# Numerical Analysis

Jaden Wang

August 20, 2020

# Contents

## 0.1 Bisection

## 0.2 Fixed point

$e_n = r - x_n = g(r) - g(x_{n-1}) = (r - x_{n-1})g'(\xi) = e_{n-1}g'(\xi) \approx e_{n-1}g'(r)$.

where $\zeta \in (x_n n - 1, r)$.

$$g(r) = g(x_{n-1}) + (r - x_{n-1})g'(x_{n-1}) + \dots.$$

.

This is linear convergence.

$$\frac{e_{n+2}}{e_{n+1}} \approx \frac{e_{n+1}}{e_n}.$$

## 0.3 Secant Method

$p_{n+1} = p_n - \frac{f(_{n+1})(p_{n+1} - p_n)}{f(p_{n+1}) - f(p_n)}$

"False Position"

### 0.3.1 "Sins"

- Do not subtract numbers that has very small differences
- Do not divide with a piece of "garbage"
- Do not set stopping criterion to equal 0, use $\varepsilon$

## 0.4 Newton's Method

$$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}.$$

Problems with this method:

- As $p_n \to p*$, $f(p_n) \to 0$ . So the convergence gets slower.
- really small slope near the r
- local minimum
- inflection point

Convergence:

$$e_n = \frac{f''(\xi)}{2f'(r)}e_{n-1}^2 \approx \frac{f''(r)}{2f'(r)}e_{n-1}^2.$$

If $e_{n+1}/e_n^\alpha = \lambda$, we call $\alpha$ the order of convergence.

Aitken's $Delta^2$ Method

Steffensen's Method

- Pick $p_0$

- $p_1 = g(p_0)$
- $p_2 = g(p_1)$
- compute $\hat{p_0}$
- $p_0 = \hat{p_0}$ repeat

If denominator $p_{n+2} - 2p_{n+1} + p_n \approx 0$, then let $p_0 = p_2$ in the next iteration instead.

Consider $f(x) = ax^2 + bx + c$. If $b^2 >> 4ac$ then error occurred in two roots might differ significantly. To compute the "small" root more accurately, we multiply the root equation by 1 using the conjugate. Assume $b > 0$, the small root should be computed by

$$x = \frac{2c}{b + (b^2 - 4ac)^{.5}}.$$

Instead of using secant, wouldn't it be better to use a parabola? Try $s(x) = a(x - x_2)^2 + b(x - x_2) + c$

$$f_0 = a(x_0 - x_2)^2 + b(x_0 - x_2) + cf_1 = \ldots f_2 = c.$$

Now we want the root $s(x_3) = 0$ that is closer to $x_2$, which is the "small" root.

$$x_3 - x_2 = -\frac{2c}{b + sgn(b)\sqrt{b^2 - 4ac}}.$$

This is cubic convergence.

$$p(x) = (x - z)[(x - z)Q_1(x) + R_1] + R_0.$$

$R_0 = p(z)$

## 0.5 Horner's Method

$$\begin{aligned} P(x) &= (x - z)Q_0(z) + R_0 \\ &= (x - z)[b_n x^{n-1} + \ldots + b_1] + b_0 \\ a_n x^n + \ldots + a_0 &= (b_n x^n + b_{n-1}x^{b-1} + \ldots + b_1 x) - z(b_n x^{n-1} + \ldots + b_1)) + b_0 \end{aligned}$$

$$\begin{aligned} b_n &= a_n \\ b_{n-1} &= a_{n-1} + zb_n \\ &\ldots \\ b_1 &= a_1 + zb_2 \\ b_0 &= a_0 \end{aligned}$$

Let $b_{n+1} = 0$
Do this again

$$\begin{aligned} c_n &= b_n \\ &\ldots \\ c_1 &= b_1 + zc_2 \end{aligned}$$

Let $c_{n+1} = 0$, $c_k = b_k + zc_{k+1}$.

Root deflation: roots found in the end suffer more from numerical errors.

