CP315: Introduction to Scientific Computing

BY SCOTT KING

Dr. Ilias Kotsireas

1 Introduction

CP315 is a set of methods for solving mathematical problems with computers; fair enough - we will be using Maple and MatLab. Fundamental operations that are used: addition and multiplication. These are needed to evaluate a polynomial at a specific value. As we know, polynomials are basic objects in scientific computing $\rightsquigarrow$ efficient evaluation.

1.1 Polynomial Evaluation

Consider a general, fourth-degree polynomial:

$$P(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4$$

i. Find $P\left(\frac{1}{2}\right)$ naively requires substituting $\frac{1}{2}$ into $P(x) \rightsquigarrow$ 10 multiplications and 4 additions comes to a total of 14 operations.

ii. Store powers of $\frac{1}{2}$ progressively $\rightsquigarrow$ 3 multiplications (from the powers) + 4 multiplications (from the coefficients) and 4 additions. The new total is 11 operations.

iii. *Horner's Method*: Rewrite $P(x)$ "backwards":

$$P(x) = c_0 + x(c_1 + x(c_2 + x(c_3 + x(c_4))))$$

This brings it down to 8 total operations.

Fact: A degree d polynomial can be a evaluated in d multiplications and d additions.

Portfolio Part 1: Implement Horner's Method in Maple and/or MatLab.

1.1.1 Variation on the Theme

Evaluate:

$$\begin{aligned} P(x) &= x^5 + x^8 + x^{11} + x^{14} \\ &= x^5(1 + x^3 + x^6 + x^9) \\ &= x^5(1 + x^3(1 + x^3 + x^6)) \\ &= x^5(1 + x^3(1 + x^3(1 + x^3))) \end{aligned}$$

We get a total of 6 multiplications by 3 additions, thus 9 operations.

Overview of Calculus

Theorem 1. *Intermediate Value Theorem*

If $f(x)$ is continuous in $[a,b]$ then $\forall\ y$, such that, $f(a) \leq y \leq f(b)\ \exists\ c$, such that $a \leq c \leq b$ and $f(c) = y$.

***Corollary* 2.** *If $f(a), f(b) < 0$, then $\exists\ c$, such that $f(c) = 0$. Where c is a root of $f(x) = 0$.*

Theorem 3. *Mean Value Theorem*

If $f(x)$ is differentiable in $[a,b]$ then $\exists\ c$, such that $f'(c) = \frac{f(a) - f(b)}{b - a}$. Thus, there is a point where we will be able to calculate the slope at c.

***Corollary* 4.** *Rolle's Theorem*

If $f(x)$ is differentialable at $[a,b]$ then $\exists\ c$, such that $a \leq c \leq b$ and $f'(c) = 0$.

Theorem 5. *Taylor's Theorem*

If $f(x)$ is $(k+1)$-differentiable in $[x_0, x]$, $\exists\ c$, such that:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots + \frac{f^{k+1}(x_0)}{(k+1)!}(x - x_0)^{k+1} + R$$

where $R = \frac{f^{(k+1)}(c)}{(k+1)!}(x - x_0)^{k+1}$, is the remainder. If we know $f(x_0)$, then we can find nearby values $f(x)$ as a polynomial of degree k.

Example 6. $f(x) = \sin(x)$. Find a degree-4 Taylor polynomial (approximation) about $x_0 = 0$.

$$P_4(x) = x - \frac{x^3}{6}$$

with a remainder is $R = \frac{x^5}{120}\cos(c)$. Now, we need to estimate the size of the remainder term:

$$|R| \leq \frac{|x|^5}{120}$$

If $|x| \leq 10^{-4}$ then $|R| \leq \frac{10^{-20}}{120}$. This tells us that for all numbers $\leq 10^{-4}$, R is close to zero and thus the Taylor approximation is accurate.

Theorem 7. *Mean Value Theorem for Integrals*

If f is continuous in $[a,b]$ and g is integrable in $[a,b]$ and does not change sign in $[a,b]$ then, $\exists\ c$ such that $a \leq c \leq b$ and

$$\int_a^b f(x) g(x) \mathrm{dx} = f(c) \int_a^b g(x) \mathrm{dx}$$

Note: *This helps because this result gives us a way to evaluate $\int f(x)g(x)$ - as there is no defined way to do this.*

2 Floating Point Representation of Real Numbers ($\mathbb{R}$)

IEEE 754 is a standard to model floating point arithmetic on a computer. The problem is that we have finite-precision memory locations to represent infinite-precision numbers, YIKES.

IEEE 754 is a set of **binary** representations of real numbers.

A floating point, or real, number has three parts:

1. Sign ($\pm$) - s
2. Mantissa (AKA significant digits) - m
3. Exponent - e

These three parts are stored in a word. There are three common precision types:

1. Single: 32 bits, (s: 1, m: 8, e: 23)
2. Double: 64 bits (s: 1, m: 11, e: 52)
3. Long-double: 80 bits, (s: 1, m: 15, e: 64)

Definition 2.1. *A normalized IEEE 754* ***floating point number*** *is the following:*

$$\pm 1.b_1 b_2 \ldots b_N \times 2^p$$

where p *is an* M*-bit binary number; where*

$$b_i \in \{0,1\}, i = 1, \ldots, N$$

Example 2.2. 9 decimal and we want to convert to an IEEE FLP number.

$$\begin{aligned} 9 &\rightarrow 1001\,(\text{binary}) \\ +1 &\;.\; 001 \times 2^3 \\ N &= 3 \\ P &= 3 \end{aligned}$$

Multiplication by power of 2 $\equiv$ a shift.

Typical double precision parameters in C/MatLab: $M = 11$, $N = 52$.

Example 2.3. We want to represent 1.

$$\begin{aligned} 1 &\rightsquigarrow 0001 \\ +1 &\;.\; 0 \ldots 0_{52} \times 2^0\,(52\,\text{zeroes}) \end{aligned}$$

What is the "next" number we can represent? The answer is: $+1.0\ldots0_{51}1 \times 2^0 \rightsquigarrow 1 + 2^{-52}$, this is 51 zeroes.

Definition 2.4. *Machine epsilon,* E_{mach}*, is the distance between* 1 *and the smallest FLP number greater than 1.*

Remark **2.5.** *For IEEE 754, double precision, we have* $\mathrm{E}_{\text{mach}} = 2^{52}$.

2.1 IEEE Nearest Rounding Rule

Example 2.6. 9.4 in decimal $\rightarrow 1001.\overline{0110}$

The binary representation of 0.4 $\approx\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^6}+\frac{1}{2^7}+\ldots=\sum_{k=1}^{\infty}\left(\frac{1}{2^{4k+2}}+\frac{1}{2^{4k+3}}\right)$

We need to fit this precision number in 52 bits.

$$1.\underline{001}011001100110\ldots0110\,\underline{0}\times 2^3$$

We have the three bits in the beginning following by 12 sets of 0110:

$$3\,\text{bits}+12\times 4\,\text{bits}=51\,\text{bits}$$

RMR: Look at the 53rd bit to the right of the radix point: $\begin{cases}1\to\text{add 1 to bit 52}\\0\to\text{do nothing}\end{cases}$

So in our example: 53rd bit is 1, so we add 1 to 52.

Thus, 9.4 is represented as:

$$+1.001\underline{0110}\,\mathbf{1}\times 2^3$$

which is actually $9.4+0.2\times 2^{-49}$ in decimal.

Remark 2.7. The IEEE double precision number associated with 9.4 using RNR is:

$$fl(9.4)=9.4+0.2\times 2^{-49}$$

where 0.2×2^{-49} is the error.

Definition 2.8.

$$\begin{aligned}x_c &= \text{computed value of } x\\ \text{absolute error} &= |x_c-x|\\ \text{relative error} &= \frac{|x_c-x|}{|x|}\end{aligned}$$

***Remark* 2.9.** *Relative error in IEEE 754 is bounded by:*

$$\frac{|fl(x)-x|}{|x|}\leq\frac{1}{2}E_{\text{mach}}$$

2.2 Loss of Significant Digits

Example 2.10. $E_1=\frac{1-\cos(x)}{\sin^2(x)}$ and $E_2=\frac{1}{1+\cos(x)}$. $\therefore E_1=E_2$ in exact arithmetic. Evaluate E_1 and E_2 numerically for $x=1.000\ldots$, $x=0.100\ldots$, $x=0.010\ldots$.

Remark 2.11. For values of $x<10^{-5}$, E_1 losses signifcant digits. For $x<10^{-8}$, E_1 has no correct significant digits. Well, we are subtracting numbers that are nearly equal.

Example 2.12. $x^2+9^{12}x-3=0$, with $a=1$, $b=9^{12}$, $c=-3$.

$$\begin{aligned}\Delta &= \sqrt{b^2-4\text{ac}}\\ x &= \frac{-b\pm\Delta}{2a}\\ \oplus\rightsquigarrow\ x= &\ \frac{-b+b}{2a}=0\end{aligned}$$

But how?! We need to restructure the formula, using the conjugate quantity:

$$\begin{aligned}\frac{-b+\sqrt{\Delta}}{2a} &\times \left(\frac{-b+\sqrt{\Delta}}{-b+\sqrt{\Delta}}\right)\\ &= \frac{\Delta-b^2}{2a\left(b+\sqrt{\Delta}\right)^2}\\ &= \frac{-4\text{ac}}{2a\left(b+\sqrt{\Delta}\right)}\\ &= \frac{-2c}{b+\sqrt{\Delta}}\end{aligned}$$

Note: This formula only applies for degree-2 polynomials.

3 Equation Solving

- We will explore iterative methods to locate solutions of $f(x)=0$
- Convergence, complexity

We are also going to look at three different methods of solving equations:

1. Bisection
2. Fixed-point
3. Newtons's method

3.1 Bisection Method

- We are looking to solve $f(x)=0$
- Means find r, st $f(r)=0$
- Existence of r: IVT

Steps:

1. Find $[a,b]$ st $f(a)\times f(b)<0$
2. Then, $\exists r\colon a<r<b$ st $f(r)=0$

Example 3.1. $f(x)=x^3+x-1$, we know $f(0)=-1$, $f(1)=1$ and thus:

$$\rightsquigarrow \exists r\in[0,1]\,\text{st}\,f(r)=0$$

Also:

$$f\left(\frac{1}{2}\right)<0\rightsquigarrow f\left(\frac{1}{2}\right)\times f(1)<0\rightsquigarrow r\in\left[\frac{1}{2},1\right]$$

Next step in the interation:

$$f\left(\frac{1}{2}\right)>0\rightsquigarrow f(0)\times f\left(\frac{1}{2}\right)<0\rightsquigarrow r\in\left[0,\frac{1}{2}\right]$$

And thus we know:

$$f\left(\frac{1}{2}\right) < 0$$

We now know that $\frac{1}{2} < f\left(\frac{1}{2}\right) < 1$. We know can check the midpoint of $\left[\frac{1}{2}, 1\right]$ which is $\frac{3}{4}$. Next interation:

$$f\left(\frac{3}{4}\right) > 0 \rightsquigarrow r \in \left[\frac{1}{2}, \frac{3}{4}\right]$$

Portfolio Part 2: Implement Bisection Method in Maple and/or MatLab.

Algorithm 3.1

Bisection Method
Input: `f`, `a`, `b` st. $f(a) \times f(b) < 0$; tolerance (ἐπσιλον') - `e`
Output: approximate root `r`, in $[a, b]$, $f(r) = 0$

```
while (b-a)/2 > e do
     r=(a+b)/2
     if f(r)=0 then return r
     if f(a)*f(r) < 0
          b=r
          else
               a=r
     return (a+b)/2
```

Example 16 cont.

ϵ	#`while` step	approx `r`
10^{-4}	13	0.6823
10^{-5}	16	0.6823
10^{-6}	19	0.68232
10^{-7}	23	0.68232780

Definition 3.2. *An approximate solution is correct to* <u>*p decimal places*</u> *if the error*

$$|x_c - r| < \frac{1}{2} 10^{-p}$$

3.1.1 Error Analysis

- Start $[a, b]$
- After n bisection steps $[a_n, b_n]$

$$x_c = \frac{a_n + b_n}{2} \rightsquigarrow |x_c - r| < \frac{b-a}{2^{n+1}}$$

Question 3.3. *How many bisection steps are needed to compute a solution correct to 6 decimal places?*

Answer. *Error after n bisection steps: $\frac{1}{2^{n+1}}$ and thus*

$$\begin{aligned}\frac{1}{2^{n+1}} &< \frac{1}{2}10^{-6}\\ 10^6 &< 2^n\\ \log(10^6) &< \log(2^n)\\ 6\times\log(10) &< n\times\log(2)\\ 6 &< n\times\log(2)\\ 19.9 &< n\end{aligned}$$

And thus we need 20 steps to compute 0.739085.

3.2 Fixed-Point Iteration

Definition 3.4. *r is a fixed point (fp) of a function $g(x)$, iff $g(r)=r$.*

Example 3.5. $g(x)=x^3$. We have three fixed points: $0, \pm 1$.

Observation. Finding a fp of $g(x) \Leftrightarrow$ solving the equation: $g(x)-x=0$ where we can define $g(x)-x$ as $f(x)$.

Algorithm 3.2

FPI
Input: $f(x)=g(x)-x$, initial guess, x_0
Output: approximate solution of $f(x)=0$, (ie. a fp of $g(x)$)

```
for i = 0..k
   $x_{i+1}$=g($x_i$)
```

If the sequence x_0, x_1, x_2, ... converges to a value, r, then r is a fp of $g(x)$. For some j: $|x_{j+1}-x_j| <$ E.

Question 3.6. *Can any fct, $f(x)$ be written as $g(x)-x$?*

Answer. *Yes, and often in more ways than one.*

Example 3.7. $x^2+x-1=0$

$$x=1-x^3 \tag{3.1}$$

$$x=(1-x)^{\frac{1}{2}} \tag{3.2}$$

$$x=\frac{1+2x^3}{1+3x^2} \tag{3.3}$$

Use fp iterations with $x_0=0.5$.

1. The interates flip-flop from 0 to 1, **no convergence**

2. The iterates converge to 0.6823 in 25 iterations

Explanation: $|g'(r)>1, <1|$

Example 3.8.

$$\begin{aligned}g_1(x) = & -\frac{3}{2}x+\frac{5}{2} \quad \text{with } r=1 \text{ and } |g_1'(1)|=\frac{3}{2}>1\\ g_2(x) = & -\frac{1}{2}x+\frac{3}{2} \quad \text{with } r=1 \text{ and } |g_2'(1)|=\frac{1}{2}<1\end{aligned}$$

Thus, we have $x_{i+1}=g_1(x_i)$. Consider $g(x) \rightsquigarrow$

$$x_{i+1}-1=-\frac{3}{2}(x_i-1)$$

denote by $e_i=|1-x_i|$ then $e_{i+1}=\frac{3}{2}e_i \rightsquigarrow$ error increases, divergent.

Consider $g_2(x)$ with $x_{i+1}=g_2(x_i) \rightsquigarrow$

$$x_{i+1}-1=-\frac{1}{2}(x_i-1)$$

then $e_{i+1}=\frac{1}{2}e_i$. 1

Definition 3.9. *Denote by e_i, the error at step i, of an iterative method.*

$$e_i=|r-x_i|$$

*The method **converges linearly** with rate, S, if:*

$$\lim_{i\to\infty}\frac{e_{i+1}}{e_i}=S$$

and $S<1$.

Observation. *f-p iteration for $g_2(x)$ converges linearly with rate $S=\frac{1}{2}$.*

Theorem 3.10. *Assume g is differentiable.*

$$\begin{aligned}g(r) = & \quad r \qquad \text{and } r \text{ is an fp of } g\\ |g'(r)| = & \quad S<1\end{aligned}$$

Then, the fp iteration for g, conerges linearly with rate, S to r. For initial guesses, x_0, sufficiently close to r.

Example 3.11. $f(x)=x^3+x-1$ in the form of $g(x)=x$.

1. $g_1(x)=1-x^3$, now $|g_1'(x)|=3x^2 \longrightarrow x=0.6823 \longrightarrow >1$
2. $g_2(x)=(1-x)^{\frac{1}{3}}$, now $|g_2'(x)|=\frac{1}{3}(1-x)^{-\frac{2}{3}}+1 \longrightarrow x=\ldots \longrightarrow <1$ $\quad \therefore$ converges
3. $g_3(x)=\frac{1+2x^3}{1+3x^2}$, now $|g_3'(x)|=\frac{(6x^2)(1+3x^2)+(6x)(1+2x^3)}{(1+3x^2)^2} \longrightarrow x=\ldots \longrightarrow 0<1$ We have a linear convergence with rate, $S=0$

3.2.1 Stopping Criteria for FPI

Where do we need to stop the iteration?

