L4 Nonlinear Equation Systems

02.11.2015

Oscar Ramirez

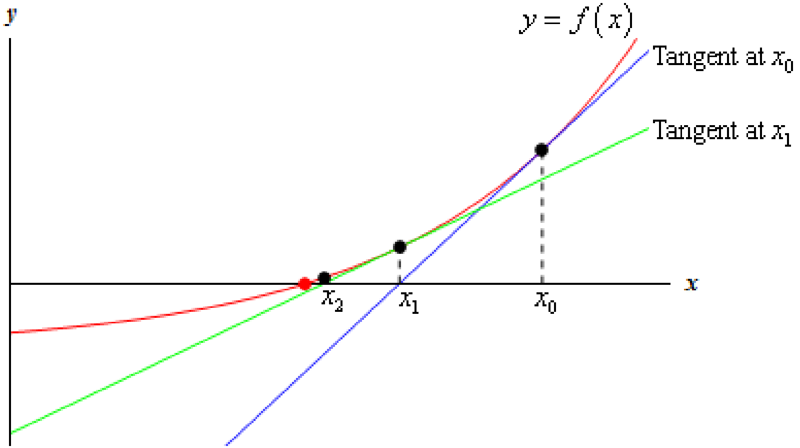
Newton's Method (5.5.3)

Algorithm 1 Newton's Method One Dimension (5.2)

$x_0 =$ initial guess;
for $k = 0, 1, 2, \ldots$ **do**
    $x_{k+1} = x_k - f(x_k)/f'(x_k)$
end for

- Iterative method that produces increasingly accurate approximations to a solution
- Methods differ mostly on their convergence rates

Newton's Method



Newton's Method (5.6.2)

Algorithm 2 Newton's Method for a System of NE (5.4)

$\mathbf{x}_0$ = initial guess;
for $k = 0, 1, 2, \ldots$ **do**
  Solve $\mathbf{J}_f(\mathbf{x}_k)\mathbf{s}_k = -\mathbf{f}(\mathbf{x}_k)$ for $\mathbf{s}_k$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$
end for

- $\mathbf{J}$ is the Jacobian matrix of $\mathbf{f}$

$$J = \begin{bmatrix} \dfrac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \dfrac{\partial f_1(\boldsymbol{x})}{\partial x_m} \\ \vdots & & \vdots \\ \dfrac{\partial f_k(\boldsymbol{x})}{\partial x_1} & \cdots & \dfrac{\partial f_k(\boldsymbol{x})}{\partial x_m} \end{bmatrix}$$

Broyden's Method (5.5)

Algorithm 3 Broyden's Method

$\mathbf{x}_0 =$ initial guess;
$\mathbf{B}_0 =$ initial Jacobian approximation;
for $k = 0, 1, 2, \ldots$ **do**
    Solve $\mathbf{B}_k \mathbf{s}_k = -\mathbf{f}(\mathbf{x}_k)$ for $\mathbf{s}_k$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$
    $\mathbf{y}_k = \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_x)$
    $\mathbf{B}_{k+1} = \mathbf{B}_k + ((\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k)\mathbf{s}_k^T)/\mathbf{s}_k^T \mathbf{s}_k)$
end for

Preparing for lab Assignment

```
addpath '/cshome/vis/data'
robot3D('new')
```

Exercises

Implement the Newton and Broyden methods and make sure they are working. Use the data from Example 5.15 and 5.16.

References

[1] Heath, M. T.
Scientific computing.
McGraw-Hill, 2001.