## 0.6 Chapter 3 Interpolation: Lagrange Polynomials

Let

$$P(x) = \sum_{k=1}^{n} P_k(x) = \sum_{k=0}^{n} L_{n,k} f(x_k).$$

where

$$L_{n,k} = \frac{(x - x_0)(x - x_1)\dots(x - x_{k-1})(x - x_{k+1}\dots(x - x_n))}{(x_k - x_0)\dots(x_k - x_{k-1})(x_k - x_{k+1})(x_k - x_n)))}.$$

$$e = \frac{f^{n+1}(\xi)}{n+1}!(x - x_0)(x - x_1)\dots(x - x_n).$$

## 0.7 Neville's Method

## 0.8 Cubic Spline

Linear doesn't work because it's not smooth at the junctions. Quadratic misses one condition for the 6 parameters. Cubic is the sweet spot where 8 parameters have 8 reasonable conditions. Conditions for two consecutive cubic polynomials:

- matches function values at two end points, for both functions
- matches each other's values at the middle point
- matches derivatives at the middle point
- matches 2nd derivatives at the overlapped points

# Chapter 4 Numerical Derivatives

Using Taylor's expansion:

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x) - \frac{h^2}{2}f''(x) - \dots}{h} \\ &= \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h) \end{aligned}$$

This is the **forward difference**.

Now try with $f(x - h)$:

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h).$$

Now subtracting forward and backward differences:

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3}f'''(x)$$
$$f'(x) = \frac{1}{2h}(f(x+h) - f(x-h)) + \mathcal{O}(h)$$

This is the **central difference**.

Now consider the second derivatives by adding the forward and backward differences.

$$f''(x) = \frac{f(x+h) + f(x-h) - 2f(x)}{h} + \mathcal{O}(h^4).$$

Using three-point Lagrange Polynomials: see lecture notes. Differences are just a weighted average. Errors are all $\mathcal{O}(h^3)$.

For second derivative using Lagrange polynomials, we obtain similar answer as from Taylor. But for third derivatives, Lagrange wouldn't work anymore because all three differences are zero now.

Midpoint for 2nd derivatives from book:

$$f''(x_0) = \frac{1}{h^2}\left[f(x_0 - h) - 2f(x_0) + f(x_0 + h)\right] - \frac{h^2}{12}f^{(4)}(\xi)$$

for some $\xi$, where $x_0 - h < \xi < x_0 + h$

Taylor Matching for forward difference:

$$\begin{aligned} f'(x) &= af(x+h) + bf(x) \\ &= a[f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \ldots] \\ &+ b[f(x)] \end{aligned}$$

Trying to match/cancel them as much as possible, going down the derivatives of the function.

## 0.9 Integration

### 0.9.1 Trapezoidal Rule

We can build a degree one Lagrange polynomial for every two points.

$$A_1 = \frac{h}{2}(f_0 + f_1)$$
$$\ldots$$

### 0.9.2 Simpson's Rule

Now use every three points for Lagrange polynomial.

$$P_2 = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f_2.$$

We can use substitution $s = \frac{x-x_1}{h}$, so the equation simplifies to:

$$P_2 = \frac{1}{2}s(s-1)f_0 - (s+1)(s-1)f_1 + \frac{1}{2}(s+1)sf_2.$$

After integration we obtain

$$A_1 = \frac{h}{3}(f_0 + 4f_1 + f_2).$$

$$\int_a^b f(x)dx = \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n].$$

### 0.9.3 Newton-Cotes

After substitution $s = \frac{x-x_0}{h}$, and resetting $x = s$, consider

$$\int_0^1 f(x)dx = af(0) + bf(1).$$

Expand $f(x)$ as

$$f(x) = f(0) + xf'(0) + \frac{x^2}{2}f''(0) + \ldots.$$

Integrate this expansion from 0 to 1, and expand $f(1)$, we can solve for $a, b$, we obtain $a = b = \frac{1}{2}$. This leads to the trapezoidal rule.

Similarly, consider

$$\int_{-1}^1 f(x)dx = af(-1) + bf(0) + cf(1).$$

By doing the "grand accounting" again, we obtain the Simpson's rule, with error of $\mathcal{O}(h^5)$proportional to the 4th derivatives.

### 0.9.4 Example

Consider:

$$\int_0^1 e^{(-x^2)}dx.$$

## 0.10 Direct method

Consider

$$\int_{-1}^1 f(x)dx = af(-1) + bf(0) + cf(1).$$

Now pretend $f(x) = 1, x, x^2, \ldots$, keep plugging in the next order until we get inconsistency. Then we obtain the same coefficient as Simpson's rule.

But we don't have to stick with $x = -1, 0, 1$. If we let $x = -\frac{2}{3,0,\frac{2}{3}}$, then we get something different.

