Collocation Method

Want to solve the two-point BVP

\[
\begin{align*}
  w'' &= f(t, u, u'), \quad a < t < b \\
  u(a) &= \alpha, \quad u(b) = \beta
\end{align*}
\]

Idea: (1) "Pretend" the solution is of the form

\[
w(t) \approx \sum_{i=1}^{n} c_i \phi_i(t)
\]

"basis functions" defined on \([a,b]\)

Possible choices of \(\phi_i's:

- polynomials,
- B-splines,
- trig functions

(2) Then \(w''(t) \approx \sum_{i=1}^{n} c_i \phi_i''(t)\) but also

\[
w''(t) = f(t, u, u')
\]

\[
\approx f(t, \sum_{i=1}^{n} c_i \phi_i(t), \sum_{i=1}^{n} c_i \phi_i'(t))
\]

So that

\[
\sum_{i=1}^{n} c_i \phi_i''(t) \approx f(t, \sum_{i=1}^{n} c_i \phi_i(t), \sum_{i=1}^{n} c_i \phi_i'(t))
\]

Unknown \quad Known \quad Known \quad Known
So we have $n$ unknowns & we'd like to create $n$-equation

(3) $\Rightarrow$ define a set of $n$ **collocation** points $a = t_1 < t_2 < \ldots < t_n = b$
and solve the system:

$$
\begin{align*}
\sum_{i=1}^{n} c_i \phi_i(t_j) &= \alpha \\
\sum_{i=1}^{n} c_i \phi_i''(t_j) &= f(t_j) \sum_{i=1}^{n} c_i \phi_i'(t_j) - \sum_{i=1}^{n} c_i \phi_i(t_j) \\
\sum_{i=1}^{n} c_i \phi_i(t_n) &= \beta
\end{align*}
$$

$\Rightarrow$ we now have the $c_i$'s

$\Rightarrow$ we now have $u(t) = \sum_{i=1}^{n} c_i \phi_i(t)$

**Example:**

$$
\begin{align*}
\frac{d^2 u}{dt^2} &= 6t, & 0 < t < 1 \\
\end{align*}
$$

$u(0) = 0, \quad u(1) = 1$

Let's use $t_1 = 0, \quad t_2 = \frac{1}{2}, \quad t_3 = 1$
and let's use $\phi_1(t) = 1, \quad \phi_2(t) = t, \quad \phi_3(t) = t^2$
so that our approx. soln

\[ w(t) = c_1 + c_2 t + c_3 t^2 \]

\[ \Rightarrow w'(t) = c_2 + 2c_3 t \quad , \quad w''(t) = 2c_3 \]

We'd like \( \{ w''(t_i) = 6t_i \quad , \quad i=2 \}

\[ \{ w(0)=0 \quad , \quad w(1)=1 \}

\[ \Rightarrow \begin{cases} 2c_3 = 6(\frac{1}{2}) \\ c_1 + c_2(0)=0 \quad , \quad c_1 + c_2^* + c_3^2 = 1 \end{cases} \]

\[ \Rightarrow c_1 = 0 \quad , \quad c_2 = -0.5 \quad , \quad c_3 = 1.5 \]

and \( w(t) = 0 + -0.5t + 1.5t^2 \)

(note that the true soln is \( \dot{u} = 6t \)

\[ \Rightarrow \dot{u}' = 3t^2 + a \]

\[ \Rightarrow u = \frac{3}{2} t^3 + at + b \]

\[ u(0)=0 \Rightarrow b = 0 \]

\[ u(1) = 1 \Rightarrow 1+a = 1 \]

\[ \Rightarrow a = 0 \]

\[ \Rightarrow u(t) = t^3 \]
Example:

\[
\begin{align*}
\ell \quad & u'' + p(t)u' + q(t)u = f(t) \\
& u(a) = \alpha, \quad u(b) = \beta
\end{align*}
\]

Use $B$-splines $B_k^{i}$, $k \geq 3$ 
$(\Rightarrow 2$ cont's derivatives $)$ \( (170\beta) \)

Choose the knots such that 
\[ t_i = t_{i-1} + h, \quad t_1 = a \]

and use the knots as collocation points.

