MATH 269A Notes

Ryan Anderson

2023-10-02

Lecture 1: Euler's Method

Setup and Estimation Error

Given an initial value problem of an ODE where we have $\frac{dy}{dt} = f(t, y), y(t_n) = y_0, t \in [t_0, T]$, we can obtain approximate solutions using Euler's method.

In pseudocode, Euler's method looks like the following. For $n = 1, 2, 3, \ldots$

$$y_n = y_{n-1} + h * f(y_{n-1}, t_{n-1})$$

Given some pretty mild assumptions on $f(t, y)$ we can get good estimation results. In particular, given such a function, we know that there exists N' and a constant C such that $\forall N > N'$,

$$\max_{0 \leq n \leq N} (|y_n - y(t_n)|) \leq Ch$$

Thus, choosing a number of steps N is equivalent to choosing a timestep h.

Since our bound is of the form Ch, we call Euler's method a first-order process.

Generalizations

Generalizations of Euler's method lead to different families of numerical methods. As an example, we could take two previous points, y_{n-1}, y_{n-2} instead of one and fit a polynomial between them to predict our next point y_n.

Discussion 1: Systems of ODEs

Converting to Autonomous ODEs

Consider the ordinary differential equation $g(t, y, y', y'', \ldots, y^{(n)}) = 0$. This is an nth-order ODE, with $y^{(n)} = F(t, y, \ldots, y^{(n-1)})$.

A first-order ODE has $y' = f(t, y)$, and if f has no dependence on t, then we call the ODE autonomous.

In a system of ODEs, our goal is to find vector-valued functions having

$$\frac{d}{dt}\vec{y}^{(n)}(t) = \vec{F}(t, \vec{y}, \ldots, \vec{y}^{(n-1)})$$

Any system of ODEs can be made autonomous.

Separation of Variables

Consider a setup

$$\frac{dy}{dt} = f(y)g(t)$$

We can separate this equation into two sides which depend only on one variable.

$$\frac{1}{f(y)}\frac{dy}{dt} = g(t)$$

Then, integrating with respect to t, we get

$$\int \frac{1}{f(y)}\frac{dy}{dt}dt = \int g(t)dt \rightarrow \int \frac{1}{f(y)}dy = \int g(t)dt \rightarrow F(y) = G(t) + C$$

We can solve for C with initial conditions $y(0) = y_0$.

Example: Consider the equation $y' = 2t(1 + y^2), y(0) = 1$.

We separate to obtain

$$\frac{y'}{1+y^2} = 2t \rightarrow \int \frac{dy/dt}{1+y^2}dt = \int 2tdt$$

Solving these integrals, we get $arctan(y) = t^2 + C$. Plugging in the initial condition we can observe $arctan(1) = C \rightarrow C = k\pi + \frac{\pi}{4}, k \in \mathbb{Z}$.

Constant Coefficients

Consider the problem $\vec{y}' = A\vec{y}$, and let A be diagonalizable.

Then $A = PDP^{-1}$, and we can rewrite

$$\frac{d}{dt}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

This allows us to separate again, getting $\frac{dy_i}{dt} = \lambda_i y_i$, which results in

$$y_1 = C_1 e^{\lambda_1 t} y_2 = C_2 e^{\lambda_2 t}$$

In general for such a setup, let P be the matrix of eigenvectors of A. Then the solution to an ODE $\vec{y}' = A\vec{y}$ has the general form

$$\vec{y}(t) = P\vec{w}(t)$$

where $\vec{w}(t)$ is given by the solution to

$$\frac{d}{dt}\vec{w} = D\vec{w}$$

Lecture 2: Derivations of Euler's Method

Recall that Euler's method prescribes

We can derive Euler's method four ways: 1. via Taylor series, which leads to other Taylor series-based methods 2. via tangent approximation, which leads to Runge-Kutta methods 3. via numerical integration 4. via numerical differentiation, the latter two of which lead to multistep methods.

Derivation via Taylor Series

Consider a point $y_n \simeq y(t_n)$. The idea is to expand linearly around (t_{n-1}, y_{n-1}).

Then y_n is given by

$$y_n = y_{n-1} + \frac{dy}{dt}|_{t_{n-1}} h + \frac{d^2y}{dt^2}|_{t_{n-1}} \frac{h^2}{2} + \dots$$

Then let $f(t_{n-1}, y_{n-1}) = \frac{dy}{dt}|_{t_{n-1}}$. Then we have $y_n = y_{n-1} + hf(t_{n-1}, y_{n-1})$, as desired.

We generalize by relaxing the step in which we considered all higher-order terms as $O(h^2)$. To do this, we need to calculate more derivatives of $y(t)$.

$$\frac{dy}{dt} = f(t, y(t)) \rightarrow \frac{d^2y}{dt^2} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt}$$

and the $\frac{dy}{dt}$ is just f again.

Expanding to the second-order, we get another version of Euler's method obtained via Taylor series techniques:

$$y_n = y_{n-1} + hf(t_{n-1}, y_{n-1}) + \frac{h^2}{2}(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f(t_{n-1}, y_{n-1})) + O(h^3)$$

Derivation via Tangent Approximation

Consider the line tangent to the curve at y_{n-1}, and compute y_n via extending that line a distance of h. Then

$$\frac{y_n - y_{n-1}}{h} = \frac{dy}{dt}|_{t_{n-1}} = f(t_{n-1}, y_{n-1})$$

We can read off Euler's method directly here.

To generalize here, we need to consider more past information than is represented by the line tangent to the curve at y_n.

Consider the line tangent to the curve at (t_{n-1}, y_{n-1}) and extend the line to a new point at (t_n, y^*). Then take $f(t_n, y^*)$, which is the slope of the line tangent to $y(t)$ through the point y^*.

y^* is given by $y^* = y_{n-1} + hf(t_{n-1}, y_{n-1})$. Then we can obtain

$$y_n = y_{n-1} + \frac{h}{2}(f(t_{n-1}, y_{n-1}) + f(t_n, y^*))$$

This is Huen's method, which is based on taking an average of tangents. Extensions lead you to Runge-Kutta methods. We can prove that Huen's method is an improvement over Euler's method, but not yet.

Derivation via Numerical Integration

If we solve the original ODE, we get the following:

$$\frac{dy}{dt} = f(t, y(t)) \rightarrow \int_{t_{n-1}}^{t_n} y' ds = \int_{t_{n-1}}^{t_n} f(s, y(s)) ds$$

Via the First Fundamental Theorem, we get the LHS is just $y_n - y_{n-1}$. The RHS needs to be approximated via an integrator - funny that the solution to a differential equation requires a numerical integrator, but it is the case.

Basic setup for the numerical integrators is to use (left) Riemann sums. Create a partition of the interval $(t_{n-1}, t_n) = \cup_i (t_{n-i}, t_{n-i+1})$. Then we can approximate the integral as

$$\int_{t_{n-1}}^{t_n} f(s, y(s))ds \simeq (t_n - t_{n-1}) f(t_{n-1}, y_{n-1})$$

And since $h = t_n - t_{n-1}$, we recover Euler's method immediately. In the case of using left Riemann sums, this is the *forward Euler method*

With right Riemann sums the formula for Euler's method changes a bit - then we get $y_n = y_{n-1} + hf(t_n, y_n)$. This is the *backward Euler method*, sometimes called an implicit method to contrast with the explicit forward Euler.

Generalization comes via alternative approaches to appoximating our integral - using the trapezoidal rule, we get

$$\int_{t_{n-1}}^{t_n} f(s, y(s))ds \simeq (t_n - t_{n-1}) \frac{(f(t_{n-1}, y_{n-1}) + f(t_n, y_n))}{2}$$

The trapezoidal rule has a form analogous to the one we obtained via second-order Taylor expansion, and indeed has similar error.

