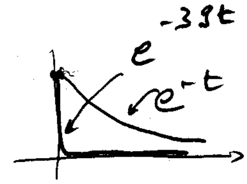
§ 5.11 Stiff differential equations

stiffness for a system = very different time scales for components in vector of solutions.

Typical example:

$$\begin{cases} \underline{y}'(t) = A\underline{y} \\ \underline{y}(0) = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \end{cases} \qquad \text{where } A = \begin{bmatrix} -20 & -19 \\ -19 & -20 \end{bmatrix}$$



solution:

$$\underline{y}(t) = \begin{bmatrix} e^{-39t} + e^{-t} \\ e^{-39t} - e^{-t} \end{bmatrix}$$

e^{-39t} is a <u>transient</u> function: after a short time it becomes $\ll e^{-t}$. however this transient solution will dictate the step size!

The numerical difficulties one encounters with stiff eq. can be illustrated with <u>Euler's method</u>. $(y_{n+1} = y_n + hf(t_n, y_n))$

Consider the simple problem:

$$\begin{cases} y' = \lambda y \\ y(0) = 1 \end{cases} \qquad y(t) = e^{\lambda t}, \text{ with } \lambda < 0 \text{ so that } e^{\lambda t} \to 0 \text{ as } t \to \infty.$$

Euler's method applied to (*) gives:

$$\begin{aligned} y_0 &= 1 \\ y_1 &= y_0 + h\lambda y_0 = (1 + h\lambda) y_0 \\ y_2 &= y_1 + h\lambda y_1 = (1 + h\lambda)^2 y_0 \\ &\vdots \\ y_n &= \dots = (1 + h\lambda)^n y_0 = (1 + h\lambda)^n \end{aligned}$$

The only way we can have $y_n \to 0$ as $n \to \infty$ (as the true sol $e^{\lambda t}$):

$$|1 + h\lambda| < 1 \iff -1 < 1 + h\lambda \underset{\text{automatic}}{<} 1$$

$$\hookrightarrow h\lambda > -2$$

$$\boxed{h < -2/\lambda} \text{ since } \lambda < 0.$$

So if $\lambda = -20$ we need $h < 0.1$ even though solution is mostly flat (zero) for $t > 0$. Ideally one would expect a good numerical method to increase time steps when sol is flat and decrease time step to track a transient behavior.

Implicit Euler's method:

$$y_{n+1} = y_n + f(t_{n+1}, y_{n+1})$$ applied to $(*)$ gives..

$$\begin{cases} y_0 = 1 \\ y_{n+1} = y_n + h\lambda y_{n+1} \end{cases} \Rightarrow y_{n+1} = (1 - h\lambda)^{-1} y_n$$

Hence: $y_n = (1 - h\lambda)^{-n}$

which $\to 0$ when $|1 - h\lambda|^{-1} < 1$, which is true since $h > 0$ and $\lambda < 0$.

Systems of DE The numerical method must perform well on simple linear cases:

$$(**) \begin{cases} \underline{y}' = \overbrace{\begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix}}^{=A} \underline{y} \\ \underline{y}(0) = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \end{cases} \Rightarrow \underline{y}(t) = \begin{bmatrix} e^{(\alpha+\beta)t} + e^{(\alpha-\beta)t} \\ e^{(\alpha+\beta)t} - e^{(\alpha-\beta)t} \end{bmatrix}$$

Euler's method applied to $(**)$:

$$\begin{cases} \underline{y}_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \\ \underline{y}_{n+1} = \underline{y}_n + hA\underline{y}_n \end{cases}$$

Solution is:

$$\underline{y}_n = \begin{bmatrix} (1 + \alpha h + \beta h)^n + (1 + \alpha h - \beta h)^n \\ (1 + \alpha h + \beta h)^n - (1 + \alpha h - \beta h)^n \end{bmatrix}$$

So we need $|1 + \alpha h + \beta h| < 1$ and $|1 + \alpha h - \beta h| < 1$ to get convergence when $\alpha < 0$, $\beta < 0$.

