Systems of Nonlinear equations

Root finding for n-nonlinear equations w/ n unknowns

$$\left.\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned}\right\} \underline{f}(\underline{x})$$

Methods: 1) fixed-point
2) Multidimensional Newton-Rhaphson

Fixed point

For one equation: $f(x) = g(x) + ax = 0$

$$\Rightarrow x_{n+1} = -\frac{1}{a} g(x_n)$$

Multidimensional: $\underline{f}(\underline{x}) = \underset{\substack{\uparrow \\ \text{nonlinear}}}{\underline{g}(\underline{x})} + \underline{A}\,\underline{x} = \underline{0}$

Iteration: $\underline{g}(\underline{x}_n) + \underline{A}\,\underline{x}_{n+1} = \underline{0}$

$\underline{x}_n$ = current (potential) solution

$\underline{x}_{n+1}$ = next (potential) solution

If $\det(\underline{A}) \neq 0 \Rightarrow \underline{x}_{n+1} = -\underline{A}^{-1}\underline{g}(\underline{x}_n)$

Convergence if $\|\underline{x}_{n+1} - \underline{x}_n\| < \varepsilon_0$

or if $\|\underline{f}(\underline{x}_{n+1})\| < \varepsilon_1$

Note: let $f_1(x_1, x_2) = \sin(x_1)\, x_2 + 3x_1 - 2x_2$

$$f_2(x_1, x_2) = \underbrace{x_1^2}_{g(\underline{x})} + \underbrace{4x_1 - x_2}_{\underline{A}\,\underline{x}}$$

$$g(\underline{x}) = \begin{bmatrix} \sin(x_1)\, x_2 \\ x_1^2 \end{bmatrix} \qquad \underline{A} = \begin{bmatrix} 3 & -2 \\ 4 & -1 \end{bmatrix}$$

$$\Rightarrow \underline{x}_{n+1} = -\begin{bmatrix} 3 & -2 \\ 4 & -1 \end{bmatrix}^{-1} \begin{bmatrix} \sin(x_1)\, x_2 \\ x_1^2 \end{bmatrix} \leftarrow$$

To avoid inverting a matrix let

$$\underline{A} = \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix} \Rightarrow g(x) = \begin{bmatrix} \sin(x_1)\, x_2 + 2x_2 \\ x_1^2 - 4x_1 \end{bmatrix}$$

Multidimensional Newton-Rhapson

Look at $f_1(x_1^i, x_2^i, \ldots, x_n^i) = f_1(\underline{x}_i)$

$i =$ iteration #, <u>not</u> power.

Taylor Series about $\underline{x}_i$

$$f_1(\underline{x}) = f_1(\underline{x}_i) + \frac{\partial f_1}{\partial x_1}(\underline{x}_i)(x_1 - x_1^i)$$

$$+ \frac{\partial f_1}{\partial x_2}(\underline{x}_i)(x_2 - x_2^i) + \frac{\partial f_1}{\partial x_3}(\underline{x}_i)(x_3 - x_3^i)$$

$$+ \ldots \underbrace{\frac{\partial f_1}{\partial x_n}(\underline{x}_i)(x_n - x_n^i)}_{} + \cancel{H.O.T.}$$

let $\nabla f_1^i = \left[\begin{bmatrix} \partial f_1/\partial x_1 \\ \partial f_1/\partial x_2 \\ \vdots \\ \partial f_1/\partial x_n \end{bmatrix}\right]_{\underline{x}_i}$

$$\Rightarrow f_1(\underline{x}) = f_1(\underline{x}_i) + (\nabla f_1^i)^T(\underline{x} - \underline{x}_i) = 0$$

$$\left.\begin{aligned} f_2(\underline{x}) &= f_2(\underline{x}_i) + (\nabla f_2^i)^T(\underline{x} - \underline{x}_i) = 0 \\ &\vdots \\ f_n(\underline{x}) &= f_n(\underline{x}_i) + (\nabla f_n^i)^T(\underline{x} - \underline{x}_i) = 0 \end{aligned}\right\}$$

let $\underline{f}_i = \begin{bmatrix} f_1(\underline{x}_i) \\ f_2(\underline{x}_i) \\ \vdots \\ f_n(\underline{x}_i) \end{bmatrix}$

Jacobian of $\underline{f}(\underline{x}) = \begin{bmatrix} -(\nabla f_1)^T- \\ -(\nabla f_2)^T- \\ \vdots \\ -(\nabla f_n)^T- \end{bmatrix}$ = A matrix

$$= \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 & \cdots & \partial f_1/\partial x_n \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 & \cdots & \partial f_2/\partial x_n \\ \vdots & & & \\ \partial f_n/\partial x_1 & \cdots & & \partial f_n/\partial x_n \end{bmatrix} = \underline{J}(\underline{x})$$

$\underline{J}_i = \underline{J}(\underline{x}_i)$

Given $\underline{f}(\underline{x})$, $\underline{J}(\underline{x})$ is known
Given $\underline{x}_i$, $\underline{J}(\underline{x}_i)$ is a matrix of numbers.