Another alternative is via polynomial interpolation. Given the points (t_{n-i}, y_{n-i}), we can interpolate a polynomial $p(s)$ and recover

$$\int_{t_{n-1}}^{t_n} f(s, y(s))ds \simeq = \int_{t_{n-1}}^{t_n} p(s)ds$$

All these end up in statements of the below form:

$$y_n = y_{n-1} + h \sum_{j}^{m} \beta_j f(t_{n-j}, y_{n-j})$$

These are known as *linear multistep methods*, or LMM (sometimes also called Adam-Bashforth or Adam-Moulton methods).

Numerical Differentiation

Recall that for any t^*, the ODE tells us that

$$\frac{dy}{dt}|_{t^*} = f(t^*, y(t^*))$$

Instead of solving and approximating the integral as above, we can approximately evaluate the derivative here.

Pick $t^* = t_{n-1}$. We can approximate $\frac{dy}{dt}|_{t^*}$ via expressions of the form $(y_n - y_{n-1})/h$. This gives us the forward Euler method. If we instead pick $t^* = t_n$, we get the backward Euler method.

For backward methods, we can extend via polynomial interpolation and clever approximations thereof.

Let $t^* = t_n$. Then our RHS is given as $f(t_n, y_n)$. Our LHS will be given by $\frac{d}{dt}p(t)$, where $p(t)$ is a polynomial interpolant of $y(t)$.

These approaches end in methods of the form

$$\sum_{i=0}^{m} \alpha_i y_{n-i} = \beta_0 h f(t_n, y_n)$$

Note that the above is always implicit. For this reason they are referred to as *backward differentiation formulas* (BDF-methods), with the backward Euler method known sometimes as BDF-1.

Lecture 4: Error in Euler's Method

Error Evolution Between Steps

We can examine how error propagates across steps in Euler's method. Let the error at step n be given by $\bar{e}_n$. Then we have

$$\bar{e}_n = \bar{e}_{n-1} + h(f(\bar{y}_n) - f(y_{n-1})) + \tau_{n-1}$$

Taking absolute values, we can obtain an inequality:

$$|\bar{e}_n| \leq |\bar{e}_{n-1}| + h|f(\bar{y}_n) - f(y_{n-1})| + |\tau_{n-1}|$$

Whenever trying to bound a difference of functions, recall Lipschitz condition. $\frac{df}{dy}$ bounded implies that f is Lipschitz with $k = max_y|\frac{df}{dy}|$.

Then we get

$$|\bar{e}_n| \leq |\bar{e}_{n-1}| + h|k(\bar{y}_n - y_{n-1})| + |\tau_{n-1}|$$

But our error at the last step is precisely $\bar{y}_n - y_{n-1}$, so we have

$$|\bar{e}_n| \leq (1 + hk)|\bar{e}_{n-1}| + |\tau_{n-1}|$$

We can now obtain a closed form solution by stepping backwards until e_0.

$$|\bar{e}_n| \leq (1 + hk)^n|\bar{e}_0| + \sum_{j=0}^{n-1}(1 + hk)^j|\tau_{n-(j+1)}|$$

This holds for all single-step methods. We can eliminate the step length h by noticing that $(1 + hk)^n \leq e^{nhk} \leq e^{Nhk}$, where N is our total number of steps. But $Nh = T - t_0$, so we get instead

$$|\bar{e}_n| \leq e^{(T-t_0)k}|\bar{e}_0| + \sum_{j=0}^{n-1}(1 + hk)^j|\tau_{n-(j+1)}|$$

For Euler's method, we can pick $y_0 = \bar{y}_0 \rightarrow \bar{e}_0 = 0$, so the first term is just 0.

For the second term, start by noting that we can pull out the τ and obtain

$$\sum_{j=0}^{n-1}(1 + hk)^j|\tau_{n-(j+1)}| \leq max_{0\leq j\leq N}|\tau_j| \sum_{j=0}^{n-1}(1 + hk)^j$$

Taking the max over all τ_j gives a result of the form mh^2. Meanwhile, the sum is geometric, leaving us with

$$\sum_{j=0}^{n-1}(1 + hk)^j|\tau_{n-(j+1)}| \leq mh^2\frac{-1 + (1 + hk)^n}{1 - (1 + hk)}$$

Simplifying, we note that this bound is independent of N. Indeed our final bound is a statement of the form

$$|\bar{e}_n| \leq Ch \rightarrow max_{0\leq n\leq N}|y(t_n) - y_n| \leq Ch$$

This is the *global error bound*.

Discussion 2

Error Bound Estimation

Consider the ODE given by $\frac{dy}{dt} = f(y), y(t_0) = y_0$. We obtain estimates via Euler's method with $y_n = y_{n-1} + hf(y_{n-1})$.

To obtain an estimate of the error bound, we only need two conditions - (1) that $f(y)$ is k-Lipschitz, and (2) that $\exists m \geq 0$ s.t. $\frac{1}{2}|y''(t)| \leq m$. Then

$$\max_{0 \leq i \leq N} |y(t_i) - y_i| \leq e^{(T-t_0)k}|e_0| + \frac{mh}{k}e^{(T-t_0)k}$$

More important is to understand the argument behind the proof. This takes place in three steps - (1) specifying the *local truncation error* or LTE, (2) obtaining the recursive relation between e_n and e_{n-1} and (3) proving that $|e_i| = O(h)$.

The LTE is given by the difference between the estimated value and the true value of y at step n. That is, we have the true value of y evaluated at t_n as $\bar{y}_n = \bar{y}_{n-1} + hf(\bar{y}_{n-1}) + \tau_{n-1}$. Taylor's Theorem tells us that the error is all contained in the τ_{n-1} term.

We can continue to expand the Taylor series to get

$$\bar{y}_n = \bar{y}_{n-1} + hf(\bar{y}_{n-1}) + \frac{h^2}{2}\frac{d^2}{dt^2}y_{n-1} + O(h^3)$$

This implies that $\tau_{n-1} = \frac{h^2}{2}\frac{d^2}{dt^2}y_{n-1} + O(h^3)$.

Our error at step n is given by $e_n = \bar{y}_n - y_n$. The recursive relation between e_n and e_{n-1} can be obtained via the following.

$$\begin{aligned} y_n &= y_{n-1} + hf(y_{n-1}) \\ \bar{y}_n &= \bar{y}_{n-1} + hf(\bar{y}_{n-1}) + \tau_{n-1} \\ e_n = \bar{y}_n - y_n &\rightarrow |e_n| \leq |e_{n-1}| + hk|e_{n-1}| + \tau_{n-1} \end{aligned}$$

Simplifying, we get that $|e_n| \leq (1 + hk)|e_{n-1}| + \tau_{n-1}$.

Lastly, we prove that $|e_n| = O(h)$. From our above bound, we concatenate backwards to get $|e_n| \leq (1 + hk)^n|e_0| + \sum_{j=0}(1 + hk)^j|\tau_{n-(j+1)}|$.

$(1 + hk)^n \leq e^{nhk}$, which we get by simply noting that thanks to Taylor expansion we see that $e^x \geq (1 + x)$, so taking both sides to any power preserves the inequality. That means we can replace the first term with $e^{Nhk}|e_0|$.

