Nonlinear Algebraic Equations Example

Continuous Stirred Tank Reactor (CSTR).
Look for steady state concentrations & temperature.

*In*: $N_s$ species with concentrations $c_{i}^{(\text{in})}$, heat capacities $c_{p,i}^{(\text{in})}$ and temperature $T^{(\text{in})}$

*Inside*: $N_r$ reactions with stoichiometric coefficients $\sigma_{\alpha,i}$ and reaction constants $r_{\alpha}$.

*Out*: $N_s$ species with concentrations $c_{i}^{(\text{in})}$ (c-s may be equal to zero), heat capacities $c_{p,i}$ and temperature $T$. 
Nonlinear Algebraic Equations Example

\[
\frac{(F/V)}{V}\left[c_{i}^{(in)} - c_i\right] + \sum_{\alpha=1}^{N_r} \sigma_{\alpha,i} r_\alpha = 0 \quad i = 1, 2, ..., N_s
\]

Mass balance for spieces 1, 2, ..., N_s

\[
\frac{(F/V)}{V}\sum_{i=1}^{N_s} \left[ c_{i}^{(in)} c_{p,i}^{(in)} T^{(in)} - c_i c_{p,i} T \right] - \sum_{\alpha=1}^{N_r} \Delta H_\alpha r_\alpha = 0
\]

Energy balance

Column of unknown variables: \( x = [c_1, c_2, ..., c_{N_s}, T]^T \)
Nonlinear Algebraic Equations Example

Each of the above equations may be written in the general form:

\[ f_i(x_1, x_2, \ldots, x_N) = 0 \]

In vector form:

\[ f(x) = 0 \]

\[ f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_N(x) \end{bmatrix} \]

Let \( \hat{x} \) be the solution satisfying \( f(x)=0 \).

We do not know \( \hat{x} \) and take \( x^{[0]} \) as initial guess.
Nonlinear Algebraic Equations

We need to form a sequence of estimates to the solution: $x^{[1]}, x^{[2]}, x^{[3]}, \ldots$ that will hopefully converge to $x^\wedge$.

Thus we want: $\lim_{m \to \infty} x^{[m]} = x^\wedge$ \[\lim_{m \to \infty} \left\| x^\wedge - x^{[m]} \right\| = 0\]

Unlike with linear equations, we can’t say much about existence or uniqueness of solutions, even for a single equation.
Single Nonlinear Equation

We assume \( f(x) \) is infinitely differentiable at the solution \( \hat{x} \).

Then \( f(x) \) may be Taylor expanded around \( \hat{x} \):

\[
f(x) = f(\hat{x}) + (x - \hat{x})f'(x) + \frac{1}{2}(x - \hat{x})^2f''(x) + \frac{1}{3!}(x - \hat{x})^3f'''(x) + \ldots
\]

Assume \( x^{[0]} \) being close to \( \hat{x} \) so that the series may be truncated:

\[
f(x^{[0]}) \approx (x^{[0]} - \hat{x})f'(x^{[0]}), \quad \text{or}
\]

\[
f(x^{[0]}) \approx (x^{[0]} - x^{[1]})f'(x^{[0]}) \quad \text{or} \quad x^{[1]} = x^{[0]} - \frac{f(x^{[0]})}{f'(x^{[0]})}
\]

– Newton’s method

Example: \( f(x) = (x - 3)(x - 2)(x - 1) = x^3 - 6x^2 + 11x - 6 \)
Single Nonlinear Equation

Plot of $f(x) = x^3 - 6x^2 + 11x - 6$
Single Nonlinear Equation

Convergence of $f(x) = x^3 - 6x^2 + 11x - 6$ and "Newton step" $-\frac{f(x)}{f'(x)}$
Single Nonlinear Equation

For \( f(x) = (x-3)(x-2)(x-1) = x^3 - 6x^2 + 11x + 6 = 0 \), we see that Newton’s method converges to the root at \( x = 2 \) only if \( 1.6 < x^0 < 2.4 \).

We may look at the direction of the first step and see why: \( x^{[1]} - x^{[0]} = -f(x^{[0]})/ f'(x^{[0]}) \).

So, we have 2 “easy” roots: \( x = 1 \), \( x = 3 \) and a more difficult one.

