Numerical Methods II, Courant Institute, Spring 2024
http://www.math.nyu.edu/faculty/goodman/teaching/NumericalMethodsII2024.html
Jonathan Goodman, February, 2024

ODE solvers, Linear Multistep methods

1 Introduction

We are still discussing numerical methods for the initial value problem for the ODE

$$\dot{x} = f(x) \ . \tag{1}$$

For the first part of this class we assume a fixed time step Δt and use notation from last class: $t_n = n\Delta t$ and $X_n \approx x(t_n)$. *Multi-step* ODE solvers with maximum lag s have the form

$$X_{n+1} = \Psi(X_n, \cdots, X_{n-s}, \Delta t) \ .$$

Runge Kutta methods have zero lag. Using lags allows for higher order of accuracy without multiple evaluations of f per time step. This if Δt is the same and the method is fourth order accurate (say), this allows fourth order accuracy with one evaluation per time step rather than the four needed for Runge Kutta.

There are practical disadvantages of multistep methods too. One is how to start. If you are given X_0, you can compute X_1 with a Runge Kutta method but not with a multi-step method. *Stability* is a primary and subtle issue for multi-step methods. There was no issue with stability of explicit Runge Kutta methods because they all have the form

$$\Psi(x, \Delta t) = x + \Delta t \widetilde{\Psi}(x, \Delta t) \ .$$

The part that multiplies Δt is build from f

Warning: Starting with Section 3, we will use the convention of writing multistep methods "in the future" rather than "in the past". We will change the index n so that an explicit multi-step method has the form

$$X_{n+s} = \Psi(X_{n+s-1}, \cdots, X_n, \Delta t) \ . \tag{2}$$

I believe this convention was adopted because it makes the *characteristic polynomial* of the *recurrence relation* (defined below) seem more natural. If that's true (?dunno?), it means that somebody thought it was better to confuse the person implementing the method than to confuse the person doing the analysis. Section 2 does not use this convention.

2 Motivating examples

The same ideas that motivated multi-stage methods can be used to motivate the first multi-step methods. We find new ways to modify forward Euler to get to second order, and then ask how to generalize those ideas.

One trick is to *center* the scheme about t_n using values from t_{n-1} and t_{n+1}.

$$\dot{x}(t_n) = \frac{1}{2\Delta t} \left[x(t_{n+1}) - x(t_{n-1}) \right] + O(\Delta t^2) \ .$$

This means that the exact solution satisfies

$$x(t_{n+1}) = x(t_{n-1}) + 2\Delta t f(x(t_n)) + O(\Delta t^3) \ .$$

This suggests that we can build a second order method by neglecting the error term:

$$X_{n+1} = X_{n-1} + 2\Delta t f(X_n) \ . \tag{3}$$

This is not a one step method because you have to know both X_n and X_{n-1} to compute X_{n+1}. It is often called *leapfrog* because of a children's game with that name in which kids take turns jumping over each other[1] It is a second order accurate method that uses just one f evaluation per time step.

Another idea involving centering starts from

$$f(x(t_{n+\frac{1}{2}})) = \dot{x}(t_{n+\frac{1}{2}}) = \frac{x(t_{n+1}) - x(t_n)}{\Delta t} + O(\Delta t^2) \ . \tag{4}$$

The notation for the midpoint of the interval $[t_n, t_{n+1}]$ is

$$t_{n+\frac{1}{2}} = (n + \tfrac{1}{2}) \, \Delta t \ .$$

We can make this into a practical second order method if you predict the midpoint value $f(t_{n+\frac{1}{2}})$ using earlier values of f. We have evaluated f at times t_n and t_{n-1}, so linear extrapolation would get to time $t_{n+\frac{1}{2}}$ with second order accuracy. The linear approximation to $F(t) = f(x(t))$ using these values is

$$\begin{aligned} \widetilde{F}(t) &= F(t_n) + s_n(t - t_n) \\ s_n &= \frac{F(t_n) - F(t_{n-1})}{t_n - t_{n-1}} \ . \end{aligned} \tag{5}$$

We want to evaluate at $t = t_{n+\frac{1}{2}}$, which is

$$\begin{aligned} \widetilde{F}(t_{n+\frac{1}{2}}) &= F(t_n) + \frac{1}{2} \left(F(t_n) - F(t_{n-1}) \right) \\ &= \frac{3}{2} F(t_n) - \frac{1}{2} F(t_{n-1}) \ . \end{aligned}$$

[1] Look at the Wikipedia page. The image might be that you jump over t_n to go from t_{n-1} to t_{n+1}.

Substituting this for $f(x(t_{n+\frac{1}{2}}))$ in (4) gives

$$x(t_{n+1}) = x(t_n) + \Delta t \left[\frac{3}{2} f(x(t_n)) - \frac{1}{2} f(x(t_{n-1})) \right] + O(\Delta t^3) \ .$$

As for the leapfrog scheme, we get a practical method by neglecting the error term:

$$X_{n+1} = X_n + \Delta t \left[\frac{3}{2} f(X_n) - \frac{1}{2} f(X_{n-1}) \right] \ . \tag{6}$$

This is the second order *Adams Bashforth* scheme. It achieves second order accuracy using lagged values of f rather than lagged values of X.

3 Adams Bashforth Methods

A *linear multistep method* of order s gets the new X using s trajectory and function values. It gets X_{n+s} as a linear combination of s most recent trajectory values, $X_{n+s-1}, \cdots, X_n$, and corresponding function values $F_k = f(X_k)$.