So, we want to approx. the soln $u(t)$ by 

\[ u(t) = \sum_{j=1}^{n} c_j \phi_j(t) \]

where 

\[ u(t_i) = \sum_{j=1}^{n} c_j \phi_j(t_i) \]

\[ \text{chosen well} \]
satisfies
\[
\begin{align*}
  w''(t_i) + p(t)w'(t_i) + q(t)w(t_i) &= z(t_i) \\
  \text{for } i &= 1, \ldots, n-2 \\
  w(a) &= \alpha, \quad w(b) = \beta
\end{align*}
\]

Using prop's of B-splines this leads to a system of linear equations for the coeff's \( \mathbf{c} = (c_1, \ldots, c_n) \)
\[
A \mathbf{c} = \mathbf{b}
\]
where
\( A \) is a banded matrix,
\( \implies \) fast inversion is possible.
Collocation for IVP's

Want to solve

\[
\begin{cases}
    y'(t) = f(t, y) \\
    y(t_0) = \alpha
\end{cases}
\]

over, say \([t_0, t_0+h]\)

Idea: approximate the sol'n \(y(t)\)

by a polynomial \(p(t)\)

of degree \(n\)

\(\Rightarrow n+1\) parameters needed

Want

\[
\begin{cases}
    p'(t_k) = f(t_k, p(t_k)), \\
    p(t_0) = \alpha \\
    \text{n+1 equations}
\end{cases}
\]

Example \((n = 2 =)\) trap. rule!

\[
\begin{cases}
    p(t_0) = y_0 \\
    p'(t_0) = f(t_0, p(t_0)) \\
    p'(t_0+h) = f(t_0+h, p(t_0+h))
\end{cases}
\]
write \( p(t) = c_3(t - t_0)^2 + c_2(t - t_0) + c_1 \)

and so the for \( c_1, c_2, c_3 \)

\[
\begin{align*}
\Rightarrow \quad & c_1 = y_0 \\
\Rightarrow \quad & c_2 = f(t_0, p(t_0)) \\
\Rightarrow \quad & 2c_3(h) + c_2 = f(t_0 + h, p(t_0 + h)) \\
\Rightarrow \quad & c_3 = \frac{f(t_0 + h, p(t_0 + h)) - f(t_0, p(t_0))}{2h} \\
\Rightarrow \quad & \text{we now have} \\
p(t_0 + h) = y_0 + f(t_0, p(t_0)) h \\
& + \frac{f(t_0 + h, p(t_0 + h)) - f(t_0, p(t_0))}{2h} (h)^2 \\
\Rightarrow \quad & p(t_0 + h) = y_0 + \frac{h}{2} \left[ f(t_0 + h, y_1) + f(t_0, y_0) \right] \\
y_1 = y_0 + \frac{h}{2} \left[ f(t_0 + h, y_1) + f(t_0, y_0) \right] 
\end{align*}
\]
Implicit eq'n in $y_1$

solve and repeat!

(to get $y_2, \ldots$)
Stiff Equations

Solving IVP’s using numerical methods leads to errors that involve a higher derivative of the solution.

Problems can happen if the magnitude of the derivative increases (but the solution does not) IVP’s with this type of issue are called stiff.

Example 8 Consider the IVP

\[
\begin{align*}
  x_1' &= 9x_1 + 24x_2 + 5\cos t - \frac{1}{3}\sin t \\
  x_2' &= -24x_1 - 51x_2 - 3\cos t + \frac{1}{3}\sin t \\
  x_1(0) &= \frac{1}{3}, \quad x_2(0) = \frac{2}{3}
\end{align*}
\]
This has the unique sol'n
\[ x_1(t) = 2e^{-3t} - e^{-33t} + \frac{1}{3} \cos t \]
\[ x_2(t) = -e^{-3t} + 2e^{-33t} - \frac{1}{3} \cos t \]

Here, the \( e^{-33t} \) term causes the equation to be stiff.

(derivatives of \( e^{ct} \to ce^{ct} \to \ldots \to c^n e^{ct} \))

How does this affect numerical methods?

In the example above, RK4 with \( h=0.05 \) works just fine but RK4 with \( h=0.1 \) blows up \( \lim_{t \to \infty} x_1(t) \to -\infty \) \( \lim_{t \to \infty} x_2(t) \to \infty \)
Q: How can we predict/understand "stiffness" when seeking numerical solutions?

