Solving Systems of Nonlinear Equations

Dr. José Ernesto Rayas-Sánchez

1

Outline

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

Dr. J. E. Rayas-Sánchez

2

Systems of Nonlinear Equations (SNLEq)

Given $\boldsymbol{f}: \Re^n \rightarrow \Re^n$,

find $\boldsymbol{x}^* \in \Re^n$ such that

$\boldsymbol{f}(\boldsymbol{x}^*) = \mathbf{0}$

$\boldsymbol{x}^*$ is called a solution or a root of the SNLEq

The system $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ may have a unique solution, many solutions, or no solution

Solving SNLEq *vs* Unconstrained Optimization

- A solution for $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ is also a minimizer of

$$\min_{\boldsymbol{x}} \|\boldsymbol{f}(\boldsymbol{x})\|_2^2 = \min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x})$$

(nonlinear least squares problem)

- However, a minimizer of

$$\min_{\boldsymbol{x}} \|\boldsymbol{f}(\boldsymbol{x})\|_2^2 = \min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x})$$

is not necessarily a solution of $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$

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 $\Delta x = x - x_0$

$$f(x) = f(x_0 + \Delta x) \approx f(x_0) + \Delta x f'(x_0) = m(x)$$

- For a multidimensional scalar function u around point $\boldsymbol{x}_0$, with $\Delta \boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_0$

$$u(\boldsymbol{x}) = u(\boldsymbol{x}_0 + \Delta \boldsymbol{x}) \approx u(\boldsymbol{x}_0) + \Delta \boldsymbol{x}^T \nabla u(\boldsymbol{x}_0) = m(\boldsymbol{x})$$

- For a multidimensional vector function $\boldsymbol{f}$ around point $\boldsymbol{x}_0$, with $\Delta \boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_0$

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_0 + \Delta \boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{x}_0) + \boldsymbol{J}(\boldsymbol{f}(\boldsymbol{x}_0))\Delta \boldsymbol{x} = \boldsymbol{m}(\boldsymbol{x})$$

Newton's Method for SNLEq (cont)

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

$$\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_i) + \boldsymbol{J}(\boldsymbol{f}(\boldsymbol{x}_i))(\boldsymbol{x} - \boldsymbol{x}_i)$$

$$\boldsymbol{m}(\boldsymbol{x}_{i+1}) = \boldsymbol{f}(\boldsymbol{x}_i) + \boldsymbol{J}(\boldsymbol{f}(\boldsymbol{x}_i))(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) = \boldsymbol{0}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \boldsymbol{J}(\boldsymbol{f}(\boldsymbol{x}_i))^{-1}\boldsymbol{f}(\boldsymbol{x}_i)$$

Dr. J. E. Rayas-Sánchez 7

Newton's Algorithm for Solving SNLEq

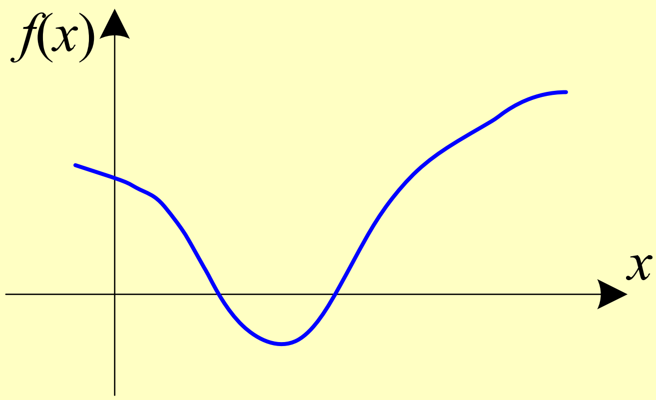
$\boldsymbol{x}^* = \text{Newton}(\boldsymbol{f}, \boldsymbol{x}_0)$
$\boldsymbol{f}: \Re^n \rightarrow \Re^n$; $\boldsymbol{x}_0, \boldsymbol{x}^* \in \Re^n$

```
begin
    i = 0, x_i = x_0
    repeat until StoppingCriteria
        solve J(f(x_i))h_i = -f(x_i) for h_i
        x_{i+1} = x_i + h_i
        i = i + 1
    end
    x* = x_i
end
```

