Math 128A, Summer 2019

Lecture 24, Today's Date 8/1/2019

1 Review

Recall from yesterday that we have a definition that f is dissapative means that

$$(f(u) - f(v))^T (u - v) \leq 0.$$

Additionally, recall that we have a theorem that if f is dissipative and $u'(t) = f(u)$ and $v'(t) = f(v)$, then

$$||u(t) - v(t)|| \qquad \text{decreasing}$$

This is a desirable because this means that our method forgets our initial conditions. This is good if we want to 'destroy' evidence, assuming we know where we want to be. To put these neatly, we include the following from lecture 23 yesterday.

Definition: Dissipative -

We say f is dissipative if

$$\left(f(u) - f(v)\right)^T (u - v) \leq 0$$

Theorem 1.1. If function f is dissipative and $u' = f(u), v' = f(v)$, then

$$||u(t) - v(t)||$$

is decreasing.

Definition: B-stable -

We say a numerical method is B-stable if

$$||u_{n+1} - v_{n+1}|| \leq ||u_n - v_n|| \qquad (\text{"contractive"})$$

whenever f is dissipative.

2 Nonlinear Stability Theory

We claim the following.

Theorem 2.1. B-stable implies A-stable.

Proof. Notice that B-stable is a nonlinear property, whereas A-stable is a linear property. Recall that if a method is A-stable, we have $y' = \lambda y, \mathrm{Re}\lambda \leq 0, |u_{n+1}| \leq |u_n|$.

$$\begin{aligned} y &= u + iv \\ y' &= u' + iv' \\ &= (\alpha + i\beta)(u + iv) \\ &= \alpha u - \beta v + i(\alpha v + \beta u) \\ \begin{bmatrix} u \\ v \end{bmatrix}' &= \underbrace{\begin{bmatrix} \alpha & -\beta \\ \alpha & \beta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}}_{f(u,v)} \end{aligned}$$

Applying the fact that these are linear, with $\alpha \leq 0$, consider:

$$\begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} \alpha u - \beta v \\ \beta u + \alpha v \end{bmatrix} = \alpha u^2 - \beta uv + \beta uv + \alpha u^2 \leq 0,$$

and we are done.

□

3 Algebraic Stability

Now we want to check for B-stability for Runge-Kutta methods by **algebra** only.

$$k_i = f(\underbrace{u_n + h \sum a_{i,j} k_j}_{=:q_i})$$

Hence

$$\begin{aligned} u_{n+1} &= u_n + h \sum b_i f(q_i) \\ &= u_n + h \sum b_i k_i \end{aligned}$$

We start somewhere else,

$$\begin{aligned} k_i^* &= f(u_n^* + h \sum a_{i,j} k_j^*) \\ u_{n+1}^* &= u_n^* + h \sum b_i k_i^*, \end{aligned}$$

which gives a different solution, u^*.
Taking the difference between these two, we have:

$$\begin{aligned} \underbrace{u_{n+1} - u_{n+1}^*}_{=:\Delta u_{n+1}} &= u_n - u_n^* + h \sum b_i (k_i - k_i^*) \\ \Delta u_{n+1} &= \Delta u_n + h \sum b_i \Delta k_i \\ \Delta k_i &= f(q_i) - f(q_i^*) \end{aligned}$$

The reason for inserting these q_i is that if we dot the difference between the q's and multiply by k_i, then this will be ≤ 0 if f is dissipative. That is, if f is disappative,

$$\Delta q_i^T \Delta k_i \leq 0$$
$$(q_i - q_i^*)^T (f(q_i) - f(q_i^*)) \leq 0$$

Taking norms squared, we have:

$$||\Delta u_{n+1}||^2 = ||\Delta u_n||^2 + 2h(\Delta u_n)^T \sum b_i \Delta k_i + h^2 \left(\sum_i b_i \Delta k_i\right)^T \left(\sum_j b_j \Delta k_j\right)$$

To help, consider:

$$q_i = u_n + h \sum a_{i,j} k_j$$
$$\Delta q_i = \Delta u_n + h \sum a_{i,j} \Delta k_j$$
$$\Delta u_n = \Delta q_i - h \sum a_{i,j} \Delta k_i.$$

Now we should write down the important part with the $2h$ in front:

$$\Delta u_n^T \Delta k_i = \Delta q_i^T \Delta k_i - h \sum a_{i,j} \Delta k_j^T \Delta k_i$$

And we can now rewrite the whole thing with what we figured out:

$$||\Delta u_{n+1}||^2 = ||\Delta u_n||^2 + 2h \sum b_i \left(\Delta q_i^T \Delta k_i - h \sum_j a_{i,j} \Delta k_i^T \Delta k_j\right) + h^2 \sum b_i \Delta k_i^T \sum b_j \Delta k_j$$

Because the method is B-stable, f is dissipative, and thus $\Delta q_i^T \Delta k_i \leq 0$, which gives us:

$$||\Delta u_{n+1}||^2 \leq ||\Delta u_n||^2 + h^2 \sum_i \sum_j \left[(b_i b_j - 2b_i a_{i,j}) \Delta k_i^T \Delta k_j\right]$$