A: 1) Fix the numerical method
   2) Examine the error it produces when applied to the test equation

\[
\begin{align*}
  x' &= \lambda x \\
  x(0) &= 1
\end{align*}
\]  

which has the sol'n \( x(t) = e^{\lambda t} \) 

(interested in \( \lambda < 0 \))

Example: Euler's Method

\[
\begin{align*}
  w_0 &= 1 \\
  w_{n+1} &= w_n + h f(t_n, w_n)
\end{align*}
\]  

\( \lambda w_n \) in (*)
\[
\begin{align*}
\Rightarrow \quad w_{n+1} &= w_n + h \lambda w_n \\
&= (1 + h \lambda) w_n \\
\Rightarrow \quad w_{n+1} &= (1 + h \lambda)^n w_0 \\
\end{align*}
\]

Euler's method approx.

True solution at \((n+1)h\): \( x((n+1)h) \)

Reindexing: the error is 
\[
| x(t_n) - w_n | = | e^{\lambda n} - (1 + h \lambda)^n | 
\]

- when \(x > 0\)
- this decays to zero
- only decays to zero if \(|1+h\lambda| < 1\)

\[
\Rightarrow \quad \text{we need } -1 < 1 + h \lambda < 1 \\
\Rightarrow \quad -2 < h \lambda < 0 \\
\Rightarrow \quad h < -\frac{2}{\lambda} \quad \text{for Euler}
\]

So we need \[ h < -\frac{2}{\lambda} \] for Euler.
In other words bigger \(|\lambda|\) requires smaller step size even though the solution goes very fast!
true

Example: Taylor method of order \(k\)

Here \(w_n = (1 + h\lambda + \frac{1}{2} h^2 \lambda^2 + \ldots + \frac{1}{n!} h^n \lambda^n)w_{n-1}\)

So we'd want

\[
|(1 + h\lambda + \frac{1}{2} h^2 \lambda^2 + \ldots + \frac{1}{n!} h^n \lambda^n)| < 1
\]

\[
\ldots
\]

Example: Implicit Euler method

\[w_{n+1} = w_n + hf(t_{n+1}, w_{n+1})\]

\[\Rightarrow \begin{cases} w_0 = 1 \\ w_{n+1} = w_n + h\lambda w_{n+1} \quad \text{— using (*)} \end{cases}\]
\begin{align*}
\Rightarrow \quad w_{n+1} &= (1 - hI)^{-1} w_n \\
\Rightarrow \quad w_n &= (1 - hI)^{-n} w_0
\end{align*}

so now, we want

$$\left| \frac{e^{\gamma n}}{(1-hI)^n} \right| \text{ small}$$

$$\Rightarrow 0$$

as \( n \to \infty \)

want \( |1-h\lambda|^{-1} < 1 \)

Always true! \( |1-h\lambda|^{-1} < 1 \) \( \Rightarrow \) \( |1-h\lambda| > 1 \)

Systems & Stiffness

Same Idea = Fix a method, examine on a test case

Example: \( \lambda < 0 \) \( \beta < 0 \)
\[ \begin{align*}
\dot{x} &= \alpha x + \beta y, \quad x(0) = 2 \\
\dot{y} &= \beta x + \alpha y, \quad y(0) = 2 
\end{align*} \]

(True soln: \[ \begin{align*}
x(t) &= e^{(\alpha+\beta)t} + e^{(\alpha-\beta)t} \\
y(t) &= e^{(\alpha+\beta)t} - e^{(\alpha-\beta)t} 
\end{align*} \]

Euler's method here would yield

\[ \begin{align*}
\omega_{n+1} &= \omega_n + h(\alpha \omega_n + \beta \nu_n), \quad \omega_0 = 2 \\
\nu_{n+1} &= \nu_n + h(\beta \omega_n + \alpha \nu_n), \quad \nu_0 = 1 
\end{align*} \]

\[ \begin{align*}
\omega_n &= (1 + \alpha h + \beta h)^n + (1 + \alpha h - \beta h)^n \\
\nu_n &= (1 + \alpha h + \beta h)^n - (1 + \alpha h - \beta h)^n 
\end{align*} \]

Want \( \omega_n \) & \( \nu_n \) to decay!