The second term we handle by noting that we can bound the sequence of τ_i with $\max_i \tau_i$ and then extract it from the sum. $\max_i |\tau_i|$ is itself bounded by mh^2. This leaves

$$\max_i |\tau_i| \sum_j (1 + hk)^j \rightarrow mh^2 \frac{1 - (1 + hk)^n}{1 - (1 + hk)} \leq \frac{mh}{k}(1 - e^{nhk}) \leq \frac{mh}{k}(1 - e^{Nhk})$$

Now we have

$$|e_n| \leq e^{Nhk}|e_0| + \frac{mh}{k}(1 - e^{Nhk}) = e^{Nhk}(|e_0| - \frac{mh}{k}) + \frac{mh}{k}$$

Often, we can just set $|e_0| = 0$ as we're given initial values. One last simplification is to note that $Nh = T - t_0$. This makes the error bound independent of N and gives

$$|e_n| \leq \frac{m}{k}(1 - e^{(T-t_0)k})h \rightarrow |e_n| = O(h)$$

Lecture 5: More Error Estimates

Huen's Method and Runge-Kutta Families

Consider the numerical method given by

$$y^* = y_{n-1} + hf(y_{n-1})$$

$$y_n = y_{n-1} + \frac{h}{2}(f(y_{n-1}) + f(y^*))$$

We can obtain the local truncation error as follows. Consider that we can expand the method and eliminate y^* as

$$y_n = y_{n-1} + \frac{h}{2}f(y_{n-1}) + \frac{h}{2}f(y_{n-1} + hf(y_{n-1}))$$

We convert to exact values of $\bar{y}_n, \bar{y}_{n-1}$ and expand around $\bar{y}_{n-1}$.

$$\bar{y}_n = \bar{y}_{n-1} + h\frac{dy}{dt}|_{t_{n-1}} + \frac{h^2}{2}\frac{d^2y}{dt^2}|_{t_{n-1}} + \frac{h^3}{6}\frac{d^3y}{dt^3}|_{t_{n-1}} + \dots$$

Then combining the expansion of $\bar{y}_n$ on the LHS with the existing RHS, we get

$$\bar{y}_{n-1} + h\frac{dy}{dt}|_{t_{n-1}} + \frac{h^2}{2}\frac{d^2y}{dt^2}|_{t_{n-1}} + \frac{h^3}{6}\frac{d^3y}{dt^3}|_{t_{n-1}} + \dots = \bar{y}_{n-1} + \frac{h}{2}f(y_{n-1}) + \frac{h}{2}f(y_{n-1} + hf(y_{n-1}))$$

It's easier to consider by examining terms according to their power of $O(h)$.

$O(1)$ term gives us $\bar{y}_{n-1} - \bar{y}_{n-1} = 0$.

$O(h)$ term gives us $\frac{dy}{dt}|_{t_{n-1}} = \frac{1}{2}f(y_{n-1}) + \frac{1}{2}f(y_{n-1} + hf(y_{n-1}))$. If we Taylor expand the last f term, we get

$$f(y_{n-1} + hf(y_{n-1})) = f(\bar{y}_{n-1}) + hf(\bar{y}_{n-1})\frac{dy}{dt}|_{t_{n-1}} + \frac{h^2}{2}f(\bar{y}_{n-1})^2\frac{d^2y}{dt^2}|_{t_{n-1}} + O(h^3)$$

Resubstituting, we get

$$\frac{dy}{dt}|_{t_{n-1}} = \frac{1}{2}f(\bar{y}_{n-1}) + \frac{1}{2}f(\bar{y}_{n-1}) \to 0$$

$O(h^2)$ term gives us

$$\frac{1}{2}\frac{d^2y}{dt^2}|_{t_{n-1}} = \frac{1}{2}f(\bar{y}_{n-1})\frac{dy}{dt}|_{t_{n-1}}$$

But this holds as $\frac{d^2y}{dt^2} = \frac{d}{dt}f(y_{n-1}) = \frac{df}{dy}\frac{dy}{dt} = \frac{df}{dy}f(y_{n-1}) \to 0$.

We calculate the $O(h^3)$ term and see that we need to evaluate

$$\frac{1}{6}\frac{d^3y}{dt^3} - \frac{1}{4}f(\bar{y}_{n-1})^2\frac{d^2y}{dt^2}|_{t_{n-1}}$$

This works out to an expression of the form

$$-\frac{1}{12}\frac{d^2f}{dy^2}f^2 + \frac{1}{6}(\frac{df}{dy})^2 f$$

Ultimately, we end up with Huen's method as having non-zero error in the h^3 term, meaning our LTE bound gives

$$|\tau_{n-1}| \leq mh^3,$$

where $m = \max|-\frac{1}{12}\frac{d^2f}{dy^2}f^2 + \frac{1}{6}(\frac{df}{dy})^2 f|$.

To obtain a global error bound, we just need to subtract $\bar{y}_n - y_n$.

This gives

$$\bar{e}_n = \bar{e}_{n-1} + \frac{h}{2}(f(\bar{y}_{n-1}) - f(y_{n-1})) + \frac{h}{2}(f(\bar{y}_{n-1} + hf(\bar{y}_{n-1})) - f(y_{n-1} + hf(y_{n-1}))) + \tau_{n-1}$$

Recall that when we see differences of functions, we want to convert them into differences of arguments by using Lipschitz bounds.

Bounding the first term is easy: $\frac{h}{2}(f(\bar{y}_{n-1}) - f(y_{n-1})) \leq \frac{h}{2}K|\bar{y}_{n-1} - y_{n-1}| = \frac{h}{2}K|\bar{e}_{n-1}|$.

For the second term, work from outside in:

$$|f(\bar{y}_{n-1}+hf(\bar{y}_{n-1}))-f(y_{n-1}+hf(y_{n-1}))| \leq K|\bar{y}_{n-1}+hf(\bar{y}_{n-1})-(y_{n-1}+hf(y_{n-1}))| \leq K|\bar{y}_{n-1}-y_{n-1}|+K|hf(\bar{y}_{n-1})-hf(y_{n-1})$$

For the last term, we just apply another Lipschitz bound to get

$$K|hf(\bar{y}_{n-1}) - hf(y_{n-1})| \leq hK^2|\bar{y}_{n-1} - y_{n-1}| = hK^2|\bar{e}_{n-1}|$$

Finally, we can see that we end up with

$$\bar{e}_n \leq \bar{e}_{n-1} + \frac{h}{2}K|\bar{e}_{n-1}| + \frac{h}{2}hK^2|\bar{e}_{n-1}| + \tau_{n-1} = (1 + \frac{hK}{2} + \frac{h^2K^2}{2})|e_{n-1}| + \tau_{n-1}$$

