May 7, 2020

Numerical Analysis

Last time: Analysis of one-step methods for solving

$$y' = f(t, y(t))$$
$$y(t_0) = y_0$$

Goal is to approximate y on the interval $[t_0, T]$.

All explicit one-step methods are of the form:

$$y_{k+1} = y_k + h\,\psi(t_k, y_k, h).$$

Definitions:
- Consistent
- Stable scheme
- Convergent

Stiff equations:
- System of initial value problems with the solutions having two different time scales.
- Ex: $y_2(t) = c_1 \underline{e^{-100t}} + c_2 \underline{e^{-t/10}}$ go to zero at very different rates.

Stability Analysis

Consider the model problem:

$$y' = \lambda y$$

Exact solution is $y(t) = C e^{\lambda t}$

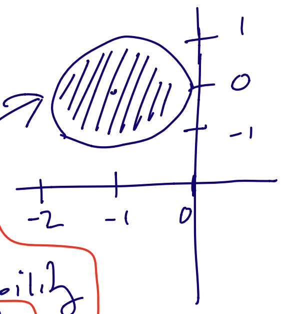
$\to 0$ iff $Re(\lambda) < 0$.

Discretize using (for example) the forward Euler method:

$$y_{k+1} = y_k + h \cdot \lambda y_k$$
$$= (1 + h\lambda)\, y_k$$
$$= (1 + h\lambda)^{k+1} y_0 \longrightarrow 0 \quad \text{iff} \quad |1 + h\lambda| < 1$$

Stability condition.

$\Rightarrow$ In this example, $|1 + h\lambda| < 1$ iff



region of absolute stability $z = \lambda h$ s.t. $y_h \to 0$.

Note: This analysis only makes sense if $\mathrm{Re}(\lambda) < 0$.

Example 2

Let $\vec{y}\,'(t) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \vec{y}(t)$

Let the eigenvalues of A be

$$\left.\begin{aligned} \lambda_1 &= -1 + 10i \\ \lambda_2 &= -10 - 10i \end{aligned}\right\} \Rightarrow$$

This means that the solutions y_1, y_2 are of the form:

$$y_i = C_{1i}\, e^{\lambda_1 t} + C_{2i}\, e^{\lambda_2 t}$$

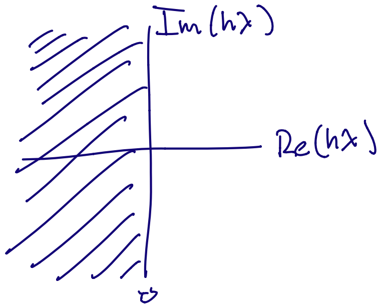
The following conditions must be met for stability of Forward Euler:

$$|1 + h\lambda_1| < 1 \Rightarrow |1 + h(-1 + 10i)| < 1$$
$$|1 + h\lambda_2| < 1 \Rightarrow |1 + h(-10 - 10i)| < 1$$

$$\Leftrightarrow \quad (1-h)^2 + 100h^2 < 1 \Rightarrow h < 2/101$$
$$(1 - 10h)^2 + 100h^2 < 1 \Rightarrow h < \frac{1}{10}$$

This condition is what dictates the maximum step size.

If the region of stability is the entire left half-plane



then this means that for any λ with $Re(\lambda) < 0$, any choice of $h > 0$ yields a stable solution, i.e. $y_k \to 0$. Schemes of this type are called A-STABLE.

Example 3 Backward Euler.

Backward Euler replaces

$$y'(t) = f(t, y(t)) \qquad \text{with} \qquad \frac{y_{k+1} - y_k}{h} = f(t_{k+1}, y_{k+1})$$
$$y(t_0) = y_0$$

$$\Rightarrow \quad y_{k+1} = y_k + h \cdot f(t_{k+1}, y_{k+1})$$

Implicit Scheme.

Apply Backward Euler to $y' = \lambda y$

$$\Rightarrow \quad y_{k+1} = y_k + h\lambda y_{k+1}$$

Solve for y_{k+1}

$$y_{k+1} - h\lambda y_{k+1} = y_k$$
$$y_{k+1} = \frac{1}{1-h\lambda} y_k$$
$$= \frac{1}{(1-h\lambda)^{k+1}} y_0$$

For stiff equations, Backward Euler is very popular

Region of stability is

$$\frac{1}{|1-h\lambda|} < 1 \iff 1 < |1-h\lambda|$$

$\Rightarrow$ A-Stable scheme.

If $Re(\lambda) < 0$, then $|1-h\lambda| > 1$ for any $h > 0$.

Fourier Series

Any function $f \in L^2[0,2\pi]$ and periodic can be written as:

$$f(\theta) = \sum_{n=-\infty}^{\infty} a_n e^{in\theta} \quad \text{continuous Fourier series.}$$

$$\Rightarrow \quad a_n = \frac{1}{2\pi} \boxed{\int_0^{2\pi} f(\theta) e^{-in\theta} \, d\theta.}$$

(By the orthogonality of $e^{in\theta}$).

Goal: Compute the coefficients a_n.

This integral has a periodic integrand.