We can generalize even further. Consider

$$\int_{-1}^{1} f(x) \sin \frac{\pi}{2} x dx = af(-1) + bf(0) + cf(1).$$

Repeat the same procedure and we obtain the weighted values.

Transforming integrals: Let $t = \frac{2x-a-b}{b-a}$, hence

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{t(b-a)+a+b}{2}\right)\frac{b-a}{2}dt.$$

## 0.11 Gaussian Quadrature

We want to find:

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} c_i f(x_i).$$

$c_i and x_i$ give us $2n$ parameters to choose, so the polynomial is at most $2n-1$ degree.

### 0.11.1 Legendre Polynomials

They are orthogonal with respect to the inner product $\int_{-1}^{1} P(x)P_n(x)dx$, where $P_n(x)$ is the nth Legendre polynomial.

Example:

$$\int_0^1 e^{(-x^2)}dx = \frac{1}{2}\int_{-1}^{1} e^{(-\frac{t+1}{2})^2}dt.$$

This is a lot less work than Simpson's.

Advantage: good accuracy
Disadvantage: uneven spacing, so if we don't know $f(x)$ there might be too much interpolation.

## 0.12 Improper Integrals

Consider the integration of functions with a singularity at $x = a$ (left endpoint) of the form:

$$f(x) = \frac{g(x)}{(x-a)^p}.$$

where $g(x)$ is continuous on $[a, b]$. And we want $\int_a^b f(x)$. Note this converges iff $0 < p < 1$.

For right singularity, you can just flip the expression. For middle singularity, break it into two parts.

If infinity appears, change variable to $t = \frac{1}{x}$.

## 0.13 Taylor expansion for multivariable functions

$f(x+dx, y+dy, z+dz) = [e^{(dx\frac{\partial}{\partial x} + dy\frac{\partial}{\partial y} + dz\frac{\partial}{\partial z})}]f(x, y, z)$

## 0.14 1st order ODE IVP

### 0.14.1 Euler's Method

$y' = f(x, y), y(a) = y_a, a \leq x \leq b$. Divide $[a, b]$ into equal parts, $h = \frac{b-a}{n}$, so for each step, we have $x_i$ and $x_{i+1}$.

Then

$$\int_{x_i}^{x_{i+1}} \frac{dy}{dx} dx = \int_{x_i}^{x_{i+1}} f(x, y(x)) dx.$$

Then approximately we get

$$y_{i+1} = y_i + f(x_i, y_i) \cdot h.$$

Using Taylor's expansion, we can get

$$y(x+h) = y(x) + \frac{h}{1!} y'(x) + \frac{h^2}{2!} y''(x) + \ldots \approx y(x) + hy'(x) = y(x) + hf(x, y).$$

which gives us the same thing. The error is $\mathcal{O}(h)$.

### 0.14.2 Higher order Taylor method

We know that $y^{(n)} = f^{(n-1)}(x_i, y_i)$. Let $T^{(n)}(x_i, y_i) = \sum_{k=1}^{n} \frac{h^{k-1}}{k!} f^{(k-1)}(x_i, y_i)$. Then we can write $w_{i+1} = w_i + hT^{(n)}(x_i, y_i)$.

### 0.14.3 Runge-Kutta Methods

Taylor needs to compute the derivative of the function. We can do something similar to Gaussian quadrature, to find the ideal points so it is equivalent as evaluating the slope. Consider

$$w_{i+1} = w_i + h[a \cdot f(x_i + \alpha, w_i + \beta)].$$

When $n = 2$,

$$\begin{aligned} T^{(2)}(x, y) &= f(x, y) + \frac{h}{2} f'(x, y) \\ &= f(x, y) + \frac{h}{2} \left[ \frac{\partial f}{\partial x} \frac{\partial x}{\partial x} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial x} \right] \\ &= f(x, y) + \frac{h}{2} [f_x(x, y) \frac{\partial x}{\partial x} + f_y(x, y) f(x, y)] \end{aligned}$$

and

$$af(x + \alpha, y + \beta) = a[f(x, y) + \alpha f_x + \beta f_y] + \ldots.$$

Now we compare and get $a = 1, a \cdot \alpha = \frac{h}{2}, a \cdot \beta = \frac{h}{2} f(x, y)$. Therefore,

$$w_{i+1} = w_i + hf(x_i + \frac{h}{2}, w_i + \frac{h}{2} f(x_i, w_i)).$$

Now the local truncation error is $\mathcal{O}(h^2)$. This is called 2nd order R-K method, or midpoint method. It uses the slope at the midpoint.