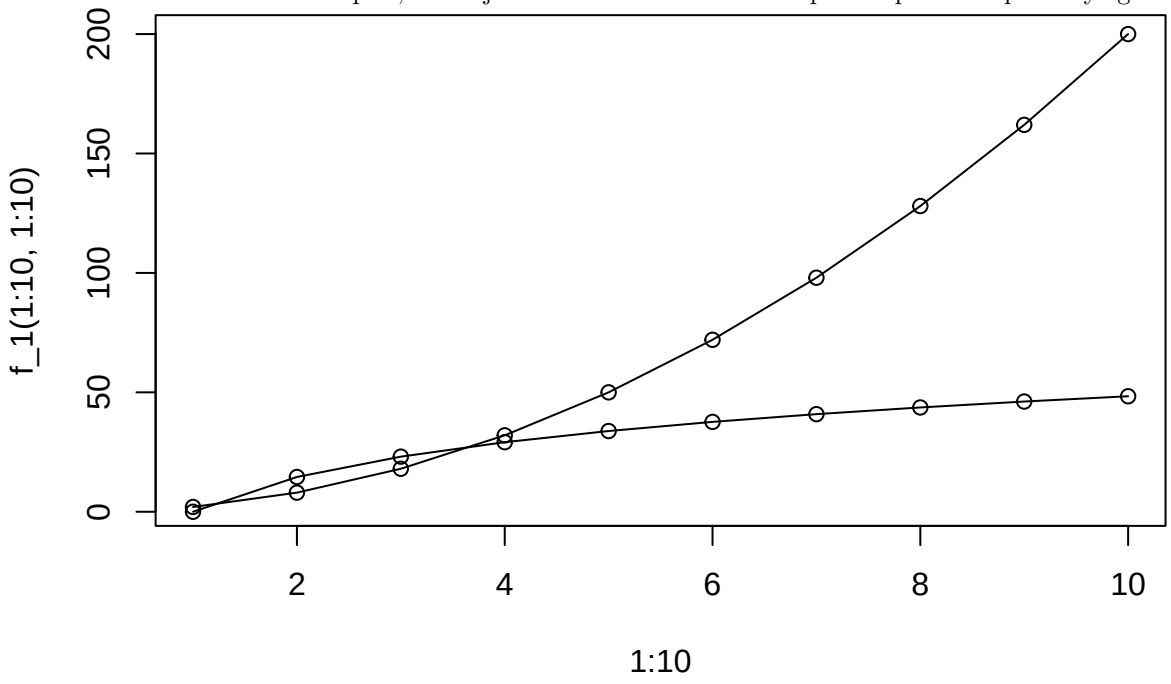
Lecture 6: Error for Systems

Plotting Solutions for Systems of ODEs

Consider the ODE given by

$$\frac{dv}{dt} = F(v) = \begin{pmatrix} F_1(v_1, v_2) \\ F_2(v_1, v_2) \end{pmatrix}$$

We could construct the "time series" plot, which just shows each function component plotted separately against time.



Another method is with phase-plane plots, which plot $v_1(t)$ vs $v_2(t)$. This is often done with dynamical systems.

Yet another is given by particle trajectory plots. Consider a system of p particles in 2D. The positions are given by $x = (x_1, y_1, x_2, y_2, \ldots, x_p, y_p)$. The velocities can be stated in general form as

$$\frac{dx_i}{dt} = \sum_{k=1}^{p} G(x_i, x_k)$$

We plot these as actual animations of the systems.

Euler's Method for Systems

Consider a system of ODEs and let F be Lipschitz. All norms are the 2-norm. We also require that F and ∇F exist and are continuous.

The result of multidimensional Euler's method is that there exists C independent of N such that

$$\max_{0 \le n \le N} \|v(t_n) - v_n\|_2 \le Ch$$

We obtain the local truncation error as

$$\tau_{n-1} = \frac{h^2}{2} \frac{d^2 v}{dt^2}|_{t^*}$$

Note that since $\frac{dv}{dt} = (F_1(v(t)), F_2(v(t)), \ldots)^T$, then the second derivative is given by

$$\frac{d^2 v}{dt^2} = \begin{pmatrix} \langle \nabla F_1, \frac{dv}{dt} \rangle \\ \langle \nabla F_2, \frac{dv}{dt} \rangle \\ \vdots \end{pmatrix}$$

We can obtain the error-timestep recurrence via Lipschitz bounds again:

$$|\bar{e}_n| \le |\bar{e}_{n-1}| + hK|\bar{e}_{n-1}| + \tau_{n-1} = (1 + hK)|\bar{e}_n| + \tau_{n-1}$$

This is all obviously the same as in the 1D case - doing second-order or higher methods requires multi-dimensional Taylor expansion, which is awful.

Lecture 7: Higher-Order Methods

Higher order methods are useful due to efficiency. One goal of numerical analysis is to obtain a solution at the endpoint with a given accuracy as efficiently as possible.

Runge-Kutta Methods

Euler's method is a 1st order Runge-Kutta method. We have two stages:

$$k_1 = f(y_{n-1})$$
$$y_n = y_{n-1} + hk_1$$

The general structure of p-stage Runge-Kutta methods is given as

$$\begin{aligned}
k_1 &= f(y_{n-1}) \\
k_2 &= f(y_{n-1} + ha_{21}k_1) \\
k_3 &= f(y_{n-1} + ha_{31}k_1 + ha_{32}k_2) \\
&\vdots \\
k_p &= f(y_{n-1} + h\sum_{j=1}^{p-1} a_{pj}k_j) \\
y_n &= y_{n-1} + h\sum_{j=1}^{p} \beta_j k_j
\end{aligned}$$

Obvious question is how to choose the coefficients a_{pj}, β_j. The most obvious solution is to choose the coefficients to maximize the order of the LTE.

Validation

Given a set of errors and timesteps, we can use the data to estimate our error p.

If we have an exact solution y, we can compare approximations at two different time scales $y_h, y_{0.5h}$.

Then we can construct the global errors $e_h, e_{0.5h}$ and by an argument of expanding the error we get that

$$\frac{|e_h|}{|e_{0.5h}|} = 2^p \rightarrow p = log(\frac{|e_h|}{|e_{0.5h}|})$$

Discussion 3

Error-Bound for One Step Method

Obtaining the global error bound for a one step method takes three steps:

1. Find the LTE of order $q+1$.
2. Get the error relation $e_{n+1} = |y_{n+1} - \bar{y}_{n+1}| = c(h)e_n$
3. Solve for global error of order $q, e_j \leq Ch^q, j \in (0, N)$.

As an example, we can get for the forward Euler the following: $LTE = O(h^2), e_j = O(h)$. For the Runge-Kutta 2-step method, we write out our method as

$$\begin{aligned}
y^* &= y_n + \alpha h f(y_n) \\
y_{n+1} &= y_n + \frac{h\beta_1}{2} f(y_n) + \frac{h\beta_2}{2} f(y^*)
\end{aligned}$$

Then if $\alpha\beta_2 = \frac{1}{2}$ then we have a 2nd order method.

If $\beta_1 + \beta_2 = 1$ we have a 1st order method.

For the method which takes a convex combination of $f(yn-1), f(y_n)$, we observe that for $\theta = 1, 0$ we get a 1st order method, but for $\theta = \frac{1}{2}$ we get a 2nd-order method.

This can be shown as follows: construct the exact value of $\bar{y}_n$ as

$$\bar{y}_n = \bar{y}_{n-1} + h\theta f(\bar{y}_{n-1}) + h(1-\theta)f(\bar{y}_n) + \tau_{n-1}$$

Now we Taylor expand both the LHS and the $f(\bar{y}_n)$ term. The LHS becomes

$$\bar{y}_n = \bar{y}_{n-1} + h\bar{y}'_{n-1} + \frac{h^2}{2}\bar{y}''_{n-1} + \frac{h^3}{6}\bar{y}'''_{n-1} + O(h^4)$$

while the term in the RHS becomes

$$f(\bar{y}_n) = \frac{dy}{dt}|_{t_{n-1}} = \bar{y}_{n-1} + h\bar{y}'_{n-1} + \frac{h^2}{2}\bar{y}''_{n-1} + O(h^3)$$

Resubstituting, we get

$$\bar{y}_{n-1}+h\bar{y}'_{n-1}+\frac{h^2}{2}\bar{y}''_{n-1}+\frac{h^3}{6}\bar{y}'''_{n-1}+O(h^4) = \bar{y}_{n-1}+h\theta f(\bar{y}_{n-1})+h(1-\theta)(\bar{y}'_{n-1}+h\bar{y}''_{n-1}+\frac{h^2}{2}\bar{y}'''_{n-1}+O(h^3))+\tau_{n-1}$$

Immediately, we see that the h^0 and h^1 terms cancel out. That leaves us with

$$\frac{h^2}{2}\bar{y}''_{n-1} + \frac{h^3}{6}\bar{y}'''_{n-1} + O(h^4) = h(1-\theta)(h\bar{y}''_{n-1} + \frac{h^2}{2}\bar{y}'''_{n-1} + O(h^3)) + \tau_{n-1}$$

Solving for the LTE, we get

$$\tau_{n-1} = h^2(\frac{1}{2} - (1-\theta))\bar{y}''_{n-1} + h^3(\frac{1}{6} - \frac{1-\theta}{2})\bar{y}'''_{n-1} + O(h^4)$$

For $\theta = \frac{1}{2}$, we get that LTE is 3rd-order, which means the global error is of order 2.