Apply the trapezoidal rule to this integral:

$$a_n \approx \frac{1}{2\pi} \frac{2\pi}{N} \sum_{l=0}^{N-1} f(\theta_l) e^{-in\theta_l}, \quad \theta_l = \frac{\cancel{l}\,2\pi}{N} \quad \text{equispaced points on } [0,2\pi).$$

$$= \frac{1}{N} \sum_{l=0}^{N-1} f(\theta_l) e^{-2\pi i n l/N}$$

Define the Discrete Fourier Transform:

$$\hat{f}_k = \sum_{l=0}^{N-1} e^{-2\pi i k l/N} f_l \quad \text{for } k = 0, \ldots, N-1.$$

think of f_l as $f(\theta_l)$.

This is a matrix vector product:

$$\begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \vdots \\ \hat{f}_{N-1} \end{pmatrix} = \begin{pmatrix} w_N^0 & w_N^0 & \cdots & w_N^0 \\ w_N^0 & w_N^1 & w_N^2 \cdots & w_N^{N-1} \\ \vdots & & & \\ w_N^0 & w_N^{N-1} & \cdots & \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix}$$

Let $w_N = e^{-2\pi i/N}$

$$e^{-2\pi i k l/N} = w_N^{kl}$$

Goal: Compute this matrix vector product fast.
Direct: $O(N^2)$ flops.

For $k = 0, .., N-1$ (Assume $N = 2^L$).

$$\hat{f}_k = \sum_{l=0}^{N-1} f_l \, \omega_N^{-kl}$$

$$= \sum_{l \text{ even}} f_l \, \omega_N^{-kl} + \sum_{l \text{ odd}} f_l \, \omega_N^{-kl}$$

$$= \sum_{l=0}^{N/2} f_{2l} \, \underbrace{\omega_N^{-k(2l)}} + \sum_{l=0}^{N/2} f_{2l+1} \, \underbrace{\omega_N^{-k(2l+1)}}$$

$$e^{-2\pi i k 2l/N} = e^{-2\pi i k l/N/2} = \omega_{N/2}^{kl}$$

$$e^{-2\pi i k(2l+1)/N} = e^{-2\pi i k/N} \, e^{-2\pi i k 2l/N} = \omega_N^{k} \, \omega_{N/2}^{kl}$$

$$= \underbrace{\sum_{l=0}^{N/2} f_{2l} \, \omega_{N/2}^{kl}} + \omega_N^k \underbrace{\sum_{l=0}^{N/2} f_{2l+1} \, \omega_{N/2}^{kl}}$$

Looks like a discrete Fourier Transform of size $N/2$ instead of N.

But k goes from $k = 0, .., N-1$

What happens to $\omega_{N/2}^{kl}$ when $k > \frac{N}{2} - 1$?

Let $k = \frac{N}{2} + j$, $j \geq 0$

$$\omega_{N/2}^{kl} = e^{-2\pi i (N/2 + j) \cdot l / N/2}$$

$$= e^{-2\pi i l} \cdot e^{-2\pi i j l / N/2}$$

$$= e^{-2\pi i j l / N/2} = \omega_{N/2}^{jl}$$

Let F_N be the matrix of the discrete Fourier transform of size N.

So $F_N \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix} = F_N \vec{f}$

$$= \begin{pmatrix} F_{N/2}\, \vec{f}_{even} + W_N F_{N/2}\, \vec{f}_{odd} \\ F_{N/2}\, \vec{f}_{even} \underbrace{- W_N}_{} F_{N/2}\, \vec{f}_{odd} \end{pmatrix} \qquad W_N = \begin{pmatrix} \omega_N^0 & & & \\ & \omega_N^1 & & \\ & & \ddots & \\ & & & \omega^{N/2-1} \end{pmatrix}$$

Since $\omega_N^{N/2+j} = -\omega_N^j$

$$= \begin{pmatrix} F_{N/2} & W_N F_{N/2} \\ F_{N/2} & -W_N F_{N/2} \end{pmatrix} \begin{pmatrix} \vec{f}_{even} \\ \vec{f}_{odd} \end{pmatrix}$$

$$= \begin{pmatrix} F_{N/2} & W_N F_{N/2} \\ F_{N/2} & -W_N F_{N/2} \end{pmatrix} \underbrace{\begin{pmatrix} 1\,0\,0\,\cdots \\ 0\,0\,1\,\cdots \\ 0\,0\,0\,0\,0\,1\,\cdots \\ \vdots \\ 0\,1\,0\,0\,0\,0\,\cdots \\ 0\,0\,0\,1\,\cdots \\ \vdots \end{pmatrix}}_{\text{permutation matrix}} \vec{f}$$

The cost of doing this is

$$\underbrace{2 \cdot \mathcal{O}\left(\left(\tfrac{N}{2}\right)^2\right)}_{\substack{\text{2 DFTs of} \\ \text{size } N/2}} + \underbrace{2 \cdot \tfrac{N}{2}}_{\substack{\text{scaling by} \\ W_N}} + 2 \cdot N/2$$

$$= \underbrace{\mathcal{O}\left(\frac{N^2}{2}\right)}_{\text{Reduction in cost by factor of 2.}} + \mathcal{O}(N).$$

Since we assumed $N = 2^L$, we can split $F_{N/2}$ again, and repeat the procedure $\Rightarrow$ There will be $\log_2 N$ splittings.

$\Rightarrow$ Eventually arrive at the cost $\mathcal{O}(N \log_2 N) \Rightarrow$ Fast Fourier Transform. (FFT).

Two important facts: (1) FFT is an exact algorithm which relies on algebraic properties of $e^{2\pi i k l/N}$.

(2) The FFT is at the heart of <u>all</u> digital signal processing in electrical engineering.