The *Adams Bashforth* method of order s does this by (approximately) integrating the ODE (1) over over the time interval from the most recent computed time to the next time. The formula for the exact solution is

$$x(t_{n+s}) = x(t_{n+s-1}) + \int_{t_{n+s-1}}^{t_{n+s}} f(x(t))\, dt \ . \tag{7}$$

Adams methods replace the integrand $f(x(t))$ with an interpolating polynomial $p(t)$, of degree $s-1$ that satisfies the s interpolation conditions

$$p(t_k) = f(X_k) \ , \quad k = n, n+1, \cdots, n+s-1 \ .$$

There is a unique polynomial of degree $s-1$ that satisfies these conditions.[2] This interpolation is a linear operation in the sense that p depends linearly on the interpolation values $f(X_k)$, so the integral does too. This implies that there are numbers b_j so that

$$\int_{t_{n+s-1}}^{t_n} p(t)\, dt = \Delta t \left[\, b_{s-1}\, f(X_{n+s-1}) + \cdots + b_0\, f(X_n) \, \right] \ . \tag{8}$$

For example, the second order scheme (6) has $s = 2$, and $b_1 = \frac{3}{2}$, and $b_0 = -\frac{1}{2}$. Adams Bashforth methods use the right side of (8) instead of the exact integral on the right side of (7), which leads to

$$X_{n+s} = X_{n+s-1} + \Delta t \left[\, b_{s-1} f(X_{n+s-1}) + \cdots + b_0 f(X_n) \, \right] \ . \tag{9}$$

This is the Adams Bashforth method of order s.

[2] The theory of polynomial interpolation is described in the book *Numerical Methods* by Dahlquist and Björk.

The right side of (8) has an overall factor Δt and coefficients b_j that do not depend on Δt. You can see that it should be this way by first asking what would happen with $\Delta t = 1$. This could be used to define the numbers b_j. When you rescale time by Δt, the integration interval shrinks by a factor of Δt and $p(t)$ scales in the same way. The only change is the Δt factor on the right of (8).

The second order method (6) was derived using extrapolation rather than integration. It has order $s = 2$ because, in our numbering convention, it gets X_{n+2} from X_{n+1} and the function values $f(X_{n+1})$ and $f(X_n)$. The function $\widetilde{F}(t)$ in (5) is the interpolating polynomial. It has degree $s-1 = 1$ (linear). The integral of the constant term of $\widetilde{F}$ is $\Delta t f(X_n)$. The integral of the linear term is the area of the triangle with the slope on top s_n, which is

$$\frac{1}{2}\Delta t^2 s_n = \frac{1}{2}\Delta t \left[f(X_n) - f(X_{n-1}) \right] .$$

Combining these gives the second order Adams Bashforth (6). The integration approach here leads to Adams Bashforth methods of any order, s.

It is also possible to derive the values of the coefficients b_j from the point of view of Section 2. We explain this using the numbering convention of Section 2, in which we get X_{n+1} using X_n and $f(X_n)$, $\cdots$, $f(X_{n-s+1})$. We look for a formula for the finite difference

$$\frac{x(t+\Delta t) - x(t)}{\Delta t}$$

in terms of the true derivative at lagged times, and you want the formula to be accurate to a certain order:

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = b_{s-1}\,\dot{x}(t) + \cdots + b_0\,\dot{x}(t-(s-1)\Delta t) + O(\Delta t^s) . \qquad (10)$$

The ODE allows us to replace derivatives $\dot{x}$ with corresponding values of f, which puts (10) in the Adams Bashforth form

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = b_{s-1}\,f(x(t)) + \cdots + b_0\,f(x(t-(s-1))\Delta t) + O(\Delta t^s) .$$

The algebra for finding the coefficients in (10) is an example of the method of undetermined coefficients for finding finite difference approximations of all kinds.[3] You assume an approximation of a certain order with unknown coefficients, insert Taylor approximations to the appropriate order, derive a set of equations for the coefficients by matching terms (see example), and then solve the equations for the unknown coefficients.

We simplify the LaTeX version of the algebra by changing notation. The independent variable will be called x instead of t. The function will be $u(x)$ instead of $x(t)$. The step will be h instead of Δt. Derivatives will be u' and u'' instead of $\dot{x}$, $\ddot{x}$, etc.

[3] Chapter 7 of *Numerical Methods* by Dahlquist and Björk has a beautiful discussion of this kind of thing.

We first re-derive the second order method of Section 2 in this framework. This calls for an approximation of the form

$$\frac{u(x+h)-u(x)}{h} = b_1 u'(x) + b_0 u'(x-h) + O(h^2) \; . \tag{11}$$

We use Taylor approximations, and write u for $u(x)$, etc.:

$$\begin{aligned} u(x+h) &= u + hu' + \frac{h^2}{2}u'' + O(h^3) \\ u'(x-h) &= u' - hu'' + O(h^2) \; . \end{aligned}$$

We substitute these in and calculate:

$$\begin{aligned} \frac{u + hu' + \frac{h^2}{2}u'' + O(h^3) - u}{h} &= b_1 u' + b_0 \left(u' - hu'' + O(h^2) \right) \\ u' + \frac{h}{2}u'' + O(h^2) &= b_1 u' + b_0 u' - h\, b_0 u'' + O(h^2) \end{aligned}$$