$\Rightarrow$ Our system (Taylor Series) is

$$\overset{\wedge}{\underline{f}} \leftarrow \text{Order-1 approximation}$$

$$\hat{\underline{f}}(\underline{x}) = \underline{f}(\underline{x}_i) + \underline{J}(\underline{x}_i)(\underline{x} - \underline{x}_i) = \underline{0}$$

Iteration is then $\underline{f}(\underline{x}_i) + \underline{J}(\underline{x}_i)(\underline{x}_{i+1} - \underline{x}_i) = \underline{0}$

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

General Procedure: Given $\underline{f}(\underline{x})$ & $\underline{x}_i$

1) Solve $\underline{J}(\underline{x}_i)\underline{\delta}_i = -\underline{f}(\underline{x}_i) \leftarrow$
2) $\underline{x}_{i+1} = \underline{x}_i + \underline{\delta}_i$
3) Convergence if $||\underline{x}_{i+1} - \underline{x}_i|| = ||\underline{\delta}_i|| < \varepsilon_0$
   $||\underline{f}(\underline{x}_{i+1})|| < \varepsilon_1$

Notes: $\underline{J}_i = \underline{J}(\underline{x}_i)$ changes every iteration.

If $\underline{J}(\underline{x}_i)$ does not change then it's a linear system, not nonlinear.

This requires the solution of a different Linear system each iteration.

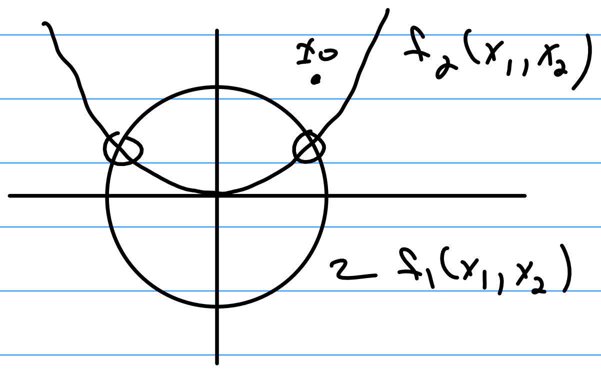
What if a single equation: $f_1(x_1) = 0$?

$$\underline{J} = \left[\partial f_1 / \partial x_1\right] = \frac{df}{dx_1} = f'(x_1)$$

$$\Rightarrow x_{i+1} = x_i - J_i^{-1} f(x_i) = x_i - f(x_i)/f'(x_i)$$

a.) $f_1(x_1, x_2) = x_1^2 + x_2^2 - 1 = 0$

$f_2(x_1, x_2) = x_1^2 - x_2 = 0$



Initial guess of $\underline{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$\underline{f}_0 = \underline{f}(\underline{x}_0) = \begin{bmatrix} 1^2 + 1^2 - 1 \\ 1^2 - 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\underline{J}(\underline{x}) = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 & 2x_2 \\ 2x_1 & -1 \end{bmatrix}$$

$$\underline{J}_0 = \underline{J}(\underline{x}_0) = \begin{bmatrix} 2 & 2 \\ 2 & -1 \end{bmatrix} \qquad \|\underline{f}_0\|_2 = 1$$

$$\underline{x}_1 = \underline{x}_0 - \underline{J}_0^{-1} \underline{f}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1/6 & 1/3 \\ 1/3 & -1/3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 5/6 \\ 2/3 \end{bmatrix} = \underline{x}_1$$

$$\underline{f}_1 = \underline{f}(\underline{x}_1) = \begin{bmatrix} 5/36 \\ 1/36 \end{bmatrix} \qquad \|\underline{f}_1\|_2 \sim 1.4 \times 10^{-1}$$

$$\underline{x}_2 = \underline{x}_1 - \underline{J}_1^{-1} \underline{f}_1 = \begin{bmatrix} 5/6 \\ 2/3 \end{bmatrix} - \begin{bmatrix} 5/3 & 4/3 \\ 5/3 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 5/36 \\ 1/36 \end{bmatrix} = \begin{bmatrix} 331/420 \\ 13/21 \end{bmatrix}$$