1. Bounded absolute error:

$$|x_{i+1}-x_i|<\mathrm{E}$$

2. Bounded relative error:

$$\frac{|x_{i+1}-x_i}{|x_{i+1}|}<\mathrm{E}$$

Example 3.12. $\begin{cases} g(x)=\frac{x+\frac{2}{x}}{2} \\ x_0=1 \end{cases}$. Set up FPI as $g(x)=x$.

$$g(\sqrt{2})=\frac{\sqrt{2}+\frac{2}{\sqrt{2}}}{2}=\sqrt{2}$$

3.2.2 Forward and Backward Error

Example 3.13. $f(x)=x^3-2x^2+\frac{4}{3}x-\frac{8}{27}$. Use the bisection method to compute a root with 6 correct significant digits. We have $f(0)\,f(1)<0$ and $f\left(\frac{2}{3}\right)=0$.

Observation. 16 steps $\rightsquigarrow$ 0.6666641. We cannot get the 6th digit correct.

IEEE double precision, there are many float point numbers within 10^{-5} of the correct root $r=\frac{2}{3}$.

Definition 3.14. *Suppose a function, f, with root r and $f(r)=0$. Also, x_a is an approximation to r computed by a root-finding method.*

Backwards error (BE): $|f(x_a)|$

Forward error (FE): $|r-x_a|$

BE amount by which we need to change $f(x)$ so that x_a is a solution. FE amount by which we need to change the approximate solution to make it correct.

***Remark* 3.15.** *The problem we encountered with the previous example is that the BE is near* $\mathrm{E}_{\mathrm{mach}}=10^{-16}$ *and the FE* $\approx 10^{-5}$.

Definition 3.16. *Multiple Roots*

f, a differentiable function, with root r and $f(r)=0$ if

$$o=f(r)=f'(r)=f''(r)=\ldots=f^{(m-1)}(r)$$

and $f^{(m)}(r)\neq 0$. Then, f, has a root of multiplicity m, at r; where r is a multiple root of order m of f.

$$\begin{cases} m=1; r\,\text{single root} \\ m=2; r\,\text{double root} \\ m=3; r\,\text{triple root} \end{cases}$$

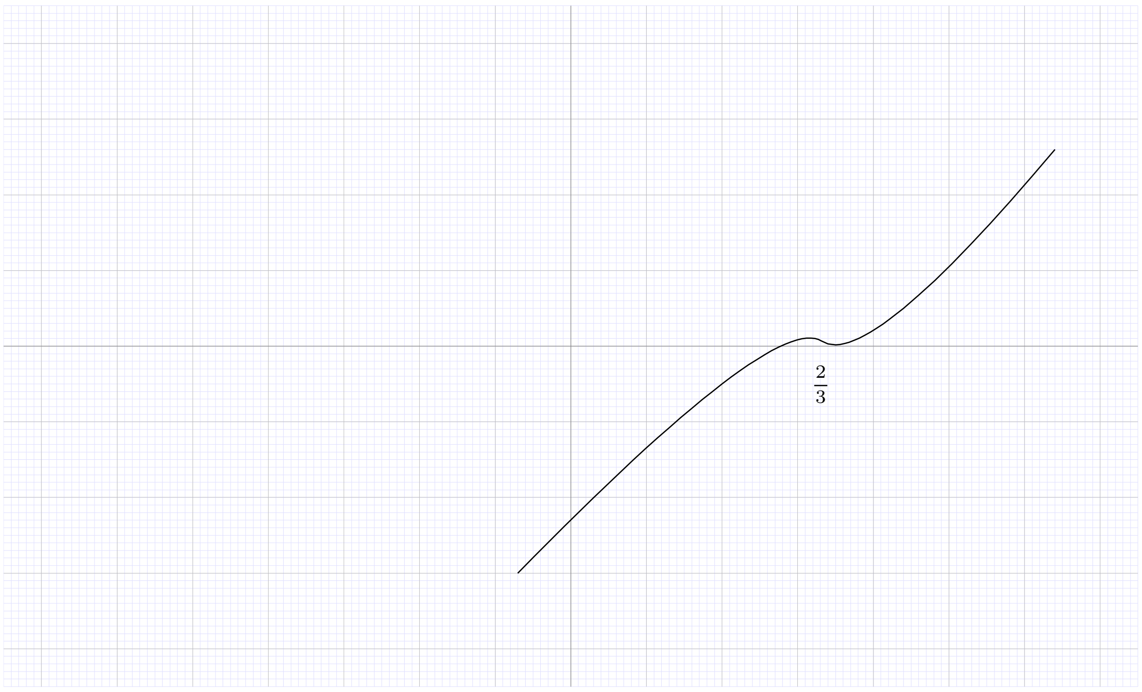
Example 3.17. $f(x)=x^2$, has a double root at $x=0$.

$$\begin{array}{ccc} f(0) & = & 0 \\ f'(0) & 2x|_{x=0} & =0 \\ f''(0) & = & 2\neq 0 \end{array}$$

Also, $f(x)=x^3$ has a triple root at $x=0$.

Geometric Intuition.

Graph of $f(x) = \left(x - \frac{2}{3}\right)^3$.



$\frac{2}{3}$ is a triple root. The graph is [supposed to be] flat around the triple root.

Consequence. Large disparity between BE and FE.

$$\text{BE} \ll \text{FE}$$

where BE is the vertical dimension and FE is the horizantal dimension.

Example 3.18. $f(x) = \sin(x) - x$ has a triple root at $r = 0$ and $x_a = 0.001$ is the approx. root. Compute BE and FE.

$$\begin{aligned} \text{BE:} \quad |f(x_a)| &= 10^{-3} \\ \text{FE:} \quad |r - x_a| &= |\sin(0.001) - 0.001| \\ &\approx 1.6667 \times 10^{-10} \end{aligned}$$

Stopping Criteria. Either make FE small or make BE small.

$$\text{Bisection:} \begin{cases} \text{BE: known} \\ \text{FE:} < \frac{b-a}{2} \end{cases}$$

```
> if abs(f(x_a)) < E then ...
> if abs((b − a)/2) < E then ...
```

$$\text{Fixed point:} \begin{cases} \text{BE: known} \\ \text{FE: not known} \end{cases}$$

3.2.3 Wilkinson Polynomial

$$\begin{aligned} W(x) &= (x-1)(x-2)\ldots(x-20) \\ &= \prod_{i=1}^{20} (x - i) \end{aligned}$$

It has 20 (simple) roots, $x = 1, \ldots, 20$. To compute numerical approximations to the roots of the expanded $W(x)$ is very hard. $W(x) = x^{20} \pm \ldots$

3.2.4 Sensitivity of Root Finding

A problem is called sensitive if small errors in the input (the eq. we are solving). This leads to large errors in the output (solution).

We need to measure how far a root is moved when the eq. is changed (perturbed).

Proposition 3.19. *Supposed, we change $f(x) \to f(x) + \epsilon\, g(x)$.*

Let Δ_r be the corresponding change to the root r.

$$f(r + \Delta_r) + \epsilon\, g(r + \Delta_r) = 0$$

D-d-d-d-d-drop the Taylor and neglect the higher order terms (H.O.T).

$$\Delta_r \approx -\frac{\epsilon\, g(r)}{f'(r)}$$

for $\epsilon \ll f'(r)$.

Example 3.20. Estimate the largest root of

$$P(x) = \left(\prod_{i=1}^{6} (x - i) \right) - 10^{-6} x^7$$

Get all emotional and used the sensitivity formula.

$$\begin{aligned} \epsilon &= -10^{-6} \\ g(x) &= x^7 \\ f(x) &= P(x) \end{aligned}$$

Now:

$$\Delta_r \approx \frac{\epsilon\, 6^7}{5!} = -2332.8\, \epsilon$$

and thus:

$$6 + \Delta_r = 6.0023$$

3.3 Newton's Method

Problem. Find a root of a fn, $f(x) = 0$.

The first thing we need to do is start with an initial guess of x_0. Draw the tangent line to f at x_0 and identify the point where the tangent intersects with the x-axis.

We can get the equation of the tangent line at $(x_0, f(x_0))$.

$$y - f(x_0) = f'(x_0)\, (x - x_0)$$

where $f'(x_0)$ is the slope. The intersection of the above equation and the x-axis $\rightsquigarrow y = 0$, we obtain:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Algorithm 3.3

Input: $f(x)$, x_0 (initial guess)
Ouput: approximation to root r st $f(r)=0$
The way it would iterate:

$$x_{i+1}=x_i-\frac{f(x_i)}{f'(x_i)}$$

where $i=0,1,2,...$

Example 3.21. $f(x)=x^3+x-1$.

First compute $f'(x)=3x^2+1$. Newton iteration is:

$$\begin{aligned} x-\frac{f(x)}{f'(x)} &= x-\frac{x^3+x-1}{3x^2+1} \\ &= \frac{2x^3+1}{3x^2+1} \end{aligned}$$

Thus we have a generalized form: $x_{i+1}=\frac{2x_i^3+1}{3x_i^2+1}$ with $x_0=0.1$.

We can perform 6 iterations to give you the root with the correct 8 significant digits: 0.68232780.

Definition 3.22. *Denote e, as the error at step i: $|r-x_i|$.*

The iterative method converges quadratically $\Longleftrightarrow$

$$\lim_{i\to\infty}\frac{e_{i+2}}{e_i^2}=M$$

***Remark* 3.23.** *If $f(r)=0$, and $f'(r)=0$, then Newton's method converges quadratically to r and thus, $M=\frac{f''(r)}{2f'(r)}$, where **M** denotes the rate of convergence.*

Example 40 Continued. We then have $f''(x)=6x$ with $r=0.6823$ and $M\approx 0.85$.

Example 3.24. $f(x)=x^2$, with $r=0$ being a double root.

$$x_{i+1}=x_i-\frac{f(x_i)}{f'(x_i)}=x_i-\frac{x_i^2}{2x_i}=\frac{x_i}{2}$$

We can make an educated guess for $x_0=1$.

i	x_i	$e_i \longrightarrow \frac{e_i}{e_{i-1}}$
0	1.000	$1.000 \longrightarrow$ —
1	0.500	$0.500 \longrightarrow 0.500$
2	0.250	$0.250 \longrightarrow 0.500$
3	0.125	$0.125 \longrightarrow 0.500$

Remark 3.25. If f has a root with multiplicity, m, then Newton's method is locally convergent to r, with rate: $\frac{m-1}{m}$:

$$\frac{e_{i+1}}{e_i}\approx\frac{m-1}{m}$$

Example 3.26.

1. $f(x) = \sin(x) + x^2\cos(x) - x^2 - x \longrightarrow$ find the mult. of root $r = 0$

   - First check that $f(0) = 0$, $f'(0) = 0$, ... We have computed the multiplicity when $f^{(n)}(0) \neq 0$. In this case, $f''(0) \neq 0 ... f'''(0) \neq 0$. Thus, we have a triple root $\rightarrow$ with a convergence rate of $\frac{2}{3}$. The error decreases by $\frac{2}{3}$ at each iteration.

2. Estimate the number of Newton iterations to converge with 6 significant digits

   - Suppose that we choose $x_0 = 1$. The estimate is $\left(\frac{2}{3}\right)^n < \frac{1}{2}10^{-6} \longrightarrow n > 35$

3.3.1 Modified Newton's Method

If $f(x)$ has a root of mult. m, then:

$$x_{i+1} = x_i - m\frac{f(x_i)}{f'(x_i)}$$

converges quadratically to r.

3.3.2 Alternative Derivation to Newton's Method

Assume $f(x) = 0$, x_0. The Taylor series expansion about x_0 is:

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0)\, f'(x_0) + \underline{\text{HOT}} \rightsquigarrow \\ x &= x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

where the HOT are neglected.

Example 3.27. Solve $f(x) = x^2 + 1 \rightsquigarrow r = \pm i$. Now, $x_0 = 0.1 \pm i$. This converges to i in 4 iterations.

3.3.3 Newton's Method Can Fail

Example 3.28. $f(x) = 4x^4 - 6x^2 - \frac{11}{4} = 0$ with $x_0 = \frac{1}{2}$ (bi-quadratic equation). We then have the Newton iteration being:

$$x_{i+1} = x_i - \frac{f(x_i)}{16x_i^3 - 12x_i}$$

$$\begin{aligned} x_1 &= -\frac{1}{2} \\ x_2 &= \frac{1}{2} \\ x_3 &= \frac{1}{2} \\ x_4 &= -\frac{1}{2} \\ &\vdots \end{aligned}$$

There is no convergence. Roots are ± 1.3667. We have an even function st $f(x) = f(-x)$ and thus

$$f\left(\frac{1}{2}\right) = f\left(-\frac{1}{2}\right) = -4$$

The tangent lines at $\left(\frac{1}{2}, 4\right)$ and $\left(-\frac{1}{2}, 4\right)$ will intersect at the x-axis at $-\frac{1}{2}, \frac{1}{2}$.

Example 3.29. $f(x) = \sin(2x)$, with $x_0 = 0.75$. Newton's method converges to the root -2π. We convrege to and missed the closer root, 0.

Why? This occurs, if $f(x_i)$ is very small, then the tangent line is almost horizantal.

Example 3.30. $f(x) = x\,e^x$. Newton iteration with $x_0 = 2$. The Newton iteration is:

$$x_{i+1} = x_i - \frac{x_i\,e^{-x_i}}{e^{x_i} - x_i\,e^{-x_i}} = \frac{x_i^2}{x_i - 1}$$

$$\begin{aligned} x_1 &= 4 \\ x_2 &= \frac{16}{3} \\ x_3 &= \ldots \\ &\vdots \\ &\to \infty \end{aligned}$$

This converges to infinity and thus fails.

3.3.4 Multivariate Newton's Method

Example 3.31. $f_1(x_1, x_2) = x_1^2 + x_2^2 - 8x_1 - 4x_2 + 11 = 0$ and $f_2(x_1, x_2) = x_1^2 + x_2^2 - 20x_1 + 75 = 0$.

These equations give us two circles. We need to solve the system $\left\{ \begin{smallmatrix} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{smallmatrix} \right.$. The solution of the system will give us the two points where the circles intersect (twice).

Let's start with the initial condition $(x_1^\circ = 2, x_2^\circ = 4)$. First step: compute 4 partial derivatives.

$$\begin{aligned} \frac{\delta f_1}{\delta x_1} &= 2x_1 - 8 \\ \frac{\delta f_2}{\delta x_1} &= 2x_2 + 20 \end{aligned}$$

And then:

$$\begin{aligned} \frac{\delta f_1}{\delta x_2} &= 2x_2 - 4 \\ \frac{\delta f_2}{\delta x_2} &= 2x_2 \end{aligned}$$

This gives us: $f_1(x_1^\circ, x_2^\circ) = -1$ and $f_2(x_1^\circ, x_2^\circ) = 55$. We can then put these values in a Jacobian matrix: $J(x_1, x_2) = (f_1, f_2) =$

$$\begin{pmatrix} 2x_1 - 8 & 2x_2 - 4 \\ 2x_1 - 20 & 2x_2 \end{pmatrix}$$

We need compute it's value at $(2, 4)$. This gives us:

$$J_{(f_1, f_2)}(2, 4) = \begin{pmatrix} -4 & 4 \\ -16 & 8 \end{pmatrix}$$

To compute the next iterate (x_1^1, x_2^1). To do this, we compute:

$$\begin{aligned} x_1^1 &= x_1^\circ + \Delta x_1 \\ x_2^1 &= x_2^\circ + \Delta x_2 \end{aligned}$$

Now solve:

$$\begin{pmatrix} -4 & 4 \\ -16 & 8 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -55 \end{pmatrix}$$

Using the values of $\Delta x_1 = 7.125$ and $\Delta x_2 = 7.375$. This gives us:

$$\begin{aligned} x_1^1 &= 2 + 7.125 = 9.125 \\ x_2^1 &= 4 + 7.375 = 11.375 \end{aligned}$$

This converges in 8 iterations.