We may “factorize out” the roots we already know.
Single Nonlinear Equation

Introduce \( g(x) = \frac{f(x)}{(x-3)(x-1)} \)

\[
g'(x) = \frac{f'(x)}{(x-3)(x-1)} - \frac{f(x)}{(x-3)(x-1)} \left( \frac{1}{x-3} + \frac{1}{x-1} \right)
\]

We now use Newton’s method to find the roots of \( g(x) \):

\[
x^{[i+1]} = x^{[i]} - \frac{g(x)}{g'(x)} \quad \rightarrow \quad \text{get } x = 2.
\]
Systems of Nonlinear Equations

Let’s extend the method to multiple equations:

\[ f_{1}(x_1, x_2, x, \ldots, x_N) = 0 \]
\[ f_{2}(x_1, x_2, x, \ldots, x_N) = 0 \quad \Rightarrow \quad f(x) = 0 \]

\[ \ldots \]

\[ f_{N}(x_1, x_2, x, \ldots, x_N) = 0 \]

As before expand each equation at the solution \( \hat{x} \) with \( f(\hat{x}) = 0 \):

\[ f_{i}(x) = f_{i}(\hat{x}) + \sum_{j=1}^{N} (x_j - \hat{x}_j) \frac{\partial f_{i}(x)}{\partial x_j} + \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} (x_j - \hat{x}_j) \frac{\partial^2 f_{i}(x)}{\partial x_j \partial x_k} (x_k - \hat{x}_k) + \ldots \]

Assume \( x^{[0]} \) is close to \( \hat{x} \) and discard quadratic terms:

\[ f_{i}(x) \approx \sum_{j=1}^{N} (x_j - \hat{x}_j) \frac{\partial f_{i}(x)}{\partial x_j} \]
Systems of Nonlinear Equations

Let's define the Jacobian matrix $J(\hat{x})$ with the elements:

$$J_{ij}(\hat{x}) = \frac{\partial f_i(\hat{x})}{\partial x_j}$$

Then our approximate expansion may be written as:

$$f_i(x) \approx \sum_{j=1}^{N} J_{ij}(\hat{x})(x_j - \hat{x}_j) = \left(J(\hat{x})(x - \hat{x})\right)_i$$

This gives us the linear system:

$$f(x) \approx J(\hat{x})(x - \hat{x})$$
Systems of Nonlinear Equations

\[ f(x^{[i]}) = J(x^{[i]})(x^{[i]} - x^{[i+1]}) = \left( J(x)(x - x) \right)_i \]

Note, Jacobian is evaluated at the position of an old iteration, not at an unknown solution

Defining \( \Delta x^{[i]} = x^{[i+1]} - x^{[i]} \), we rewrite the equation as \( J(x^{[i]})\Delta x^{[i]} = -f(x^{[i]}) \) or just \( J^{[i]}\Delta x^{[i]} = -f^{[i]} \)

The iterations are continued until some convergence criteria are met:

relative error \( \|f^{[i]}\| \leq \delta_{\text{rel}} \|f^{[0]}\| \)

absolute error \( \|f^{[i]}\| \leq \delta_{\text{abs}} \)
Newton’s method does not always converge. If it does, can we estimate the error?

Let our function $f(x)$ be continuously differentiable in the vicinity of $x$ and the vector connecting $x$ and $x + p$ all lies inside this vicinity.

$$J(x + sp) = \left. \frac{\partial f}{\partial x^T} \right|_{x+sp}$$

$$f(x + p) = f(x) + \int_0^1 J(x + sp) p \, ds$$

– use path integral along the line $x - x+p$, parametrized by $s$. 
Systems of Nonlinear Equations

Add and subtract $J(x)p$ to RHS:

$$f(x + p) = f(x) + J(x)p + \int_{0}^{1} [J(x+sp) - J(x)] \, p \, ds$$

In Newton’s method we ignore the integral term and choose $p$ to estimate $f(x+p)$. Thus the error in this case is:

$$\int_{0}^{1} [J(x+sp) - J(x)] \, p \, ds = f(x + p)$$

What is the upper bound on this error?

$f(x)$ and $J(x)$ are continuous:

$$\left\| J(x+sp)-J(x) \right\| \to 0 \text{ as } p \to 0 \text{ for all } 0 \leq s \leq 1.$$
Systems of Nonlinear Equations

The norm of the matrix is defined as: \[ \|A\| = \max_{v \neq 0} \frac{\|Av\|}{\|v\|}, \]

so for any \( y \) \( \|A\| \geq \frac{\|Ay\|}{\|y\|} \), or \( \|Ay\| \leq \|A\| \|y\| \), therefore

\[ \left\| \int_0^1 [J(x + sp) - J(x)] \, p \, ds \right\| \leq \int_0^1 \| [J(x + sp) - J(x)] \| ds \|p\| \]