Lecture 9: Validating Interpolations

Asymptotic Error Expansion

Asymptotic error expansion allows us to approximate the global error with the result

$$|y(T) - y_h(T)| \simeq C_p(T)h^p$$

Asymptotic error expansion is very different from traditional error bound, which is loose.

If we don't have an analytic solution, then we need to figure out new ways of estimating the order p.

Consider approximated solutions at diff time scales $y_h(t), y_{\frac{h}{2}}(t), y_{\frac{h}{4}}(t)$.

Then consider the following ratio:

$$\frac{y_{\frac{h}{2}}(t) - y_h(t)}{y_{\frac{h}{4}}(t) - y_{\frac{h}{2}}(t)} = \frac{(y(t) - y_h(t)) - (y_t - y_{\frac{h}{2}}(t))}{(y(t) - y_h(t)) - (y_t - y_{\frac{h}{4}}(t))}$$

We can restate each term as an asymptotic error expansion C

This ends up giving us the following estimate for p:

$$p \simeq \log(|\frac{y_{\frac{h}{2}}(t) - y_h(t)}{y_{\frac{h}{4}}(t) - y_{\frac{h}{2}}(t)}|)$$

No matter what, we need $C_{p+k}h^{p+k} << C_p h^p$ in order to choose h.

Need to be in the asymptotic regime for this to work. When h is too small, above estimating ratio can result in catastrophic cancellation as we're taking difference of things that are approximately equal.

Lecture 11: Choice of Timestep & Qualitative Behavior

Consider the model ODE $\frac{dy}{dt} = \lambda \to y(t) = e^{Re(\lambda)}e^{Im(\lambda)}y_0$.

Want to establish conditions on h that govern asymptotic behavior of y_n.

Ex.: Euler - take the numerical method $y_n = (1 + h\lambda)y_{n-1} \to |y_n| = |(1 + h\lambda)^n|y_0$.

We can see that $\lim_{h\to\infty} |y_n| = 0 \to |1 + h\lambda| \leq 1$. If $\lambda \leq 0$, we have $h\lambda \in (-2, 0)$, so we need $h \leq \frac{2}{\lambda}$ in order to have $\lim |y_n| = 0$.

When we have $\lambda \geq 0$, $|1 + h\lambda| \geq 1$ always, so we have no restriction on choice of h.

We are often interested in this set, which we call the *region of absolute stability*, given by

$$S = \{h\lambda \in \mathbb{R} | \lim_{h\to\infty} |y_n| = 0 \text{ when method applied to model ODE}\}$$

Alternately, you could have $h\lambda \in \mathbb{C}$.

Ex. Huen's method - $y^* = y_{n-1} + hf(y_{n-1}), y_n = y_{n-1} + \frac{h}{2}f(y^*) + \frac{h}{2}f(y_{n-1})$. Let our ODE be given by $f(y) = \lambda y$.

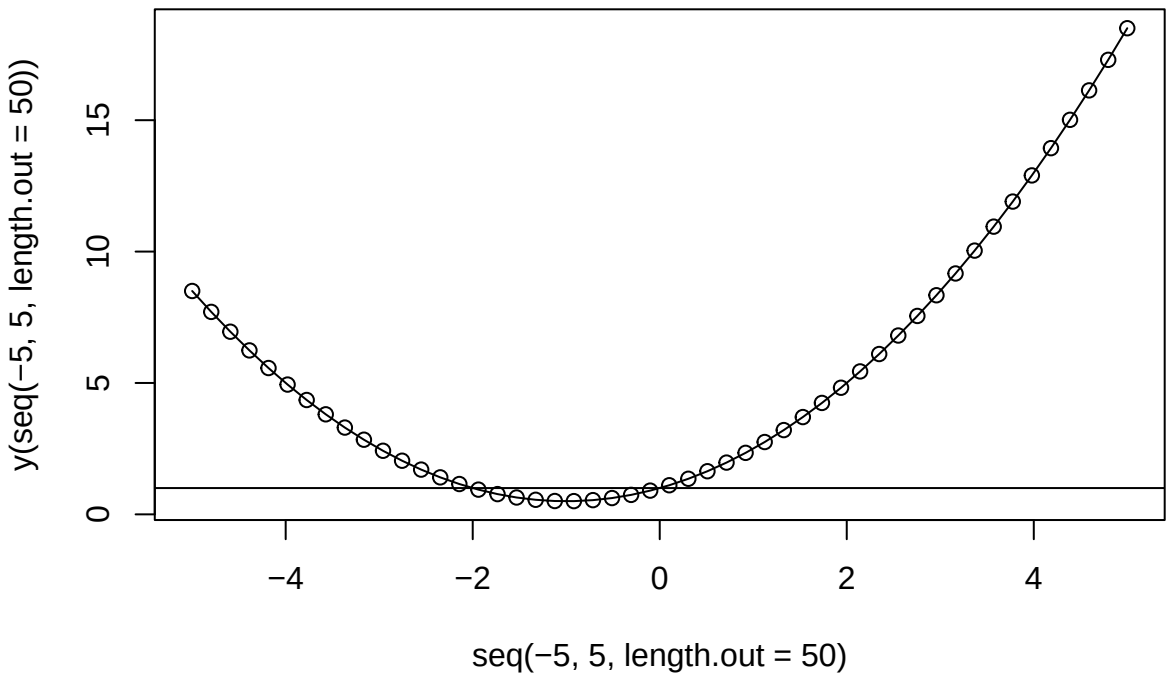
We know that Huen's method gives $|y_n| = |(1 + h\lambda + \frac{(h\lambda)^2}{2})||y_{n-1}|$.

Then we must have that $|(1 + h\lambda + \frac{(h\lambda)^2}{2})| \leq 1$, which gives us our region of absolute stability:

$$S = \{h\lambda \in \mathbb{R} \text{ s.t. } |(1 + h\lambda + \frac{(h\lambda)^2}{2})| \leq 1\}$$

But consider the graph of the function $y = 1 + x + 0.5 * x^2$. This shows that the region is given by $x = h\lambda \in (-2, 0)$, which turns out to be the exact same as we got in Euler's method.

```
y <- function(x) { 1 + x +0.5*x^2}
plot(seq(-5,5,length.out=50),y(seq(-5,5,length.out=50)))
lines(seq(-5,5,length.out=50),y(seq(-5,5,length.out=50)))
abline(h=1)
```



Discussion 4: Review of Homework Problems

HW2, P1

Consider the ODE given by $\frac{dy}{dt} = f(y)$. We Taylor expand the exact solution to get

$$\bar{y}_n = \bar{y}_{n-1} + \bar{y}'_n h + \frac{h^2}{2}\bar{y}''_n + \frac{h^3}{6}\bar{y}'''_n + O(h^4)$$

We want to have our LTE be 4th-order, meaning our method has to use the first three terms.

We start by replacing $\bar{y}'_n$ with $f(\bar{y}_n)$, then need to derive expressions for $\bar{y}''_n, \bar{y}'''_n$.