It may fail if $||\boldsymbol{f}(\boldsymbol{x}_0)||_2 / ||\boldsymbol{J}(\boldsymbol{f}(\boldsymbol{x}_0))||_F$ is too large, or if $\boldsymbol{J}(\boldsymbol{f}(\boldsymbol{x}_0))$ is close to singular

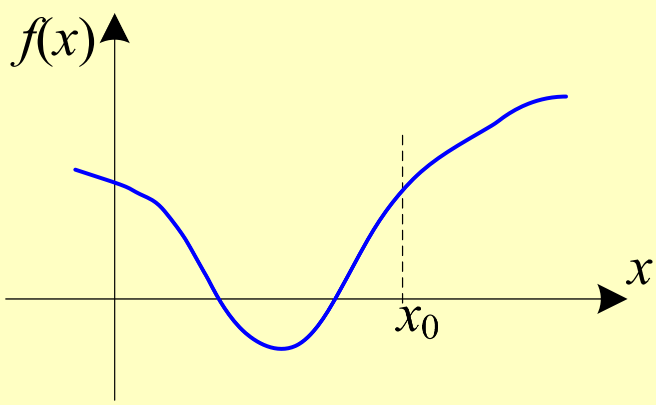
Dr. J. E. Rayas-Sánchez 8

Newton’s Algorithm: Method of Tangents



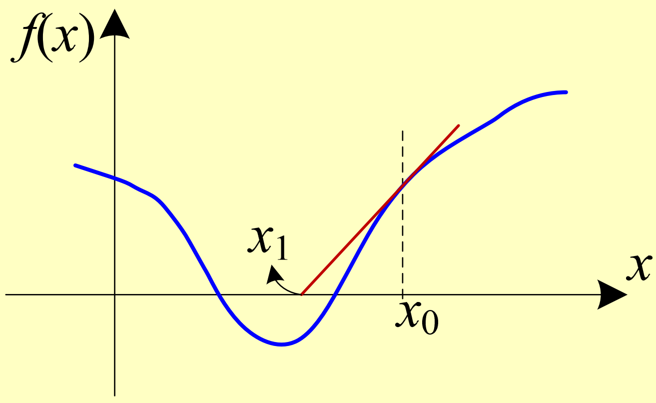
Dr. J. E. Rayas-Sánchez 9

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

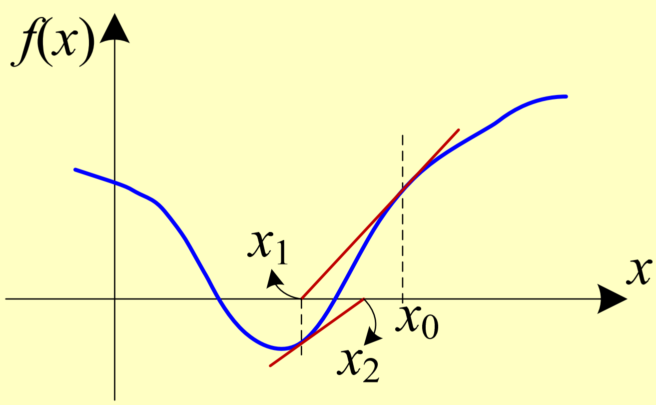


Dr. J. E. Rayas-Sánchez 10

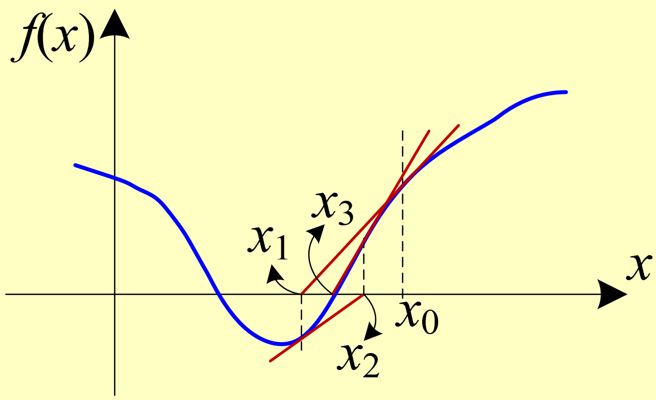
Newton's Algorithm: Method of Tangents (cont.)