$\|\underline{f}(\underline{x}_2)\|_2 \sim 4.8 \times 10^{-3}$

Convergence is 2^{nd}-order

As before you need to worry
about oscillations & if
$\det(\underline{J}_i) \approx 0$ (same as $df/dx \sim 0$ in 1D)

Damped Iterations

let $\underline{x}^*$ be the root of $\underline{f}(\underline{x})$: $\underline{f}(\underline{x}^*) = \underline{0}$

If $\underline{x}_0$ is not near $\underline{x}^*$, iteration may not converge

$\uparrow$ initial guess

Introduce a Damped iteration

let $\underline{\delta}_i := \underline{x}_{i+1} - \underline{x}_i$ be a full step for a given method. $\Rightarrow$ $\underline{x}_{i+1} = \underline{x}_i + \underline{\delta}_i$

Fixed point: $\underline{\delta}_i := -\underline{A}^{-1} \underline{g}(\underline{x}_i) - \underline{x}_i$

Newton: $\underline{\delta}_i := -\underline{J}_i^{-1} \underline{f}(\underline{x}_i)$

A Damped iteration takes a partial-step

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

$\uparrow$ Damping factor

In practice α_i can be fixed ($\alpha_i = 0.5$) or it can be computed to meet some criteria such as $\|\underline{f}(\underline{x}_i + \alpha_i \underline{\delta}_i)\| < \|\underline{f}(\underline{x}_i)\|$

One Line Search methods (more later)

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

$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

Fixed Point:

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

$$\underline{x}_{i+1} = \underline{g}(\underline{x}_i)$$

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

i	x_1	x_2	x_3	ε
0	0.5	-1	0	
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

$\downarrow$

∞

Diverges – No Solution

New $\underline{x}_0$:

i	x_1	x_2	x_3	ε
0	$-1/2$	-1	0	
1	0.25	-0.25	0.3125	1.125
2	1	0.21	0.03	1.214

$\Downarrow$ 1

error stagnates → No Solution

Try $\underline{x}_0 = \begin{bmatrix} -1/2 \\ -1 \\ 0 \end{bmatrix}$ w/ $\alpha_i = 0.5$

: $\underline{x}_{i+1} = 0.5\, \underline{x}_i + 1/2\, \underline{g}(\underline{x}_i)$

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

No Damping → Stagnates
Damping → A Solution

Damping does <u>not</u> help $\underline{x}_0 = \begin{bmatrix} 1/2 \\ -1 \\ 0 \end{bmatrix}$

Try Newton

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

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

Try w/ $\underline{x}_0 = \begin{bmatrix} -1/2 \\ -1 \\ 0 \end{bmatrix}$ w/ no damping

i	x_1	x_2	x_3	ε
1	-1.25	-1	0.5	0.56
4	-1.0465	-0.99	0.475	0.186
6	-1	-0.99	0.49	1.3×10^{-5}
7	-1	-1	0.5	3×10^{-9}

Solution will depend on x_0, any damping & the method used.

Try $\underline{x}_0 = \begin{bmatrix} 1/2 \\ -1 \\ 0 \end{bmatrix}$ $i = 5$ $\underline{x} = \begin{bmatrix} 0.577 \\ 0.153 \\ 0.0893 \end{bmatrix}$

$\varepsilon \sim 10^{-11}$

Matlab Functions

1D: fzero(fun, x0) ← initial guess

↑ function handle

System: fsolve(fun, x0)

fun = function handle that returns the residual vector and possibly the Jacobian matrix evaluated at an input vector $\underline{x}$.

x0 = initial guess vector.

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

```
function [f] = Sample(x)
    f = zeros(3,1)
    f(1) = x(1)^2 + x(1)*x(2) + x(1) -1;
    f(2) = x(1)*x(2) + x(2) + x(3)^2 -0.25;
    f(3) = x(1)^2 + x(2)^2 - 4*x(3);
end
```

$$\text{fsolve}(@\text{Sample}, [0; 1; 0]) = \begin{bmatrix} 0.577 \\ 0.1534 \\ 0.0893 \end{bmatrix}$$

fsolve (@ sample, [-1; -1; 0]) = $\begin{bmatrix} -1 \\ -1 \\ -1/2 \end{bmatrix}$

