Methods-VI

Use BE on the Stiff ODE (cont.)

Stiff Problems and Implicit Methods-VII

Applying BE to an Example

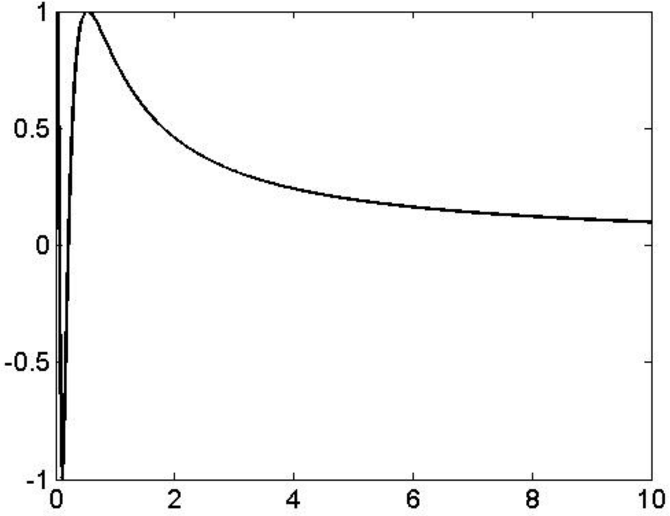
Consider the problem for $y(t)$

$$\frac{dy}{dt} = t + y^2 \quad \text{with } y(0) = 1$$

approximate $y(0.1)$ using one step of BE time stepping.

Adaptive Time Stepping

An Example



Adaptive Time Stepping-II

If you were trying to compute approximations of the last problem, you would have to use small h to resolve the solution near $t = 0$ accurately. But for larger t you could use a larger h for the same accuracy.

Adaptive Time Stepping-III

Idea of Adaptive Time Stepping

- Suppose you are using a p'th order method to approximate an ODE.
- You want the local trunction error to be smaller that ϵ at each step.
- You compute one step with h_o.
- By magic you determine the error δ you made in the solution with that step size. (The magic is estimating the error using different schemes).
- If $\delta > \epsilon$ you want to decrease h so the scheme is more accurate.
- If $\delta < \epsilon$ you want to increase h so the scheme is more efficient.
- Safety margins are built into most schemes.

Adaptive Time Stepping-IV

Idea of Adaptive Time Stepping (cont.)

MATLAB routines

There are three MATLAB routines that are standard ODE solvers. All use variable time step strategies based on error approximation.

ODE23: A second order RK scheme with an extra stage that allows for error estimation.

ODE45: A fourth order RK scheme with an extra stage that allows for error estimation.

ODE15s: A variable order implicit scheme for stiff problems. Solves the implicit systems using a Newton-type method, based on numerically computed derivates.

The Bad News

No ODE solver is perfect. All of the routines above will "fail" if applied to the simple problem for $x(t)$

$$\ddot{x} + x = 0 \quad \text{with } x(0) = 0 \text{ and } \dot{x}(0) = 1$$

(solution $x(t) = \sin t$) and computed for *very long* times.

ODE23: The approximate solution will grow slowly in time.

ODE45: The approximate solution will decay slowly in time.

ODE15s: The approximate solution will decay slowly in time.

Very specialized (symplectic) schemes for this specific problem will avoid the growth and decay, but there will still be phase drift in time. Also, no numerical method can accurately calculate solutions to chaotic problems for large times.