Algorithm 3.4

Input: $f_1(x_1, \ldots, x_n) = 0$ (square system of non-linear equations)
$\vdots$
$f_n(x_1, \ldots, x_n) = 0$
ἐπσιλον', initial point $(x_1^\circ, \ldots, x_n^\circ)$
Output: Approx. solution $\vec{r} = (r_1, \ldots, r_n)$ st $f_1(\vec{r}) = \ldots = f_n(\vec{r}) = 0$

```
i = 1
while ($\left|f_j(\vec{r})\right| \geq$ eps, j = 1) do
    i = i+1
    solve system of linear eqs
    $J_{(f_1:::f_n)}(x_1^i; :::; x_n^i) \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} = \begin{pmatrix} -f_1(\vec{x^i}) \\ \vdots \\ -f_n(\vec{x^i}) \end{pmatrix}$
    $x_j^{(i)} = x_j^{(i-1)} + \Delta x_j$, j=1..n
```

Let's talk Jacobian dude.

Jacobian $n \times n$ Matrix

$$J_{(f_1, \ldots, f_n)}(x_1, \ldots, x_n) = \begin{pmatrix} \frac{\delta f_1(\vec{x})}{\delta x_1} & \cdots & \frac{\delta f_1(\vec{x})}{\delta x_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_n(\vec{x})}{\delta x_1} & \cdots & \frac{\delta f_n(\vec{x})}{\delta x_n} \end{pmatrix}$$

3.4 Secant Method (root-finding without derivatives)

We want to replace the tangent with the *secant* line:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

So we replace the tangent line with a discrete approximation.

Algorithm 3.5

Input: $f(x)$, x_0, x_1, ϵ (two initial guesses)
Output: approximation to the root, r

$x_{i+1} = x_i - f(x_i) \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} = x_i - f(x_i) \frac{1}{f'(x_i)}$

Where $i = 1, 2, 3, \ldots$

Error Analysis:

If $f'(r) \neq 0$ (simple root), and the secant method converges, then $e_{i+1} \approx c\, e_i^{\phi}$. This is called *superlinear convergence*.

We have the golden ratio, $\phi = \frac{1+\sqrt{5}}{2} \approx 1.6$

4 Systems of Linear Equations

$$\begin{cases} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=b_2 \\ \vdots \\ a_{n1}x_1+a_{n2}x_2+\ldots+a_{nn}x_n=b_n \end{cases}$$

In an $n \times n$ system, there are n equations and n variables. The matrix form of a system:

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ & & \vdots & \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Solution methods:

Direct methods:

- Gauss elimination
- Gauss-Jordan
- LU decomposition

Iterative methods:

- Jacobi
- Gauss-Seidel
- Relaxtion

4.1 Vector, Matrix Norms

If we have $\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, we need to measure the size of $\vec{x}$ by the *Euclidean* norm:

$$\|\vec{x}\| = \sqrt{x_1^2+\ldots x_n^2}$$

Then we have the L_p norm:

$$\|\vec{x}\| =^p \sqrt{|x_1|^p+\ldots+|x_n|^p}$$

The following results occur:

$$\begin{aligned} p=1 &\rightsquigarrow \text{sum of absolute values of the elements of } \vec{x} \\ p=2 &\rightsquigarrow L_p = \text{Euclidean norm} \\ p=\infty &\rightsquigarrow \max|x_i| \end{aligned}$$

We can denote $A = (\, a_{ij} \,)$, where A is an $m \times n$ matrix (rows $\times$ columns).

We have the *Frobenius* norm:

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} |a_{ij}|^2}$$

where the $\|A\|_F$ has $m \times n$ terms.

p-norms:

$$\|A\|_p = \max \left\|A\vec{x}\right\|_p$$

where $\left\|\vec{x}\right\| = 1$.

Next, we look at 4 square matrices $n \times n$:

$$\|A\|_2 = \max \lambda_i$$

where $i = 1, ..., n$. λ_i are the eigenvalues of A.

Next, the *spectral* norm:

$$\begin{aligned} \|A\|_1 &= \max_{1 \leq j \leq n} \left(\sum_{i=1}^{n} |a_{ij}| \right) \\ \|A\|_\infty &= \max_{1 \leq i \leq n} \left(\sum_{j=1}^{m} |a_{ij}| \right) \end{aligned}$$

4.1.1 Norm Inequalities

1. $\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2$
2. $\frac{1}{\sqrt{n}} \|A\|_\infty \leq \|A\|_2 \leq \sqrt{n}\|A\|_\infty$
3. $\frac{1}{\sqrt{n}} \|A\|_1 \leq \|A\|_2 \leq \sqrt{n}\|A\|_1$

4.1.2 Solution Concepts

Remember: $A \times \vec{x} = \vec{b}$ where the sizes are $n \times n$, $n \times 1$ and $n \times 1$ respectively.

1. If $\vec{b} = \vec{0}$ then it is a homogeneous system
   - Trivial solution of $\vec{x} = \vec{0}$
   - A non-trivial solution exists if $\det(A) = 0$
2. If $\vec{b} \neq \vec{0}$, then it is a non-homogeneous system, denoted as *augmented* matrix

$$A' = \left[A/\vec{b}\right]$$

   - System has a solution $\Longleftrightarrow \text{rank}(A) = \text{rank}(A')$
   - Solution is unique if $\text{rank}(A) = n$; A is considered **full rank**
   - If $\text{rank}(A) < n$, then it is denoted as **inconsistent**

Definition 4.1. *Linearly dependent rows and equations*

$$A = \begin{pmatrix} 1 & -1 & 1 \\ 2 & 1 & -1 \\ 8 & 1 & -1 \end{pmatrix} \rightsquigarrow \begin{cases} r_1 \\ r_2 \\ r_3 \end{cases}$$

And we have: $2r_1 + 3r_2 = r_3$. *And* r_1, r_2 *and* r_3 *are linearly dependent.*

4.2 Rank and Nullspace of a Matrix ($\boldsymbol{A}$: $\boldsymbol{m} \times \boldsymbol{n}$)

We can have:

$$\begin{aligned} \text{range}(A) &= \left\{ \vec{y} \in \mathbb{R}^m : \vec{y} = A\vec{x}, \text{for some } \vec{x} \in \mathbb{R}^n \right\} \\ \text{rank}(A) &= \dim(\text{range}(A)) \\ \text{nullspace}(A) &= \left\{ \vec{x} \in \mathbb{R}^n : A\vec{x} = \vec{0} \right\} \end{aligned}$$

Theorem 4.2. $\dim(\text{nullspace}(A)) + \text{rank}(A) = n$

Where the $\text{rank}(A)$ *is the maximum, linearly independent number of rows.*

Definition 4.3. *A is a full rank if*

$$\text{rank}(A) = \min(m, n)$$

A is rank-defficient if

$$\text{rank}(A) < \min(m, n)$$

4.3 Cramer's Method

We have $A\vec{x} = \vec{b}$ where A is an $n \times n$ matrix. We define:

$$A_i = \left(\vdots \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \vdots \right)$$

with $i = 1, \ldots, n$. Then, we can have $x_i = \frac{|A_i|}{|A|}$ with $i = 1, \ldots, n$.

Example 4.4. Given:

$$\begin{aligned} x_1 - x_2 + x_3 &= 3 \\ x_1 + x_2 - x_3 &= 0 \\ 3x_1 + 2x_2 + 2x_3 &= 15 \end{aligned}$$

That looks like:

$$|A| = \begin{pmatrix} 1 & -1 & 1 \\ 2 & 1 & -1 \\ 3 & 2 & 2 \end{pmatrix} = 1 \times \begin{pmatrix} 1 & -1 \\ 2 & 2 \end{pmatrix} - (-1) \times \begin{pmatrix} 2 & -1 \\ 3 & 2 \end{pmatrix} + 1 \times \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}$$

We then have:

$$|A_1| = \begin{pmatrix} 3 & -1 & 1 \\ 0 & 1 & -1 \\ 15 & 2 & 2 \end{pmatrix} = 12, x_1 = \frac{|A_1|}{|A|} = \frac{12}{12} = 1$$

Next:

$$|A_2| = \begin{pmatrix} 1 & 3 & 1 \\ 2 & 0 & -1 \\ 3 & 15 & 2 \end{pmatrix} = 24, x_2 = \frac{24}{12} = 2$$

And:

$$|A_3| = \begin{pmatrix} 1 & -1 & 3 \\ 2 & 1 & 0 \\ 3 & 2 & 15 \end{pmatrix} = 48, x_3 = \frac{48}{12} = 4$$

4.4 Gauss Elimination

The **strategy** is reduce to [upper] triangular form, and then run back substitution. This reduction is carried out by *elementary row operations*:

- Multiply divide by any row or constant value
- Any row can be added/subtracted to/from any row
- Interchange rows

Example 4.5. Suppose we have the system:

$$\begin{aligned} 2x_1 - x_2 + x_3 &= 4 \\ 4x_1 + 3x_2 - x_3 &= 6 \\ 3x_1 + 2x_2 + 2x_3 &= 15 \end{aligned}$$

Use elementary row operations to obtain:

$$\begin{aligned} x_3 &= 4 \\ x_2 &= 2 \\ x_1 &= 1 \end{aligned}$$

4.4.1 Determinant Computation via GEM

If we have $A \times \vec{x} = \vec{b}$ and by GE we reduce it to $\tilde{A} \times \vec{x} = \vec{b}$. We can then say:

$$\det(A) = \det\left(\tilde{A}\right)$$

We can then define the property:

$$\det(B) = \prod_{i=1}^{n} b_{i_i}$$

where B is triangular. From the previous example, we have

$$\det(A) = 2 \times 5 \times \frac{13}{5} = 26$$

where those numbers come from the diagonal of the [upper] triangle matrix.

4.5 Gauss-Jordan

It is an extension of GEM. Use a *pivot* to reduce (to 0). Element below and above the main diagonal - apply back substitution as needed.

$$A \overset{\text{GJ}}{\rightsquigarrow} I_n$$

What happens is we end up finding the solution, in the augment, by transforming A in I_n.

4.5.1 Inverse Matrix Computation via GJE

$$\left(\ A|I_n\ \right)\overset{\text{GJ}}{\rightsquigarrow}\left(\ I_n|A^{-1}\ \right)$$

The way this one works, is transforming $A\rightarrow I_n$ and the augment now has the inverse.

4.6 LU Decomposition

$$A=L\times U$$

Where L is a lower triangular matrix and U is an upper triangular matrix. We can start by saying A has n^2 elements, L has $\frac{n(n+1)}{2}$ and U has $\frac{n(n+1)}{2}$.

We have n extra variables in LU wrt A. There are 3 LU variations:

1. $u_{ii}=1$ where $i=1,...,n\longrightarrow$ Crout
2. $l_{ii}=1$ where $i=1,...,n\longrightarrow$ Doolittle
3. $l_{ii}=u_{ii}$ where $i=1,...,n\longrightarrow$ Choleski

This proposes 2 issues:

1. Find LU given A
2. Solve $A\times\vec{x}=\vec{b}$

The second:

$$\left(\text{LU}\times\vec{x}\right)=\vec{b}$$

and denote $U\times\vec{x}=\vec{z}$. We then have $l\times\vec{z}=\vec{b}$.

Example 4.6. Let's say we have $A=\begin{pmatrix} a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{pmatrix}$. Find the LU decomposition, of A, using Crout's method.

$$\begin{array}{ccc} A & = & L\times U\rightsquigarrow\\ \begin{pmatrix} a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{pmatrix} & = & \begin{pmatrix} l_{11} & 0 & 0\\ l_{21} & l_{22} & 0\\ l_{31} & l_{32} & l_{33}\end{pmatrix}\times\begin{pmatrix} 1 & u_{12} & u_{13}\\ 0 & 1 & u_{23}\\ 0 & 0 & 1\end{pmatrix}\\ 9\,a_{ij}\text{ variables} & & 6\,l_{ij}\text{ variables and }3\,u_{ij}\text{ variables}\end{array}$$

Computation:

$$=\begin{pmatrix} l_{11} & l_{11}u_{12} & \ldots\\ l_{21} & l_{21}u_{12}+l_{22} & \ldots\\ l_{31} & l_{31}u_{12}+l_{32} & \ldots\end{pmatrix}$$

Our objective is to find l_{ij} and u_{ij} in terms of a_{ij}. Equate A and $LU \rightsquigarrow$

$$\begin{aligned} l_{11} &= a_{11} \\ l_{21} &= a_{21} \\ l_{31} &= a_{31} \\ u_{12} &= \frac{a_{12}}{a_{11}} \\ \vdots & \vdots \ \vdots \end{aligned}$$

We can find the rest of the matrix entries but solving the rest of the computation.

Example 4.7. Use LU decomposition of A to solve the linear system $A \times \vec{x} = \vec{b}$.

$$A = \begin{pmatrix} 2 & -1 & 1 \\ 4 & 3 & -1 \\ 3 & 2 & 2 \end{pmatrix}, \vec{b} = \begin{pmatrix} 4 \\ 6 \\ 15 \end{pmatrix}$$

$$A = \ L \times U = \begin{pmatrix} 2 & 0 & 0 \\ 4 & 5 & 0 \\ 3 & \frac{7}{2} & \frac{13}{5} \end{pmatrix} \times \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 1 & -\frac{3}{5} \\ 0 & 0 & 1 \end{pmatrix}$$

We can then denote $\vec{z} = U \times \vec{x}$ (2), and then the original system becomes: $L \times \vec{z} = \vec{b}$ (1). We can then solve (1) to obtain:

$$\vec{z} = \begin{pmatrix} 2 \\ -\frac{2}{5} \\ 4 \end{pmatrix}$$

and solve (2) to obtain:

$$\vec{x} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$$

Note: LU is especially efficient when we need to solve several linear systems with the same A and different constant vectors, $\vec{b_i}$ because we need to find the LU decomposition of A, only once; and use it k times.

Theorem 4.8. *An $n \times n$ matrix, A, has an LU decomposition if* $\det(A(1{:}k, 1{:}k)) \neq 0$ *for $k = 1, \ldots,$ $n - 1$.*

***Notation* 4.9.** *This $A(1{:}\,k, 1{:}\,k)$ is called a **leading principal minor** of A. If $k = 1$, we take the first element of the matrix being the leading principal minor of $k = 1$. If $k = 2$, then we take 4 elements, and that is the leading principal minor for $k = 2$. This increases by an exponential factor of 2. Then can continue until $n - 1$.*

Example 4.10. $n = 3$, $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 7 \\ 3 & 5 & 3 \end{pmatrix}$.

This matrix does not have a LU decomposition, because, for the previous theorem, for $k = 2$, there is a determinant of $\begin{vmatrix} 1 & 2 \\ 2 & 4 \end{vmatrix} = 0$.

Example 4.11. We have $A = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 6 & 4 \\ 1 & -4 & 5 \end{pmatrix}$. For $k = 1, 2$ the leading principal minors have $\det \neq 0$.

NOTE: If a_{11}=0, we cannot compute the LU decomposition.

4.7 Iterative Methods to Solve Linear Systems

1. Jacobi Method, $\boldsymbol{A \times \vec{x} = \vec{b}}$

$$\begin{aligned} a_{11}x_1 + a_{21}x_2 + \ldots + a_{1n}x_n &= b_1 \rightsquigarrow x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - \ldots - a_{1n}x_n) \\ \vdots &= \vdots \\ a_{n1}x_1 + \ldots + a_{nn}x_n &= b_n \rightsquigarrow x_n = \frac{1}{a_{nn}}(b_n - a_{n1}x_1 - \ldots - a_{nn-1}x_{n-1}) \end{aligned}$$

In summary, $x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{n} a_{ij}x_j\right)$, for $i = 1, \ldots, n$. We need to exclude $j = i$ from the sum, so we have $n-1$ terms.

Our initial guess is: $x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}$. For example, we could have $x_1^{(1)} = 0, x_2^{(1)} = 0, \ldots, x_n^{(1)} = 0$.

We then substitute our initial guess into the Jacobi iteration formula and obtain $x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)}$. This would be the next approximation to the solution.

In general:

Substitute $x_i^{(k)}$ for $i = 1, \ldots, n$ into the Jacobi iteration formula to obtain $x_i^{(k+1)}$, the next iteration.

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{n} a_{ij}x_j^{(k)}\right)$$

excluding $j = i$ so we have $n-1$ terms.

Convergence/stopping criterion:

$$|x_i^{(k+1)} - x_i^{(k)}| \leq \epsilon, \forall i = 1, \ldots, n$$

There is a condition for the convergence of the Jacobi matrix: if A, is *diagonally dominant*, then Jacobi will converge to the solution.