In order for this formula to be true, the coefficients of various powers of h on the left and right sides have to agree. Specifically, the terms independent of h (order h^0) and h^1 terms on the two sides must agree separately. It happens that (no coincidence) the order h^0 terms both have coefficient u' and the order h^1 terms both have u''. Thus, the accuracy condition are

$$\text{coefficient of } u' : \qquad 1 = b_1 + b_0 \tag{12}$$

$$\text{coefficient of } u''h : \qquad \frac{1}{2} = -b_0 \tag{13}$$

Equation (13) gives

$$b_0 = -\frac{1}{2} \; .$$

Substituting this into equation (12) gives

$$b_1 = \frac{3}{2} \; .$$

This gives the approximation

$$u(x+h) = u(x) + h \left[\frac{3}{2}u'(x) - \frac{1}{2}u'(x-h) \right] \; .$$

This is yet another derivation of the second order Adams Bashforth method (6).

The third order Adams Bashforth method is built from an approximation like (11) that uses one more lag and is accurate to one more order:

$$\frac{u(x+h)-u(x)}{h} = b_2 u'(x) + b_1 u'(x-h) + b_0 u'(x-2h) + O(h^3) \; .$$

Working this out requires Taylor approximations of one more order. On the left side is:

$$\frac{u + hu' + \frac{h^2}{2}u'' + \frac{h^3}{6}u''' + O(h^4) - u}{h} = u' + \frac{h}{2}u'' + \frac{h^2}{6}u''' + O(h^3) \; .$$

On the right side is (check this carefully)

$$\begin{aligned} & b_2 u' + b_1 \left(u' - hu'' + \frac{h^2}{2}u''' \right) + b_0 \left(u' - 2hu'' + 2h^2 u''' \right) \\ &= \left(b_2 + b_1 + b_0 \right) u' + \left(-b_1 - 2b_0 \right) hu'' + \left(\frac{1}{2}b_1 + 2b_0 \right) h^2 u''' + O(h^3) \; . \end{aligned}$$

We equate corresponding terms from the left and right to get three accuracy conditions:

$$\text{coefficient of } u' : \qquad 1 = b_2 + b_1 + b_0 \tag{14}$$

$$\text{coefficient of } u''h : \qquad \frac{1}{2} = -b_1 - 2b_0 \tag{15}$$

$$\text{coefficient of } u'''h^2 : \qquad \frac{1}{6} = \frac{1}{2}b_1 + 2b_0 \tag{16}$$

We solve the equations by elimination. Adding (15) and (16) eliminates b_0 and leaves

$$\frac{2}{3} = -\frac{1}{2}b_1 \implies b_1 = -\frac{4}{3} \; .$$

Substitute the b_1 value back into (15), and you get

$$\frac{1}{2} = \frac{4}{3} - 2b_0 \implies b_0 = \frac{5}{12} \; .$$

Finally, use the known b_1 and b_0 in (14), and you get

$$1 = b_2 - \frac{4}{3} + \frac{5}{12} \implies b_2 = \frac{23}{12} \; .$$

The finite difference approximation is

$$\frac{u(x+h) - u(x)}{h} = \frac{23}{12}u'(x) - \frac{4}{3}u'(x-h) + \frac{5}{12}u'(x-2h) + O(h^4) \; .$$

You translate this back to the $x(t)$ and $n+s$ notation and get

$$x(t_{n+3}) = x(t_{n+2}) + \Delta t \left(\frac{23}{12}f(x(t_{n+2})) - \frac{4}{3}f(x(t_{n+1})) + \frac{5}{12}f(x(t_n)) \right) + O(\Delta t^4) \; .$$

The corresponding third order Adams Bashforth method is

$$X_{n+3} = X_{n+2} + \Delta t \left(\frac{23}{12}f(X_{n+2}) - \frac{4}{3}f(X_{n+1}) + \frac{5}{12}f(X_n)) \right) \; . \tag{17}$$

You can find the same third order Adams Bashforth method (17) using the interpolating polynomial method. First, you write a formula for the three point quadratic interpolating polynomial. This is a quadratic generalization of the two point linear interpolating function (5). Then you integrate each of the three terms over the interval $[t_{n+2}, t_{n+3}]$. Some more algebra would put the result in the form (17).

These calculations are typical of what people do to find high order methods. The calculations take hours or days and the coefficients turn out to be complicated rational numbers. This can be automated using symbolic algebra software, also called *computer* algebra.

4 General linear multistep methods

The general *linear multi-step method* gets the new value using lagged values of X and f. We saw that Adams Bashforth methods use lags only on f. The second order leapfrog method (3) uses lags on X but not f. The general form that for an order s method that uses both X and f lags is

$$X_{n+s} + a_{s-1}X_{n+s-1} + \cdots + a_0 X_n = \Delta t \, [\, b_{s-1} f(X_{n+s-1}) + \cdots + b_0 f(X_n) \,] \quad (18)$$

The leapfrog method (3) has this form with $s = 2$, $a_1 = 0$, $a_0 = -1$, $b_1 = 2$, and $b_0 = 0$.