Notice that we can swap i and j, because the matrix $\Delta k_i^T \Delta k_j$ is symmetric (via effective dot product):

$$\sum_i \sum_j b_i a_{ij} \Delta k_i^T \Delta k_j = \sum_i \sum_j b_j a_{ji} \Delta k_i^T \Delta k_j$$

and now because we have two, we take one of each representation.

$$||\Delta u_{n+1}||^2 \leq ||\Delta u_n||^2 + h^2 \sum_i \sum_j \overbrace{(b_i b_j - b_i a_{ij} - b_{ji} a_{ji})}^{M_{ij}} \Delta k_i^T \Delta k_j$$

We conclude that if $b_i \geq 0$ and $x^T M x \leq 0$ for all vectors x, then the method is B-stable. We call this **algebraic stability**.
How do we check $x^T M x \leq 0$ for all x? Because M is a symmetric matrix, M has a basis of eigenvectors, say $e_1, \ldots, e_s$.
This gives:

$$e_j^T M e_j = e_j^T \lambda_j e_j,$$

if $||e_j|| = 1$. Hence this is equivalent to checking $\lambda_j \leq 0$ for $\forall j$. This is the condition for M to be negative semi-definite.

3.1 Example: Trying Implicit Euler

Recall that the Butcher Array Implicit Euler is simply $\begin{bmatrix}1\\1\end{bmatrix}$, where the lower value is $1 \geq 0$. Hence

$$M = 1 \cdot 1 - 1 - 1 = -1 \leq 0,$$

and hence Implicit Euler is indeed B-stable (which we already knew).

3.2 Example: Trying Trapezoidal Rule

$$\begin{aligned} k_1 &= f(u_n) \\ k_2 &= f(u_{n+1}) = f\left(u_n + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2)\right)\right) \\ u_{n+1} &= u_n + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right), \end{aligned}$$

which gives us the Butcher array:

$$\begin{bmatrix} 0 & 0 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

where the bottom row is all ≥ 0. Now we need to evaluate the matrix M:

$$\begin{aligned} M &= \begin{bmatrix}1/2\\1/2\end{bmatrix}\begin{bmatrix}1/2 & 1/2\end{bmatrix} - \begin{bmatrix}1/2 & 0\\0 & 1/2\end{bmatrix}\begin{bmatrix}0 & 0\\1/2 & 1/2\end{bmatrix} - \begin{bmatrix}0 & 1/2\\0 & 1/2\end{bmatrix}\begin{bmatrix}1/2 & 0\\0 & 1/2\end{bmatrix} \\ &= \begin{bmatrix}-1/4 & 1/4\\1/4 & 1/4\end{bmatrix} - \begin{bmatrix}0 & 0\\1/4 & 1/4\end{bmatrix} - \begin{bmatrix}0 & 1/4\\0 & 1/4\end{bmatrix} = \begin{bmatrix}1/4 & 0\\0 & -1/4\end{bmatrix} \end{aligned}$$

We aren't sure of this result (that Trapezoidal Rule is not B-stable), so we check the Midpoint rule. If this is the case, Trapezoidal Rule is A-stable but not B-stable.

3.3 Midpoint Rule

$$\begin{aligned} u_{n+1} &= u_n + hf\left(\frac{u_n + u_{n+1}}{2}\right) \\ k_1 &= f\left(\frac{u_n}{2} + \frac{1}{2}\left(u_n + hk_1\right)\right) \\ &= f\left(u_n + \frac{1}{2}hk_1\right) \end{aligned}$$

which gives the butcher array

$$\begin{bmatrix}1/2\\1\end{bmatrix}$$

and we write:

$$M = 1 - \frac{1}{2} - \frac{1}{2} = 0 \leq 0,$$

so the Midpoint rule is B-stable but Trapezoidal rule is **not** B-stable (technically we only failed to prove it is B-stable, and not exactly proved it isn't).

4 Hammer-Hollingsworth

This is a fourth-order, two stage implicit Runge-Kutta method which is B-stable (4-order 2-stage IRK).
We write this as

$$\int_{t_n}^{t_{n+1}} y'(s)\ ds = w_1 y'(s_1) + w_2 y'(s_2) = y_{n+1} - y_n + \underbrace{O(h^5)}_{h\tau_n}$$

To get the RHS of the above, let's say we use 2-point Gaussian integration, which we recall from before takes on points at $\pm\frac{1}{\sqrt{3}}$ with weights that add to 2. Let s_1, s_2 be the Gaussian Quadrature points:

$$s_1 = t_n + \frac{h}{2} + \frac{h}{2}\left(\frac{-1}{\sqrt{3}}\right)$$
$$s_2 = t_n + \frac{h}{2} + \frac{h}{2}\left(\frac{+1}{\sqrt{3}}\right)$$

So we take $w_1 = w_2 := \frac{h}{2}$.
How do we find the stages?

$$k_1 \approx y'(s_1)$$

This means that we evaluate:

$$\underbrace{f(u_n + h\sum_{j=1}^{2} a_{ij}k_j)}_{y(s_1)},$$

where

$$y(s_1) = y_n + \int_{t_n}^{s_1} y'(s)\ dt$$

The stages
We observed that there is a chance of getting fourth order accuracy if we put the stages precisely at the Gauss points. This tells us what the c's are, but it doesn't give us the a's that we need. The row sums of the Butcher array are:

$$c_1 = \frac{1}{2} - \frac{1}{2\sqrt{3}}$$
$$c_2 = \frac{1}{2} + \frac{1}{2\sqrt{3}}$$

and we have some Butcher array:

$$\begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

for some matrix up top.