Since $y' = f$, we have $y'' = \frac{d}{dt}f(y(t))$, which by the chain rule gives us $y'' = f'y' = f'f$.

Similarly, $y''' = \frac{d}{dt}y'' = \frac{d}{dt}(f'f)$. That gives us $y''' = f''f^2 + (f')^2 f$.

Replacing our expansion with the above terms we get

$$y_n = y_{n-1} + hf(y_{n-1}) + \frac{h^2}{2}(f'(y_{n-1})f(y_{n-1})) + \frac{h^3}{6}(f''(y_{n-1})f(y_{n-1})^2 + (f'(y_{n-1}))^2 f(y_{n-1}))$$

This is guaranteed to give us $\tau_{n-1} = O(h^4)$, which gives 3rd-order error.

HW2, P2

Consider the same ODE as above but with a method giving

$$y^* = y_{n-1} + \frac{h}{2}f(y_{n-1})$$

$$y_n = y_{n-1} + hf(y^*)$$

We want to compute the LTE and the global error.

To find the LTE, we start with the exact expression

$$\bar{y}_n = \bar{y}_{n-1} + hf(\bar{y}_{n-1} + \frac{h}{2}f(\bar{y}_{n-1})) + \tau_{n-1}$$

To access the term in the middle, we need to give a Taylor expansion of f at the point $\bar{y}_{n-1} + \frac{h}{2}f(\bar{y}_{n-1})$:

$$f(\bar{y}_{n-1} + \frac{h}{2}f(\bar{y}_{n-1})) = f(\bar{y}_{n-1}) + f'(\bar{y}_{n-1})(\frac{h}{2}f(\bar{y}_{n-1})) + \frac{1}{2}f''(\bar{y}_{n-1})(\frac{h}{2}f(\bar{y}_{n-1}))^2 + O(h^3)$$

Returning to the original exact expansion we get

$$\bar{y}_n = \bar{y}_{n-1} + h[f(\bar{y}_{n-1}) + f'(\bar{y}_{n-1})(\frac{h}{2}f(\bar{y}_{n-1})) + \frac{1}{2}f''(\bar{y}_{n-1})(\frac{h}{2}f(\bar{y}_{n-1}))^2 + O(h^3)] + \tau_{n-1}$$

From above, we know the relationships between the products of f, f', f'' and y', y'', y'''. We can therefore simplify to

$$\bar{y}_n = \bar{y}_{n-1} + hf(\bar{y}_{n-1}) + \frac{h^2}{2}f'(\bar{y}_{n-1})(f(\bar{y}_{n-1})) + \frac{h^3}{8}f''(\bar{y}_{n-1})(f(\bar{y}_{n-1}))^2 + O(h^3) + \tau_{n-1}$$

and thus to

$$\bar{y}_n = \bar{y}_{n-1} + h\bar{y}'_{n-1} + \frac{h^2}{2}\bar{y}''_{n-1} + \frac{h^3}{8}\bar{y}'''_{n-1} + O(h^4) + \tau_{n-1}$$

Comparing with the above result for an expansion of y_n, we note that the h^0, h^1, h^2 terms cancel out, leaving only that

$$\tau_{n-1} = h^3(\frac{1}{24}\bar{y}'''_{n-1}) + O(h^4)$$

To get the global error, we subtract our exact and method values for y_n. This gives

$$|e_n| = |e_{n-1}| + h(f(\bar{y}_{n-1}) - f(y_{n-1})) + \tau_{n-1}$$

f is given as K-Lipschitz, so we can impose the below bound:

$$|e_n| \leq |e_{n-1}| + hK(|e_{n-1}| + \frac{h}{2}K|e_{n-1}|) + |\tau_{n-1}| = (1 + hk + \frac{h^2}{2})|e_{n-1}| + |\tau_{n-1}|$$

Concatenating from $t = 0, \ldots, n$, we get

$$|e_n| \leq (1 + hk + \frac{h^2}{2})^n|e_0| + \sum_{j=0}^{n-1}(1 + hk + \frac{h^2}{2})|\tau_{n-(j+1)}|$$

Using some additional bounds, we eventually get

$$|e_n| \leq e^{K(T-t_0)}|e_0| + mh^2e^{K(T-t_0)}$$

HW3 P2

Consider a numerical method for a system of ODEs given by

$$\begin{pmatrix} v_1^n \\ v_2^n \end{pmatrix} = \begin{pmatrix} v_1^{n-1} \\ v_2^{n-1} \end{pmatrix} + \begin{pmatrix} f(v_1, v_2) \\ 1 \end{pmatrix} h$$

Note that in particular this gives $v_2^n = v_2^{n-1} + h$, meaning that for $v_2^0 = 0, v_2^n = nh$.

If instead we have

$$\begin{pmatrix} v_1^n \\ v_2^n \end{pmatrix} = \begin{pmatrix} v_1^{n-1} \\ v_2^{n-1} \end{pmatrix} + \begin{pmatrix} f(v_1, v_2) \\ 1 + \frac{\epsilon}{h} \end{pmatrix} h$$

Then we see that the error propagates to give us $v_2^n = nh + n\epsilon$.

Lecture 12: Timestep Restrictions for General ODEs

Consider a method applied to the model problem which can be expressed as $y_n = g(h\lambda)y_{n-1}$. Then the following are equivalent definitions of stability:

$$S = \{h\lambda \in \mathbb{C} | g(h\lambda) < 1\}$$
$$S = \{h\lambda \in \mathbb{C} | \lim_{n\to\infty} |y_n| = 0\}$$
$$S = \{h\lambda \in \mathbb{C} | |y_n| < |y_{n-1}|\}$$

Timestep Restriction for General Equations

Consider $\frac{dy}{dt} = f(y,t), y(t_0) = y_0$. Our goals are to identify the qualitative behavior of the analytic solutions, and calibrate our method by identifying the restrictions on the timestep necessary for the numerical method to match the behavior of the analytic solution. In particular, we are interested in finding out whether nearby solution trajectories expand or contract.

Let $y(t)$ be a solution of $\frac{dy}{dt} = f(y,t), y(t_0) = y_0$ for $t \in [t_0, T]$. Let $t^* \in [t_0, T]$ and define $y^* = y(t^*)$.

Now let $y^A(t), y^B(t)$ be solutions nearby to y^* at t^*, i.e.,

$$\frac{dy^A}{dt} = f(t, y^A), y^A(t^*) = y^* + \epsilon_A$$
$$\frac{dy^B}{dt} = f(t, y^B), y^B(t^*) = y^* + \epsilon_B$$

Note that these separate trajectories never intersect, as given by the Picard-Lindelöf theorem. If $\frac{\partial f}{\partial y}|_{(t^*,y^*)} < 0$, then for $t > t^*$, we can see that $|y^A(t) - y^B(t)|$ decreases, meaning nearby trajectories contract. Otherwise, the difference of the nearby trajectories will grow, meaning the solutions expand.

Now consider numerical solutions. Consider a one-step method with associated region of absolute stability S. Let the method be applied to $\frac{dy}{dt} = f(y,t), y(t_0) = y_0$ for $t_n \in [t_0, T]$ using a uniform timestep $h = \frac{T-t_0}{N}$. Let t_k be a timestep, y_n^A, y_n^B numerical solutions starting nearby to y_k at t_k, with

$$y_n^A = y_k + \epsilon_A$$
$$y_n^B = y_k + \epsilon_B$$

Then we know if $\frac{\partial f}{\partial y}|_{(t^*,y^*)} < 0$ and $h(\frac{\partial f}{\partial y}|_{(t^*,y^*)}) \in S$, then numerical trajectories contract. Otherwise, if $\frac{\partial f}{\partial y}|_{(t^*,y^*)} < 0$ and $h(\frac{\partial f}{\partial y}|_{(t^*,y^*)}) \in \bar{S}^c$, then numerical trajectories expand.