Definition 4.12. *Matrix A, is* ***diagonally dominant*** *if*

$$|a_{ii}| > \sum_{j=1}^{n} |a_{ij}|$$

excluding $j = i$ and $\forall i = 1, \ldots, n$.

So basically, if the diagonal element of that row is larger than the sum of the absolute value of all other elements in the same row, and this occurs for every row in the matrix, then the matrix is considered diagonally dominant. Sexy.

Example 4.13. We have:

$$\begin{aligned} -5x_1 - x_2 + 2x_3 &= 1 \\ 2x_1 + 6x_2 - 3x_3 &= 2 \\ 2x_1 + x_2 + 7x_3 &= 32 \end{aligned}$$

Check if this is diagonally dominant. The first row:

$$|-5| > |-1| + |2|$$

This checks out.

$$|6|>|2|+|-3|$$

This also checks out.

$$|7|>|2|+|1|$$

Nice! It works. Let's call it a *domimatrix*. Sorry. It is diagonally dominant.

$$\begin{aligned} x_1 &= -\frac{1}{5}(1+x_2-2x_3) \\ x_2 &= \frac{1}{6}(2-2x_1-3x_3) \\ x_3 &= \frac{1}{7}(32-2x_1-x_2) \end{aligned}$$

Our initial approximation is $x_1^{(1)}=0, x_2^{(1)}=0, x_3^{(1)}=0$. The next approximation is:

$$\begin{aligned} x_1^{(2)} &= -\frac{1}{5} \\ x_2^{(2)} &= \frac{1}{3} \\ x_3^{(2)} &= \frac{32}{7} \end{aligned}$$

The next approximation is:

$$\begin{aligned} x_1^{(3)} &= 1.56 \\ x_2^{(3)} &= 2.68 \\ x_3^{(3)} &= 4.58 \end{aligned}$$

The Jacobi converges to the solution $x_1=1$, $x_2=2$ and $x_3=4$.

2. Gauss-Seidel

This is an improvement upon the Jacobi method: $x_i^{(k+1)}$ are generated for $x_i^{(k)}$.

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$$

If A is diagonally dominant, the $G-S$ method converges to the solution.

5 Ill-conditions and Condition Numbers

Definition 5.1. *A system of equations is said to be* ***ill-conditioned*** *if small changes in the coefficients lead to very large changes in the solution.*

Example 5.2.

$$\begin{aligned} x_1-x_2 &= 5 \\ k\,x_1-x_2 &= 4 \end{aligned}$$

There are k parameters, $k\neq 1$ and $k=1$ has no solutions.

$$\det(A)=\begin{vmatrix} 1 & -1 \\ k & -1 \end{vmatrix}=k-1$$

The solution is:

$$x_1 = \frac{1}{1-k}, x_2 = \frac{5\,k-4}{1-k}$$

We then have (k approaching 1 from negative and positive side):

$$\begin{array}{l}\text{if } k \to 1^- \quad \text{then} \quad x_1 \in [3000, 10000] \\ \text{if } k \to 1^+ \quad \text{then} \quad x_2 \in [-10000, -3000]\end{array}$$

Thus we have an ill-condition around $k = 1$.

We can quantify ill-conditionness using the **condition number**.

Definition 5.3.

$$\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$$

(using various norms). This is a mesasure of how ill-conditioned a problem is.

Properties:

$$\begin{array}{rcl}\text{cond}(I_n) & = & 1 \\ \text{cond}(A) & \geq & 1\end{array}$$

If we have a large $\text{cond}(A)$*, example 10000, then we say the linear system is ill-conditioned.*

Example 65. (cont.)

$$A^{-1} = \frac{1}{\det(A)} \times \begin{pmatrix} -1 & 1 \\ -k & 1 \end{pmatrix}$$

We then have two cases: [we will use the ∞ norm - max. row sum]

1. $k > 1$

$$\begin{array}{rcl}\|A\|_\infty & = & k+1 \\ \|A^{-1}\|_\infty & = & \dfrac{k+1}{k-1}\end{array}$$

   We then have $\text{cond}(A) = \frac{(k+1)^2}{k-1}$.

$$\begin{array}{l}k = 1.0001 \rightsquigarrow \text{cond}(A) \approx 40000 \\ k = 1.0002 \rightsquigarrow \text{cond}(A) \approx 20000\end{array}$$

2. $k < 1$

$$\begin{array}{rcl}\|A\|_\infty & = & 2 \\ \|A^{-1}\|_\infty & = & \dfrac{2}{1-k}\end{array}$$

   We have $\text{cond}(A) = \frac{4}{1-k}$.

$$\begin{array}{l}k = 0.998 \rightsquigarrow \text{cond}(A) \approx 20000 \\ k = 0.999 \rightsquigarrow \text{cond}(A) \approx 40000\end{array}$$

The condition number is $\approx 10^s$. Thus, we may end up losing s significant digits.

5.1 Positive Definite Matrices

Definition 5.4. *A $(n \times n)$ matrix is* ***positive-definite*** *if:*

define $\overrightarrow{x}$ as an $n \times 1$.

$$\overrightarrow{x} \cdot A \cdot \overrightarrow{x} > 0, \forall \overrightarrow{x} \neq \overrightarrow{0}$$

Example 5.5. Show that $A = \left(\begin{smallmatrix} 2 & 2 \\ 2 & 5 \end{smallmatrix}\right)$ is positive-definite. Let's use $n = 2$.

Take $\overrightarrow{x} = \left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right)$ and prove that:

$$\begin{pmatrix} x_1 & x_2 \end{pmatrix} \times \begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} > 0$$

for every x_1, x_2. Then $\rightsquigarrow 2(x_1 + x_2)^2 + 3\,x^2 \geq 0$. This will be equal to 0 when

$$\begin{cases} x_1 + x_2 = 0 \longrightarrow x_1 = 0 \\ x_2 = 0 \longrightarrow x_2 = 0 \end{cases}$$

Example 5.6. Show that $A = \left(\begin{smallmatrix} 2 & 4 \\ 4 & 5 \end{smallmatrix}\right)$ is **not** positive-definite. Thus, exhibit $\overrightarrow{x}$ such that

$$\overrightarrow{x} \cdot A \cdot \overrightarrow{x} \leq 0$$

<u>Fact:</u>

Suppose A is symmetric and positive-definite, then A is non-singular.

<u>Fact:</u>

If A is symmetric and positive-definite then $\exists$ upper triangular matrix U s.t. $A = U^T \cdot U$.

6 Matrix Eigenvalue Problems

Standard eigenvalue problem:

$$A \cdot \overrightarrow{x} = \lambda \cdot \overrightarrow{x}$$

where:

$$\begin{aligned} \overrightarrow{x} &= n \times 1 \, \text{column vector} \longrightarrow \text{the eigenvector} \\ \lambda &= \text{scalar, the eigenvalue} \\ A &= n \times n \, \text{matrix} \end{aligned}$$

We have an equivalent formulation where:

$$(A - \lambda \cdot I_n) \cdot \overrightarrow{x} = \overrightarrow{0}$$

This is a homogenous system of linear equations.

Note: If we want a non-trivial solution to exist, then $\det(A) = 0$ and thus the system is homogenous.

Then: ** $|A-\lambda \cdot I_n| = 0$.

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22}-\lambda & \dots & a_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ a_{1n}-\lambda & \dots & \dots & a_{nn}-\lambda \end{vmatrix} = 0$$

** is the characteristic polynomial of the standard equation eigen problem has degree n and its roots are the eigenvalues:

$$\lambda_1, \dots, \lambda_2$$

We have then, for each eigenvalue λ_i, we can determine an eigenvector $\vec{x}_{(i)}$ such that $A \cdot \vec{x}_{(i)} = \lambda_i \cdot I_n$ and $\left(\lambda_i, \vec{x}_{(i)}\right)$ come in pairs. Thus, $\vec{x}_{(i)}$ is the eigenvector associated to the eigenvector associated to the eigenvalue λ_i.

Example 6.1. $A = \left(\begin{smallmatrix} 4 & 1 \\ 3 & 2 \end{smallmatrix}\right)$. Solve the standard eigen problem.

$$\begin{aligned} |A-\lambda \cdot I_2| &= 0 \\ \begin{pmatrix} 4-\lambda & 1 \\ 3 & 2-\lambda \end{pmatrix} &= 0 \\ \rightsquigarrow \lambda^2 - 6\lambda + 5 &= 0 \,(\text{char poly}) \\ \rightsquigarrow \lambda_1 = 1 \quad &, \quad \lambda_2 = 5 \end{aligned}$$

Thus, we have two (eigenvalue, eigenvector) pairs.

$$\begin{aligned} A \cdot \vec{x}_{(1)} &= \lambda_1 \cdot \vec{x}_{(1)} \rightsquigarrow \vec{x}_{(1)} = \begin{pmatrix} -\frac{1}{3} \\ 1 \end{pmatrix} \\ A \cdot \vec{x}_{(2)} &= \lambda_2 \cdot \vec{x}_{(2)} \rightsquigarrow \vec{x}_{(2)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{aligned}$$

Thus:

$$\begin{aligned} \begin{pmatrix} -\frac{1}{3} \\ 1 \end{pmatrix} &= \begin{pmatrix} -\frac{1}{3} \\ 1 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} &= 5 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{aligned}$$

6.1 Generalized Eigenvale Problem

$$A \cdot \vec{x} = B \cdot \vec{x}$$

Where A, B are symmetric matrices, λ is the eigenvalue and $\vec{x}$.

1. B is diagonal, $B = C \cdot C$ with C diagonal, $c_{ii} = \sqrt{b_{ii}}$

$$C^{-1} \cdot A \cdot C^{-1} \cdot \vec{x} = \lambda \cdot \vec{x}$$

2. B is symmetric and positive-definite, $B = U^T \cdot U$

$$(U^T)^{-1} \cdot A \cdot U^{-1} \cdot \vec{x} = \lambda \cdot \vec{x}$$

Characteristics of Eigen Problems:

A	$\lambda_1, \ldots, \lambda_n$
singular	at least one $\lambda_i = 0$
non-singular	all $\lambda_i \neq 0$
symmetric	all λ_i are real
diagonal	$\lambda_i = a_{ii}$, $i = 1, \ldots, n$
A^{-1}	eigenvalues are $\frac{1}{\lambda_i}$
$P^T \cdot A \cdot P$ (orthogonal)	remain unchanged

6.2 Computational Methods

- Determinant method
- F-L method
- Power method

6.2.1 Determinant Method

$$\begin{aligned}
A \cdot \vec{x} &= \lambda \cdot \vec{x} \rightsquigarrow \\
|A - \lambda \cdot {}_n| &= 0 \rightsquigarrow \\
\text{characteristic poly.} &\rightarrow (-1)^n (\lambda^n - p_1 \cdot \lambda^{n-1} - \ldots - p_{n-1} \cdot \lambda - p_n) = 0 \\
p_1 &= a_{11} + a_{22} + \ldots + a_{nn} \\
&= \mathrm{Trace}(A)
\end{aligned}$$

Example 6.2. We have $A = \begin{pmatrix} 12 & 6 & -6 \\ 6 & 16 & 2 \\ -6 & 2 & 16 \end{pmatrix}$, with $n = 3$. A is symmetric, so the eigenvalues are real. We need to determine p_1, p_2, p_3:

$$\begin{array}{ccc}
|A - \lambda I_3| = 0 & , & \begin{pmatrix} 12-\lambda & 6 & -6 \\ 6 & 16-\lambda & 2 \\ -6 & 2 & 16-\lambda \end{pmatrix} = 0 \\
\text{polynomial} & \rightsquigarrow & -\lambda^3 + 44\lambda^2 - 564\lambda + 1728 = 0 \\
\lambda_1 = 18, \quad \lambda_2 = 13 + \sqrt{73}, \quad \lambda_3 = 13 - \sqrt{73} & &
\end{array}$$

Since all λ_i are positive, we conclude that A is positive-definite.

$$p_1 = \mathrm{Trace}(A) = 12 + 16 + 16 = 44$$

The eigenvector associated to $\lambda_1 = 18$ comes after we solve: $A \cdot \vec{x} = 18 \cdot \vec{x} \rightsquigarrow \vec{x} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$.

6.2.2 Faddeev-Leverrier Method

The idea behind this method is to generate the coefficients $p_1, \ldots, p_n$ of the characteristic polynomial by trace computations of the matrices $P_1, \ldots, P_n$.

$$\begin{array}{rcl}
P_1 = A & \rightsquigarrow & p_1 = \mathrm{Trace}(P_1) = \mathrm{Trace}(A) \\
P_2 = A\,(P_1 - p_1 I_n) & \rightsquigarrow & p_2 = \frac{1}{2}\mathrm{Trace}(P_2) \\
\vdots & \vdots & \vdots \\
P_i = A\,(P_{i-1} - p_{i-1} I_n) & \rightsquigarrow & p_i = \frac{1}{i}\mathrm{Trace}(P_i) \\
\vdots & \vdots & \vdots \\
P_n = A\,(P_{n-1} - p_{n-1} I_n) & \rightsquigarrow & p_n = \frac{1}{n}\mathrm{Trace}(P_n)
\end{array}$$

Example 6.3. Use the power method to compute the dominant eigenvalue; $A = \begin{pmatrix} 12 & 6 & -6 \\ 6 & 16 & 2 \\ -6 & 2 & 16 \end{pmatrix}$.
The initial trial vector, $\vec{X}_1 = \begin{pmatrix} 1.0 \\ 0.5 \\ -0.5 \end{pmatrix}$.

We then have: $\vec{X}_2 = A \cdot \vec{X}_1 = \begin{pmatrix} 18.0 \\ 13.0 \\ -13.0 \end{pmatrix}$ with $c = 18.0 \rightsquigarrow \begin{pmatrix} 1 \\ 0.722 \\ -0.722 \end{pmatrix}$.

The next iteration will be: $\vec{X}_3 = A \cdot \vec{X}_2 = \begin{pmatrix} 20.66 \\ 16.11 \\ 16.11 \end{pmatrix}$ with $c = 20.66 \rightsquigarrow \begin{pmatrix} 1 \\ 0.779 \\ -0.779 \end{pmatrix}$.

This will go through 8 iterations to obtain $\lambda_1 = 21.5439$.

6.3 Symmetric Tridiagonal Matrices

$A = \begin{pmatrix} a_1 & b_2 & 0 & 0 \\ b_2 & a_2 & \ddots & 0 \\ 0 & \ddots & \ddots & b_n \\ 0 & 0 & b_n & a_n \end{pmatrix}$, with $2n-1$ elements and $O(n)$.

We will compute the characteristic polynomial of A by a recursive construction.

Note: The char. polynomial of $A = |A - \lambda \cdot I_n|$.

$$\begin{aligned} f_0(\lambda) &= 1 \\ f_1(\lambda) &= a_1 - \lambda \\ f_2(\lambda) &= f_1(\lambda) \cdot (a_2 - \lambda) - b_2^2 \cdot f_0(\lambda) \\ \vdots & \ \vdots \ \vdots \\ f_n(\lambda) &= f_{n-1}(\lambda) \cdot (a_n - \lambda) - b_n^2 \cdot f_{n-2}(\lambda) \end{aligned}$$

We than have that $f_n(\lambda)$ is the characteristic polynomial of A.

