Systems of Non linear Equations

$$f_1(x_1, x_2, \dots, x_n) = 0$$
$$\vdots$$
$$f_n(x_1, x_2, \dots, x_n) = 0$$

$$\underline{f}(\underline{x}) = \underline{0}$$

(1) Fixed Point: If $\underline{f}(\underline{x})$ has a portion linear to $\underline{x}$,

Set $\underline{g}(\underline{x}) = \underline{x} - \underline{f}(\underline{x})$, then

iterate $\underline{x}_{n+1} = \underline{g}(\underline{x}_n)$

(2) Newton - Rhapson Method.

Focus on $f_1(x_1, x_2, \dots, x_n) = 0$

Given $f_1^i, x_1^i, \dots, x_n^i$ $i \in$ iteration, not power

Write

Derivative $\frac{\partial f_1}{\partial x_1}$ evaluated at $x_1^i, x_2^i, \dots$

$$f_1^{i+1} = f_1^i + \frac{\partial f_1}{\partial x_1}^i (x_1^{i+1} - x_1^i) + \frac{\partial f_1}{\partial x_2}^i (x_2^{i+1} - x_2^i)$$
$$+ \frac{\partial f_1}{\partial x_3}^i (x_3^{i+1} - x_3^i) + \dots$$

Since we want the $f_1^{i+1} = 0 \rightarrow$

$$-f_1^i = \frac{\partial f_1^i}{\partial x_1}(x_1^{i+1} - x_1^i) + \frac{\partial f_1^i}{\partial x_2}(x_2^{i+1} - x_2^i) + \dots$$

Do this for every function in $\underline{f}(\underline{x})$:

$$\underbrace{\begin{bmatrix} \frac{\partial f_1^i}{\partial x_1} & \frac{\partial f_1^i}{\partial x_2} & \cdots & \frac{\partial f_1^i}{\partial x_n} \\ \frac{\partial f_2^i}{\partial x_1} & \frac{\partial f_2^i}{\partial x_2} & \cdots & \frac{\partial f_2^i}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n^i}{\partial x_1} & \cdots & \cdots & \frac{\partial f_n^i}{\partial x_n} \end{bmatrix}}_{\text{Jacobian Matrix of } \underline{f}(\underline{x})} \begin{bmatrix} x_1^{i+1} - x_1^i \\ \vdots \\ x_n^{i+1} - x_n^i \end{bmatrix} = \begin{bmatrix} -f_1^i \\ \vdots \\ -f_n^i \end{bmatrix}$$

$$\underline{J}_i \, \underline{\delta}_i = -\underline{f}_i$$

$\underline{J}_i$ = Jacobian of $\underline{f}(\underline{x})$ evaluated using information at iteration i,

$\underline{\delta}_i = \underline{x}_{i+1} - \underline{x}_i$

$\underline{f}_i = \underline{f}(\underline{x}_i) \leftarrow$ "residual"

$\Rightarrow \underline{\delta}_i = -\underline{J}_i^{-1} \underline{f}_i$

$\Rightarrow \underline{x}_{i+1} - \underline{x}_i = -\underline{J}_i^{-1} \underline{f}_i$

$\Rightarrow \boxed{\underline{x}_{i+1} = \underline{x}_i - \underline{J}_i^{-1} \underline{f}_i}$

Note: $\underline{J}_i$ will be a <u>linear</u> matrix

$\Rightarrow$ Every iteration of this method requires the solution of a matrix that varies!

ex.) $f_1(x_1, x_2) = x_1^2 + x_2^2 - 1$

$f_2(x_1, x_2) = x_1^2 - x_2$

$$\underline{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 & 2x_2 \\ 2x_1 & -1 \end{bmatrix}$$

$\Rightarrow$ let $\underline{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, then $\underline{J}_0 = \begin{bmatrix} 2 & 2 \\ 2 & -1 \end{bmatrix}$

Note! If the initial guess $\underline{x}_0$ is
not close to the solution, $\underline{x}^*$,
it might not converge.

Also, sometimes you might need to do
a damped Newton-method.

