Solving Nonlinear Equations

Carlos Hurtado

Department of Economics
University of Illinois at Urbana-Champaign
hrtdmrt2@illinois.edu

Nov 2nd, 2017

On the Agenda

1. System of Nonlinear Equations
2. Nonlinear Solvers
3. In Practice

On the Agenda

System of Nonlinear Equations

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as being nonlinear when it does not satisfy the superposition principle:

$$f(x_1 + \cdots + x_n) \neq f(x_1) + \cdots + f(x_n)$$

- Now that we know what the term nonlinear refers to we can define a system of nonlinear equations.
- A system of nonlinear equations is a set of equations as the following:

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_n) &= 0, \\ f_2(x_1, x_2, \cdots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, x_2, \cdots, x_n) &= 0 \end{aligned}$$

where $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ and each f_i is a nonlinear function, for $i = 1, 2, \cdots, n$.

System of Nonlinear Equations

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as being nonlinear when it does not satisfy the superposition principle:

$$f(x_1 + \cdots + x_n) \neq f(x_1) + \cdots + f(x_n)$$

- Now that we know what the term nonlinear refers to we can define a system of nonlinear equations.
- A system of nonlinear equations is a set of equations as the following:

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_n) &= 0, \\ f_2(x_1, x_2, \cdots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, x_2, \cdots, x_n) &= 0 \end{aligned}$$

where $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ and each f_i is a nonlinear function, for $i = 1, 2, \cdots, n$.

System of Nonlinear Equations

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as being nonlinear when it does not satisfy the superposition principle:

$$f(x_1 + \cdots + x_n) \neq f(x_1) + \cdots + f(x_n)$$

- Now that we know what the term nonlinear refers to we can define a system of nonlinear equations.
- A system of nonlinear equations is a set of equations as the following:

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_n) &= 0, \\ f_2(x_1, x_2, \cdots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, x_2, \cdots, x_n) &= 0 \end{aligned}$$

where $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ and each f_i is a nonlinear function, for $i = 1, 2, \cdots, n$.

System of Nonlinear Equations

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as being nonlinear when it does not satisfy the superposition principle:

$$f(x_1 + \cdots + x_n) \neq f(x_1) + \cdots + f(x_n)$$

- Now that we know what the term nonlinear refers to we can define a system of nonlinear equations.
- A system of nonlinear equations is a set of equations as the following:

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_n) &= 0, \\ f_2(x_1, x_2, \cdots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, x_2, \cdots, x_n) &= 0 \end{aligned}$$

where $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ and each f_i is a nonlinear function, for $i = 1, 2, \cdots, n$.

System of Nonlinear Equations

- Example of a nonlinear system:

$$3x_1 - \cos(x_2 x_3) - \frac{1}{2} = 0$$
$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1 = 0$$
$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi}{3} = 0$$

- The terms root or solution are frequently use to describe the final result of solving the systems of equations.
- The root or solution is a point $a = (a_1, a_2, \cdots, a_n) \in \mathbb{R}^n$ such that $f_1(a) = f_2(a) = \cdots = f_n(a) = 0$

System of Nonlinear Equations

- Example of a nonlinear system:

$$
\begin{aligned}
3x_1 - \cos(x_2 x_3) - \frac{1}{2} &= 0 \\
x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1 &= 0 \\
e^{-x_1 x_2} + 20x_3 + \frac{10\pi}{3} &= 0
\end{aligned}
$$

- The terms root or solution are frequently use to describe the final result of solving the systems of equations.
- The root or solution is a point $a = (a_1, a_2, \cdots, a_n) \in \mathbb{R}^n$ such that $f_1(a) = f_2(a) = \cdots = f_n(a) = 0$

System of Nonlinear Equations

- Example of a nonlinear system:

$$
\begin{aligned}
3x_1 - \cos(x_2 x_3) - \frac{1}{2} &= 0 \\
x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1 &= 0 \\
e^{-x_1 x_2} + 20x_3 + \frac{10\pi}{3} &= 0
\end{aligned}
$$

- The terms root or solution are frequently use to describe the final result of solving the systems of equations.