The error goes down at least as fast as \( \|p\| \) because for a continuous Jacobian \( \|J(x + sp) - J(x)\| \to 0 \) as \( p \to 0 \).
Systems of Nonlinear Equations

If we suggest that there exist some $L>0$ such that

$$\|J(y)-J(z)\| \leq L \|y - z\|$$

or there is some upper bound on the "stretching" effect of $J$.

$$\|f(x + sp)\| = \left\| \int_0^1 [J(x + sp) - J(x)]p \, ds \right\| \leq \left\| \int_0^1 [J(x + sp) - J(x)] \, ds \right\| \|p\|$$

and

$$\left\| \int_0^1 [J(x + sp) - J(x)] \, ds \right\| \leq L \|s\| \|p\|,$$

so in this case

$$\|f(x + sp)\| = \left\| \int_0^1 [J(x + sp) - J(x)]p \, ds \right\| \leq (L \|s\| \|p\|) \|p\|$$

$$\leq (L \|s\|) \|p\|^2 = O(\|p\|^2)$$
Systems of Nonlinear Equations

Thus if we are at distance $p$ from the solution, our error scales as $||p||^2$.

What about convergence? How the error scales with the number of iterations?

The answer is:

$$
\left\| x^{[i+1]} - \hat{x} \right\| = \mathcal{O}\left(\left\| x^{[i]} - \hat{x} \right\|^2 \right)$$

- local quadratic convergence, a very fast one! Works when you are close enough to the solution.
Systems of Nonlinear Equations

\[ \left\| x^{[i]} - \hat{x} \right\| \sim 0.1 = 10^{-1} \]
\[ \left\| x^{[i+1]} - \hat{x} \right\| \sim 0.01 = 10^{-2} \]
\[ \left\| x^{[i+2]} - \hat{x} \right\| \sim 0.00001 = 10^{-4} \]
\[ \left\| x^{[i+3]} - \hat{x} \right\| \sim 0.000000001 = 10^{-8} \]

Works if we are close enough to an isolated (non-singular) solution: \( \det \left[ J(\hat{x}) \right] \neq 0 \).
Systems of Nonlinear Equations, Example

Let’s examine how Newton’s method works for a simple system:

\[ f_1 = 3x_1^3 + 4x_2^2 - 145 = 0 \]
\[ f_2 = 4x_1^2 - x_2^3 + 28 = 0 \]

The Jacobian is:

\[
J = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\
\frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2}
\end{bmatrix}
= \begin{bmatrix}
9x_1^2 & 8x_2 \\
8x_1 & -3x_2^2
\end{bmatrix}
\]
Systems of Nonlinear Equations, Example

At each step we have the following system to solve:

\[
J^{[i]} = \begin{bmatrix}
9(x_1^{[i]})^2 & 8x_2^{[i]} \\
8x_1^{[i]} & -3(x_2^{[i]})^2
\end{bmatrix}
\]

\[
f^{[i]} = \begin{bmatrix}
3(x_1^{[i]})^3 + 4(x_2^{[i]})^2 - 145 \\
4(x_1^{[i]})^2 - (x_2^{[i]})^3 + 28
\end{bmatrix}
\]

\[
J^{[i]}\Delta x^{[i]} = -f^{[i]} \\
x^{[i+1]} = x^{[i]} + \Delta x^{[i]}
\]
Systems of Nonlinear Equations, Example

Let us examine performance of Newton’s method with the convergence criterion \( \| f \| = \max \{ |f_1|, |f_2| \} < \delta_{\text{abs}} \)
\[
\delta_{bs} = 10^{-10}
\]
Newton’s Method

Newton’s method works well close to the solution, but otherwise takes large erratic steps, shows poor performance and reliability.

Let’s try to avoid such large steps: employ reduced step Newton’s algorithm.

“Full” Newton’s step gives $x^{[i+1]}=x^{[i]}+p^{[i]}$. We’ll use only fraction of of the step: $x^{[i+1]}=x^{[i]}+\lambda_i \, p^{[i]}$, $0<\lambda_i<1$. 
Newton’s Method

How do we choose $\lambda_i$?

Simplest way - weak line search:
- start with $\lambda_i = 2^{-m}$ for $m=0,1,2,…$

As we reduce the value of $\lambda_i$ by a factor of 2 at each step, we accept the first one that satisfies a descent criterion:

$$\|f(x^{[i]} + \lambda_ip^{[i]})\| < \|f(x^{[i]})\|$$

It can be proven that if the Jacobian is not singular, the correct solution will be found by the reduced step Newton’s algorithm.