$\underline{\delta}_i = -J^{-1} \underline{f}_i$, but then

$\underline{x}_{i+1} = \underline{x}_i + \alpha_i \underline{\delta}_i$, $\alpha_i \in (0, 1]$ that

moves $\underline{x}_{i+1}$ closer to $\underline{x}^*$.

---

ex1) $\underline{f}(\underline{x}) = \begin{bmatrix} x_1^2 + x_1 x_2 + x_1 - 1 \\ x_1 x_2 + x_2 + x_3^2 - 0.25 \\ x_1^2 + x_2^2 - 4x_3 \end{bmatrix}$ $\quad \underline{f}(\underline{x}) = \underline{0}$

Fixed point:
$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -x_1^2 - x_1 x_2 + 1 \\ -x_1 x_2 - x_3^2 + 0.25 \\ 0.25(x_1^2 + x_2^2) \end{bmatrix}$$

Define $\varepsilon_i = \|\underline{f}(\underline{x}_i)\|_\infty$

Try $\underline{x}_0 = \begin{bmatrix} 0.5 \\ -1 \\ 0 \end{bmatrix}$

i	x_1	x_2	x_3	ε
1	1,25	0,75	0,3125	2,75
2	−1,5	−0,78	0,53	0,927
3	−2,42	−1,21	0,72	5,4
4	−7,83	−3,2	1,84	~77
				↓
				$\to \infty$

Diverges → No Solution obtained

Try $x_0 = \begin{bmatrix} -1/2 \\ -1 \\ 0 \end{bmatrix}$

i	x_1	x_2	x_3	ε
1	0,25	−0,25	0,3125	1,125
2	1	0,21	0,03	1,214
3	−0,21	0,034	0,26	1,17
4	0,961	0,188	0,01	1,06
				⋮
				~1

Does not diverge or converge,

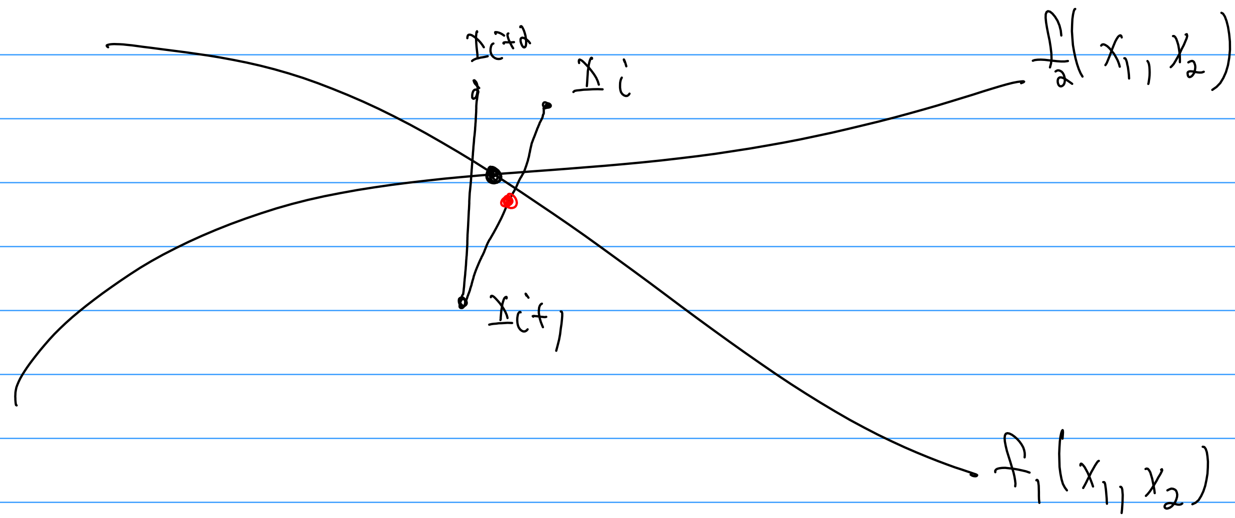
Try a damped iteration,

$$\hat{x}_{i+1} = g(\underline{x}_i) \Rightarrow \underline{x}_{i+1} = 1/2\, \underline{x}_i + 1/2\, \hat{\underline{x}}_{i+1}$$

Damped:

i	x_1	x_2	x_3	ε
10	0.577	0.153	0.0896	1.2×10^{-3}
27	$\sim$0.577	$\sim$0.153	$\sim$0.0896	1.3×10^{-8}

Doing this damped iteration does <u>not</u> help the first $\underline{x}_0$.