In general, there is no one magic location that gives you the equivalence of 4th order Taylor. We need four.

When $n = 4$

$$\begin{aligned} k_1 &= hf(x_i, w_i) \\ k_2 &= hf(x_i + \frac{h}{2}, w_i + \frac{k_1}{2}) \\ k_3 &= hf(x_1 + \frac{h}{2}, w_i + \frac{k_2}{2}) \\ k_4 &= hf(x_i + h, w_i + k_3) \\ w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

The local truncation error is $\mathcal{O}(h^4)$).

## 0.15 Extrapolation

## 0.16 Higher order

## 0.17 Multi-step Methods

### 0.17.1 Adams-Bashforth 3-step

$$y_{i+1} = y_i + h[af(x_i, y_i) + bf(x_{i-1}, y_{i-1}) + cf(x_{i-2}, y_{i-2})].$$

Using Taylor's series expansion on two previous points, -h and -2h away from $x_i$,

$$\begin{aligned} y_{i+1} &= y_i + hy_i' + \frac{h^2}{2}y_i''' + \frac{h^3}{6} + \dots. \\ f(x_{i-1}, y_{i-1}) &= y_i' - hy_i'' + \frac{h^2}{2}y_i'' - \frac{h^3}{6}y_i''' + \dots. \\ f(x_{i-2}, y_{i-2}) &= y_i' - 2hy_i'' + 2h^2 yi''' - \frac{4}{3}h^3 y_i''''. \end{aligned}$$

Plugging back into the first equation, we obtain

$$\begin{aligned} h &: y_i' = ay_i' + 6y_i' + cy_i' \\ h^2 &: \frac{1}{2}y_i'' = -by_i'' - 2cy_i'' \\ h^3 &: \frac{1}{6}y_i''' = \frac{b}{2}y_i''' + 2cy_i''' \\ h^4 &: \frac{1}{24} = \dots \text{ error} \end{aligned}$$

And the grand accounting gives us the coefficients.

$$y_{i+1} = y_i + \frac{h}{12}[23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})].$$

Let $\phi = \frac{h}{12}[23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})]$. The local truncation error is $\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - \frac{\phi}{h}$ and is $\mathcal{O}(h^3)$.

### 0.17.2 Adams-Moulton Methods 2-step

We can also let

$$y_{i+1} = y_i + h[af(x_{i+1}, y_{i+1}) + bf(x_i, y_i + cf(x_{i-1}, y_{i-1})].$$

Note that this uses two old and one new points. Adams-Bashforth only uses old points. "Step" refers to old points only.

### 0.17.3 Predictor-Corrector Method/Modified Euler Method

We would like to use explicit method to find implicit solution.

$$y_{i+1}* = y_i + hf(x_i, y_i)y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1}*)].$$

Average the slopes between two points.

### 0.17.4 Generating seed values

Suppose $y' = y + x$. If $x$ is small, then $\frac{y'}{y} \approx 1$ and $y(0) = 0$, whose solution is $y = 0$, but this solution violated our assumptions. If $y$ is small, we get $y = \frac{x^2}{2}$. Since $x >> y = \frac{x^2}{2}$ for $|x| << 1$, so we can approximate $y \approx \frac{x^2}{2}$

## 0.18 Project: Cooling of Flat Plate

$$\eta = \frac{y}{\sqrt{\frac{V_x}{U_\infty}}}$$

Does heat diffuse faster than momentum? High viscosity ⇒ low heat diffusivity, high momentum diffusivity.

$$G(\eta) = \frac{t - t_\infty}{t_w - t_\infty}.$$

We can transform the PDEs into two coupled ODEs.

$$F''' + \frac{1}{2}FF'' = 0, F(0) = 0, F'(0) = 0, F'(\infty) = 1.$$

$$G'' + \frac{P_r F G')}{2} = 0, G(0) = 1, G(\infty) = 0.$$

where

$$P_r = \frac{V}{\alpha}.$$

This is a boundary value problem, not an IVP. We need to convert it to an IVP.

Using the garden hose technique, we adjust the value of $f''(0)$ until $F'(\infty) \to 1$. Then we can transform a third-order ODE into a system of three first order ODEs.