The *residual* (also called *local truncation error*) is the amount by which the exact solution values $x(t_n)$ fail to satisfy the discrete equations (18). The method has *formal* order of accuracy p if the residual has order Δt^{p+1}. In that case, we pull out the factor Δt^{p+1} and define the residual coefficient function $r(t, \Delta t))$ by

$$\begin{aligned} x(t_{n+s}) + \cdots + a_0 x(t_n) \; &= \Delta t \, [\, b_{s-1} f(x(t_{n+s-1})) + \cdots + b_0 f(x(t_n)) \,] \\ &\quad + \Delta t^{p+1} r(t_n, \Delta t) \; . \end{aligned}$$

This is the definition of r, since x is the solution of the ODE. Taylor asymptotic calculations (long ones) give asymptotic expansions

$$r(t, \Delta t) \sim r_0(t) + \Delta t \, r_1(t) + \cdots \; .$$

The order of accuracy is exactly p if r has this form and r_0 is not zero. For example, when we say a method is first order we normally mean to imply that it is not second order.

Some tricks from Section 3 help with the algebra. One is to use the ODE and substitute $\dot{x}$ for $f(x)$ on the right. This gives

$$x(t_{n+s}) + \cdots + a_0 x(t_n) \; = \; \Delta t \, [\, b_{s-1} \dot{x}(t_{n+s-1}) + \cdots + b_0 \dot{x}(t_n) \,] + \Delta t^{p+1} r(t_n, \Delta t) \; . \quad (19)$$

Then we ask that (19) should be true for any smooth function $x(t)$ without referring to the ODE that $x(t)$ satisfies. Also, although the solution $x(t)$ to an

ODE usually has many components, (19) does not couple different components. Therefore, we may assume that $x(t)$ has just one component. Next, (19) holds for any smooth $x(t)$ if it holds exactly (with $r = 0$) when x is a power of t up to t^p. You can see this using the Taylor approximation of $x(t + k\Delta t)$ up to order p. The relation (19) will be satisfied if the coefficients of Δt^m match for m up to p. This happens if (and only if) it is true when $x(t)$ is exactly a power or t up to t^p. Finally, $x(t)$ is a power of t, you may take $t = 0$ and $\Delta t = 1$. All of these simplifications imply we need to get $r = 0$ when

$$\begin{aligned} x(t_{n_k}) &= \quad k^m \\ \dot{x}(t_{n+k}) &= mk^{m-1} \\ m &= 0, \cdots, p \; . \end{aligned}$$

With these simplifications, the conditions are

$$\begin{array}{lrcl} m = 0: & 1 + a_{s-1} & + \cdots + a_1 + a_0 = & 0 \\ m = 1: & s + a_{s-1}(s-1) + \cdots + a_1 & = & b_{s-1} + \cdots + b_0 \\ & & \text{etc.} & \end{array}$$

These equations are systematic and easy to formulate and solve. By contrast, the accuracy conditions for Runge Kutta methods are complicated and do not seem that systematic.

There are *characteristic polynomials* associated to linear multistep methods

$$\rho(z) = z^s + a_{s-1}z^{s-1} + \cdots + a_0 \tag{20}$$

$$\sigma(z) = \quad b_{s-1}z^{s-1} + \cdots + b_0 \; . \tag{21}$$

These are convenient ways to describe the deeper theory of linear multi-step methods. For example, the first two accuracy conditions may be written as

$$\begin{aligned} \rho(1) &= 0 \\ \rho'(1) &= \sigma(1) \; . \end{aligned} \tag{22}$$

5 Convergence and accuracy theory

Convergence and accuracy theory for linear multistep methods, like it was for Runge Kutta methods, is about showing the error is on the order of the residual. This was always true for Runge Kutta methods, but not here. Linear multistep methods can be *unstable*. An unstable method can have errors that amplify the residual by factors of g^n with $g > 1$. This allows the error to look like

$$g^n \Delta t^{p+1} r \; .$$

If you fix $T = t_n = n\Delta t$ and do some algebra, this becomes

$$\frac{g^n}{n^{p+1}} T^p r \; . \tag{23}$$

The fraction goes to infinity as $n \to \infty$ no matter the order of accuracy, p. We will show that linear multistep methods converge with an error related to the residual if the method is *stable*.

A scheme is *unstable* if solutions of the discrete equations (18) grow too rapidly. This growth is not related to growth of solutions of the ODE. Solutions of an ODE may grow in time, but that growth is not a function of Δt. By contrast, the growth factor (23) goes to infinity as $\Delta t \to 0$ for any positive T. For this reason, you can try to understand the stability/instability of a linear multistep method by asking what it does for the trivial ODE

$$\dot{x} = 0 \ .$$

A method that cannot solve this ODE correctly will not be much use on harder problems. When $\dot{x} = 0$, the b_k coefficients are irrelevant. *Zero stability* is the condition that the numerical solution X_n remains bounded, as the solution of the ODE does, trivially.

Definition. A linear multi-step method is *zero stable* if there is a C so that solutions of the recurrence relation

$$X_{n+s} + a_{s-1} X_{n+s-1} + \cdots + a_0 X_n = 0 \tag{24}$$

satisfy, for all $n > 0$,

$$|X_n| \leq C \max \{ \, |X_0| \, , \cdots , |X_{s-1}| \, \} \ . \tag{25}$$

The theory of linear recurrence relations gives a criterion for this, which involves the *characteristic polynomial* of the recurrence. This happens to be the ρ characteristic polynomial (20) of the method (check this).[4] If z_k is a *root* of ρ, that is, $\rho(z_k) = 0$, then $X_n = z_k^n$ satisfies the recurrence relation (24), because z_k^n factors out. If $|z_k| > 1$ then $X_n = z_k^n$ makes the left side of (25) go to infinity while the right side obviously does not. Thus: a root of the characteristic polynomial outside the unit circle in the complex plane implies that the method is not zero stable.