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

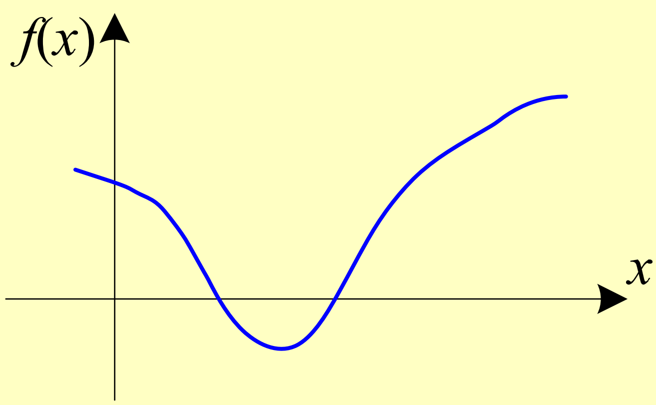


Newton’s Algorithm: Method of Tangents (cont.)

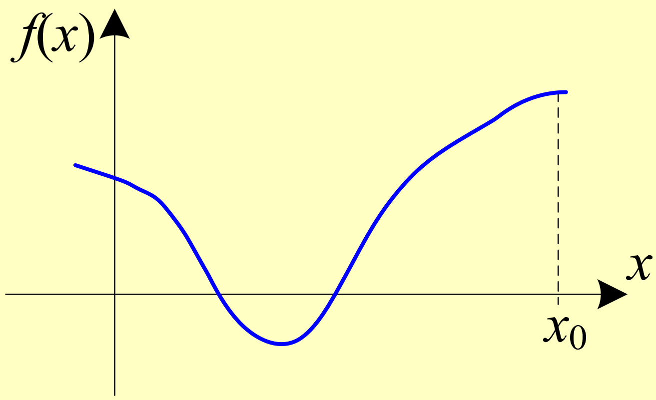


Good starting point

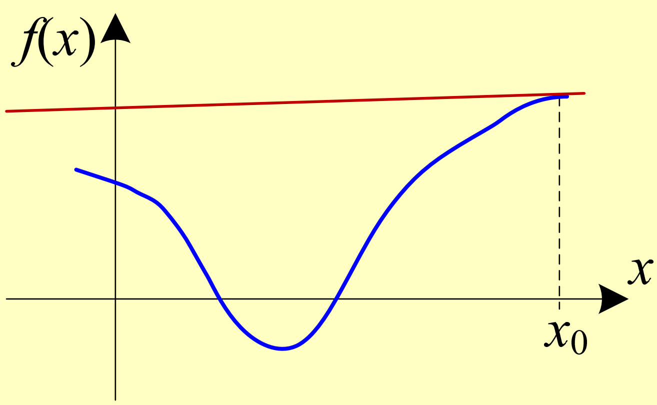
Newton’s Algorithm: Method of Tangents (cont.)



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



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



Bad starting point

Globally Convergent Newton Method

$\boldsymbol{x}^* = \text{GCNewton}(\boldsymbol{f}, \boldsymbol{x}_0)$
$\boldsymbol{f}: \Re^n \rightarrow \Re^n$; $\boldsymbol{x}_0, \boldsymbol{x}^* \in \Re^n$

```
begin
    i = 0, x_i = x_0, set r
    repeat until StoppingCriteria
        solve J(f(x_i))h_i = -f(x_i) for h_i
        x^test = x_i + h_i ; α = 1
        while ||f(x^test)||_2 > ||f(x_i)||_2
            α := rα
            x^test = x_i + αh_i
        end
        x_{i+1} = x^test
        i = i + 1
    end
    x* = x_i
end
```

($0 < r < 1$)