Example 6.4. Given $A = \begin{pmatrix} 3 & -2 & 0 & 0 \\ -2 & 5 & -3 & 0 \\ 0 & -3 & 7 & -4 \\ 0 & 0 & -4 & 9 \end{pmatrix}$.

$$\begin{aligned} f_0(\lambda) &= 1 \\ f_1(\lambda) &= 3 - \lambda \\ f_2(\lambda) &= \lambda^2 - 8\lambda + 11 \\ \vdots & \ \vdots \ \vdots \\ f_4(\lambda) &= \lambda^4 - 24\lambda^3 - 177\lambda^2 - 444\lambda + 274 \end{aligned}$$

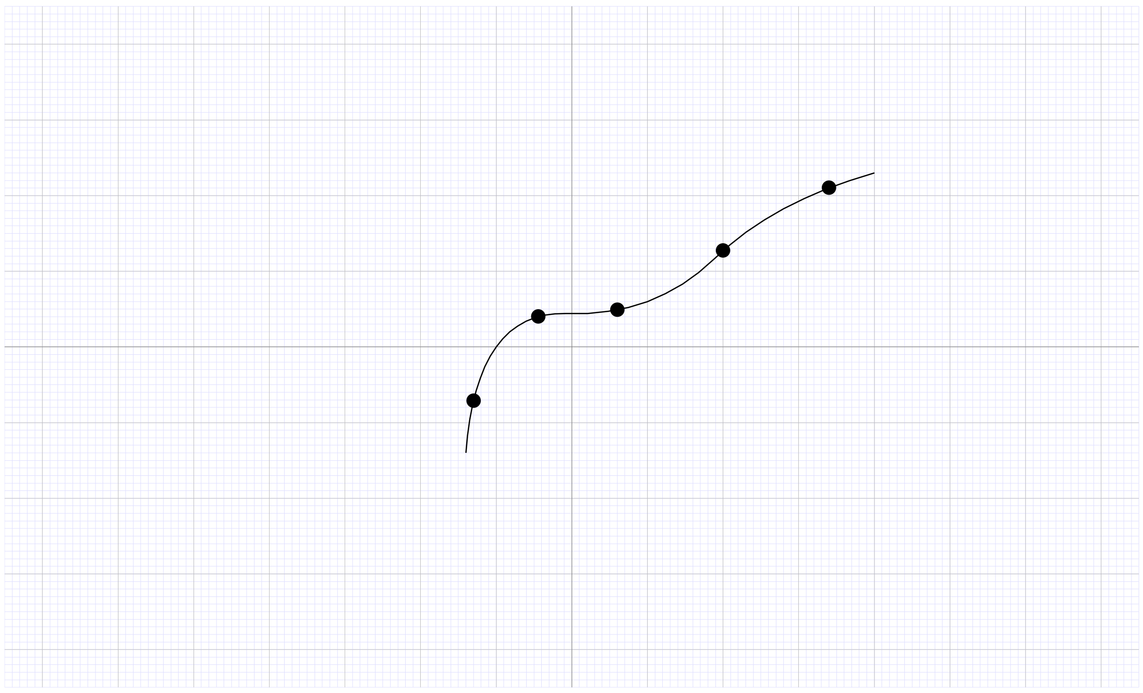
We know that it has 4 positive roots in $[0, 13]$. Thus, we know it is positive definite.

7 Interpolation (AKA Curve Fitting)

The **main idea** behind interpolation is that we have data available as discrete points, and we want to apply a smooth/continuous curve to this data.

7.0.1 1$^{\text{st}}$ Approach: Collocation

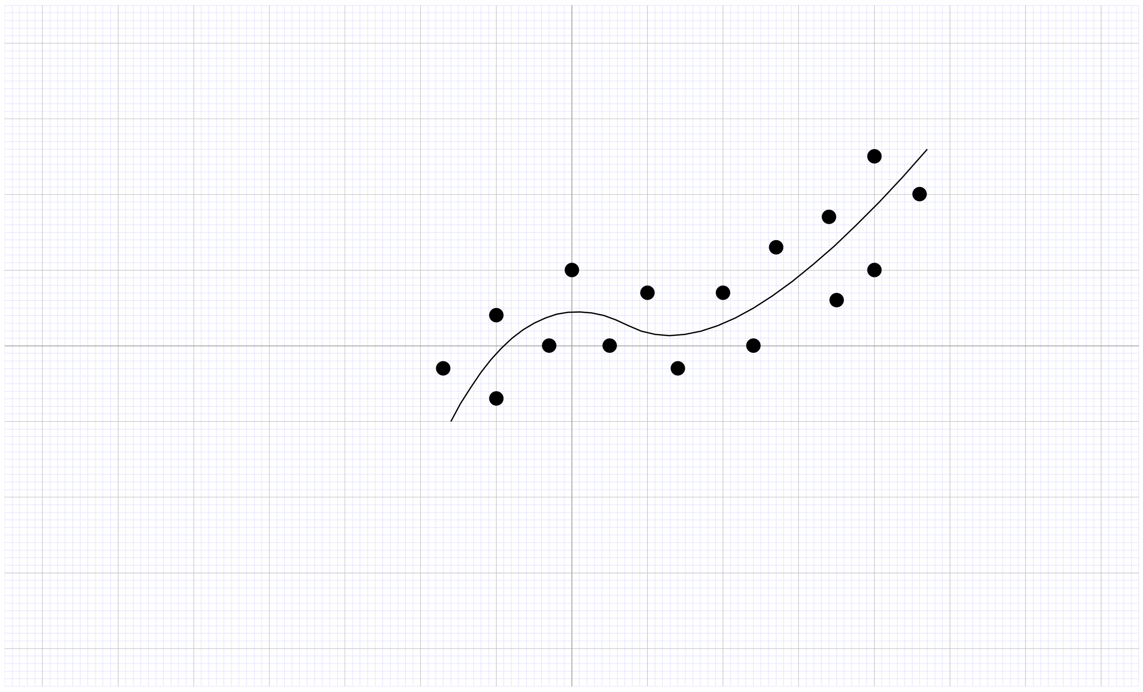
The curve passes *exactly* through all the data points.



$$f(x) = a_n \cdot x^n + \ldots + a_x \cdot x + a_0$$

7.0.2 2nd Approach: Least Squares

The curve indicates a general trend of the data.



Interpolation: Find $f(x) = y$.

Extrapolation: Predict/deduce data values (y-values) outside of the given range. **Ex.** What is $(0, ??)$, or $(11, ??)$.

Method 1.

Example 7.1. Collocation, polynomial fit.

We have $n+1$ data points: $(x_0, y_0), \ldots, (x_n, y_n)$.

$$y = f(x) = a_n \cdot x^n + \ldots + a_1 \cdot x + a_0$$

With:

$$f(x_0) = y_0, \ldots, f(x_n) = y_n$$

Above, is a set of $n+1$ linear equations in $a_0, \ldots a_n$.

We will aim to get the form of $B \cdot \vec{a} = \vec{y}$, where $\vec{a} = \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix}$ and $\vec{y} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$. We can understand that B is a Vandermonde matrix.

$$B = \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix}$$

And thus:

$$|\det(B)| = \prod_{\substack{i>j \\ i=0,\ldots,n \\ j=0,\ldots,n}} (x_i - x_j)$$

Note: There is a **unique** polynomial of degree n, that passes through $n+1$ data points.

7.1 Lagrange Interpolation

The **idea** is to avoid solving a linear system by using a *semi-factored* form of the interpolation polynomial.

Example 7.2. Fit a quadratic through 3 (=2+1) data points:

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2$$

With

$$y = f(x) = a_0(x - x_1)(x - x_2) + a_1(x - x_0)(x - x_2) + a_2(x - x_0)(x - x_1)$$

And the three data points are:

$$\begin{array}{c} (x_0, y_0) \\ (x_1, y_1) \\ (x_2, y_2) \end{array}$$

We have

$$\begin{array}{ll} f(x_0) = y_0 & , \quad a_0(x_0 - x_1)(x_0 - x_2) = y_0 \rightsquigarrow a_0 = \dfrac{y_0}{(x_0 - x_1)(x_0 - x_2)} \\ f(x_1) = y_1 & , \quad a_1(x_1 - x_0)(x_1 - x_2) = y_1 \rightsquigarrow a_1 = \dfrac{y_1}{(x_1 - x_0)(x_1 - x_2)} \\ f(x_2) = y_2 & , \quad \ldots = y2 \rightsquigarrow a_2 = \dfrac{y_2}{(x_2 - x_0)(x_2 - x_1)} \end{array}$$

Example 7.3. We have :

$$\begin{aligned}(x_0, y_0) &= (0,0)\\(x_1, y_1) &= (1,1)\\(x_2, y_2) &= (2,3)\end{aligned}$$

We have $n+1=3 \rightsquigarrow n=2$. Further, writing this in the Lagrange form:

$$y=f(x)=\sum_{i=0}^{2} a_i \cdot \left(\prod_{\substack{j=0\\j\neq i}}^{2} (x-x_j)\right)$$

We then have:

$$\begin{aligned}a_0=\frac{y_0}{(x_0-x_1)\,(x_0-x_2)} &= \frac{0}{\ldots}=0\\a_1=\frac{y_1}{(x_1-x_0)\,(x_1-x_2)} &= \frac{1}{(1-0)(1-2)}=-1\\a_2=\frac{y_2}{(x_2-x_0)\,(x_2-x_1)} &= \frac{3}{(2-0)\,(2-1)}=\frac{3}{2}\end{aligned}$$

Thus, we have the Lagrange interpolation polynomial is

$$\begin{aligned}y=f(x) &= -1(x-0)\,(x-2)+\frac{3}{2}(x-0)\,(x-1)\\&= \frac{x(x+1)}{2}\end{aligned}$$

Using the Vandermonde method, we have $B\cdot\vec{a}=\vec{y}$ with

$$\begin{pmatrix}1&0&0\\1&1&1\\1&2&4\end{pmatrix}\cdot\begin{pmatrix}a_0\\a_1\\a_2\end{pmatrix}=\begin{pmatrix}0\\1\\3\end{pmatrix}$$

We need to solve this.

7.1.1 Lagrange Interpolation for $n+1$ Data Points (with a polynomial of degree n)

$$y=f(x)=\sum_{i=0}^{n} y_i \cdot \left(\prod_{\substack{j=0\\j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}\right)$$

7.1.2 Defficiency in Lagrange Interpolation

We cannot reuse previous computations when the number of data points changes.

7.2 Newton's Divided Differences - Interpolation Polynomial

Note: $f_i \equiv f(x_i)$.

7.2.1 Linear Interpolation

$$(x_0, f_0), (x_1, f_1)$$

Connect these two points with a straight line $f_1(x)=a_0+a_1(x-x_0)$; where $a_0=f_0, a_1=\frac{f_1-f_0}{x_1-x_0}$.

Thus, we have the requirements:

$$\begin{aligned} f_1(x_0) &= f_0 \\ f_1(x_1) &= f_1 \end{aligned}$$

7.2.2 Quadratic Interpolation

$$(x_0, f_0), (x_1, f_1), (x_2, f_2)$$

Fit the data with a degree-2 polynomial:

$$f_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$

We then have these requirements:

$$\begin{aligned} f_2(x_0) = f_0 &\leadsto a_0 = f_0 \\ f_2(x_1) = f_1 &\leadsto a_1 = \frac{f_1 - f_0}{x_1 - x_0} \\ f_2(x_2) = f_2 &\leadsto a_2 = \frac{1}{x_2 - x_0} \cdot \left\{ \frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0} \right\} \end{aligned}$$

a_0 and a_1 are identical with linear interpolation.

7.2.3 n-th Order Newton Divided Differences Interpolation Polynomial

We will have $n+1$ data points to begin: $(x_0, f_0), ..., (x_n, f_n)$

Where the polynomial is:

$$f_n(x) = a_0 + a_1(x - x_0) + ... + a_n \cdot \left(\prod_{i=0}^{n-1} (x - x_i) \right)$$

The requirements follow:

$$f_n(x_i) = f_i \quad , \quad i = 0, ..., n$$

7.2.4 Recursive Construction of the Coefficients $a_0, ..., a_n$

We shall denote as such:

$$\begin{aligned} a_0 = f_0 &= g(x_0) \\ a_1 = \frac{f_1 - f_0}{x_1 - x_0} &= g(x_1, x_0) \\ \vdots \quad \vdots \; &\; \vdots \\ a_n &= g(x_n, ..., x_0) \end{aligned}$$

These g values are called *forward finite divided differences.*

We have $a_0 = g(x_0) = f_0$ as the O-th order forward divided difference (fdd). Also, let's define $g(x_1) = f_1$. We can continue:

$$\begin{aligned} a_1 = g(x_1, x_0) &= \frac{g(x_1) - g(x_0)}{x_1 - x_0} \leadsto 1^{\text{st}} \text{order fdd} \\ a_2 = g(x_2, x_1, x_0) &= \frac{g(x_2, x_1) - g(x_1, x_0)}{x_2 - x_0} \\ \vdots \quad \vdots \; &\; \vdots \\ a_n = g(x_n, ..., x_0) &= \frac{g(x_n, ..., x_1) - ... - g(x_{n-1}, ..., x_0)}{x_n - x_0} \leadsto n^{\text{th}} \text{order fdd} \end{aligned}$$

<u>Another form:</u> [of the polynomial]

$$f_n(x) = g(x_0) + g(x_1, x_0)(x - x_0) + \ldots + g(x_n, \ldots, x) \cdot \left(\prod_{i=0}^{n-1} (x - x_i) \right)$$

More concise...?

$$f_n(x) = \sum_{j=0}^{n} \left(g(x_j, \ldots, x_0) \cdot \prod_{i=0}^{j-1} (x - x_i) \right)$$

Example 7.4. We want to developer the Newton's divided differences interpolation polynomial.

$$\begin{aligned} (x_0, f_0) &= (1, 2) \\ (x_1, f_1) &= (2, 3) \\ (x_2, f_2) &= (3, 5) \end{aligned}$$

We have a degree-2 polynomial ($n = 2$). We need to compute a_0, a_1, a_2 ($g(x_0), g(x_1, x_0), g(x_2, x_1, x_0)$).

We have $g(x_0) = f_0 = 2$ and $g(x_1) = f_1 = 3$ and $g(x_2) = f_2 = 5$ that are given. Let's do stuff! First order first:

$$\begin{aligned} g(x_1, x_0) = \frac{g(x_1) - g(x_0)}{x_1 - x_0} &= 1 \\ g(x_2, x_1) = \frac{g(x_2) - g(x_1)}{x_2 - x_1} &= 2 \end{aligned}$$

Time for second order:

$$g(x_2, x_1, x_0) = \frac{g(x_2, x_1) - g(x_1, x_0)}{x_2 - x_0} = \frac{1}{2}$$

The polynomial is:

$$\begin{aligned} f_2(x) &= 2 + 1 \cdot (x - 1) + \frac{1}{2} \cdot (x - 1)(x - 2) \\ &= \frac{x^2}{2} - \frac{x}{2} + 2 \end{aligned}$$

Example 7.5. Polynomial approximation. Given $f(x) = e^{2x^2}$. Develop a third order Newton divided differences interpolation polynomial based on the data points:

$$\begin{aligned} x_0 = 0, f_0 = f(x_0) &= 1 \\ x_1 = 2, f_1 = f(x_1) &= e^8 \\ x_2 = 4, f_2 = f(x_2) &= e^{32} \\ x_3 = 5, f_3 = f(x_3) &= e^{50} \end{aligned}$$

O-th order forward divided difference:

$$\begin{aligned} g(x_0) = f_0 &= 1 \\ g(x_1) = f_1 &= e^8 \\ g(x_2) = f_2 &= e^{32} \\ g(x_3) = f_3 &= e^{50} \end{aligned}$$

1st order fdd:

$$\begin{aligned} g(x_1, x_0) &= \ldots \\ g(x_2, x_1) &= \ldots \\ g(x_3, x_2) &= \ldots \end{aligned}$$

2nd order fdd:

$$\begin{aligned} g(x_2, x_1, x_0) &= \ldots \\ g(x_3, x_2, x_1) &= \ldots \end{aligned}$$

3rd order fdd:

$$g(x_3, x_2, x_1, x_0) = \ldots$$

7.2.5 Uniformly Spaced Data

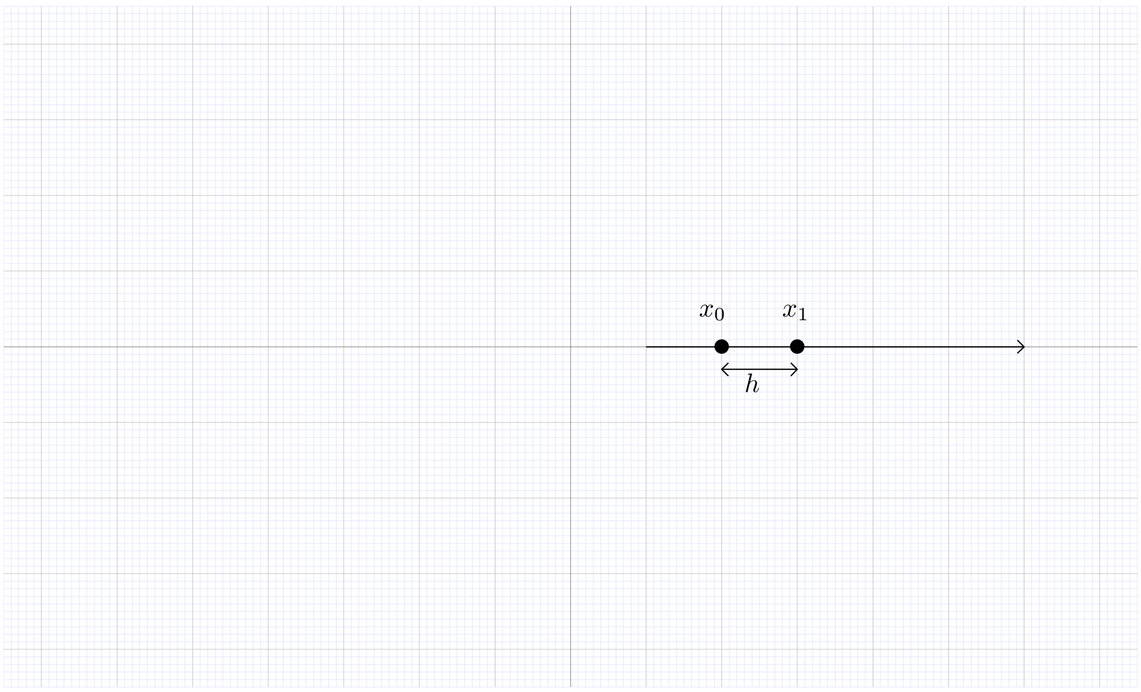
In short, this means that x_i is equally spaced along the x-axis.

$$h = x_{i+1} - x_i$$

Where h is the interval between any two consecutive x_i values. We can denote:

$$x_i = x_0 + i\,h$$

where $i = 0, \ldots, n$.