Let $P_r = 5$, guess $F''(0) = 0.332057, G'(0) = -0.576689$, use step size of 0.1 with $\eta$ from 0 to 10 (a number large enough to be considered as infinity),

$$
\begin{aligned}
F &= y_1 \\
F' &= y_1' = y_2 \\
F'' &= y_2' = y_3 \\
F''' &= y_3' = \frac{1}{2} F F'' \\
G &= y_4 \\
G' &= y_4' = y_5 \\
G'' &= y_5' = \frac{1}{2} P_r F G'
\end{aligned}
$$

where $y_1(0) = 0, y_2(0) = 0, y_3(0) = 332057, y_4(0) = 1, y_5(0) = 0.576689$.

We need $F'(10) \to 1, G'(10) \to 0$ by adjusting $F''(0), G''(0)$. And notice that F"(0) is positive, G'(0) is negative and bounded regardless of $P_r$.

Plot $F(\eta), G(\eta)$ vs $\eta$, and $\eta$ vs $P_r$.

## 0.19 Project 2

$$
\begin{aligned}
k &= Be^{-\frac{E}{RT}} \\
\frac{dA_f}{dt} &= -kA_f e^{-\frac{E}{RT}} \qquad T = T(t)
\end{aligned}
$$

### 0.19.1 Non-adiabotic explosion

$$
\begin{aligned}
\frac{dE}{dt} &= \frac{dQ}{dt} - S \qquad S = H(T - T_0) \\
C_N \frac{dT}{dt} &= -k\frac{dA_f}{dt} - H(T - T_0) \qquad T(0) = T_0
\end{aligned}
$$

where the LHS is the increase in internal energy, first term of RHS is the rate of heat release, and the last term is the rate of heat loss. Define $\hat{T} = \frac{T}{T_0} = 1 + \varepsilon\theta$, $\tau = \frac{t}{t_r}$, so that $\hat{T}(0) = 1$. $\varepsilon = \frac{T_0 R}{E}$.

$$
\frac{d\theta}{d\tau} = e^{\theta} - \frac{\theta}{\delta}.
$$

where $\delta \propto \frac{1}{H}$.

Let $\tau = \delta\sigma$ and we obtain

$$
\frac{d\theta}{d\sigma} = \delta e^{\theta} - \theta \qquad \theta(0) = 0.
$$

The more heat is lost to the environment, the more delay there is for the explosion time. We can do this to prevent explosion for forever. It's called a fizzle.

If $\delta e^{\theta} > \theta$, then $\theta$ always grows exponentially with time. If $\delta e^{\theta} < \theta$, then $\theta$ converges. At osculation point, both the magnitude and slope are equal:

$$\begin{aligned} \delta^* e^{\theta^*} &= \theta^* \\ \delta^* e^{\theta^*} &= 1 \end{aligned}$$

If $\delta > \frac{1}{e}$, explosion; If $\delta < \frac{1}{e}$, fizzle.

- $\theta << 1$ and $\sigma << 1$: we can use taylor expansion on $e^{\theta}$,giving

$$\begin{aligned} \frac{d\theta}{d\sigma} &= \delta(1 + \theta + \frac{\theta^2}{2} + \ldots) - \theta \\ &\approx \delta + (\theta - 1)\delta \\ \theta &= \frac{\delta}{\delta - 1}(e^{(\delta-1)\sigma} - 1) \end{aligned}$$

- $\sigma \to \infty$ and $\theta \to \theta_{\infty}$, so $\frac{d\theta}{d\sigma} << 1$

$$\begin{aligned} \frac{d\theta}{d\sigma} &\approx \theta \\ \frac{e^{\theta_f}}{\theta_f} &= \end{aligned}$$

- we can swap independent and dependent variables to avoid a stiff problem using RK4, solve for $\sigma$

$$\sigma = \frac{1}{\delta - 1} \ln \left[ \frac{\theta + \frac{\delta}{\delta - 1}}{\frac{\delta}{\delta - 1}} \right].$$

  This is zero divide by zero, so we need to use L'Hopital's Rule with respect to $\delta$ and get

$$\theta = \sigma \text{ early solution.}$$

- $\theta \to \infty$ and $\sigma \to \sigma_t$:

$$\begin{aligned} \frac{d\theta}{d\sigma} &\approx \delta e^{\theta} \\ -e^{-\theta} &= \delta\sigma + C, \text{let } C = -\delta\sigma_e \\ \sigma &= \sigma_e - \frac{e^{-\theta}}{\delta} \text{ explosion limit solution} \end{aligned}$$

We still need to find $\sigma_e$

$$\begin{aligned}\frac{d\sigma}{d\theta} &= \frac{1}{\delta e^{\sigma} - \theta} \\ \int \frac{d\sigma}{d\theta} &= \int_{\theta_0}^{\theta} \\ &= \int_0^{\eta} \frac{dx}{\delta e^x - x} \\ \sigma - 0 &= \int_0^{\eta} \frac{dx}{\delta e^x - x}\end{aligned}$$

## 0.20 Matrices

### 0.20.1 Round-off error

Ill-conditioned: if the columns are almost dependent. A small change in the RHS can yield drastically different solutions as an almost parallel line shifted.

If $\gamma$ is the angle between two linear equations, then

$$\tan\gamma = \frac{a_{11}a_{22} - a_{12}a_{21}}{a_{11}a_{12} + a_{21}a_{22}}.$$

As this quantity $\to 0$, $\gamma \to 0$.

Symptoms of ill-conditioned:

- if $|detA| << \max|a_{ij}| or \max|b_i|$
- poor approximation solutions with small residuals
- elements of $A^{-1}$ are large compared to elements of

Signs of well-conditioned: |diagonal elements|»|off-diagonal elements|

To tackle this problem:

- want the largest coefficient in all rows to be comparable, rescale the rows
- rearrange the rows to place the largest elements on diagonal

### 0.20.2 Gauss-Seidel Method

$$\begin{aligned}x_i &= \frac{b_i}{a_{ii}} \\ &\dots \\ x_i^* &= \frac{b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j}{a_{ii}} \\ x_i* &= \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^* - \sum_{j=i+1}^{n} a_{ij}x_j^*}{a_{ii}}\end{aligned}$$

### 0.20.3 Residuals

$$
\begin{aligned}
r &= b - Ax \\
r_i &= b_i - \sum_{j=1}^{n} a_{ij}x_j \\
&= b_i - \sum_{j=1}^{i-1} a_{ij}x_j - a_{ii}x_i - \sum_{j=i+1}^{n} a_{ij}x_j \\
&= b_i - a_{ii}x_i - (\sum_{j=1}^{i-1} a_{ij}x_j * + \sum_{j=i+1}^{n} a_{ij}x_j) \\
&= a_{ii}x_i^* - a_{ii}x_i \\
x_i^* &= x_i + \frac{r_i}{a_{ii}}
\end{aligned}
$$

### 0.20.4 Relaxation Methods

$$x_i^* = x_i + \omega\frac{r_i}{a_{ii}}.$$

where $\omega$ is the relaxation constant. If $0 < \omega < 1$, it is under-relaxation; if $\omega > 1$, it is over-relaxation. Some systems do not converge unless we use $0 < \omega < 1$. When using systems to solve PDEs can use over-relaxation to speed up convergence.

### 0.20.5 Matrix Inversion

#### Iterative Method

For the problem $x \cdot a = 1$, we can solve it using Newton's method on $f(x) = \frac{1}{x} - a = 0$. Then we have

$$x_{i+1} = x_i(2 - ax_i).$$

Can we extend this finding to matrices? Yes but it's only guaranteed to converge if all eigenvalues of $I - Ax$ $|\lambda_i| < 1$.

$$X_{i+1} = X_i(2I - AX_i).$$

This $X_i$ will eventually give us $A^{-1}$. The error is also $\mathcal{O}(h^2)$, for each entry.

### 0.20.6 Eigenvalues and Eigenvectors

The set of all eigenvalues are called spectrum. And $|\lambda_{\max}|$ is called the spectral radius.

#### Gershgorin Theorem

Let $\lambda$ be an eigenvalue of an arbitrary matrix $A = (a_{ij})$. Then

$$|a_{ii} - \lambda| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

The eigenvalues lie in the union of the Gershgorin disks (Gershgorin domain) centered at the diagonal entries with radius of the sum of the off-diagonal entries of that row.

**Collatz Theorem**

Let $A = (a_{ij})$ be a real square matrix with positive elements, and $x$ be any real vector with positive components, $y$ be the components of $y = Ax$. Then the closed interval bounded by $\left|\frac{y_i}{x_i}\right|_{\min}$ and $\left|\frac{y_i}{x_i}\right|_{\max}$ contains at least an eigenvalue of $A$.