You might object that the definition of zero stability (25) is about real numbers and the root z_k might not be real. However, if z_k is not real then the real part and imaginary parts are real solutions

$$X_n = \mathrm{Re}(z_k^n) \ , \quad \text{or } X_n = \mathrm{Im}(z_k^n) \ .$$

If $|z_k^n| \to \infty$, then either the real part or the imaginary part also blows up, because

$$|z|^2 = [\mathrm{Re}(z)]^2 + [\mathrm{Im}(z)]^2 \ .$$

If there is an "unstable" complex solution to the recurrence relation (24), then there is an unstable real solution.

[4] The strange indexing in (18) used to describe linear multi-step methods was chosen to make the indexing of the characteristic polynomial have coefficient corresponding to power: $a_k z^k$.

Roots on the unit circle, $|z_k| = 1$, $z_k = e^{i\theta}$, have $|z_k^n| = 1$ for all n. Zero stability does not forbid roots of the characteristic polynomial on the unit circle. In fact, the consistency condition (22) shows that any linear multistep method, even a first order accurate one, has a root $z_1 = 1$ on the unit circle. What makes stability subtle is the possibility of double roots of ρ on or inside the unit circle. Multiple roots inside are OK while roots on the unit circle are bad.

Theorem. The recurrence relation is zero stable in the sense of (25) if and only if all roots $\rho(z_k) = 0$ have $|z_k| \leq 1$ and if all roots with $|z_k| = 1$ are simple. That is: all roots must be in the (closed) unit circle of the complex plane (more properly, unit *disk*) and roots on the unit circle must be simple.

For example, all Adams Bashforth methods are stable in this sense. With s lags, they have

$$\rho(z) = z^s - z^{s-1} = (z-1)z^{s-1} \ .$$

This polynomial has a root on the unit circle, $z_1 = 1$ and the rest of the roots $z_k = 0$. This is not a simple root, but is is not on the unit circle.

The leapfrog method (3) has $\rho(z) = z^2 - 1$. The roots are $z_1 = 1$ and $z_2 = -1$. These are simple roots on the unit circle.

The theorem an "if" and "only if" theorem. If there is a bad root then the recurrence is unstable. If there are no bad roots than the recurrence is stable in the sense (25). We saw that a root $|z_k| > 1$ makes the recurrence unstable. A multiple root with $|z_k| = 1$ leads to solutions with polynomial rather than exponential growth as a function of n. We will see, for example, that if z_k is a double root, then $X_n = nz_k^n$ satisfies the recurrence. either the real or imaginary part (or both) of this sequence grows linearly with n. These are *weak* instabilities. A method that is only weakly unstable might work for some problems, but it is like a Rube Goldberg device in that you have to be lucky for it to work.

The main point of the theorem is that the behavior of solutions like z_k^n and nz_k^n, which are *power law* and *generalized power law* solutions, determines the behavior of all solutions of the recurrence relation. This is because power law and generalized power law solutions form a basis for the set of all solutions.

5.1 An unstable method

You don't have to look far to find an unstable linear multi-step method. Just look for the explicit three step ($s = 2$) method with the highest order of accuracy for such a method. That one is unstable. Here's the math.

The method in question would have the form

$$X_{n+2} + a_1 X_{n+1} + a_0 X_n = \Delta t \left[b_1 f(X_{n+1}) + b_0 f(X_n) \right] \ .$$

This method doesn't have a name because nobody uses it, because it's unstable.

As in Section 3, we find the coefficients using the method of undetermined coefficients and Taylor approximations. We also make the algebra quicker using

$u(x)$ instead of $x(t)$. After some trial and error, it seems that the best order of accuracy is $p = 3$. We find coefficients so that

$$u(x+h) + a_1 u(x) + a_0 u(x-h) = h\,[\,b_1 u'(x) + b_0 u'(x-h)\,] + O(h^4)\ . \quad (26)$$

The Taylor expansions for this need to include up to h^3 terms for u and up to h^2 terms for u' (because the u' terms have an h pre-factor). Again, we write u for $u(x)$, etc. The calculation starts with

$$u + hu' + \frac{h^2}{2}u'' + \frac{h^3}{6}u''' + a_1 u + a_0\left(u - hu' + \frac{h^2}{2}u'' - \frac{h^3}{6}u''' \right)$$
$$= h\left[b_1 u' + b_0\left(u' - hu'' + \frac{h^2}{2}u''' \right) \right] + O(h^4)\ .$$

Next, we equate coefficients of powers of h as before:

coefficient of u :	$1 + a_1 + a_0 = 0$
coefficient of hu' :	$1 - a_0 = b_1 + b_0$
coefficient of $h^2 u''$:	$\frac{1}{2} + \frac{1}{2}a_0 = -b_0$
coefficient of $h^3 u'''$:	$\frac{1}{6} - \frac{1}{6}a_0 = \frac{1}{2}b_0$

The last two equations give $a_0 = -5$ and $b_0 = 2$. Then, the first two equations give $a_1 = 4$ and $b_1 = 4$. The method (translating back to the appropriate x and t variables)

$$X_{n+2} + 4X_{n+1} - 5X_n = \Delta t[\,4f(X_{n+1}) + 2f(X_n)\,]\ . \quad (27)$$

People would use this method and it would have a name if it were stable, because it would give the same order of accuracy of the Adams Bashforth method with fewer lags.