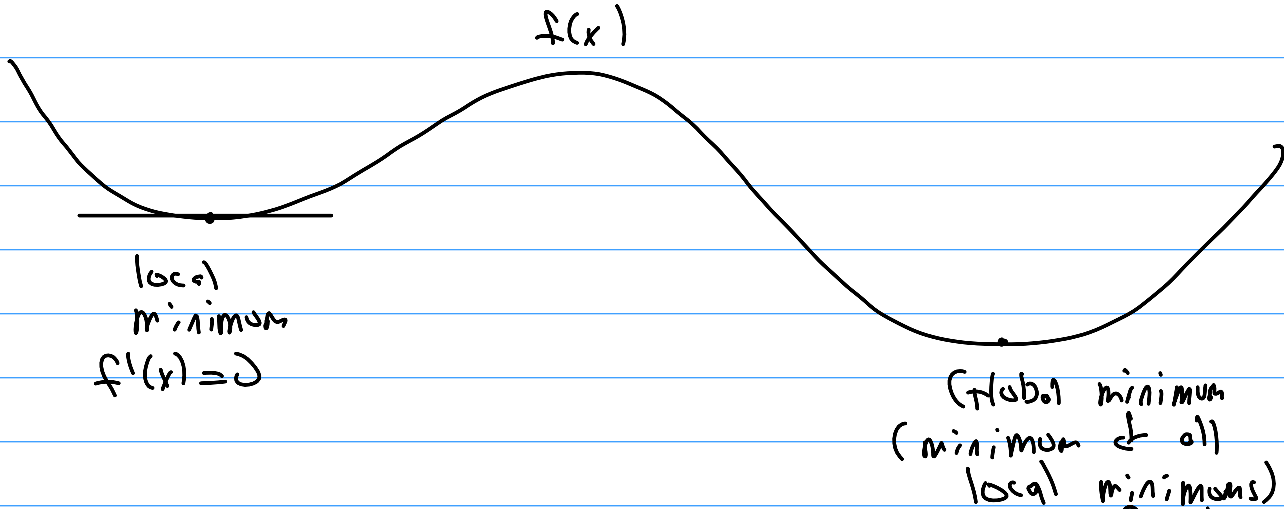
w/ Jacobian :

```
[f, J] = Sample(x)
    f = zeros(3,1)
    f(1) = ...
    f(2) = ...
    f(3) = ...

->  if nargout == 2
        J = zeros(3,3)
        J(1,1) = 1 + 2 * x(1) + x(2);
        J(1,2) = x(2);
        etc.
    end
end
```

Minimization / Optimization

let $f(x)$ be an objective function to be minimized. Assume $f(x)$ is continuous.



To maximize, do the minimum of $-f(x)$

Methods for local minimum:

- Brent's method
- Newton's method
- Steepest Gradient Descent
- others (Praxis, Nelder-Mead, Simplex, etc.)

Global minimums

- Simulated Annealing
- Genetic Algorithms
- others (MLSL, Direct-L, etc.)

Focus on unconstrained local minimums
find $f'(x) = 0$

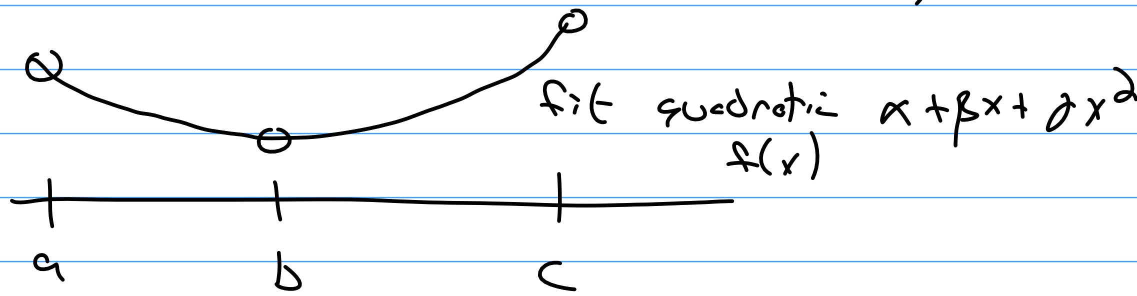
Constrained minimization: minimize $f(x)$
subject to $g(x) = 0$

Note: In addition to Matlab I
recommend Nlopt & TAO & GSL

Brent's Method

A bracketing Method

let $a < b < c$ w/ $f(a) > f(b) < f(c)$



fit $f(a)$, $f(b)$, & $f(c)$ to a quadratic
equation $f(x) = \alpha + \beta x + \gamma x^2$

Solve $f'(m) = 0 \Rightarrow m = -\frac{\beta}{2\gamma}$

If $a < m < b$ use $[a, m, b]$ & repeat
if $b < m < c$ use $[b, m, c]$ & repeat

Iterate until $|c - a| < \varepsilon_0$ or $|f'(m)| < \varepsilon_1$

might fail: