Ch 2 Finite Differences + Parabolic Eq.

First, some background on numerical methods for ODEs (Initial Value Problems + Boundary Value Problems : IVPs + BVPs).

IVPs

One-dimensional case

$$\begin{cases} \frac{dy}{dt} = f(t,y) \\ y(t_0) = y_0 \end{cases} \qquad \begin{array}{l} y \in \mathbb{R} \\ f : \mathbb{R}^2 \to \mathbb{R} \end{array}$$

e.g. $\frac{dy}{dt} = \lambda y$, $y(0) = 1$ (see ch. 1 notes also)

Objective: Approximate the true solution $y(t)$ at a discrete set of points t_i by discrete values y_i.

True Solution	Approximation
$y(t)$	$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \approx \begin{bmatrix} y(t_0) \\ y(t_1) \\ y(t_2) \\ \vdots \\ y(t_N) \end{bmatrix}$
continuous function	discrete vector

For an IVP we know the value of y at $t=t_0$ $[y_0]$ and we know $\frac{dy}{dt}$ at $t=t_0$ $[f(t_0,y_0)]$.

This allows us to predict the solution at a future point in time.

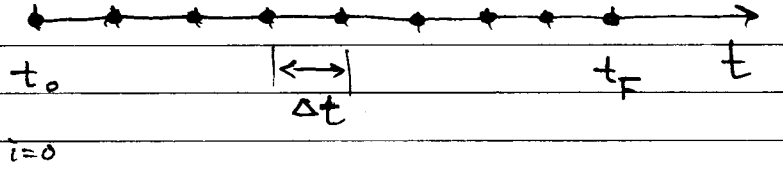
There are many numerical methods for IVPs. The simplest one is Euler's Method.

Euler's Method (also "Forward Euler")

Approximate $\left.\frac{dy}{dt}\right|_{t=t_i}$ by $\frac{y_{i+1}-y_i}{\Delta t}$, and write

$$\frac{y_{i+1}-y_i}{\Delta t} = f(t_i, y_i)$$

OR $$\boxed{y_{i+1} = y_i + \Delta t\, f(t_i, y_i)} \quad i=0,1,2,\ldots$$



Comments:

- this method is explicit — the approximation can be advanced to the next time step by using an explicit formula.
- very simple to implement numerically
- numerical stability issues are a concern [MORE LATER]

Example

Solve $\begin{cases} \frac{dy}{dt} = -5y \\ y(0) = 1 \end{cases}$

using Euler's Method:

$$y_{i+1} = y_i + \Delta t(-5y_i) = (1-5\Delta t)y_i$$

$y_0 = 1$ approximates $y(0)$

$y_1 = (1-5\Delta t)y_0 = (1-5\Delta t)$ approximates $y(\Delta t)$

$y_2 = (1-5\Delta t)y_1 = (1-5\Delta t)^2$

$\vdots$

$y_N = (1-5\Delta t)y_{N-1} = (1-5\Delta t)^N$ approximates $y(N\Delta t)$

$\vdots$

Note that the exact solution is $y(t) = e^{-5t}$

If $t_F \equiv N\Delta t$ (final time)

$$\Delta t = \frac{t_F}{N}$$

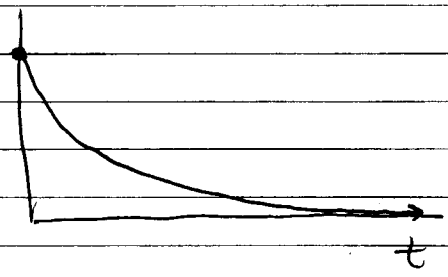
so $$y_N = \left(1 - 5\frac{t_F}{N}\right)^N$$

Note: $\lim_{N\to\infty}\left(1-\frac{5t_F}{N}\right)^N = e^{-5t_F}$

$\therefore$ Euler's Method converges to exact solution as $N\to\infty$, fixed t_F.

Further comments:

- In this example the exact solution decays to zero as $t \to \infty$.



- Note that $y_N = (1-5\Delta t)^N \to 0$ only if

$$|1-5\Delta t| < 1$$

or $-1 < 1-5\Delta t < 1$

$-2 < -5\Delta t < 0$

$0 < \Delta t < \frac{2}{5}$ ← time step restriction for numerical stability!

if $\Delta t > \frac{2}{5}$ y_N blows up!

as $N \to \infty$.

- This is a typical concern in explicit methods.

(i.e. a time step restriction for numerical stability)

A stable numerical scheme does not cause small perturbations introduced through rounding grow + dominate the solution (see section 2.6)

Backward Euler

This is an implicit numerical method closely related to the Forward Euler Method. Here

$$\frac{y_{i+1} - y_i}{\Delta t} = f(t_{i+1}, y_{i+1})$$

OR

$$\boxed{y_{i+1} = y_i - \Delta t\, f(t_{i+1}, y_{i+1})} \qquad i = 0, 1, 2, \ldots$$

Note that this is no longer an explicit formula for y_{i+1} (in general). y_{i+1} is defined implicitly. In order to advance from (t_i, y_i) to (t_{i+1}, y_{i+1}) we need to solve this equation for the unknown y_{i+1}. The effort involved depends on the form of $f(t_{i+1}, y_{i+1})$.

Example (EASY ONE)

Solve $\begin{cases} \frac{dy}{dt} = -5y \\ y(0) = 1 \end{cases}$

using Backward Euler.

$$y_{i+1} = y_i + \Delta t(-5 y_{i+1})$$

$$(1+5\Delta t) y_{i+1} = y_i$$

$$y_{i+1} = \frac{1}{(1+5\Delta t)} y_i$$

$y_0 = 1$ approximates ... $y(0)$

$y_1 = \frac{1}{(1+5\Delta t)} y_0 = \frac{1}{(1+5\Delta t)}$ $y(\Delta t)$

$y_2 = \frac{1}{(1+5\Delta t)} y_1 = \frac{1}{(1+5\Delta t)^2}$ $y(2\Delta t)$

$\vdots$ $\vdots$

$y_N = \frac{1}{(1+5\Delta t)} y_{N-1} = \frac{1}{(1+5\Delta t)^N}$ $y(N\Delta t)$

Note: $\lim_{N\to\infty} y_N = e^{-5t_F}$

Further comments

$$y_N = \frac{1}{(1+5\Delta t)^N} \to 0 \quad \text{if}$$

$$\frac{1}{|(1+5\Delta t)|} < 1$$

$|1+5\Delta t| > 1$ ← this is always true regardless of Δt (assuming $\Delta t > 0$!)

∴ there are no numerical stability restrictions on Δt.

EXAMPLE (HARDER ONE)

Solve
$$\begin{cases} \frac{dy}{dt} = -(2+\sin(y))\,y \\ y(0) = 1 \end{cases}$$

using Backward Euler

$$y_{i+1} = y_i - \Delta t\,(2+\sin y_{i+1})\,y_{i+1}$$

(solve this for y_{i+1} given y_i and Δt).

Requires numerical solution of

$$F(y_{i+1}) = y_{i+1} - y_i + \Delta t\,(2+\sin y_{i+1})\,y_{i+1} = 0$$

↓ use Newton's method, ...

Forward + Backward Euler are both <u>first order accurate</u> schemes.

$\Rightarrow$ "Global" Error $= O(\Delta t)$

"Local" Error $= O(\Delta t^2)$ = [Error at each time step if no previous error exists]

(more later...)

More accurate schemes are also common:

Fourth Order Runge-Kutta: (RK)

$$y_{i+1} = y_i + \frac{\Delta t}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right)$$

where

$$K_1 = f(t_i, y_i)$$

$$K_2 = f\left(t_i + \tfrac{1}{2}\Delta t,\ y_i + \tfrac{1}{2}\Delta t K_1\right)$$

$$K_3 = f\left(t_i + \tfrac{1}{2}\Delta t,\ y_i + \tfrac{1}{2}\Delta t K_2\right)$$

$$K_4 = f\left(t_i + \Delta t,\ y_i + \Delta t K_3\right)$$

└ This scheme is explicit.

Matlab's ode45 is based on a combined version of 4th order accurate + 5th order accurate RK (using variable step size to control the error)

[type "help ode45" at Matlab prompt for more info and other related solvers].

- The IVP methods described here can be applied generally to first order systems

$$\begin{cases} \vec{y}' = \vec{f}(t, \vec{y}) \\ \vec{y}(t_0) = \vec{y}_0 \end{cases}$$

e.g. Forward Euler:

$$\vec{y}_{i+1} = \vec{y}_i + \Delta t \vec{f}(t, \vec{y}_i)$$

· explicit vector equation

e.g. Backward Euler

$$\vec{y}_{i+1} = \vec{y}_i + \Delta t \vec{f}(t_{i+1}, \vec{y}_{i+1})$$

$$\Rightarrow \vec{F}(\vec{y}_{i+1}) \equiv \vec{y}_{i+1} - \vec{y}_i - \Delta t \vec{f}(t_{i+1}, \vec{y}_{i+1}) = 0$$

System of Nonlinear Equations for unknown $\vec{y}_{i+1}$.

EXAMPLES in Matlab

A.
$$\begin{cases} \dfrac{dy}{dt} = f(t,y) \\ y(0) = y_0 \end{cases}$$

- $f(t,y) = y + 5\cos(5t)e^t$; $y_0 = 0$
- $f(t,y) = \dfrac{1}{1+t^2} - 2ty^2$; $y_0 = 0$
- $f(t,y) = -30y^4$; $y_0 = 1$

$0 \le t \le t_F$

see
euler.m
feuler.m
beuler.m

B. Lorenz Equations

see lorenz.m , fun_lorenz.m

$$\begin{cases} y_1' = \sigma(y_2 - y_1) \\ y_2' = r y_1 - y_2 - y_1 y_3 \\ y_3' = y_1 y_2 - b y_3 \\ y_1(0) = 0 \\ y_2(0) = 1 \\ y_3(0) = 0 \end{cases}$$

$0 \le t \le t_F$

e.g. $\sigma = 10$, $b = 8/3$, $r = 28$ (vary these...)

EXACT SOLUTIONS FOR A

- $y(t) = e^t \sin(5t)$
- $y(t) = t/(1+t^2)$
- $y(t) = (1+90t)^{-1/3}$

BVPs

Recall the Sturm-Liouville problem

$$\begin{cases} \frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + q(x)u(x) + \lambda r(x)u(x) = 0 \\ \alpha_1 u(a) + \beta_1 u'(a) = 0 \\ \alpha_2 u(b) + \beta_2 u'(b) = 0 \end{cases}$$

EXAMPLE (with nonhomogeneous boundary conditions)

$(p=1, q=1, r=0, \alpha_1=\alpha_2=1, \beta_1=\beta_2=0)$

$a=0$

$$\begin{cases} \frac{d^2u}{dx^2} + u = 0 \\ u(0) = 0 \\ u(b) = \beta \end{cases}$$

Gen. Solution: $u(x) = A\cos x + B\sin x$

$u(0)=0 \Rightarrow A=0$

$u(b)=\beta \Rightarrow \beta = B\sin b$

so $u(x) = \frac{\beta}{\sin b}\sin x$ if $\sin b \neq 0$

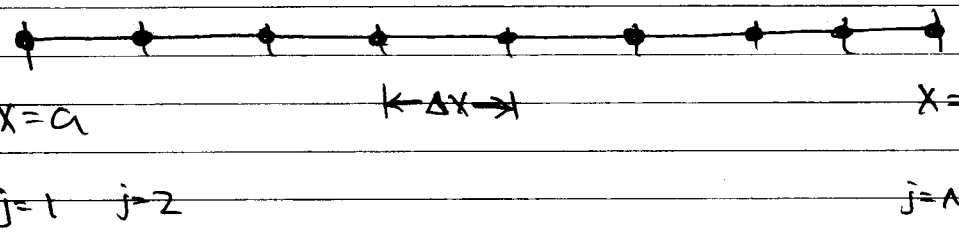
$\Rightarrow$

- if $b = \pm\pi, 2\pi, \ldots$ → NO SOLUTION if $\beta \neq 0$; → INFINITELY MANY SOL. if $\beta = 0$
- otherwise → UNIQUE SOLUTION

Consider $\begin{cases} u'' + u = 0 \\ u(0) = 0 \\ u(1) = \beta \end{cases}$

How do we address this problem numerically?

- discretize the domain into N equal intervals



a=0
b=1
in the example

$$\Delta x = \frac{b-a}{N} \qquad x_j = a + \frac{(j-1)}{(N+1)}(b-a)$$

- seek a vector $\vec{u}$ in $\mathbb{R}^{N+1}$ that will approximate the continuous function $u(x)$ at grid points $x = x_j$

$$\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N+1} \end{bmatrix}$$

- identify $N+1$ equations to determine these $N+1$ unknowns.

$j=1$: $u_1 = 0$ (enforces $u(0)=0$)

$j=2,\dots,N$: $\left(\frac{d^2u}{dx^2}\right)_j + u_j = 0$ (enforces ODE at interior points)

$j=N+1$ $u_{N+1} = \beta$ (enforces $u(1)=\beta$)

The key step is to approximate the derivatives (2^{nd} deriv. in this case) of a continuous function (the exact sol.) using the discrete set of points $u_1, u_2, \ldots, u_{N+1}$.

One possible approximation for $\frac{d^2u}{dx^2}$ at $x = x_j$ is given by

$$\left(\frac{d^2u}{dx^2}\right)_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}$$

[We'll see how to derive this and others shortly...]

- We are approximating the 2^{nd} derivative at $x = x_j$ by a difference formula involving neighboring points u_{j-1}, u_j, u_{j+1}.

Note: to approximate the first derivative one might refer back to the limit definition of derivative

$$\frac{du}{dx} = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}$$

and take

$$\left(\frac{du}{dx}\right)_j = \frac{u_{j+1} - u_j}{h}$$

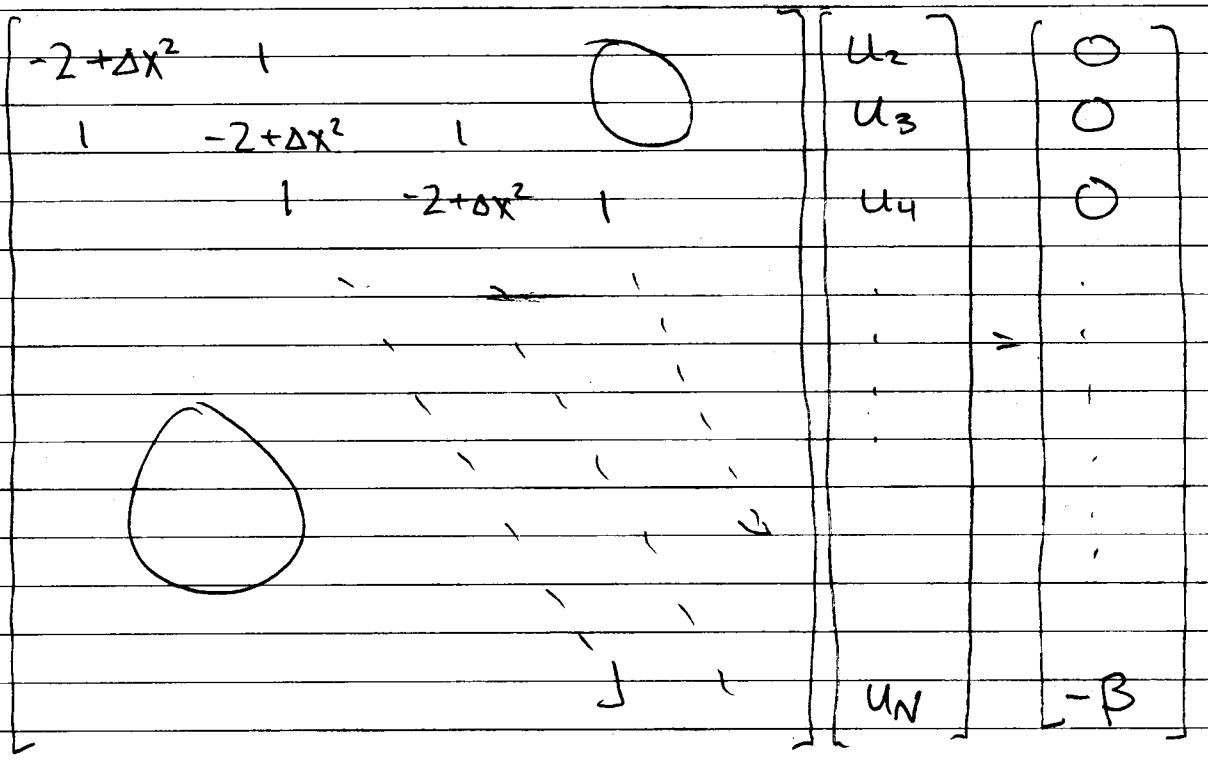
So in this example we find the unknowns $u_1, u_2, \ldots, u_{N+1}$ by solving

$(j=1)$ $\quad u_1 = 0$

$j=2,\ldots,N$ $\quad \dfrac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2} + u_j = 0$

$(j=N+1)$ $\quad u_{N+1} = \beta$

Writing this in matrix form and using the $j=1$ and $j=N+1$ information we have



tri-diagonal system for unknowns $u_2, \ldots, u_N$!

Such a system can be set up and solved numerically (e.g. see Matlab code) for different values of N.

Compare $\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}$ with exact sol.

$$u(x) = \beta \frac{\sin x}{\sin 1}$$

[See bvp_fd.m]