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

$$\boldsymbol{B}_i \approx \boldsymbol{J}(\boldsymbol{f}(\boldsymbol{x}_i))$$

$$\boldsymbol{B}_{i+1} = \boldsymbol{B}_i + \frac{\boldsymbol{f}_{i+1} - \boldsymbol{f}_i - \boldsymbol{B}_i\boldsymbol{h}_i}{\boldsymbol{h}_i^T\boldsymbol{h}_i}\boldsymbol{h}_i^T$$

where

$$\boldsymbol{h}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \qquad \boldsymbol{f}_i = \boldsymbol{f}(\boldsymbol{x}_i) \qquad \boldsymbol{f}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_{i+1})$$

Dr. J. E. Rayas-Sánchez 19

A Broyden's Method Algorithm

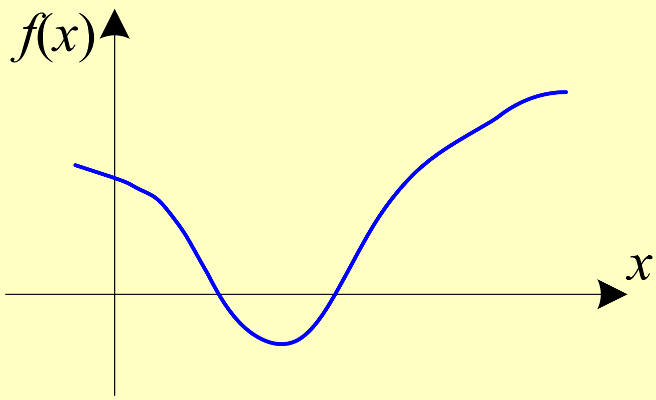
$\boldsymbol{x}^* = \text{Broyden}(\boldsymbol{f}, \boldsymbol{x}_0)$

$\boldsymbol{f}: \Re^n \rightarrow \Re^n$; $\boldsymbol{x}_0, \boldsymbol{x}^* \in \Re^n$

begin
  $i = 0$, $\boldsymbol{x}_i = \boldsymbol{x}_0$, $\boldsymbol{B}_i = \boldsymbol{I}$, $\boldsymbol{f}_i = \boldsymbol{f}(\boldsymbol{x}_i)$
  repeat until *StoppingCriteria*
    solve $\boldsymbol{B}_i\boldsymbol{h}_i = -\boldsymbol{f}_i$ **for** $\boldsymbol{h}_i$
    $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{h}_i$
    $\boldsymbol{f}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_{i+1})$
    $\boldsymbol{B}_{i+1} = \boldsymbol{B}_i + \frac{\boldsymbol{f}_{i+1}\boldsymbol{h}_i^T}{\boldsymbol{h}_i^T\boldsymbol{h}_i}$
    $i = i + 1$
  end
  $\boldsymbol{x}^* = \boldsymbol{x}_i$
end