We check our timestep by finding the interval $\frac{\partial f}{\partial y}$ lies in, and then adjusting h so that $h\frac{\partial f}{\partial y} \in S$.

Ex. Let $\frac{dy}{dt} = 4t^2\cos(y), t \in [0,5]$. Then we know that $y(t) \in [0, \frac{\pi}{2}]$.

$\frac{\partial f}{\partial y} = -4t^2\sin(y) \subset [-100, 0]$. Then for Euler's method, $S = (-2, 0)$, meaning we need $h < \frac{1}{20}$.

Lecture 12: Contraction & Expansion Results

We want to prove the above results on contraction and expansion of numerical solutions.

Let y_n^A, y_n^B be two solutions to the model problem via the method $y_n = g(h\lambda)y_{n-1}$, with

$$y_n^A = y_n + \epsilon_A$$
$$y_n^B = y_n + \epsilon_B$$

Then if $h\lambda < 0$, $|y_n^A - y_n^B|$ decreases, and otherwise for $h\lambda > 0$, $|y_n^A - y_n^B|$ increases.

Consider $z_n = y_n^A - y_n^B$. Clearly $z_n = g(h\lambda)z_{n-1}$ and we have that $z_0 = \epsilon_A - \epsilon_B$.

Lecture 13: Contraction and Expansion Results Continued

Lemma 1: For y^A, y^B, $\frac{dy}{dt} = \lambda y$, we have that if $\lambda < 0, |y^A - y^B|$ decreases, and otherwise $|y^A - y^B|$ increases.

Lemma 2: For methods of the form $y_n = g(h\lambda)y_{n-1}$, if $\lambda < 0, h\lambda \in S, |y^A - y^B|$ decreases and otherwise it increases.

Lemma 3: For y^A, y^B, $\frac{dy}{dt} = f(t,y)$, there exists $\lambda^* = \frac{df}{dy} < 0$ such that $|y^A - y^B|$ decreases, and otherwise increases.

Lemma 4 details the behavior of numerical solutions to $\frac{dy}{dt} = f(t,y)$. In particular it says that if we have $n \in (k, k+1, \dots)$ and

$$y_n^A = y_n + \epsilon_A$$
$$y_n^B = y_n + \epsilon_B$$

then there exists $\lambda^* = \frac{df}{dy}|_{(t^*,y^*)} < 0, h\lambda^* \in S$ such that $|y_n^A - y_n^B|$ decreases. Otherwise, $|y_n^A - y_n^B|$ increases.

Summary of the justification for stability: At $(t^*, y^*) = (t_k, y_k)$ nearby solution trajectories are approximate solutions to a linearized problem

$$\frac{dy}{dt} = f(t^*, y^*) + \frac{df}{dt}|_{(t^*,y^*)}(t - t^*) + \frac{df}{dy}|_{(t^*,y^*)}(y - y^*)$$

Numerical trajectories starting near y_k at t_k are approximately equivalent to applying our method to the linearized problem.

Apply the method to the linearized problem results in $\bar{z}_n = \bar{y}_n^A - \bar{y}_n^B$, which takes the form

$$\bar{z}_n = g(h\frac{df}{dy}|_{(t^*,y^*)})\bar{z}_{n-1}$$

We then derive our timestep restriction h from asserting

$$\frac{df}{dy}|_{(t^*,y^*)} < 0, h\frac{df}{dy}|_{(t^*,y^*)} \in S, |\bar{z}_n| \text{ decreases iff } |g| < 1$$

Timestep Restrictions for Linear Constant Coefficient Systems of ODEs

Consider $\frac{dv}{dt} = Av, v(0) = v_0$. Then we have that $v(t) = e^{At}v_0$.

Matrix exponentials are defined as

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

In practice there are easier ways to get approximations to any matrix exponential.

Lecture 14: Timestep Restrictions for Systems of ODEs

Recall we had $\frac{dv}{dt} = Av, v(t_0) = v_0$, with A diagonalizable. If we have eigenpairs (u_i, λ_i) of A, then we know that the zone of absolute stability is given by the intersection of the zones of absolute stability for each eigenvalue.

We can get this by writing $v(t)$ in the eigenbasis as $v(t) = \sum_{i=1}^m w_i(t)u_i(t)$, $\frac{dw_i}{dt} = \lambda_i w_i$.

If we first express a numerical method implicitly via $Q(hA)v_n = P(hA)v_{n-1}$, we can rewrite this in the eigenbasis to give $v_n = Q^{-1}(hA)P(hA)v_{n-1} = \sum_{j=1}^m (w_j)_n u_j$, where $(w_i)_n = [Q(hk)]^{-1}P(hA)(w_i)_{n-1}$.

Now we consider non-diagonalizable A. We will still have a set of eigenvectors, not all of which will be distinct from 0. It turns out that we get the same restriction, that we need to be in the intersection of all the $h\lambda_i \in S$. Whereas the trick in the diagonalizable case was to move from Euclidean space to eigenspace, the trick here is to move from Euclidean space to the *Jordan canonical form*.

When we consider the behavior of nearby trajectories in the eigenspace, we obtain a result that we can control the expansion/contraction behavior at any given point t. However, with the Jordan normal form, the asymptotic behavior only matches as $t \to \infty$.

General Linear Systems

Consider $\frac{dv}{dt} = F(v), F : \mathbb{R}^{\ltimes} \to \mathbb{R}^{\gtrdot}$.

Start by calculating the Jacobian at the value $v(t)$, $JF(t) = \frac{dF}{dv}|_{v(t)}$. The elements of the Jacobian are given by $(JF)_{ij} = \frac{\partial F}{\partial v_j}$ where v_j is the jth component of v. The eigenvalues of JF will be the eigenvalues of the problem at time t, $\lambda_i(t)$.

At t^*, the solution to $\frac{dv}{dt} = F(v)$ near $v(t^*)$ can be linearized as

$$\frac{dv}{dt} = F(v(t^*)) + JF(t^*)(v - v(t^*))$$

Now consider two nearby trajectories $v_A(t), v_B(t)$ and form $z(t) = v_A(t) - v_B(t)$. We have that $\frac{dz}{dt} = JF(t^*)z$. Thus z satisfies a constant-coefficient model problem, which is also diagonalizable and thus has a full rank set of eigenpairs.

Next write $z(t) = \sum_{i=1}^{m} s_i(t)u_i$, $\frac{ds_i}{dt} = \lambda_i(t^*)s_i$.

Hence if $Re(\lambda_i(t^*)) \leq 0, |s_i|$ decreases, which means the u_i component of the linearized solution's trajectories decreases. Since the linear problem is a good solution to the nonlinear problem near $v(t^*)$, we can conclude that the nonlinear problem's solution's nearby trajectories also decrease.

Stiffness

We have seen that timestep restrictions are a property of the method chosen, usually assessed on the model problem. The property of stiffness is a property of the problem itself - notably, of the eigenvalues of the Jacobian.

Comparing methods, we can observe a few things.

```
## Warning: fonts used in `flextable` are ignored because the `pdflatex` engine
## is used and not `xelatex` or `lualatex`. You can avoid this warning by using
## the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a compatible
## engine by defining `latex_engine: xelatex` in the YAML header of the R Markdown
## document.
```

methods	lte	stability
Forward Euler	o(h)	(-2,0)
Huen's	o(h ^2)	(-2.78,0)
Runge-Kutta 4	o(h ^4)	(-2.78,epsilon)
Backward Euler	o(h)	R ^2 - (0,2)
Trapezoidal	o(h ^2)	(-infty,0)

So far we've also seen two restrictions on the timestep: the stability restriction, $h < h\sigma$, which ensures that nearby numerical trajectories agree with the true solution, and the LTE restriction $h < h_\tau$, which ensures that the LTE is sufficiently small (sufficient up to user).

Lecture 15: Stiffness

An ODE method is *A-stable* if the zone of absolute stability contains the entire left half of the complex plane, $S \supset C^- = \{w \in \mathbb{C}, Re(w) < 0\}$.

Lecture 16: Newton's Method and Solvability

Recall that we assessed that for non-stiff ODE problems, we should choose to use explicit methods. If the ODE is stiff, we need methods that have large regions of absolute stability, which usually in practice means implicit methods like trapezoidal or backward Euler.

Newton's method gives that solutions to an equation $G(z) = 0$ are given by $z^k = z^{k-1} - [JG]^{-1}_{z^{k-1}} G(z^{k-1})$. In terms of backwards Euler, this looks like $z - hf(z) - y_{n-1} = 0$.

Lecture 17: Convergence and Fixed Point Iteration

Consider a method $z^k = \bar{g}(z^{k-1})$. As an example we have Newton's method, $z^k = z^{k-1} - \frac{f(z^{k-1})}{f'(z^{k-1})}$. We want to compare $z^k = \bar{g}(z^{k-1})$ to $z^* = \bar{g}(z^*)$ and examine the difference

$$\tilde{e}^k = z^k - z^* = \bar{g}(z^{k-1}) - \bar{g}(z^*)$$

Naturally we will want to look at things like Lipschitz bounds on $\bar{g}$. We start with the case where $\bar{g}$ is linear. Then $\bar{g}(z) = b + mz \to z^k - z^* = m(z^{k-1} - z^*) = m\tilde{e}^{k-1}$. Thus $\tilde{e}^k = m\tilde{e}^{k-1}$.

These m coefficients will depend on step size h. We want to understand relationship between m, h such that we can ensure $\lim_{k\to\infty} ||\tilde{e}^k|| = 0$.

When $\bar{g}$ is non-linear, the obvious choice is to expand $\bar{g}(z^{k-1})$ around z^*, $\bar{g}(z^{k-1}) = \bar{g}(z^*) + \bar{g}'(z^*)(z^{k-1} - z^*) + \frac{1}{2}\bar{g}''(z^*)(z^{k-1} - z^*)^2 + \ldots$. Our constant terms cancel, meaning we can then write $\tilde{e}^k = \bar{g}'(z^*)(z^{k-1} - z^*) + \frac{1}{2}\bar{g}''(z^*)(z^{k-1} - z^*)^2 + \ldots$.

It turns out that many of the derivatives in the above expression will vanish - in fact, there exists $p \in \mathbb{Z} > 0$ such that $\frac{d^r\bar{g}}{dz^r}|_{z^*} = 0 \forall r < p$. This means we can rewrite the error at z^k as

$$|\tilde{e}^k| = |z^k - z^*| \leq \max_{z \in B_\delta(z^*)} \frac{|\frac{d^p\bar{g}}{dz^p}|_z|}{p!} |z^{k-1} - z^*|^p$$

If $\bar{g}$ is smooth and z^0 is sufficiently close to z^* we can say $\exists\lambda$ s.t. $|\tilde{e}^k| \leq \lambda|\tilde{e}^{k-1}|^p$.

For non-linear equations, this p is the *rate or order of convergence*. For linear equations, we have $p = 1$ and $\lambda = m$, and so with the relation $|\tilde{e}^k| \leq \lambda|\tilde{e}^{k-1}|$, we must have that $|\lambda| < 1$ for $\lim_{k\to\infty} |\tilde{e}^k| = 0$. Then λ is the *rate of linear convergence.*

Ex: consider $\frac{dy}{dt} = -400y + \cos(y)$. Note that $\frac{df}{dy}$ is large and negative for all values y, meaning that the ODE is stiff. We can solve this using backward Euler, $z - h(-400y + \cos(y)) - b = 0$. Its fixed point expression is given as

$$z^k = h(-400z^{k-1} + \cos(z^{k-1})) + b = \bar{g}(z^{k-1})$$

Meanwhile the application of Newton's method gives us

$$z^k = z^{k-1} - \frac{z^{k-1} - h(-400z^{k-1} + \cos(z^{k-1})) - b}{h(-400 - \sin(z^{k-1}))}$$

The fixed point expression of the Newton method is given as

$$\begin{aligned} z^k - z^* &= \bar{g}(z^{k-1}) - \bar{g}(z^*) \\ \frac{d\bar{g}}{dz} &= h(-400 - \sin(z)) \\ \tilde{e}^k &\leq \max_\xi |h(-400 - sin(\xi))||\tilde{e}^{k-1}| = 401h|\tilde{e}^{k-1}| \end{aligned}$$

Then we have $\lambda = 401h$ and $p = 1$. This means that for $h < \frac{1}{401}$, we have $\lim_{k\to\infty} |\tilde{e}^k| = 0$.

Lecture: Linear Multistep Methods

The general form for a linear multistep method is given by

$$\sum_{j=0}^{k} \alpha_j y_{n-j} = h \sum_{j=0}^{k} \beta_k f(y_{n-j})$$

We can calculate the coefficients via integrating our ODE:

$$\frac{dy}{dt} = f(y) \rightarrow \int_{t_{n-1}}^{t_n} \frac{dy}{dt} = \int_{t_{n-1}}^{t_n} f(y(s))ds$$

We replace the f in the integral with a set of interpolating polynomials through (t_{n-j}, f_{n-j}) to get

$$y_n - yn - 1 \simeq \int_{t_{n-1}}^{t_n} F(s)ds = h\sum_{j=0}^{k} \beta_j f_{n-j}$$

When $\beta_0 = 0$, then we have a set of explicit methods commonly referred to as *Adams-Bashforth methods*. When implicit, these are called *Adams-Moulton methods*.

Another approach to deriving linear multistep methods comes through taking a backward finite difference approximation to $\frac{dy}{dt}|_{t_n}$, which gives us

$$\frac{dy}{dt}|_{t_n} \simeq \frac{y_n - y_{n-1}}{h} = \frac{h\sum_{j=0}^{k} \beta_j f_{n-j}}{h}$$

This is a *backward differentiation formula*. BDF methods are also generalizations of methods like backward Euler.

Lecture: Stability of Linear Multistep Methods

Given a LMM satisfying the root condition, there exists $\tilde{K}$ and

$$|y_n - y_{n-1}| \leq$$

If the root condition is satisfied and the local truncation error is of order $p + 1$ and the initial values are of order p, then the linear multistep method will converge with rate p.

Root condition/0-stability are necessary for convergence. This contrasts with one-step methods, where being in the region of absolute stability is not necessary for convergence.

BDF methods of order ≤ 6 satisfy the root condition and thus converge. However, BDF methods of order ≥ 6 do not satisfy the root condition and thus do not converge.

Proof of the Root Condition

Consider a general method

$$\sum_{j=0}^{k} \alpha_j y_{n-j} = h\sum_{j=0}^{k} \beta_j f(y_{n-j})$$

Two questions: what happens as $h \rightarrow 0$?

In both scenarios the LHS $\sum_{j=0}^{k} \alpha_j y_{n-j}$ dictates solution behavior. First we choose y_j such that $|y_j| = O(h)$ for $j = 0, 1, ..., k - 1$. Then we apply $\sum_{j=0}^{k} \alpha_j y_{n-j} = 0$ to determine $y_k, y_{k+1}, \ldots$.