The forward difference operator is denoted as Δ.

$$\begin{aligned} \Delta\, f_0 &= f_1 - f_0 \\ \Delta\, f_1 &= f_2 - f_1 \\ \vdots\ \ &\vdots\ \ \vdots \\ \Delta\, f_n &= f_{n+1} - f_n \end{aligned}$$

We need to express the vales of the Newton forward differences interpolation polynomial.

$$\begin{aligned}
a_0 = f_0 &= \frac{\Delta^0 f_0}{h^0} \\
a_1 = \frac{f_1 - f_0}{x_1 - x_0} &= \frac{\Delta^1 f_0}{h^1} \\
a_2 = \frac{f_2 - 2 f_1 - f_0}{2 h^2} &= \frac{\Delta^2 f_0}{2!\, h^2} \\
\vdots \quad &\vdots \quad \vdots \\
a_n &= \frac{\Delta^n f_0}{n!\, h^n}
\end{aligned}$$

This known as the *Newton-Gregory forward interpolation polynomial.*

Aside. For reassurance that we know what we are doing, let's calculate $\Delta^2 f_0$.

$$\begin{aligned}
\Delta^2 f_0 &= \Delta(\Delta f_0) = \Delta(f_1 - f_0) \\
&= \Delta f_1 - \Delta f_0 \\
&= (f_2 - f_1) - (f_1 - f_0) \\
&= f_2 - 2 f_1 + f_0
\end{aligned}$$

The polynomial ends up being:

$$f_n(x) = f_0 + \frac{\Delta f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!\, h^2}(x - x_0)(x - x_1) + \ldots + \frac{\Delta^n f_0}{n!\, h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Example 7.6. We have 3 equally spaced data points: $(1, f_0), (3, f_1), (5, f_2)$.

We figure out that $h = 2$.

$$\begin{aligned}
a_0 &= f_0 \\
a_1 &= \frac{\Delta f_0}{h} = \frac{f_1 - f_0}{2} \\
a_2 &= \frac{\Delta^2 f_0}{2!\, h^2} = \frac{f_2 - 2 f_1 + f_0}{8}
\end{aligned}$$

We now know that the Newton-Gregory polynomial will have the form:

$$f_2(x) = f_0 + \frac{f_1 - f_0}{2}(x - 1) + \frac{f_2 - 2 f_1 + f_0}{8}(x - 1)(x - 3)$$

To verify, let's check: $f_2(3) = f_1$ (this is done by algebraically, by showing that substituting 3 returns an answer of 3).

7.3 Chebyshev Polynomials

Definition 7.7.

$$\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
\vdots \quad &\vdots \quad \vdots \\
P_{i+1}(x) &= 2x\, P_i(x) - P_{i-1}(x)
\end{aligned}$$

where $i = 1, 2, 3, \ldots.$

Some examples:

$$\begin{aligned} P_2(x) &= 2x^2-1 \\ P_3(x) &= 4x^3-3x \end{aligned}$$

We have **properties**:

1. $\deg(P_n(x))=n$
2. The n roots of $P_n(x)$ belong to $[-1,+1]$ and they are equal to:

   $$x_j=\cos\left[\left(\frac{n-j+\frac{1}{2}}{n}\right)\pi\right]$$

   where $j=1,...,n$.

Example 7.8. We have $n=3$.

$P_3(x)$ has 3 roots:

$$-\frac{\sqrt{3}}{2},0,+\frac{\sqrt{3}}{2}$$

7.3.1 Express Powers of x as Linear Combinations of Chebyshev Polynomials

$$\begin{aligned} 1 &= P_0(x) \\ x &= P_1(x) \\ x^2 &= \frac{1}{2}[P_0(x)+P_2(x)] \\ x^3 &= \frac{1}{4}[3P_1(x)+P_3(x)] \\ \vdots & \vdots \vdots \\ x^n &= \frac{1}{2^{n-1}}[n\,P_{n-2}(x)+P_n(x)] \end{aligned}$$

To interpolate a function, $f(x)$ by a polynomial:

1. Expand $f(x)$ as a Taylor polynomial, up to degree n
2. Replace $1,x,x^2,...,x^n$ by linear combinations of Chebyshev polynomials
3. Collect the result in terms of $P_i(x)$

7.3.2 Use Orthogonality Property

$$\begin{aligned} \int_{-1}^{1}\frac{P_n(x)\cdot P_m(x)}{\sqrt{1-x^2}}\,\mathrm{dx} &= \begin{cases} 0 & , \; n\neq m \\ \pi & , \; n=m=0 \\ \frac{\pi}{2} & , \; (n=m)\neq 0 \end{cases} \\ a_i &= 2\pi\cdot\int_{-1}^{1}\frac{f(x)\cdot P_i(x)}{\sqrt{1-x^2}}\,\mathrm{dx} \end{aligned}$$

where $i=0,1,2,....$

We can then define:

$$f(x) = \frac{a_0}{2}+\sum_{i=1}^{\infty}a_i\cdot P_i(x)$$

7.3.3 Use the Roots of the Chebyshev Polynomials as the Abscissas of the Data Points in the Lagrange Interpolation Formula $(x_i, f(x_i))$ where $i = 1, ..., n$

8 Regression

8.1 Least Squares Regression (Linear)

$$(x_1, y_1), ..., (x_n, y_n)$$

We can obtain a best fit using a straight line: $y = a_0 + a_1 x$. The error between the model and the data:

$$e_i = y_i - a_0 - a_1 x$$

Next, consider the sum of squares of errors:

$$\boldsymbol{S} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 + a_1 x_i)^2$$

This *best fit* will minimize $\boldsymbol{S}$ with respect to a_0, a_1. We will take derivatives:

$$\frac{\delta \boldsymbol{S}}{\delta a_0} = 0 \ , \ \frac{\delta \boldsymbol{S}}{\delta a_1} = 0$$

We will obtain a linear system in a_0, a_1:

$$\begin{aligned} n a_0 + \left(\sum_{i=1}^{n} x_i\right) \cdot a_1 &= \sum_{i=1}^{n} y_i \\ \left(\sum_{i=1}^{n} x_i\right) \cdot a_0 + \left(\sum_{i=1}^{n} x_i^2\right) \cdot a_1 &= \sum_{i=1}^{n} x_i y_i \end{aligned}$$

where we will obtain values for a_0 and a_1.

8.1.1 Accuracy of Linear Regression

Before regression:

We have the mean and accuracy:

$$\begin{aligned} \bar{y} &= \frac{1}{n} \cdot \sum_{i=1}^{n} y_i \\ S_0 &= \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{aligned}$$

After regression:

Then:

$$\begin{aligned} S &= \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2 \\ r &= \sqrt{\left|\frac{S_0 - S}{S_0}\right|} \end{aligned}$$

Linear [least squares] regression is a good fit when r is close to to 1.

Special case: When all points are collinear, $S = 0 \rightsquigarrow r = 1$.

Example 8.1. Use linear regression to fit the 5 data points.

i	1	2	3	4	5
x_i	1	2	3	4	5
y_i	0.7	2.2	2.8	4.4	4.9

We need to find a_0 and a_1. Let's setup a 2×2 linear system.

The 4 coefficients of this 2×2 linear system are:

$$\begin{aligned} \sum_{i=1}^{5} x_i &= 15 \\ \sum_{i=1}^{5} y_i &= 15 \\ \sum_{i=1}^{n} x_i\, y_i &= 55.6 \\ \sum_{i=1}^{n} x_i^2 &= 55 \end{aligned}$$

Solve a 2×2 linear system $\rightsquigarrow a_0 = -0.18, a_1 = 1.06$. Thus, we have the line $y = -0.18 + 1.06\, x$.

The mean: $\bar{y} = 3$, $S_0 = 11.54$, $S = 0.304$.

$$\begin{aligned} r &= \sqrt{\left|\frac{S_0 - S}{S_0}\right|} \\ &= 0.9867 \end{aligned}$$

Thus, r is close to 1, $\therefore$ a good fit.

8.2 Polynomial Regression

Given points:

$$(x_1, y_1), \ldots, (x_n, y_n)$$

The equation goes as:

$$\begin{aligned} y &= a_0 + a_1\, x + \ldots + a_m\, x^m \\ e_i &= y_i - a_0 - a_1\, x - \ldots - a_m\, x^m \\ \boldsymbol{S} &= \sum_{i=1}^{n} e_i^2 \end{aligned}$$

and minimize $\boldsymbol{S}$ w.r.t $a_0, \ldots, a_m$. Computing

$$\frac{\delta \boldsymbol{S}}{\delta a_0} = \frac{\delta \boldsymbol{S}}{\delta a_1} = \ldots = \frac{\delta \boldsymbol{S}}{\delta a_m} = 0$$

gives us a $(m+1)\times(m+1)$ systems of linear equations in $a_0, \ldots, a_m$. The composite function:

$$([f(a_0)]^n)' \;=\; n\cdot f'(a_0)\, f(a_0)^{n-1}$$

The coefficients that exist:

$$\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} x_i^2, \;\ldots\; , \sum_{i=1}^{n} x_i^m$$
$$\sum_{i=1}^{n} x_i\, y_i \;\ldots$$

8.3 Non-Linear Regression

1. Trigonometry

$$y = a_1 \sin(\omega\cdot x) + a_2 \cos(\omega \cdot x)$$

where ω is the frequency (known) and a_1, a_2 are unknown values.

$$\begin{aligned}\boldsymbol{S} &= \sum_{i=1}^{n} (y_i - a_1 \sin(\omega\cdot x_i) - a_2 \sin(\omega\cdot x_i))^2 \\ \frac{\delta\,\boldsymbol{S}}{\delta\, a_1} &= \frac{\delta\,\boldsymbol{S}}{\delta\, a_2} = 0\end{aligned}$$

2. Exponential

$$y \;=\; a\cdot e^{bx}$$

We have 2 variables a and b.

$$\begin{aligned}\boldsymbol{S} &= \sum_{i=1}^{n} (y_i - a\cdot e^{bx})^2 \\ \frac{\delta\,\boldsymbol{S}}{\delta\, a} &= \frac{\delta\,\boldsymbol{S}}{\delta\, b} = 0\end{aligned}$$

The composite function here is:

$$(e^{f(x)})' \;=\; f'(x)\cdot e^{f(x)}$$

8.4 Multi-Dimensional Regression

We have:

$$(x_{1,i}, x_{2,i}, \ldots, x_{m,i}, y_i)$$

for $i = 1, 2, \ldots, n$. We could set up the equation to look like

$$y \;=\; a_0 + a_1\, x_1 + \ldots + a_m\, x_m$$

Determine $a_0, \ldots, a_m$ using the data points.

$$\begin{aligned}\boldsymbol{S} &= \sum_{i=1}^{n} (y_i - a_0 - a_1\, x_{1,i} - a_2\, x_{2,i} - \ldots)^2 \\ \frac{\delta\,\boldsymbol{S}}{\delta\, a_0} = \;\ldots\; &= \frac{\delta_1\,\boldsymbol{S}}{\delta\, a_m} = 0\end{aligned}$$

We could expand this model with degree-2 monomials.

9 Numerical Differentiation

Approximate derivatives by discrete approximations. We know the definition as:

$$\begin{aligned} f'(x_0) &= \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} \\ &\approx \frac{f(x) - f(x_0)}{\Delta x} \end{aligned}$$

We have equally spaced points: $x_0, x_1, ..., x_{i-1}, x_i, x_{i+1}, ...$ with

$$\begin{aligned} x_{i+1} - x_i &= \Delta x \\ &= h \end{aligned}$$

<u>Approximations for $f'(x_i)$:</u>

- 2-point forward approximation:

$$f'(x_i) = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}$$

- 2-point backward approximation:

$$f'(x_i) = \frac{f_i - f_{i-1}}{x_i - x_{i-1}}$$

<u>Approximations for $f''(x_i)$:</u>

- 3-point forward approximation

  Eliminate the 1st derivative from Taylor's series expansion:

$$f(x) = f(x_i) + (x - x_i) f'(x_i) + \frac{(\Delta x)^2}{2!} f_i'' + ...$$

  We will substitute: $x = x_i + \Delta x$

$$f_{i+1} = f_i + \underline{\Delta x\, f_i'} + \frac{(\Delta x)^2}{2!} f''(x_i) + ... \tag{9.1}$$

  We will substitute: $x = x_i + 2\Delta x$

$$f_{i+2} = f_i + \underline{2\Delta x\, f_i'} + \frac{(2\Delta x)^2}{2!} f_i'' + ... \tag{9.2}$$

  And then $(9.2) \to (9.2) \cdot (9.1)$:

$$f_i'' = \frac{f_{i+2} - 2f_{i+1} + f_i}{(\Delta x)^2}$$

9.1 Calculus of Finite Differences

1. Equally spaced points: $x_{i-1}, x_i, x_{i+1}, ...$

2. Works for both experimental data and function values: $f_{i-1}, f_i, f_{i+1}, \dots$

3. 2 operators Δ, ∇ (forward, backward)

$$\begin{aligned}\Delta f_i &= f_{i+1} - f_i\\ \nabla f_i &= f_i - f_{i-1}\end{aligned}$$

Operators can be composed:

$$\begin{aligned}&\Delta^k \nabla^k\\ \Delta^2 f_i &= \Delta(\Delta f_i)\\ &= \Delta(f_{i+1} - f_i)\\ &= \Delta f_{i+1} - \Delta f_i\\ &= (f_{i+2} - f_{i+1}) - (f_{i+1} - f_i)\\ &= f_{i+2} - 2 f_{i+1} + f_i\\ &\left\{\begin{array}{l}\Delta\nabla f_i\\ \nabla\Delta f_i \text{ are identical}\end{array}\right.\end{aligned}$$

9.1.1 Discrete Approximations to Derivatives Using Calculus of FDs

The 1st derivative:

$$\frac{\delta f(x)}{\delta x} f(x)|_{x=x_i} \rightsquigarrow \frac{\Delta f(x_i)}{\Delta x} = \frac{f_{i+1} - f_i}{\Delta x}$$

The 2nd derivative:

$$\begin{aligned}\frac{\delta^2 f(x)}{\delta x^2} f(x)|_{x=x_i} &\rightsquigarrow \frac{\Delta}{\Delta x}\left(\frac{\Delta}{\Delta x}\right)|_{x=x_i}\\ &= \frac{\Delta^2 f_i}{(\Delta x_i)^2} = \frac{f_{i+2} - 2 f_i + f_i}{(\Delta x)^2}\end{aligned}$$

Example 9.1. $f(x) = x^2 + x + 1$. We have 6 spaced data points: 0, ..., 5. We need to compute discrete approximations to the derivatives of the function f at the data points.

Develop a computational scheme to perform this computation:

		forward differences		
x	$f(x)$	**1st derivative approx.**	**2nd derivative approx.**	**3rd derivative approx.**
0	$f_0 = 1$	$\Delta f_0 = f_1 - f_0 = 2$	$\Delta^2 f_0 = \Delta f_1 - \Delta f_0 = 2$	0
1	$f_1 = 3$	$\Delta f_1 = f_2 - f_1 = 4$	$\Delta^2 f_1 = \Delta f_2 - \Delta f_1 = 2$	0
2	$f_2 = 7$	$\Delta f_2 = f_3 - f_2 = 6$	$\Delta^2 f_2 = \Delta f_3 - \Delta f_2 = 2$	0
3	$f_3 = 13$	$\Delta f_3 = f_4 - f_3 = 8$	$\Delta^2 f_3 = \Delta f_4 - \Delta f_3 = 2$	–
4	$f_4 = 21$	$\Delta f_4 = f_5 - f_4 = 10$	–	–
5	$f_5 = 31$	–	–	–

		backward differences		
x	$f(x)$	**1st derivative approx.**	**2nd derivative approx.**	**3rd derivative approx.**
0	$f_0 = 1$	–	–	–
1	$f_1 = 3$	$\nabla f_1 = f_1 - f_0 = 2$	–	–
2	$f_2 = 7$	$\nabla f_2 = f_2 - f_1 = 4$	$\nabla^2 f_2 = \nabla f_2 - \nabla f_1 = 2$	–
3	$f_3 = 13$	$\nabla f_3 = f_3 - f_2 = 6$	$\nabla^2 f_3 = \nabla f_3 - \nabla f_2 = 2$	0
4	$f_4 = 21$	$\nabla f_4 = f_4 - f_3 = 8$	$\nabla^2 f_4 = \nabla f_4 - \nabla f_3 = 2$	0
5	$f_5 = 31$	$\nabla f_5 = f_5 - f_4 = 10$	$\nabla^2 f_5 = \nabla f_5 - \nabla f_4 = 2$	0

9.1.2 Discrete Approximations of Partial Derivatives

If we were to do it in 2D:

$$\begin{aligned} \frac{\delta f(x,y)}{\delta x} &, \quad \frac{\delta f(x,y)}{\delta y} \\ \frac{\delta f}{\delta x}|_{(x_i,y_i)} &\approx \frac{f(x_i+\Delta x, y_j) - f(x_i - y_j)}{\Delta x} \\ \frac{\delta f}{\delta y}|_{(x_i,y_j)} &\approx \frac{f(x_i, y_j) - f(x_i - \Delta x, y_j)}{\Delta x} \end{aligned}$$

It gets it's own thing:

$$\frac{\delta^2 f}{\delta x\, \delta y} = \frac{f(x_i+\Delta x, y_j+\Delta y) - f(x_i+\Delta x, y_j-\Delta y) - f(x_i-\Delta x, y_j+\Delta y) + f(x_i-\Delta x, y_j-\Delta y)}{4\Delta x\, \Delta y}$$

10 Numerical Integration

$$I = \int_a^b f(x)\, \mathrm{dx}$$

where $f(x)$ is continuous on $[a,b]$.

Note: We can also use *quadrature*, which is talked about later.

To find I, we use:

1. f can be given by a table of values (x_i, f_i) where $i=0,...,n$.

2. I may not have an analytical solution:

$$I = \int_{-\pi}^{\pi} \cos(x^2)\, \mathrm{dx}$$

3. I may have an analytical solution, but so complicated that it really becomes useless for computational purposes, but so complicated it is rendered useless.

$$\begin{aligned} I &= \int \frac{x^3-1}{x^5-1}\, \mathrm{dx} \\ &= \frac{2}{a}\arctan\left(\frac{4x+1-\sqrt{5}}{a}\right) + \frac{2}{b}\arctan(...) + ... \end{aligned}$$

4. **Symmetry**

10.1 Newton-Cotes Formulae

$$\begin{aligned} I &= \int_a^b f(x)\, \mathrm{dx} \\ &\approx \int_a^b p_m(x)\, \mathrm{dx} \end{aligned}$$

where:

$$p_m(x) = a_m x^m + ... + a_1 x + a_0$$

And the coefficients are to be determined.

Notation 10.1. *$[a,b]$ is the* ***subdivision*** *in n of width $h=\frac{b-a}{n}$, where $x_i=a+i\,h$, where $i=0,\ldots,$ n.*

If we are looking for x_n:

$$x_n = a=b$$

Approximate $f(x)$ using:

- $m=0$, a <u>constant</u>
- $m=1$, trapezoidal rule
- $m=2$, Simpson's rule
- ...

10.1.1 $m=0$: Rectangular Rule

$f(x)$ is approximated over $[x_i,x_{i+1}]$ by f_i or f_{i+1}: [where the first one underestimates I, and the second overestimates I]

$$\begin{aligned} I &\approx h\cdot\left(\sum_{i=0}^{n-1} f_i\right) \\ I &\approx h\cdot\left(\sum_{i=0}^{n-1} f_{i+1}\right) \\ &= h\cdot\left(\sum_{i=1}^{n} f_i\right) \end{aligned}$$

10.1.2 $m=1$: Trapezoidal Rule

$$\begin{aligned} I &\approx I_1+\ldots+I_n \\ &= h\cdot\left(\frac{f_0+f_1}{2}\right)+h\cdot\left(\frac{f_1+f_2}{2}\right)+\ldots+h\cdot\left(\frac{f_{n-1}+f_n}{2}\right) \\ &= \frac{h}{2}\cdot(f_0+2f_1+\ldots+2f_{n-1}+f_n) \end{aligned}$$

Example 10.2. Approximate I using the trapezoidal rule.

$$I = \int_0^1 (1-x^2)\,\mathrm{dx}$$

Choose $n=3$. $h=\frac{b-a}{n}=\frac{1-0}{3}=\frac{1}{3}$.

$$\begin{aligned} I &\approx \frac{h}{2}(f_0+2f_1+2f_2+f_3) \\ &= \frac{1}{6}\left(1+2\cdot\frac{8}{9}+2\cdot\frac{5}{9}+0\right) \\ &= \frac{35}{54} \end{aligned}$$

Example 10.3. Approximate I using the trapezoidal rule.

$$I = \int_{-\infty}^{\infty} e^{-x^2}\,\mathrm{dx}$$

First, restrict I to $[-3, 3]$ because $f(x)$ takes almost 0 values outside it. We will get that $I \approx 1.7724 = \sqrt{\pi}$.

10.1.3 Error Analysis for Trapezoidal Rule

Truncation error:

- $n = 1$

$$\begin{aligned} E &= I - \frac{h}{2}(f(a) + f(b)) \\ E &\approx -\frac{1}{12} h^3 f''(\bar{x}) \end{aligned}$$

  where $\bar{x} = \frac{x_{i-1} + x_i}{h}$.

- $n > 1$ (multi-step method)

$$\begin{aligned} E &\approx -\frac{1}{12} h^3 \cdot \left(\sum_{i=1}^{n} f''(\bar{x_i}) \right) \\ E &\approx -\frac{1}{12} (b-a) \cdot h^2\, \overline{f''}(\ldots) \end{aligned}$$

  The error is $O(h^2)$.

10.2 $m = 2$: Simpson's $\frac{1}{3}$ Rule

$$\begin{aligned} I &= \int_a^b f(x)\,\mathrm{dx} \\ &\approx \int_a^b p_2(x)\,\mathrm{dx} \\ &= \frac{h}{3}(f_{i-1} + 4 f_i + f_{i+1}) \end{aligned}$$

where:

$$p_2(x) = c_2 x^2 + c_1 x + c_0$$

and n must be even. The truncation error is $O(h^4)$.

Multi-step:

$$I \approx \frac{h}{3}\left(f_0 + 4 \sum_{i=1,3,\ldots} f_i + 2 \sum_{i=2,4,\ldots} f_i + f_n \right)$$

10.3 $n = 3$: Simpson's $\frac{3}{8}$ Rule

We have this information:

- Use $p_3(x)$

- Truncation error: $O(h^4)$
- Multi-step version:

$$I \approx \frac{3}{8} h \cdot \left(f_0 + 3 \sum_{i=1,4,7,\ldots}^{n-2} (f_i + f_{i+1}) + \sum_{i=3,6,9,\ldots}^{n-3} f_i + f_n \right)$$

where $3, 6, 9$ is $0 \pmod 3$ and $1, 4, 7$ is $1 \pmod 3$.

Newton-Cotes:

m	name	trunc-error
0	rectangle	$O(h)$
1	trapezoidal	$O(h^2)$
2	$\frac{1}{3}$ Simpson	$O(h^4)$
3	$\frac{3}{8}$ Simpson	$O(h^4)$

10.4 Gauss Quadrature

Overall idea:

Approximate:

$$\int_{-1}^{1} f(x)\, \mathrm{dx} = \sum_{i=1}^{n} w_i \cdot f(x_i) \tag{10.1}$$

where n is the number of Gauss points $(x_1, ..., x_n)$ and w_i is the weight of x_i, where $i = 1, ..., n$. This is called *n-point Gauss quadrature*.

10.4.1 Coordinate Transformation

$$\int_a^b f(y)\, \mathrm{dy}$$

where $y = \frac{x\,(b-a) + (a+b)}{2}$, and the differential $\mathrm{dy} = \left(\frac{b-a}{2}\right) \mathrm{dx}$. These results follow:

$$x = -1 \rightsquigarrow y = a$$
$$x = 1 \rightsquigarrow y = b$$

Thus:

$$\int_a^b f(y)\, \mathrm{dy} = \int_{-1}^{1} f\left(\frac{x\,(b-a) + (a+b)}{2}\right) \cdot \left(\frac{b-a}{2}\right) \mathrm{dx}$$

To choose w_i and x_i, we require that (10.1) is exact for all polynomials, up to, and including, degree $2n - 1$.

Example 10.4. $n = 2$, choose w_1, w_2 and x_1, x_2, so that (10.1) is exact for polynomials, up to degree $2n - 1$, which is 3.

10.4.2 $2-$Point Gauss Quadrature

$$\int_{-1}^{1} f(x)\, \mathrm{dx} = w_1\, f(x_1) + w_2\, f(x_2)$$

We do up to an including degree 3.

$$\begin{aligned}
f(x) &= 1, x, x^2, x^3 \\
f(x)=1 &\rightsquigarrow w_1+w_2=2 \\
f(x)=x &\rightsquigarrow w_1 x_1 + w_2 x_2 = 0 \\
f(x)=x^2 &\rightsquigarrow w_1 x_1^2 + w_2 x_2^2 = \frac{2}{3} \\
f(x)=x^3 &\rightsquigarrow w_1 x_1^3 + w_2 x_2^3 = 0
\end{aligned}$$

Using the above calculations, we can work out that:

$$\begin{aligned}
w_1 &= w_2 = 1 \\
x_1 &= \frac{1}{\sqrt{3}} \\
x_2 &= -\frac{1}{\sqrt{3}}
\end{aligned}$$

Now, we have:

$$\int_{-1}^{1} f(x)\,\mathrm{dx} = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

10.4.3 $n-$Point Gauss Quadrature

n Gauss points x_1, ..., x_n are the roots of the *Legendre* polynomials, $P_n(x)$. And we know that Legendre polynomials are orthogonal in $[-1,1]$.

$$\int_{-1}^{1} P_n(x)\cdot P_m(x)\,\mathrm{dx} = \frac{2}{2n+1}\,\delta \mathrm{nm}$$

The *Kronecker-delta* notation works as follows: $\begin{cases} 1 & \text{if } n=m \\ 0 & \text{otherwise} \end{cases}$

where the polynomials are:

$$\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
P_2(x) &= \frac{1}{2}(3x^2-1) \\
\vdots \ & \ \vdots \ \ \vdots \\
P_n(x) &= \left(\frac{2n-1}{n}\right) x\, P_{n-1}(x) - \left(\frac{n-1}{n}\right) P_{n-2}(x)
\end{aligned}$$

The weights are <u>also</u> determined via Legendre polynomials:

$$w_i = \frac{2(1-x_i^2)}{[n\cdot P_{n-1}(x_i)]^2}$$

where $i=1,...,n$.

We get this nice **property**: $w_1+...+w_n=2$.

Example 10.5. Find the $3-$point GQ formula.

$$\int_{-1}^{1} f(x)\,\mathrm{dx} = \frac{5}{9} f\left(-\frac{\sqrt{15}}{5}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\frac{\sqrt{15}}{5}\right)$$

We shall do the computation for one of the coefficients: $\frac{8}{9}$. We that the Gauss points are $-\frac{\sqrt{15}}{5}, 0, \frac{\sqrt{15}}{5}$ and $x_2 = 0$.

$$w_2 = \frac{2(1-(0)^2)}{\left[3\cdot\left(-\frac{1}{2}\right)\right]^2} = \frac{8}{9}$$

10.4.4 Error Estimate for $n-$Point GQ

$$E \approx \frac{2^{n+1}(n!)^4}{(2n+1)((2n)!)^3}\cdot f^{(2n)}(\xi)$$

where $-1<\xi<1$. We also have $E=0$ $\forall$ polynomials with degrees up to $2n-1$.

Example 10.6. Evaluate:

$$I = \int_0^2 y\,e^{2y}\,\mathrm{dy}$$

Since this integral is **not** over $[-1, 1]$, we need to use the coordinate transformation, this $a=0$, $b=2$; our bounds of integration and $y=\frac{2x+2}{2}=x+1$. And also, $\mathrm{dy}=\mathrm{dx}$. Therefore, we have that:

$$I = \int_{-1}^{1}(x+1)\,e^{2(x+1)}\,\mathrm{dx}$$

Using $2-$point GQ, we obtain:

$$\begin{aligned} I &\approx f\left(-\frac{1}{\sqrt{3}}\right)+f\left(\frac{1}{\sqrt{3}}\right) \\ &\approx 37.9667 \\ \text{But with Maple we get....} & \\ &= \frac{1+3e^4}{4} \\ &\approx 41.1986 \end{aligned}$$

where $f(x)=(x+1)\,e^{2(x+1)}$, as that is what we calculated via coordinate transformation.

If we were to use $6-$point GQ, we could get an error that would be closer to the true value (that was calculated via Maple).

11 Ordinary Differential Equations

We have $y=y(t)$, where t is the independent variable. Then:

$$\begin{aligned} y' &= f(t,y) \\ y'(t) &= f(y, y(t) \end{aligned}$$

y is a function of t; and this is consider a 1st order ODE. We then have IVPs, *initial value problems*:

$$\begin{aligned} y' &= f(t,y) \\ y(a) &= y_a \\ t &\in [a,b] \end{aligned}$$

and these are typically solved numerically.

11.1 Euler's Method

We start with the step size, $h=\frac{b-a}{2}$, which h is the subdivision of $[a,b]$. We will have $n+1$ points in $[a,b]$, or in other words the $\underline{t-\text{axis}}$.

Euler's method is:

$$\begin{aligned} w_0 &= y_a \\ w_{i+1} &= w_i + h\, f(t_i, w_i) \end{aligned}$$

The output of Euler's method are: $w_0, w_1, \ldots, w_n$; where w_i is an approximate value for $y(t_i)$.

Example 11.1. $\begin{cases} y' = t\,y + t^3 \\ y(0) = 1 \\ t \in [0,1] \end{cases} \rightarrow$ solve this IVP, using Euler's method. We have:

$$\begin{aligned} a &= 0 \\ b &= 1 \\ n &= 5 \\ h &= \frac{b-a}{n} = 0.2 \end{aligned}$$

And our intervals are $0, \ldots, 1.0$ by 0.2. We have $e_i = |y_i - w_i|$ which is the error at point t_i.

step	t_i	w_i	y_i	e_i
0	0.0	1.00	1.00	0.00
1	0.2	1.00	1.02	0.02
2	0.4	1.04	1.08	0.04
3	0.6	1.13	1.23	0.10
4	0.8	1.31	1.49	0.18
5	1.0	1.63	1.94	0.31

11.2 Heun's Method

The point of *Heun's method* is to improve on Euler's method. We have:

$$\begin{cases} w_0 = y_0 \\ w_{i+1} = w_i + \frac{h}{2} \cdot [f(t_i, w_i) + f(t_i + h, w_i + h\, f(t_i, w_i))] \\ i = 0, 1, 2, \ldots \end{cases}$$

Example 11.2. $\begin{cases} y' = t\,y + t^3 \\ y(0) = 1 \\ t \in [0,1] \end{cases} \longrightarrow$ Solve this IVP using Heun's method. We have:

$$\begin{aligned} f(t,y) &= t\,y + t^3 \\ n &= 10 \\ h &= \frac{1-0}{10} = 0.1 \end{aligned}$$

Heun's method is: RIP this is the *El Diablo* equation

$$\begin{aligned} w_0 &= 1 \\ w_{i+1} &= w_i + \frac{0.1}{2} \cdot [(t_i\, w_i + t_i^3) + (t_i + 0.1) \cdot [w_i + 0.1 \cdot (t_i\, w_i + t_i^3)] + (t_i + 0.1)^3] \end{aligned}$$

Then we have:

step	t_i	w_i	y_i	e_i
0	0.0	1.00	1.00	0.00
1	0.1	1.00	1.00	0.00
2	0.2	1.02	1.02	0.00
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

11.3 Runge-Kutta

We will look at 1st order ODE: $\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$

The Runge-Kutta general form:

$$y_{i+1} = y_i + h\,(c_1\,k_1 + \ldots + c_n\,k_n)$$

where $n=$ the order of the RK method, $c_1, \ldots, c_n$ are constants and $k_1, \ldots, k_n$ are the recurrence relations.