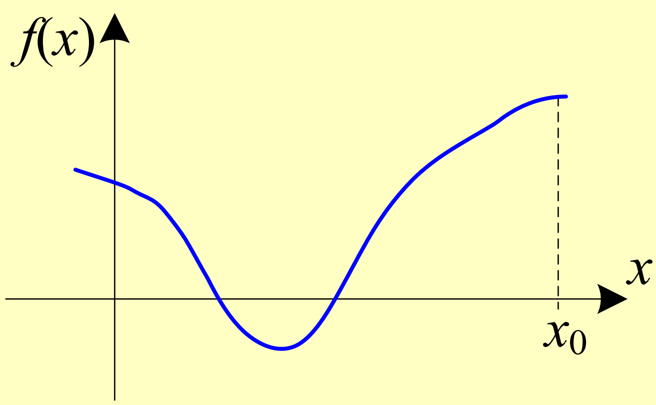
Dr. J. E. Rayas-Sánchez 20

Broyden's Algorithm: Method of Secants



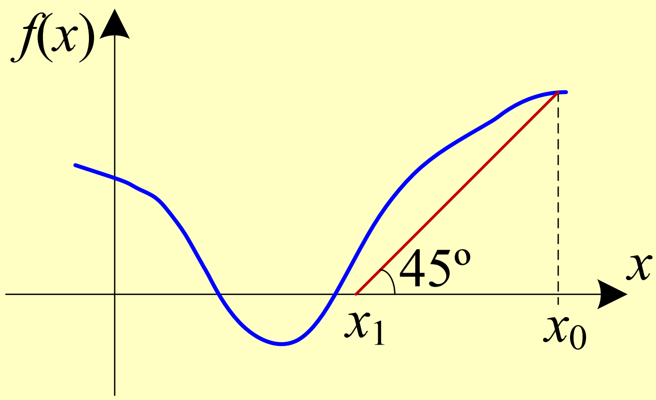
Dr. J. E. Rayas-Sánchez 21

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



Dr. J. E. Rayas-Sánchez 22

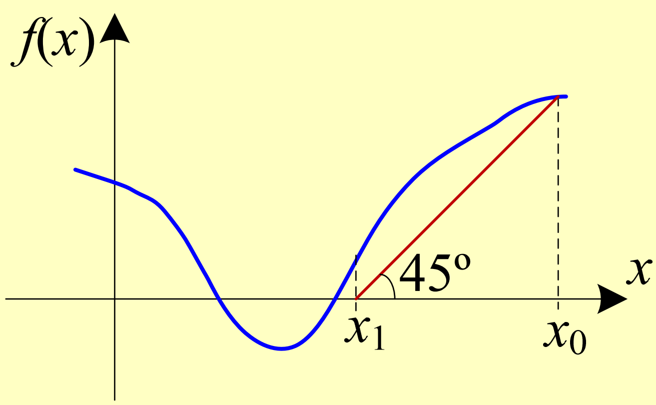
Broyden's Algorithm: Method of Secants (cont.)



Dr. J. E. Rayas-Sánchez

23

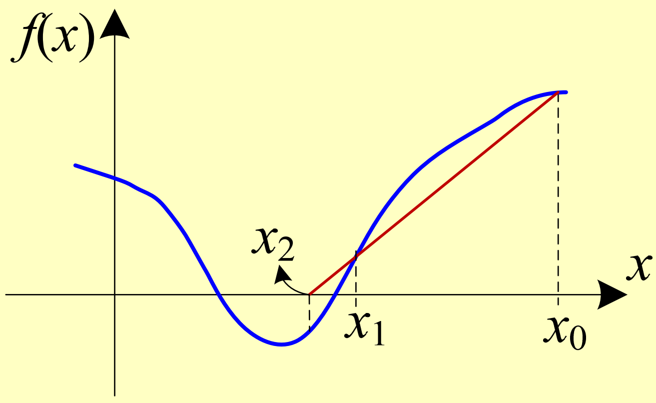
Broyden's Algorithm: Method of Secants (cont.)



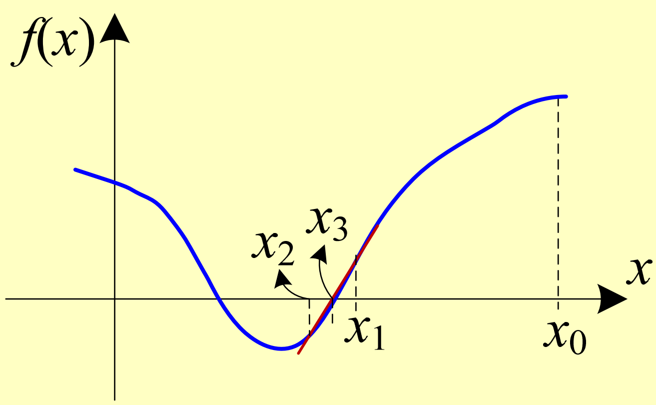
Dr. J. E. Rayas-Sánchez

24

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



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



Stopping Criteria for Solving SNLEq

- A maximum number of iterations has been reached

$$i > i_{\max}$$

- A solution has been found

$$\boldsymbol{f}(\boldsymbol{x}_{i+1})^T \boldsymbol{f}(\boldsymbol{x}_{i+1}) < \varepsilon_1 \quad \text{or} \quad \|\boldsymbol{f}(\boldsymbol{x}_{i+1})\|_\infty < \varepsilon_2$$

- The absolute change in the optimization variables is small enough

$$\| \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \|_2 < \varepsilon_3$$

- The relative change in the optimization variables is small enough

$$\| \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \|_2 < \varepsilon_4 (\| \boldsymbol{x}_i \|_2 + \varepsilon_5)$$