The characteristic polynomial equation for zero stability is

$$\rho(z) = z^2 + 4z - 5 = 0\ ,\quad z = -2 \pm 3\ .$$

The root $z = 1$ is a check on the algebra, since the characteristic polynomial for a linear multi-step method always has $z = 1$ as a root. The other root is $z = -5$, which is (far) outside the unit circle.

5.2 Solutions of linear recurrence relations

We just saw that roots of the characteristic polynomial give solutions of the recurrence relation. It would be nice if all solutions were linear combinations of these basic solutions, and that is almost true, but not quite. It is true if the roots are distinct, but the story is more complicated if there are double roots (or higher).

Double roots and higher might seem unlikely to occur in "real problems" because they are not *generic*. If you choose the coefficients by a continuous probability density, then the probability of getting a ρ with a multiple root is zero. But "real problems" are not necessarily chosen at random and we have to face the possibility of multiple roots.

Suppose z_k is a double root of ρ. This means that $\rho(z_k) = 0$ and $\rho'(z_k) = 0$. You can check that $\rho(z_k) = 0$ implies that $X_n = z_k^n$ satisfies the recurrence relation (24). You just factor out z^n:

$$\begin{aligned} z_k^{n+s} + a_{s-1} z_k^{n+s-1} + \cdots + a_0 z_k^n &= \left[z_k^s + a_{s-1} z_k^{s-1} + \cdots + a_0 \right] z_k^n \\ &= \rho(z_k) z_k^n \\ &= 0 \ . \end{aligned}$$

If $\rho'(z_k) = 0$ too, then you can differentiate this to get

$$\begin{aligned} (n+s) z_k^{n+s-1} + a_{s-1}(n+s-1) z_k^{n+s-2} + \cdots + a_0 n z_k^{n-1} \\ &= \rho'(z_k) z_k^n + n \rho(z_k) z_k^{n-1} \\ &= 0 \ . \end{aligned}$$

You can multiply both sides by z_k and see that the sequence

$$X_n = n z_k^n$$

also satisfies the recurrence relation (24). You can continue this process to see that $n^2 z_k^2$ satisfies the recurrence relation if $\rho''(z_k) = 0$, etc.

The root z_k has multiplicity m if you can factor out $(z - z_k)^m$ but not $(z - z_k)^{m+1}$. This is equivalent to

$$\left(\frac{d}{dz} \right)^r \rho(z_k) = 0 \ , \quad \text{for } r = 0, \cdots, m-1 \ , \quad \text{but } \left(\frac{d}{dz} \right)^m \rho(z_k) \neq 0 \ .$$

This is equivalent to the recurrence relation having solutions

$$X_n = n^r z_1^n \tag{28}$$

($r = 0, 1, \cdots, m-1$, but not $r = m$) that satisfy the recurrence relation. The solutions with $r = 0$ are *power law* solutions. The solutions with $r = 1, 2,$, etc., are *generalized* power law solutions. Section 5.3 will show that power law solutions correspond to eigenvectors of the *companion matrix* (30), and generalized power law solutions correspond to generalized eigenvectors. It is a theorem in linear algebra that a family of eigenvectors and generalized eigenvectors of a matrix forms a basis. Therefore the set of power law and generalized power law solutions forms a basis for the set of all solutions of the recurrence.

5.2.1 Jordan blocks

A *Jordan block* of size m with eigenvalue λ is the matrix with λ on the diagonal, 1 on the super-diagonal, and the rest zero

$$J(\lambda) = \begin{pmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \\ & & & \lambda & 1 \\ 0 & \cdots & & 0 & \lambda \end{pmatrix} \tag{29}$$

A *Jordan form* of a matrix is a block matrix with diagonal blocks of the form $J_k(\lambda_k)$ and size m_k. It is a theorem of linear algebra that any $s \times s$ matrix has a non-singular matrix V so that

$$V^{-1}AV = \begin{pmatrix} J_1(\lambda_1) & 0 & 0 \\ 0 & J_2(\lambda_2) & 0 \\ 0 & 0 & \ddots \end{pmatrix} .$$

If all the sizes m_k are equal to one, then the columns of V are eigenvectors of A and the λ_k are the corresponding eigenvalues. If A has "non-trivial Jordan structure" (some $m_k > 1$), then some of the columns of V are *generalized* eigenvectors.

We illustrate generalized eigenvectors with a Jordan block of size $m = 3$:

$$J = \begin{pmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{pmatrix} .$$

The eigenvectors and generalized eigenvectors are

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} , \quad v_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} , \quad v_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} .$$

You can calculate that

$$Jv_1 = \begin{pmatrix} \lambda \\ 0 \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \lambda v_1 .$$

Thus, v_1 is an eigenvector of J. Next:

$$Jv_2 = \begin{pmatrix} 1 \\ \lambda \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \lambda v_2 + v_1 .$$

Similarly,

$$Jv_3 = \lambda v_3 + v_2 .$$

These relations may be written in the form

$$(J-\lambda I)v_1 = 0\ ,\ \ (J-\lambda I)v_2 = v_1\ ,\ \ (J-\lambda I)v_3 = v_2\ .$$

The generalized eigenvectors v_i are not eigenvectors (except v_1). Instead they form a *Jordan chain*. You have to apply $J-\lambda I$ to v_i several times to get to zero. A diagonalizable matrix A cannot have non-trivial Jordan chains. If A is diagonalizable, λ is any number, $x \neq 0$ and $(A-\lambda I)x \neq 0$, then $(A-\lambda I)^p x \neq 0$ for any $p = 1, 2, \cdots$. You find the Jordan structure of a matrix by finding the eigenvalues, the actual eigenvectors (as opposed to generalized eigenvectors), and the Jordan chains corresponding to those eigenvectors and eigenvalues.