Now try Newton - Rhapson

$$\underline{J} = \begin{bmatrix} 1+2x_1+y_2 & x_1 & 0 \\ x_2 & 1+x_1 & 2x_3 \\ 2x_1 & 2x_2 & -4 \end{bmatrix}$$

Try $x_0 = \begin{bmatrix} -1/2 \\ -1 \\ 0 \end{bmatrix}$

i	x_1	x_2	x_3	ε
1	-1.25	-1	0.5	0.563
4	-1.04165	-0.99	0.475	0.186
6	-1	-0.99	0.49	1.3×10^{-5}
7	-1	-1	0.5	3×10^{-9}

Different Root!

Try $x_0 = \begin{bmatrix} 1/2 \\ -1 \\ 0 \end{bmatrix}$ $\quad i=5 \quad x = \begin{bmatrix} 0.5777 \\ 0.153 \\ 0.0893 \end{bmatrix}$

$\varepsilon = 1.6\times10^{-11}$

Matlab function :

fzero → Roots of one equation

fsolve → Roots of nonlinear system,

Same basic input:

func(f, x_0)

ex,) fzero(@ (x) sin(x), 3,14)

ex,) fsolve (@ F, x0)

function (f) = F(x)

⋮

Minimization

Closely related to root finding.

Start w/ one-variable functions, $f(x)$

Find the minimum (or maximum) of $f(x)$.

① Brent's Method.

Let (a, b, c) be a triplet such that $a < b < c$ and

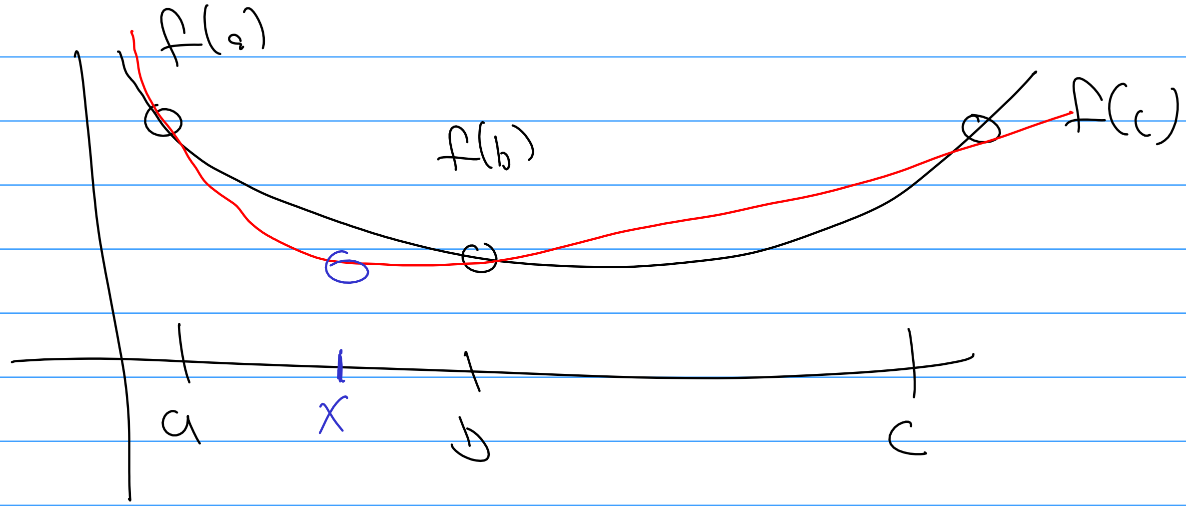
$$f(a) > f(b) \quad \text{and} \quad f(b) < f(c)$$

$\Rightarrow$ A minimum must exist in $[a, c]$

Construct a 2^{nd}-order, quadratic polynomial through

$$(a, f(a)), \ (b, f(b)), \ (c, f(c)),$$

then find where the derivative is zero.



