# 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:

$$P(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4$$

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:*

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

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

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

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

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

**Example 9.** 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 10.** We want to represent 1.

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

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

**Definition 11.** *Machine epsilon,* $\mathrm{E}_{\text{mach}}$*, is the distance between* 1 *and the smallest FLP number greater than 1.*

***Remark* 12.** *For IEEE 754, double precision, we have* $\mathrm{E}_{\text{mach}} = 2^{52}$.

## 2.1 IEEE Nearest Rounding Rule

**Example 13.** 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: $\begin{cases} 1 \rightarrow \text{add 1 to bit 52} \\ 0 \rightarrow \text{do nothing} \end{cases}$

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 14.** The IEEE double precision number associated with 9.4 using RNR is:

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

where $0.2 \times 2^{-49}$ is the error.

**Definition 15.**

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

***Remark* 16.** *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 17.** $E_1 = \frac{1 - \cos(x)}{\sin^2(x)}$ 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 18.** 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 19.** $x^2 + 9^{12} x - 3 = 0$, 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:

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

**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, $\exists r: a<r<b$ st $f(r)=0$

**Example 20.** $f(x)=x^3+x-1$, 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 1**

Bisection Method
**Input**: `f`, `a`, `b` st. $f(a) \times f(b) < 0$; tolerance $(\epsilon)$ - `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.**

| $\epsilon$ | #`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 21.** *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 22.** *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 23.** *$r$ is a fixed point (fp) of a function $g(x)$, iff $g(r) = r$.*

**Example 24.** $g(x) = x^3$. 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 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 $x_0$, $x_1$, $x_2$, ... <u>converges</u> to a value, $r$, then $r$ is a fp of $g(x)$. For some $j$: $|x_{j+1} - x_j| < \text{E}$.

**Question 25.** *Can any fct, $f(x)$ be written as $g(x) - x$?*

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

**Example 26.** $x^2 + x - 1 = 0$

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

$$x = (1 - x)^{\frac{1}{3}} \tag{2}$$

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

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 27.**

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

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 28.** *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 29.** *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 30.** $f(x) = x^3 + x - 1$ 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 $|g_2'(x)| = \frac{1}{3}(1 - x)^{-\frac{2}{3}} + 1 \longrightarrow x = ... \longrightarrow <1$ $\quad\therefore$ converges
3. $g_3(x) = \frac{1 + 2x^3}{1 + 3x^2}$, now $|g_3'(x)| = \frac{(6x^2)(1 + 3x^2) + (6x)(1 + 2x^3)}{(1 + 3x^2)^2} \longrightarrow x = ... \longrightarrow 0 < 1$ 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 31.** $\begin{cases} g(x)=\frac{x+\frac{2}{x}}{2} \\ x_0=1 \end{cases}$. 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 32.** $f(x)=x^3-2x^2+\frac{4}{3}x-\frac{8}{27}$. 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 $6^{\text{th}}$ digit correct.

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

**Definition 33.** *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* 34.** *The problem we encountered with the previous example is that the BE is near* $\mathrm{E}_{\text{mach}}=10^{-16}$ *and the FE* $\approx 10^{-5}$.

**Definition 35.** *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 $f$.*

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

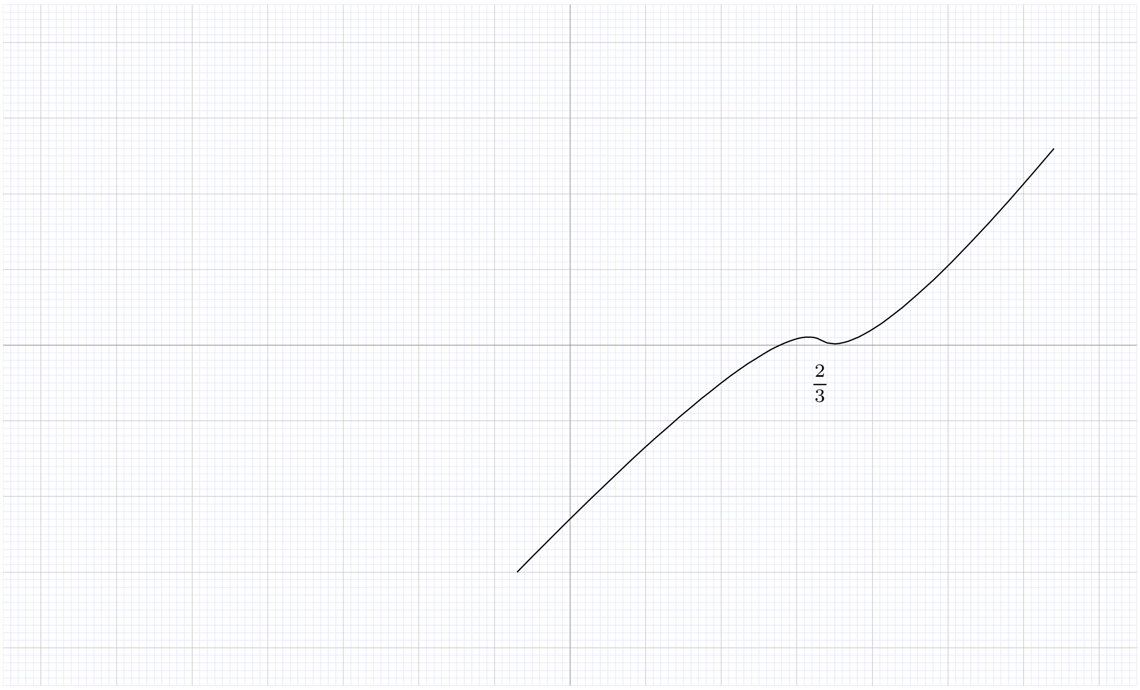
**Example 36.** $f(x)=x^2$, has a double root at $x=0$.

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

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.

$$\mathrm{BE} \ll \mathrm{FE}$$

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

**Example 37.** $f(x) = \sin(x) - x$ has a triple root at $r = 0$ and $x_a = 0.001$ is the approx. root. Compute BE and FE.

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

**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($\frac{b-a}{2}$) < 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, $x = 1, ..., 20$. To compute numerical approximations to the roots of the expanded $W(x)$ is very hard. $W(x) = x^{20} \pm ...$

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

*Let $\Delta_r$ 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 39.** 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**

**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 40.** $f(x)=x^3+x-1$.

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 41.** *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* 42.** *If* $f(r)=0$, *and* $f'(r)=0$, *then Newton's method converges quadratically to* $r$ *and thus,* $M=\frac{f''(r)}{2\,f'(r)}$, *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 43.** $f(x)=x^2$, 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 44.** 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 45.**

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 46.** 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 47.** $f(x) = 4x^4 - 6x^2 - \frac{11}{4} = 0$ 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 $\pm 1.3667$. 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 48.** $f(x) = \sin(2x)$, with $x_0 = 0.75$. Newton's method converges to the root $-2\pi$. 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 49.** $f(x) = x\,e^x$. 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 50.** $f_1(x_1, x_2) = x_1^2 + x_2^2 - 8x_1 - 4x_2 + 11 = 0$ 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 4**

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

```
i = 1
while (|f_j(r⃗)| ≥ eps, j = 1) do
   i = i+1
   solve system of linear eqs
```

$$J_{(f_1 ::: f_n)}(x_1^i;\ :::;\ x_n^i)\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix}=\begin{pmatrix} -f_1\left(\vec{x}^i\right) \\ \vdots \\ -f_n\left(\vec{x}^i\right) \end{pmatrix}$$

$x_j^{(i)} = x_j^{(i-1)} + \Delta x_j$ , j=1..n

Let's talk Jacobian dude.

**Jacobian $n \times n$ Matrix**

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

## 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 5**

**Input:** $f(x)$, $x_0$, $x_1$, $\epsilon$ (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 $i = 1, 2, 3, ...$
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:

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

Then we have the $L_p$ norm:

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

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 $i = 1, ..., n$. $\lambda_i$ 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 <u>homogeneous</u> 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 <u>non-homogeneous</u> system, denoted as *augmented* matrix

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

   - System has a solution $\Longleftrightarrow \operatorname{rank}(A) = \operatorname{rank}(A')$
   - Solution is unique if $\operatorname{rank}(A) = n$; $A$ is considered **full rank**

- If $\text{rank}(A) < n$, then it is denoted as **inconsistent**

**Definition 51.** *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 ($A: m \times n$)

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 52.** $\dim(\text{nullspace}(A)) + \text{rank}(A) = n$

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

**Definition 53.** $A$ *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:

$$A_i = \begin{pmatrix} \vdots & \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} & \vdots \end{pmatrix}$$

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

**Example 54.** Given:

$$\begin{aligned} x_1 - x_2 + x_3 &= 3 \\ x_1 + x_2 - x_3 &= 0 \\ 3x_1 + 2x_2 + 2x_3 &= 5 \end{aligned}$$

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 55.** 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

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

The way this one works, is transforming $A \to I_n$ 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 a lower 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. $l_{ii} = u_{ii}$ with $i = 1, ..., n \longrightarrow$ 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 $l \times \vec{z} = \vec{b}$.