**Rayleigh Quotient**

Given a real symmetric matrix $A$, a real and non-zero vector $x$, compute

$$x_i = Ax_{i-1}$$
$$m_i = x_i^T x_i TODO$$

Then $q = \frac{m_i}{m_{i-1}} = \frac{x_{i-1}^T x_i}{x_{i-1}^T x_{i-1}} TODO$ is an approximate to an eigenvalue of $A$. And the error $\varepsilon = q - \lambda$ is

$$|\varepsilon| \leq \sqrt{\frac{m_2}{m_0} - q^2} = \sqrt{\frac{x_i^T x_i}{x_{i-1}^T x_{i-1}} - \frac{x_{i-1}^T x_i}{x_{i-1}^T x_{i-1}}} TODO.$$

Why does it work? Multiplying lots of $A$s makes $x_{i-1}$ starting to look like $v_1$, so the Rayleigh quotient gives $\lambda_1$.

## 0.21 The Power Method

The ratio of elements at two neighboring iterations converges to the largest eigenvalue, and the vector itself converges to the associated eigenvector (scaled by some constant).

Why does it work?

We can write any arbitrary vector as a linear combination of its eigenvector basis, *i.e.*

$$w = \sum_{i=1}^{n} c_i v_i.$$

Multiplying $A$ results in

$$Aw = \sum_{i=1}^{n} c_i A v_i$$
$$= \sum_{i=1}^{n} c_i \lambda_i v_i$$

After $k$th iteration, we obtain

$$\begin{aligned} A^k w &= \sum_{i=1}^{n} c_i \lambda_i^k v_i \\ &= \lambda_1^k \sum_{i=1}^{n} c_i \frac{\lambda_i^k}{\lambda_1^k} v_i \\ &\approx c_i \lambda_1^k v_1 \\ A^{k+1} w &\approx c_i \lambda_1^{k+1} v_i \end{aligned}$$

If the arbitrary vector we initialize happens to not contain a $v_1$ component, then we would not get the first eigenvector.

Can we find the smallest eigenvalue of a positive definite matrix? Yes by using $A^{-1}$:

$$\begin{aligned} Ax &= \lambda x \\ \frac{1}{\lambda} x &= A^{-1} x \\ \text{so} \qquad \hat{\lambda}_{\max} &= \lambda_{\min} \end{aligned}$$

## 0.22 Steepest Descent

We can solve the step size that gives us the steepest descent by letting gradient of the descent step to zero. Each step would then be perpendicular to the previous step as we've exhausted descent in that direction, so any direction that is not orthogonal to this direction wouldn't maximize the descent. This method is very robust but the method is very expensive. It's linear convergence.

We can sample three points of $t$ and solve the minimum of the quadratic formed by the three points.

## 0.23 Fixed point method for a system

This might accelerate a naive fixed point method.

$$\begin{aligned} x_1^* &= g_1(x_1, x_2, \ldots, x_n) \\ x_2^* &= g_2(x_1^*, x_2, \ldots, x_n) \\ x_n^* &= g_n(x_1^*, x_2^*, \ldots, x_n) \end{aligned}$$

## 0.24 Newton's Method for root finding of a system

$$\begin{aligned} f_1(x, y, z) &= 0 \\ f_2(x, y, z) &= 0 \\ f_3(x, y, z) &= 0 \end{aligned}$$

Expand about point $(x,y,z)_{n-1}$

$$f_i(x,y,z) = f_i(x,y,z)_{n-1} + (x-x_{n-1})\frac{\partial f_i}{\partial x}(x,y,z)_{n-1} + (y-y_{n-1})\frac{\partial f_i}{\partial y}(x,y,z)_{n-1} + (z-z_{n-1})\frac{\partial f_i}{\partial z}(x,y,z)_{n-1}.$$

$$\begin{pmatrix} \frac{\partial f_1}{\partial x}(x,y,z)_{n-1} & \frac{\partial f_1}{\partial y}(x,y,z)_{n-1} & \frac{\partial f_1}{\partial z}(x,y,z)_{n-1} \\ \frac{\partial f_2}{\partial x}(x,y,z)_{n-1} & \frac{\partial f_2}{\partial y}(x,y,z)_{n-1} & \frac{\partial f_2}{\partial z}(x,y,z)_{n-1} \\ \frac{\partial f_3}{\partial x}(x,y,z)_{n-1} & \frac{\partial f_3}{\partial y}(x,y,z)_{n-1} & \frac{\partial f_3}{\partial z}(x,y,z)_{n-1} \end{pmatrix} \begin{pmatrix} x-x_{n-1} \\ y-y_{n-1} \\ z-z_{n-1} \end{pmatrix} = \begin{pmatrix} f_1(x,y,z)_{n-1} \\ f_2(x,y,z)_{n-1} \\ f_3(x,y,z)_{n-1} \end{pmatrix}.$$

$$x_n = x_{n-1} - J^{-1}(x_{n-1})F(x_{n-1}).$$

## 0.25 Heat conduction problem

Consider a rod with length $L$.

$$\rho c\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2}.$$

Where $\rho$ is the density, $c$ is the heat capacity, and $k$ is the heat conductivity.
Initial condition: $T(x,t)|_{t=0} = f(x)$
Boundary conditions: $T(x,t)|_{x=0} = T_0$, $T(x,t)|_{x=L} = T_0$

We want to nondimensionalize the problem to reduce complexity: Let $\alpha = \frac{k}{\rho c}$ be the thermal diffusivity, $\overline{x} = \frac{x}{L}$, $U = \frac{T-T_0}{T_r}$, so $x = \overline{x}L$, $T = UT_r + T_0$.

$$\begin{aligned} \frac{\partial UT_r + T_0}{\partial t} &= \alpha\frac{\partial^2 UTr + T_0}{\partial \overline{x}L^2} \\ T_r\frac{\partial U}{\partial t} &= \frac{T_r\alpha}{L^2}\frac{\partial^2 U}{\partial x^2} \\ \frac{\partial U}{\partial \frac{t\alpha}{L^2}} &= \frac{\partial^2 U}{\partial \overline{x}^2} \end{aligned}$$

Let $\tau = \frac{t\alpha}{L^2} = \frac{t}{t_{ref}}$, so $t_{ref} = \frac{L^2}{\alpha}$, and we obtain:

$$\frac{\partial U}{\partial \tau} = \frac{\partial^2 U}{\partial \overline{x}^2}.$$

$$\begin{aligned} U(\overline{x},\tau)|_{\tau=0} &= \frac{f(x)-T_0}{T_r} := F(x) \\ U(\overline{x},\tau)|_{\overline{x}=0} &= 0 \\ U(\overline{x},\tau)|_{\overline{x}=1} &= 0 \end{aligned}$$

$$U(\overline{x},\tau) = \sin(\pi\overline{x})e^{-t\pi^2}.$$

Make a grid in the $\overline{x} - \tau$ plane. Let $k = \Delta\tau$, $h = \Delta\overline{x}$. Using finite difference approximation:

$$\frac{U_{i,j+1} - U_{ij}}{k} = \frac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{h^2}.$$

the LHS is the forward difference in time, and the RHS is the central difference in space.

$$\begin{aligned} U_{i,j+1} &= U_{ij} + \frac{k}{h^2}(U_{i+1,j} - 2U_{ij} + U_{i-1,j}) \\ &= (1 - 2r)U_{ij} + r(U_{i+1,j} + U_{i-1,j}) \end{aligned}$$

where $r = \frac{k}{h^2} = \frac{\Delta\tau}{\Delta x^2}$.

Set $r = \frac{1}{4}$, $\Delta x = 0.2$, then $k = 0.01$. Set $r = 1$, $\Delta x = 0.2$, then $k = 0.04$. Starts to fall apart a little bit. Set $r = 2.5$,$\Delta x = 0.2$, then $k = 0.1$. Results violate physics.

### 0.25.1 Crank-Nicolson Method

We can fix the problem above by using an implicit method. We want to average $\frac{\partial^2 U}{\partial x^2}$ at time $j$ and $j+1$. Three points in the past and three points in the future.

$$\frac{U_{i,j+1} - U_{ij}}{k} = \frac{1}{2}\left(\frac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{h^2} + \frac{U_{i+1,j+1} - 2U_{i,j+1} + U_{i-1,j+1}}{h^2}\right)$$

$$-rU_{i+1,j+1} + (2 + 2r)U_{i,j+1} - rU_{i-1,j+1} = rU_{i+1,j} + (2 - 2r)U_{ij} + rU_{i-1,j}$$

This is much more stable. Symmetry gives us two equations for two unknowns.