Solving Systems of Nonlinear Equations

Dr. José Ernesto Rayas-Sánchez

Outline

- Systems of nonlinear equations (SNLEq)
- Solving SNLEq vs unconstrained optimization problems
- Newton method
- Broyden method
- Stopping criteria for SNLEq
Systems of Nonlinear Equations (SNLEq)

Given \( f: \mathbb{R}^n \rightarrow \mathbb{R}^n \),
find \( x^* \in \mathbb{R}^n \) such that
\[ f(x^*) = 0 \]

\( x^* \) is called a solution or a root of the SNLEq.

The system \( f(x) = 0 \) may have a unique solution, many solutions, or no solution.

Solving SNLEq vs Unconstrained Optimization

- A solution for \( f(x) = 0 \) is also a minimizer of
  \[ \min_{x} \| f(x) \|^2 \quad = \quad \min_{x} f(x)^T f(x) \]
  (nonlinear least squares problem)

- However, a minimizer of
  \[ \min_{x} \| f(x) \|^2 \quad = \quad \min_{x} f(x)^T f(x) \]
  is not necessarily a solution of \( f(x) = 0 \)
Newton’s Method for SNLEq

- It approximates a linear model to the nonlinear system of equations; the solution of the linear model becomes the next iterate
- It can fail if its starting point is bad
- It is also called “Newton-Raphson” method, or the “Method of Tangents”

Linear Models from Taylor Series Expansions

- For a unidimensional scalar function \( f \) around point \( x_0 \), with \( \Delta x = x - x_0 \)
  \[
  f(x) = f(x_0 + \Delta x) \approx f(x_0) + \Delta f'(x_0) = m(x)
  \]
- For a multidimensional scalar function \( u \) around point \( x_0 \), with \( \Delta x = x - x_0 \)
  \[
  u(x) = u(x_0 + \Delta x) \approx u(x_0) + \Delta x^T \nabla u(x_0) = m(x)
  \]
- For a multidimensional vector function \( f \) around point \( x_0 \), with \( \Delta x = x - x_0 \)
  \[
  f(x) = f(x_0 + \Delta x) \approx f(x_0) + J(f(x_0)) \Delta x = m(x)
  \]
Newton’s Method for SNLEq (cont)

Solving the linear model at the $i$-th iteration:

$$m(x) = f(x_i) + J(f(x_i))(x - x_i)$$

$$m(x_{i+1}) = f(x_i) + J(f(x_i))(x_{i+1} - x_i) = 0$$

$$x_{i+1} = x_i - J(f(x_i))^{-1}f(x_i)$$

Newton’s Algorithm for Solving SNLEq

It may fail if $\|f(x_0)\|_2 / \|J(f(x_0))\|_F$ is too large, or if $J(f(x_0))$ is close to singular
Newton’s Algorithm: Method of Tangents

Newton’s Algorithm: Method of Tangents (cont.)
Newton’s Algorithm: Method of Tangents (cont.)

\[ f(x) \]

\[ x_0 \]

\[ x_1 \]

\[ x_2 \]
Newton’s Algorithm: Method of Tangents (cont.)

Good starting point

Newton’s Algorithm: Method of Tangents (cont.)
Newton’s Algorithm: Method of Tangents (cont.)

Bad starting point
Globally Convergent Newton Method

\[ x^* = \text{GCNewton}(f, x_0) \]
\[ f: \mathbb{R}^n \rightarrow \mathbb{R}^n; \ x_0, x^* \in \mathbb{R}^n \]

\begin{align*}
\text{begin} \\
\quad i = 0, x_i = x_0, \text{ set } r \\
\quad \text{repeat until StoppingCriteria} \\
\quad \quad \text{solve } J(f(x_i)) h_i = -f(x_i) \text{ for } h_i \\
\quad \quad x^{\text{test}} = x_i + h_i ; \quad \alpha = 1 \\
\quad \quad \text{while } \left| f(x^{\text{test}}) \right| > \left| f(x_i) \right| \\
\quad \quad \quad \quad \alpha := r \alpha \\
\quad \quad \quad x^{\text{test}} = x_i + \alpha h_i \\
\quad \quad \text{end} \\
\quad x_{i+1} = x^{\text{test}} \\
\quad i = i + 1 \\
\text{end} \\
\quad x^* = x_i
\end{align*}

(0 < r < 1)

It globally converges to a local minimum, but not necessarily to a root

Broyden’s Methods

- They approximate the Jacobian with a matrix, updating it at each iteration
- They are also called “Secant” methods
- Its convergence rate is fast enough for most practical purposes, though not as fast as in Newton’s method
- It is very efficient in terms of the number of function evaluations
Broyden’s Updating Formula

\[ B_i \approx J(f(x_i)) \]

\[ B_{i+1} = B_i + \frac{f_{i+1} - f_i - B_i h_i}{h_i^T h_i} h_i^T \]

where

\[ h_i = x_{i+1} - x_i \quad f_i = f(x_i) \quad f_{i+1} = f(x_{i+1}) \]

A Broyden’s Method Algorithm

\[ \begin{align*}
  x^* &= \text{Broyden}(f, x_0) \\
  f : \mathbb{R}^n &\rightarrow \mathbb{R}^n ; \quad x_0, x^* \in \mathbb{R}^n \\
  \begin{align*}
    i &= 0, x_i = x_0, B_i = I, f_i = f(x_i) \\
    \text{repeat until } &\text{StoppingCriteria} \\
    &\text{solve } B_i h_i = -f_i \text{ for } h_i \\
    &x_{i+1} = x_i + h_i \\
    &f_{i+1} = f(x_{i+1}) \\
    &B_{i+1} = B_i + \frac{f_{i+1} h_i^T}{h_i^T h_i} \\
    &i = i + 1 \\
    \text{end} \\
  \end{align*}
\]

\[ x^* = x_i \]
Broyden’s Algorithm: Method of Secants

\[ f(x) \]

\[ x \]

Broyden’s Algorithm: Method of Secants (cont.)

\[ f(x) \]

\[ x \]

\[ x_0 \]
Broyden’s Algorithm: Method of Secants (cont.)
Broyden’s Algorithm: Method of Secants (cont.)

\[ f(x) \]

Dr. J. E. Rayas-Sánchez

Broyden’s Algorithm: Method of Secants (cont.)

\[ f(x) \]

Dr. J. E. Rayas-Sánchez
Stopping Criteria for Solving SNLEq

- A maximum number of iterations has been reached
  \[ i > i_{\text{max}} \]

- A solution has been found
  \[ f(x_i^{t+1})^T f(x_i^{t+1}) < \epsilon_1 \]
  or
  \[ \|f(x_i^{t+1})\|_e < \epsilon_2 \]

- The absolute change in the optimization variables is small enough
  \[ \|x_{i+1} - x_i\|_2 < \epsilon_3 \]

- The relative change in the optimization variables is small enough
  \[ \|x_{i+1} - x_i\|_2 < \epsilon_4 (\|x_i\|_2 + \epsilon_5) \]