5.3 Roots as eigenvalues of the companion matrix

It simplifies the discussion of recurrence relations to reformulate a degree s recurrence relation for a single component sequence x_n as a degree 1 recurrence relation for an s component sequence Y_n. The vector Y_n is defined in terms of the scalar sequence X_n by including X values at multiple times. These normally would be called "lags", but the recurrence relation (24) looks into the future rather than the past. Therefore we define:

$$Y_n = \begin{pmatrix} X_{n+s-1} \\ X_{n+s-2} \\ \vdots \\ X_{n+1} \\ X_n \end{pmatrix}\ .$$

You can check that the scalar recurrence relation (24) is equivalent to the vector recurrence

$$\begin{pmatrix} X_{n+s} \\ X_{n+s-1} \\ \vdots \\ X_{n+2} \\ X_{n+1} \end{pmatrix} = \begin{pmatrix} -a_{s-1} & -a_{s-2} & \cdots & -a_1 & -a_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} X_{n+s-1} \\ X_{n+s-2} \\ \vdots \\ X_{n+1} \\ X_n \end{pmatrix} \tag{30}$$

In vector form, this may be written as

$$Y_{n+1} = AY_n\ . \tag{31}$$

The matrix A on the left is the *companion matrix* of the recurrence relation. The top row of the matrix A on the left represents the scalar recurrence (24) (in the form $X_{n+s} = -a_{s-1}X_{n+s-1} - \cdots - a_0 X_n$). The second row represents X_{n+s-1} as the second component of Y_{n+1} and the first component of X_n. The bottom row represents X_{n+1} as the last component of Y_{n+1} and the next to last component of X_n. The companion matrix has the coefficients of ρ on the top row. The rest of A zeros except for ones on the first sub-diagonal.

We have seen that power law solutions of the scalar recurrence (24) correspond to roots of the characteristic polynomial (**??**). They also correspond to eigenvalues and eigenvectors of the companion matrix in (30). The scalar sequence $X_n = z^n$ is encoded into the vector sequence

$$Y_n = \begin{pmatrix} z^{n+s-1} \\ z^{n+s-2} \\ \vdots \\ z^n \end{pmatrix} .$$

This sequence satisfies

$$Y_{n+1} = zY_n \ .$$

If $X_n = z^n$ satisfies the scalar recurrence, then Y_n satisfies the vector recurrence (31) so

$$zY_n = AY_n \ .$$

This, Y_n is an eigenvector corresponding to the eigenvalue z. If there are n distinct roots of ρ, then there are n distinct eigenvalues of A and the corresponding eigenvectors form a basis. In particular, the eigenvectors are linearly independent.

This process of going from a high order scalar relation to a first order system is similar to the process of reformulating a ODE with high order derivatives as a larger system with only first order derivatives. The language of matrices and general linear algebra, once you've converted to it, turns out to be simpler, easier to understand, and better for computing. In fact, computers find roots of a polynomial (**??**) by forming the companion matrix and finding its eigenvalues.

5.4 Lyapunov's theorem for stable matrices

Lyapunov's theorem is a technical tool that demonstrates that linear stability is robust. A small perturbation of a stable system should also be stable. Lyapunov showed this, for strongly stable linear systems, by constructing a sense in which any strongly stable dynamics is a *contraction*. A small perturbation of a contraction mapping is also a contraction.

A square matrix A is *strongly stable* for discrete time dynamics if the eigenvalues λ_j of A are all properly inside the unit circle in the complex plane:

$$|\lambda_j| < 1 \ , \quad \text{for all } j \ .$$

For example, the matrix

$$A = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

has eigenvalues

$$\lambda = \frac{1}{2} \pm \frac{i}{2} \ , \quad |\lambda| = \sqrt{\frac{1}{4} + \frac{1}{4}} = \frac{1}{\sqrt{2}} < 1 \ .$$

The linear transformation that A represents is a *contraction* in the norm $\|\cdot\|$, with contraction factor $r < 1$ if

$$\|Ax\| \leq r \|x\| \ , \quad \text{for all } x \ .$$

For example, the A above has

$$A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(x_2 + x_1) \\ \frac{1}{2}(x_2 - x_1) \end{pmatrix} \ .$$

If $y = Ax$, and $\|x\| = \sqrt{x_1^2 + x_2^2}$, then A is a contraction with contraction factor $\frac{1}{\sqrt{2}}$ because

$$\begin{aligned} \|y\| &= \sqrt{\frac{1}{4}(x_2 + x_1)^2 + \frac{1}{4}(x_2 + x_1)^2} \\ &= \sqrt{\frac{1}{2}\left(x_1^2 + x_2^2\right)} \\ &= \frac{1}{\sqrt{2}} \|x\| \ . \end{aligned}$$