Let that minimum be x,

Choose (a, x, b) or (b, x, c) as appropriate,

Repeat until convergence,

Need to keep an eye on step size,
need to make sure that

$f(a) > f(x)$ & $f(x) < f(b)$ for example,

(2) Newton's Method for minimization

Let x_n be the current approximate solution to the true minimum location x^*,

$$f(x_n + \Delta x) = f(x_n) + \Delta x f'(x_n) + \frac{1}{2}\Delta x^2 f''(x_n) + O(\Delta x^3)$$

Minimum occurs when $\frac{\partial f}{\partial x} = 0$

$\Rightarrow$ find Δx such that $\frac{\partial f(x_n + \Delta x)}{\partial \Delta x} = 0$

$$\Rightarrow 0 = f'(x_n) + \Delta x f''(x_n) + \cancel{O(\Delta x^2)}$$

$$\Rightarrow \Delta x = -\frac{f'(x_n)}{f''(x_n)}$$ 2 is missing. Check

let $\Delta x = x_{n+1} - x_n = \dfrac{-f'(x_n)}{f''(x_n)}$

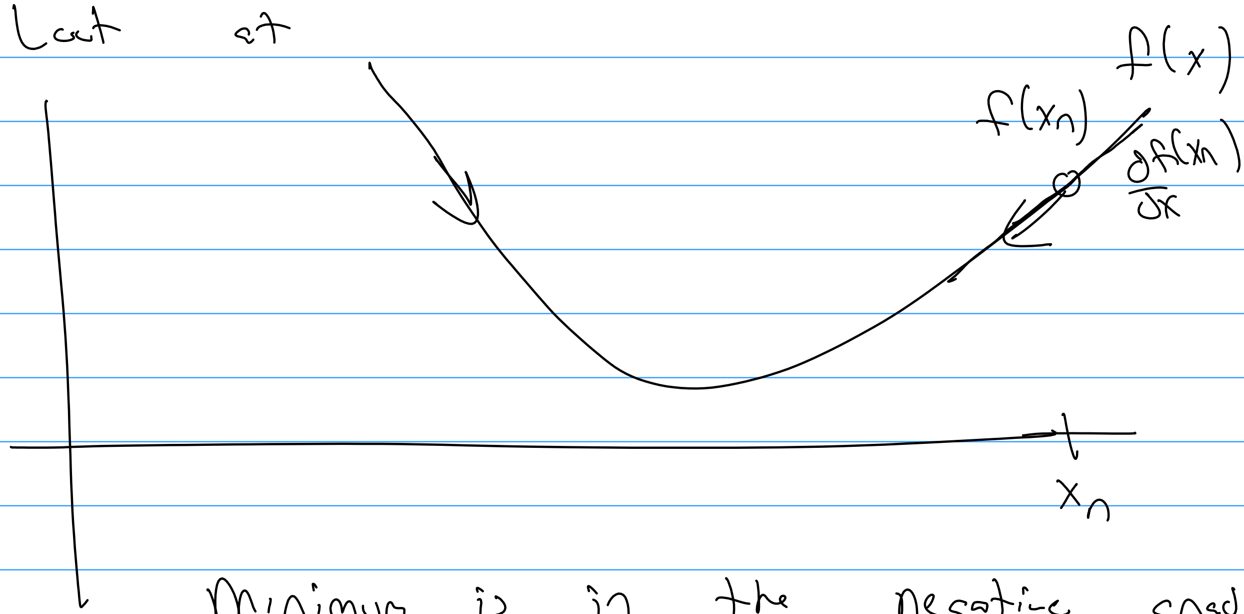
$$\Rightarrow \underbrace{x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}}$$

The newton iteration to find the root $f'(x) = 0$

③ Steepest Gradient Descent

More useful if # of variables is > 1.

Look at



Minimum is in the negative gradient direction

Construct an iteration that takes a step in $-\frac{\partial f}{\partial x}$:

$$x_{n+1} = x_n - \alpha_n \frac{\partial f(x_n)}{\partial x}$$

where α_n is chosen every time step such that $|f(x_n - \alpha_n f'(x_n))| < |f(x_n)|$

These are called line-search methods, as the minimization problem now becomes how to chose α_n.

More later.