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + p_2\,h, y_i + a_{21}\cdot h\,k_1) \\ k_3 &= f(x_i + p_3\,h, y_i + a_{31}\cdot h\,k_1 + a_{32}\cdot h\,k_2) \end{aligned}$$

Example 11.3. 3rd order ($n=3$) RK method:

$$y_{i+1} = y_i + \frac{h}{6}\,(k_1 + 4k_2 + k_3)$$

where we have:

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}\,k_1\right) \\ k_3 &= f(x_i + h, y_i - h\,k_1 + 2h\,k_2) \end{aligned}$$

Example 11.4. Solve the IVP: $\begin{cases} y' = y + 2x - 1 \\ y(0) = 1 \\ x \in [0,1] \end{cases} \longrightarrow$ using a 3rd order RK method.

The exact solution is $y(x) = 2e^x - 2x - 1$. To check if these are equal:

$$\begin{aligned} 2e^x - 2 &= 2e^x - 2x - 1 + 2x - 1 \\ &= 2e^x - 2 \\ n &= 100 \\ h &= 0.01 \end{aligned}$$

where $f(x,y) = y + 2x - 1$.

We then have:

step	x_i	k_1	k_2	k_3	$y_i \equiv w_i$	y_i exact
0	0.0	$f(0,1) = 1 - 1 = 0$	$f\left(\frac{0.01}{2}, 1 + \frac{0.01}{2}\cdot 0\right) = \ldots$	...	...	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

11.4 Systems of [1^{st} order] ODE's

$$\begin{aligned}
\frac{d\,y_1}{d\,x} &= f_1(x, y_1, \ldots, y_n)\\
\vdots\ &\ \vdots\ \ \vdots\\
\frac{d\,y_n}{d\,x} &= f_n(x, y_1, \ldots, y_n)
\end{aligned}$$

where $n = n$ unknwn functions. This comes with the associated IVP:

$$\begin{cases}
y_1(x_0) = y_1^0\\
\vdots\\
y_n(x_0) = y_n^0
\end{cases}$$

Euler's, Heun's and RK methods all apply to this IVP. And thus comes:

$$\begin{cases}
w_1^0 = y_1^0\\
\vdots\\
w_n^0 = y_n^0\\
\\
w_{1,i+1} = w_{1,i} + h\,f_1(x_i, w_{1,i}, \ldots, w_{n,i})\\
\vdots\\
w_{n,i+1} = w_{n,i} + h\,f_n(x_i, w_{1,i}, \ldots, w_{n,i})
\end{cases}$$

11.5 Higher Order Equations

Equations of motion of a body in a moving plane in Earth's gravitational field:

$$\begin{aligned}
\frac{d^2\,x}{d\,t^2} &= -G\frac{x}{(x^2+y^2)^{\frac{3}{2}}}\\
\frac{d^2\,y}{d\,t^2} &= -G\frac{y}{(x^2+y^2)^{\frac{3}{2}}}
\end{aligned}$$

Convert this into a system of order 1. We have the set:

$$\begin{aligned}
x_1 &= x\\
x_2 &= \frac{d\,x}{d\,t}\\
x_3 &= y\\
x_4 &= \frac{d\,y}{d\,t}
\end{aligned}$$

And the original system becomes the following 1^{st} order system:

$$\begin{aligned}
\frac{d\,x_1(t)}{d\,t} &= x_2(t)\\
\frac{d\,x_2(t)}{d} &= -G\frac{x_1(t)}{(x_1(t)^2+x_3(t)^2)^{\frac{3}{2}}}\\
\frac{d\,x_3(t)}{d\,t} &= x_4(t)\\
\frac{d\,x_4(t)}{d\,t} &= -G\frac{x_3(t)}{(x_1(t)^2+x_3(t)^2)^{\frac{3}{2}}}
\end{aligned}$$

11.6 Boundary Value Problems (BVP) for ODEs

We have linear 2nd order BVP:

$$\begin{cases} y''(x)+p(x)\,y'(x)+q(x)\,y(x)=r(x) \\ x\in[a,b], \qquad\qquad\qquad y(a)=\alpha \;\; y(b)=\beta \end{cases}$$

where $p(x), q(x), r(x)$ are some functions. We can derive finite-difference approximations to $y''(x)$ and $y'(x)$ using <u>central</u> differences. We can subdivide $[a,b]$ into N subintervals, $h=\frac{b-a}{N}$.

$$\begin{aligned} x_0 &= a \\ x_1 &= a+h \\ x_2 &= a+2h \\ \vdots &\;\vdots\; \vdots \\ x_n &= a+Nh=b \end{aligned}$$

thus, there are $N+1$ points on the $x-$axis.

$$\begin{aligned} y'(x_i) &= \frac{y_{i+1}-y_i}{2h} \\ y''(x_i) &= \frac{y_{i+1}-2y_i+y_{i-1}}{h^2} \\ \text{with} & \end{aligned}$$

$$\begin{cases} p(x_i)=p_i \\ q(x_i)=q_i \\ r(x_i)=r_i \end{cases} \qquad \text{with } i=0,\ldots,N$$

For every interior point x_i, where $i=1,\ldots,N-1$, the finite-difference approximation to the BVP is:

$$\left(1+\frac{h}{2}\,p_i\right)y_{i+1}+(h^2\,q_i-2)\,y_i+\left(1-\frac{h}{2}\,p_i\right)y_{i-1} \;=\; h^2\,r_i$$

Where the parts of this equation are P_i, Q_i, R_i respectively in regards to their lowercase.

We have boundary conditions such that:

$$\begin{aligned} y(a)=\alpha &\rightsquigarrow y_0=\alpha \\ y(b)=\beta &\rightsquigarrow y_N=\beta \end{aligned}$$

But *whyyyyy* is this matrix **tridiagonal**. We have:

$$\vec{y}=\begin{pmatrix} y_1 \\ \vdots \\ y_{N-1} \end{pmatrix}, \quad \vec{b}=\begin{pmatrix} h^2\,r_1-\left(1-\frac{h}{2}\,p_i\right)\alpha \\ h^2\,r_2-\ldots. \\ \vdots \\ h^2\,r_{N-1}-\left(1+\frac{h}{2}\,p_{N-1}\right)\beta \end{pmatrix}$$

$$A=\begin{pmatrix} Q_1 & P_1 & 0 & 0 \\ R_2 & Q_2 & P_2 & 0 \\ 0 & R_3 & Q_3 & P_3 \\ & \ddots & \ddots & P_{N-2} \\ 0 & 0 & R_{N-1} & Q_{N-1} \end{pmatrix}$$

In order to solve this tridiagonal system of linear equations, use **LU** decomposition.

Example 11.5. Solve the BVP: $\begin{cases} y''-y'+y=0 \\ y(0)=0 \\ y(1)=1 \\ x\in[0,1] \end{cases}$. The exact solution is: $y(x)=c\cdot e^{\frac{x}{2}}\cdot\sin\left(\frac{\sqrt{3}}{2}x\right)$.

Changing our boundary conditions $y(1)=1$ to $y(1)=0$ has the effect that $y(x)=0$.

$$\begin{aligned} N &= 10 \\ h &= \frac{1-0}{10}=0.1 \end{aligned}$$

Solve a 9×9 tridiag. system:

$$\begin{aligned} p(x) &= -1 \\ q(x) &= 1 \\ r(x) &= 0 \end{aligned}$$

The finite-differences approximation for the given BVP is:

$$\left(1-\frac{0.1}{2}\right)y_{i+1}+(0.1^2-2)\,y_i+\left(1+\frac{0.1}{2}\right)y_{i-1} = 0$$

for $i=1,...,9$. Solving this system gives us $y_1,...,y_9$ with $\begin{cases} y_0=0 \\ y_{10}=1 \end{cases}$ from the BVP.

11.7 Non-Linear 2$^{\text{nd}}$ Order BVP

$$\begin{cases} y''(x) &= f(x,y,y') \\ y(a) &= \alpha \\ y(b) &= \beta \end{cases}$$

Subdivision of $[a,b]+$ central difference approximation to $y'(x), y''(x)$.

The finite-differences approximation to this BVP is:

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} = f\left(x_i, y_i, \frac{y_{i+1}-y_{i-1}}{2h}\right)$$

where $i=1,...,N-1$, $y_0=\alpha$, $y_N=\beta$. We have a system on $N-1$ non-linear equations in $N-1$ unknowns: $y_1,...,y_{N-1}$. Can be solved with multivariate Newton's method.

11.8 Collocation Method for Solving BVPs

We have:

$$\begin{cases} y''=f(t,y,y') \\ y(a)=\alpha \\ y(b)=\beta \end{cases}$$

Choose a set of basis functions, $\varphi_1(t),...,\varphi_n(t)$; form a candidate solution of the BVP:

$$y(t) = c_1\,\varphi_1(t)+...+c_n\,\varphi_n(t)$$

and of course, determine the values of $c_1,...,c_n$. Substitute $y(t)$ to the BVP at $N-1$ interior points.

The basic functions could be: polynomials, trig, splines.

Example 11.6. $\varphi_i(t) = t^{i-1}$.

$$y(t) = \sum_{i=1}^{n} c_i \, t^{i-1}$$

for $i = 1, ..., n-1$. Next:

$$\sum_{i=1}^{n} (i-1)\,(i-2)\,c_i\,t^{i-3} = f\left(t, \sum_{i=1}^{n} c_i\,t^{i-1}, \sum_{i=1}^{n} (i-1)\,(i-2)\,t^{i-2}\right)$$

Example 11.7. $\begin{cases} y'' = 4y \\ y(0) = 0 \\ y(1) = 1 \end{cases}$. The exact solution is: $\frac{e^{2+2x} - e^{2-2x}}{e^4 - 1}$

12 Integral Equations

A linear integral equation has the following form:

$$h(x)\,u(x) = f(x) \int_a^{b(x)} K(x,t)\,u(t)\,\mathrm{dt}$$

where $K(x,t)$ is the **kernel function** and $u(x)$ is the unknown function.

12.1 Method of Successive Approximations

1. Make a reasonable 0th approximation to $u(x)$
2. Use the integral equation to find the i^{th} equation for $i = 1, 2, ...$
3. Recgonize $u_n(x)$ as a Taylor series of a known function

Example 12.1.

$$u(x) = x - \int_0^x (x-t)\,u(t)\,\mathrm{dt}$$

1. $u_0(x) = 0$
2.

$$\begin{aligned} u_1(x) &= x - \int_0^x (x-t)\,u_0(t)\,\mathrm{dt} \\ &= x - \int_0^x (x-t)\,0\,\mathrm{dt} \\ &= x \\ u_2(x) &= x - \int_0^x (x-t)\,u_1(t)\,\mathrm{dt} \\ &= x - \int_0^x (x-t)\,t\,\mathrm{dt} \\ &= x - \frac{x^3}{3!} \\ u_3(x) &= x - \int_0^x (x-t)\,u_2(t)\,\mathrm{dt} \\ &= x - \int_0^x (x-t)\left(t - \frac{t^3}{3!}\right)\mathrm{dt} \\ &= x - \frac{x^3}{3!} + \frac{x^5}{5!} \end{aligned}$$

13 Partial Differential Equations (PDE)

We can have different cases of these PDEs. Assume we have something like this, where a, b, c are constants:

$$a\frac{\delta^2 A}{\delta x^2} + b\frac{\delta^2 A}{\delta x \cdot \delta y} + c\frac{\delta A}{\delta x} + d\frac{\delta A}{\delta y} + f \cdot A(x, y) + g \;=\; 0$$

Using the quadratic formula, we know:

$$\begin{aligned} b^2 - 4\text{ac} > 0 &\;\rightsquigarrow\; \text{hyperbolic} \\ b^2 - 4\text{ac} = 0 &\;\rightsquigarrow\; \text{parabolic} \\ b^2 - 4\text{ac} < 0 &\;\rightsquigarrow\; \text{elliptic} \end{aligned}$$

13.0.1 Diffusion

$$\frac{\delta T(t, x)}{\delta t} \;=\; \kappa \frac{\delta^2 T(t, x)}{\delta x^2}$$

This type of equation is *parabolic*.

13.0.2 Wave

$$\frac{\delta^2 A(t, x)}{\delta t^2} \;=\; c^2 \frac{\delta^2 A(t, x)}{\delta x^2}$$

This type of equation is *hyperbolic*.

13.0.3 Poisson

$$\frac{\delta^2 \phi(x, y)}{\delta x^2} + \frac{\delta^2 \phi(x, y)}{\delta y^2} \;=\; -\frac{1}{\epsilon_0} \rho(x, y)$$

This type of equation is *elliptic*.

13.1 "Hey, let's talk about that diffusion equation!"

We denote $n(x, y)$ as the density, we get flux, F:

$$F \;=\; -D \cdot \nabla n(t, x)$$

where *gradient*, $\nabla = \left[\frac{\delta}{\delta x}, \frac{\delta}{\delta y}, \frac{\delta}{\delta z}\right]$. We then get the relation:

$$\frac{\delta n(t, x)}{\delta t} \;=\; \nabla \cdot F$$

13.2 Initial Value Problem (IVP)

Suppose $T(t = 0, x)$ is known, we need to find $T(t, x)$. We have *Dirichlet boundary conditions*:

$$\begin{aligned} T\left(t, x = -\frac{L}{2}\right) &\;=\; T_a \\ T\left(t, x = \frac{L}{2}\right) &\;=\; T_b \end{aligned}$$

We have *Neumann boundary conditions*:

$$\begin{aligned}-\kappa\frac{\mathrm{dT}}{\mathrm{dx}}\,\mathrm{at}\,x=-\frac{L}{2} &= F_a\\ -\kappa\frac{\mathrm{dT}}{\mathrm{dx}}\mathrm{at}\,x=\frac{L}{2} &= F_b\end{aligned}$$

Hey look, *periodic boundary conditions*:

$$\begin{aligned}T\left(t,x=-\frac{L}{2}\right) &= T\left(t,x=\frac{L}{2}\right)\\ \frac{\mathrm{dT}}{\mathrm{dx}}\mathrm{at}\,x=-\frac{L}{2} &= \frac{\mathrm{dT}}{\mathrm{dx}}\mathrm{at}\,x=\frac{L}{2}\end{aligned}$$

13.3 Forward Time Centered Space Scheme

$$\begin{aligned}T(t_n,x_i) &= T_i^n\\ t_n=(n-1)\,\tau &\rightsquigarrow \tau\equiv\Delta t\,\mathrm{timestep}\\ x_i=-\frac{L}{2}+(i-1)\,h &\rightsquigarrow h=\Delta x\,\mathrm{spacestep}\\ \mathrm{where} &\qquad h=\frac{L}{N-1}\end{aligned}$$

Our discretization comes about:

$$\begin{aligned}\frac{\delta T}{\delta t} &\Rightarrow \frac{T(t_n+\tau,x_i)-T(t_n,x_i)}{\tau}=\frac{T_i^{n+1}-T_i^n}{\tau}\\ \frac{\delta^2 T}{\delta x^2} &\Rightarrow \frac{T(t_n,x_i+h)+T(t_n,x_i-h)-2T(t_n,x_i)}{h^2}\\ &\equiv \frac{T_{i+1}^n+T_{i-1}^n-2T_i^n}{h^2}\\ \frac{T_i^{n+1}-T_i^n}{\tau} &= \kappa\,\frac{T_{i+1}^n+T_{i-1}^n-2T_i^n}{h^2}\end{aligned}$$