So, we want to do the integral:

$$\int_{t_n}^{s_i} y'(s)\ ds = h \sum \underbrace{a_{ij}}\ y'(s_j)$$

There aren't many ways to get these a_{ij}, so we say to make it exact for the function $f(x) := 1$, then for the function $f(x) = s$:

$$\begin{aligned}
&\int_{t_n}^{s_i} 1\ ds = h \sum_j a_{1j} 1 \\
\implies\ & s_i - t_n = h\left(a_{i,1} + a_{i,2}\right) \\
&\int_{t_n}^{s_i} s\ ds = h \sum_j a_{1j} s_j \\
\implies\ & \frac{1}{2} s^2 |_{t_n}^{s_i} = h\left(a_{i1} s_1 + a_{i2} s_2\right) \\
& \frac{1}{2} s_i^2 - \frac{1}{2} t_n^2 = h\left(a_{i1} s_1 + a_{i2} s_2\right),
\end{aligned}$$

and we skip the algebra and write the resulting Butcher array:

$$\begin{bmatrix}
\left(\frac{1}{2} - \frac{1}{2\sqrt{3}}\right) | & \frac{1}{4} & \left(\frac{1}{4} - \frac{1}{2\sqrt{3}}\right) \\
\left(\frac{1}{2} + \frac{1}{2\sqrt{3}}\right) | & \left(\frac{1}{4} + \frac{1}{2\sqrt{3}}\right) & \frac{1}{4} \\
& \frac{1}{2} & \frac{1}{2}
\end{bmatrix}$$

We call this **Hammer-Hollingsworth** '2-stage Gauss'. To check if this is B-stable, we do the easy part first, which is to check the bs at the bottom are ≥ 0, which is true.
Now,

$$\underbrace{\begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix}}_{bb^T} - \underbrace{\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}}_{B=\mathrm{diag}(b)} \underbrace{\begin{bmatrix} \left(\frac{1}{4}\right) & \left(\frac{1}{4} - \frac{1}{2\sqrt{3}}\right) \\ \left(\frac{1}{4} + \frac{1}{2\sqrt{3}}\right) & \frac{1}{4} \end{bmatrix}}_{A} - \underbrace{\begin{bmatrix} \frac{1}{4} & \left(\frac{1}{4} + \frac{1}{2\sqrt{3}}\right) \\ \left(\frac{1}{4} - \frac{1}{2\sqrt{3}}\right) & \frac{1}{4} \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}}_{B}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \leq 0,$$

so Hammer-Hollingsworth is indeed B-stable!

5

Now suppose $g : \mathbb{R}^m \to \mathbb{R}$, and we want to minimize $g(x)$. To do this, first notice that our result is a single result which makes this very simple. To not brute-force or perform random searches, we want from calculus:

$$\begin{aligned}
x'(t) &= -\nabla g(x) \\
&= f(y)
\end{aligned}$$

Essentially, this can work like (stochastic) gradient descent. Our question is whether f is dissipative. To see this,

$$(u - v)^T (f(u) - f(v)) = (u - v)^T (-\nabla g(u) + \nabla g(v)).$$

If we have a one-dimentional problem, we have:

$$\begin{aligned}(g'(v) - g'(u))(u - v) &= g''(\xi)(v - u)(u - v) \\ &= -g''(\xi)(u - v)^2 \\ &\leq 0,\end{aligned}$$

if $g''(\xi) \geq 0$. This is what it means for f to be dissipative (this is why dissipative systems come up a lot). Unfortunately, in multi-dimensions, there is **no reason** for the mean-value theorem to apply, as the mean value may not lie on the exact straight line between two points.
However, we still have:

$$\begin{aligned}u - v &= v + t(u - v)|_{t=0}^{t=1} \quad \text{(just verify this is true)} \\ &= \int_0^1 \frac{d}{dt}(v + t(u - v))\end{aligned}$$

Hence

$$\begin{aligned}(u - v)^T (f(u) - f(v)) &= \int_0^1 \frac{d}{dt}(v + t(u - v))\ dt^T (f(u) - f(v)) \\ &= \int_0^1 \frac{d}{dt}[v + t(u - v)]^T (f(u) - f(v))\ dt\end{aligned}$$

To solve this integral, we'll pick up on Monday!

Lecture ends here.