$\Rightarrow$ get one single condition: $0 < h < \dfrac{-2}{\alpha + \beta}$

General Multistep methods

$$a_k y_n + a_{k-1} y_{n-1} + \cdots + a_0 y_{n-k} = h\,[\,b_k f_n + b_{k-1} f_{n-1} + \cdots + b_0 f_{n-k}]$$

Applied again to (*) we get:

$$a_k y_n + a_{k-1} y_{n-1} + \cdots + a_0 y_{n-k} = h\lambda[\,b_k y_n + b_{k-1} y_{n-1} + \cdots + b_0 y_{n-k}]$$

$$\Leftrightarrow (a_k - h\lambda b_k) y_n + (a_{k-1} - h\lambda b_{k-1}) y_{n-1} + \cdots + (a_0 - h\lambda b_0) y_{n-k} = 0$$

Solutions to this difference equation will be linear combinations of basic solutions of the kind $y_n = r^n$, where r = root of poly:

$$\phi(z) = (a_k - h\lambda b_k) z^k + (a_{k-1} - h\lambda b_{k-1}) z^{k-1} + \cdots (a_0 - h\lambda b_0)$$
$$= p(z) - h\lambda\, q(z)$$

using the polynomials we introduced in stability & consistency study:

$$p(z) = a_k z^k + a_{k-1} z^{k-1} + \cdots + a_0$$
$$q(z) = b_k z^k + b_{k-1} z^{k-1} + \cdots + b_0$$

A-Stability

- Consider the slightly more general version of (*) where we allow $\lambda \in \mathbb{C}$. then $y(t) = e^{\lambda t} = e^{Re(\lambda)t} e^{i\,Im(\lambda)t}$
$$= e^{Re(\lambda)t}\left[\cos Im(\lambda)t + i \sin Im(\lambda)t\right].$$

$$y(t) \to 0 \text{ as } t \to 0 \Leftrightarrow Re(\lambda) < 0$$

- In order for our method to mimic this behavior, we want that all roots of ϕ lie in $\{z \mid |z| < 1\}$.

Def (A-Stability): When $h > 0$ and $Re(\lambda) < 0$, all roots of $\phi(z) = p(z) - \lambda h\, q(z)$ have magnitude < 1

<u>Example</u>. Implicit trapezoidal method:

$$y_n - y_{n-1} = \frac{1}{2} h \left[f_n + f_{n-1} \right]$$

$$\rho(z) = z - 1 \; ; \; \sigma(z) = \frac{1}{2}(z+1)$$

$$\phi(z) = z - 1 - \frac{\lambda h}{2}(z+1)$$

root: $z = \dfrac{2+\lambda h}{2-\lambda h}$

$$\left.\begin{array}{l} |2+\lambda h|^2 = 4 + 4\,\mathrm{Re}(\lambda h) + |\lambda h|^2 \\ |2-\lambda h|^2 = 4 - 4\,\mathrm{Re}(\lambda h) + |\lambda h|^2 \end{array}\right) \wedge \text{ if } \mathrm{Re}(\lambda h) < 0.$$

$\Rightarrow |z| < 1$.

<u>Note</u>: Dahlquist proved that all A-stable multistep methods must be <u>implicit</u> and with max. order 2.
<u>Implicit trapezoidal rule</u> is often used because it achieves this max order.

A-stability is too restrictive. $\rightarrow$

<u>Definition (Region of absolute stability)</u>

$$\{ w \in \mathbb{C} \mid \text{all roots of } \rho - w\sigma \text{ are s.t. } |z| < 1 \}$$

- A method will work well in our linear problem (*) if $\lambda h \in$ absolute stability region
- A method is A-stable iff region of absolute stability $= \{ z \in \mathbb{C} \mid \mathrm{Re}\, z < 0 \}$ = left half plane

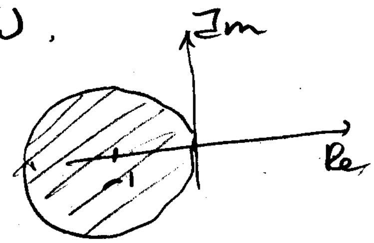
Example: Region of absolute stability of Euler's method:

$y_n - y_{n-1} = h f_{n-1}$

$\Rightarrow \rho(z) = z - 1; \; q(z) = 1; \; \phi(z) = z - 1 - \omega$, with $\omega = \lambda h$

root of ϕ is $z = 1 + \omega$.

need $|1 + \omega| < 1$



When a method that is not A-stable is used, it is hoped that $\omega = \lambda h$ will fall in stability region

- For linear systems of eq $\underline{y}' = A y$ — λ = eigenvalues of matrix A $\Rightarrow$ $h \lambda(A) \in$ region of abs. stability is desired!

- For non-linear eq λ = eigenvalue of linearization