Any contraction is strongly stable (why?), but it is possible that a strongly stable matrix is not a contraction. It is possible that a matrix is a contraction in one norm and not in another. The following matrix is strongly stable but not a contraction in the 2 norm:

$$A = \begin{pmatrix} -\frac{1}{2} & 1 \\ -1 & \frac{3}{2} \end{pmatrix}$$

This A has a double eigenvalue $\lambda = \frac{1}{2}$, which makes A strongly stable. But

$$A \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ -1 \end{pmatrix} \ , \quad \text{and} \quad \left\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\|_2 = 1 \ , \quad \text{but} \quad \left\| \begin{pmatrix} -\frac{1}{2} \\ -1 \end{pmatrix} \right\|_2 = \frac{\sqrt{5}}{2} > 1 \, .$$

We will soon show that there is a norm in which this A is a contraction.

The contraction property makes stability robust to perturbations. Let A is a contraction and B be any matrix. If ϵ is small enough then $A + \epsilon B$ is also a contraction. The following calculation shows this:

$$\begin{aligned} \|(A + \epsilon B)\, x\| &= \|Ax + \epsilon Bx\| \\ &\leq \|Ax\| + \epsilon \|Bx\| \\ &\leq r \|x\| + \epsilon \|B\| \|x\| \\ &= (r + \epsilon \|B\|) \|x\| \\ &= \widetilde{r} \|x\| \ . \end{aligned}$$

No matter what $\|B\|$ is, as long as $r < 1$ and if ϵ is small enough, $A + \epsilon B$ is a contraction because

$$\widetilde{r} = r + \epsilon \|B\| < 1 \, .$$

Lyapunov's theorem[5] states that if A is strongly stable then there is a positive definite quadratic form Q and an $r < 1$ so that

$$Q(Ax) \leq r^2 Q(x) \ . \tag{32}$$

Any positive definite quadratic form defines a norm

$$\|x\|_Q = \sqrt{Q(x)} \ .$$

We saw (when discussing variational formulations of the continuous and discrete Laplace equations) that any positive definite quadratic form is represented by a positive definite symmetric matrix M in the sense that

$$Q(x) = x^T M x \ .$$

Thus, the Q norm is also the M norm

$$\|x\|_M = \sqrt{x^T M x} \ .$$

It is convenient to go back and forth between the abstract quadratic form and the more concrete SPD matrix M.

The Q norm of Lyapunov's theorem may be defined in terms of the 2 norm of the *trajectory* defined by A and x. The trajectory is the sequence $x, Ax, A^2x, \cdots$. The 2 norm is $\|x\|_2^2 = x^T x$. The 2 norm of the trajectory is

$$Q(x) = \|x\|_2^2 + \|Ax\|_2^2 + \left\|A^2 x\right\|_2^2 + \cdots \ . \tag{33}$$

Each of the terms on the right is a quadratic form:

$$\|A^n x\|_2^2 = (A^n x)^T A^n x = x^T \left[(A^n)^T A^n \right] x \ .$$

Thus, the sum is a quadratic form. This leaves two questions: Why does the sum (33) converge? and: Why does it make A a contraction?

The second question is easier to answer. The trajectory starting from Ax is $Ax, A^2 x \cdots$. If the sum converges,

$$Q(Ax) = \|Ax\|_2^2 + \left\|A^2 x\right\|_2^2 + \cdots \ . \tag{34}$$

In particular, Q norm of Ax is less than the Q norm of x (i f$x \neq 0$) because

$$Q(Ax) = Q(x) - \|x\|_2^2 \ . \tag{35}$$

This alone does not quite make A a contraction in $\|Q\|_Q$ because the same contraction factor r has to work for any x. This is true in finite dimensions (by "compactness of the unit ball") but possibly otherwise.

[5] Lyapunov was a productive mathematician and there is more than one theorem called *Lyapunov's theorem.*

The trick is to show that subtracting $\|x\|_2^2$ makes Q go down "a lot", relative to $Q(x)$. In other words, $\|x\|_2^2$ is not too small relative to Q. To see how small it can be relative to Q, look at the minimum of the ratio

$$m = \min_{x \neq 0} \frac{\|x\|_2^2}{Q(x)} \ .$$

A vector that makes the ratio as small as possible is the eivenvector of M with largest eigenvalue μ_n.

$$Mv = \mu_n v \implies Q(v) = v^T M v = \mu_n v^T v$$

Thus, $m = \frac{1}{\mu_n}$, and

$$\|x\|_2^2 \geq \frac{1}{\mu_n} Q(x) \ .$$

We get

$$Q(Ax) \leq \left(1 - \frac{1}{\mu_n}\right) Q(X) \ ,$$

So, in the notation of the contraction condition (32)

$$\|Ax\|_Q \leq r \, \|x\|_Q \ , \ \ r = \sqrt{1 - \frac{1}{\mu_n}} < 1 \ .$$

5.5 Applying Lyapunov

The companion matrix for the linear recurrence has an eigenvalue equal to one, so it does not satisfy the hypotheses of Lyapunov's theorem.

6 References

On the FFT

The basic method for $n = 2^p$ is described in these two excellent basic books:
Numerical Methods, Anne Greanbaum, Timothy Chartier, p. 411
Numerical Methods, Germund Dahlquist, Åke Björk, p. 413

A good online reference for good modern FFT software and general algorithms
http://fftw.org/