So we want \( |1 + \alpha h + \beta h| < 1 \)

\( |1 + \alpha h - \beta h| < 1 \)

\[ h < \frac{2}{\alpha + \beta} \]

(\( \Rightarrow \) want)
General linear multi-step methods

Recall:
$$\sum_{i=0}^{k} a_i x_{n-i} = h \sum_{i=0}^{k} b_{k-i} f_{n-i}$$

A general multi-step method

Apply this to the test problem

\[
\begin{cases}
    x' = \lambda x \\
    x(0) = 1
\end{cases}
\]

(\bullet) \quad (\lambda < 0)

To get
$$\sum_{i=0}^{k} a_i x_{n-i} = \lambda h \sum_{i=0}^{k} b_{k-i} x_{n-i}$$

\[(a_k - h \lambda b_k) x_n + \cdots + (a_0 - h \lambda b_0) x_{n-k} = 0\]

So the solution is a combo of terms: $x_n = r^n$

where $r$ is a root of $\phi(z)$
\[ \phi(z) = (a_k - h\lambda b_k) z^k + (a_{k-1} - h\lambda b_{k-1}) z^{k-1} + \cdots + (a_0 - h\lambda b_0) \]

Characteristic polynomial

\[ \phi(z) = p(z) - h\lambda q(z) \]

(from stability section)

Fact: In order to obtain a decaying numerical sol'n (complex)

we need the \( r \) roots of \( \phi(z) \)

to lie in the disk given by \( |z| < 1 \)

A-stability

We've been considering \( \{ x' = \lambda x \}

with \( \lambda < 0 \)
Let us now consider complex \( \lambda : \lambda = \mu + i\nu \).

Now, the solution to (*) is
\[ x(t) = e^{\lambda t} = e^{\mu t} (\cos \nu t + is\nu t) \]

we're interested in \( \mu < 0 \).

So, for a multistep method to do well, we want the roots of \( \phi(z) \) to be in the unit disk whenever \( h > 0 \), \( \mu = \text{Re}(\lambda) < 0 \). This property is A-stability.
Example 3: * Implicit Euler is $A$-stable (check) 

* Implicit Trapezoidal method

$w_n = w_{n-1} + \frac{1}{2} h \left[ f_n + f_{n-1} \right]$ is also $A$-stable, bec.

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

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

has root $z^* = \frac{1 + h\sqrt{2}}{1 - h\sqrt{2}}$

when $h > 0$ & $\text{Re}(\lambda) < 0$

$|z^*| = \left| \frac{2 + h\lambda (\mu + iv)}{2 - h\lambda (\mu + iv)} \right| < 1$

Theorem 6: Among linear multi-step methods, only implicit method of order $\leq 2$ can be $A$-stable.
Region of Absolute Stability (for multi-step methods)

Idea: want roots of

\[ \phi(z) = p(z) - h\lambda g(z) \]

to be in unit disk, so the multi-step method can work on the test prob.

\[ x' = \lambda x \]

So, we are interested in:

Region of absolute stability is

\[ \{ w \in \mathbb{C} : \text{roots of } p(z) - wzg(z) \text{ lie in the interior of the unit disk } \} \]

(A-stable methods work for all \( h > 0 \), other methods work when \( h \) is small enough)
Example: \( x_n = x_{n-1} + hf_{n-1} \)

\( E\)uler's method

Then \( \phi(z) = (z-1) - z \frac{h}{2} \)

\[ = (z-1) - \frac{h}{2} \Rightarrow \text{Root: } 1+w \]

\( \Rightarrow \text{Region of absolute stability} \)

is \( \{ w \in \mathbb{C} : |1+w| < 1 \} \)

\[
(1+hw < 1 \Rightarrow |1+h\mu + ihw| < 1
\]
\[
\Rightarrow \sqrt{(1+h\mu)^2 + (hw)^2} < 1
\]
\[ 1 + h^2\mu^2 + 2h\mu + h^2v^2 < 1 \]
\[ h^2(\mu^2 + v^2) < -2h\mu \]
\[ h < -\frac{2\mu}{\mu^2 + v^2} \]
Practical implications

To use higher order methods, we'd like \( w - \lambda h \in \mathbb{R} \) = Region of abs. stab

When \( x' = \lambda x \) is our ODE, we just pick \( h \) so that \( \lambda h \in \mathbb{R} \)

If you have a non-linear IVP

\[
x' = f(t, x) \quad \rightarrow \quad f(t, x) \approx \lambda x
\]

linear approx