- The root or solution is a point $a = (a_1, a_2, \cdots, a_n) \in \mathbb{R}^n$ such that $f_1(a) = f_2(a) = \cdots = f_n(a) = 0$

System of Nonlinear Equations

- Example of a nonlinear system:

$$3x_1 - \cos(x_2 x_3) - \frac{1}{2} = 0$$
$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1 = 0$$
$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi}{3} = 0$$

- The terms root or solution are frequently use to describe the final result of solving the systems of equations.

- The root or solution is a point $a = (a_1, a_2, \cdots, a_n) \in \mathbb{R}^n$ such that $f_1(a) = f_2(a) = \cdots = f_n(a) = 0$

Fundamentals

- Simplest problem: Root finding in one dimension.

$$f(x) = 0 \text{ with } x \in [a, b]$$

- More generally, solving a square system of nonlinear equations.

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- There can be no closed-form answer, we need iterative methods.
- Iterative methods: initial guess x_0 and update function ϕ

$$x_{k+1} = \phi(x_k)$$

Fundamentals

- Simplest problem: Root finding in one dimension.

$$f(x) = 0 \text{ with } x \in [a, b]$$

- More generally, solving a square system of nonlinear equations.

$$\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- There can be no closed-form answer, we need iterative methods.
- Iterative methods: initial guess x_0 and update function ϕ

$$x_{k+1} = \phi(x_k)$$

Fundamentals

- Simplest problem: Root finding in one dimension.

$$f(x) = 0 \text{ with } x \in [a, b]$$

- More generally, solving a square system of nonlinear equations.

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- There can be no closed-form answer, we need iterative methods.
- Iterative methods: initial guess x_0 and update function ϕ

$$x_{k+1} = \phi(x_k)$$

Fundamentals

- Simplest problem: Root finding in one dimension.

$$f(x) = 0 \text{ with } x \in [a, b]$$

- More generally, solving a square system of nonlinear equations.

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- There can be no closed-form answer, we need iterative methods.
- Iterative methods: initial guess x_0 and update function ϕ

$$x_{k+1} = \phi(x_k)$$

Fundamentals

- A sequence of iterates x_k that converges to α has order of convergence $p > 0$ if

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = C$$

  where the constant $0 < C < 1$ is the convergence factor.

- A method should at least converge linearly, that is, the error should at least be reduced by a constant factor every iteration, for example, the number of accurate digits increases by 1 every iteration.

- A good iterative method coverges quadratically, that is, the number of accurate digits doubles every iteration!

Fundamentals

- A sequence of iterates x_k that converges to α has order of convergence $p > 0$ if

$$\lim_{k \to \infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = C$$

  where the constant $0 < C < 1$ is the convergence factor.

- A method should at least converge linearly, that is, the error should at least be reduced by a constant factor every iteration, for example, the number of accurate digits increases by 1 every iteration.

- A good iterative method coverges quadratically, that is, the number of accurate digits doubles every iteration!

Fundamentals

- A sequence of iterates x_k that converges to α has order of convergence $p > 0$ if

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = C$$

  where the constant $0 < C < 1$ is the convergence factor.

- A method should at least converge linearly, that is, the error should at least be reduced by a constant factor every iteration, for example, the number of accurate digits increases by 1 every iteration.

- A good iterative method coverges quadratically, that is, the number of accurate digits doubles every iteration!

On the Agenda

Nonlinear Solvers

- A good initial guess is extremely important in nonlinear solvers!
- Assume we are looking for a unique root $a \leq \alpha \leq b$ starting with an initial guess $a \leq x_0 \leq b$.
- A method has local convergence if it converges to a given root α for any initial guess that is sufficiently close to α (in the neighborhood of a root).
- A method has global convergence if it converges to the root for any initial guess.
- General rule: Global convergence requires a slower method but is safer.
- It is best to combine a global method to first find a good initial guess close to α and then use a faster local method.

Nonlinear Solvers

- A good initial guess is extremely important in nonlinear solvers!
- Assume we are looking for a unique root $a \leq \alpha \leq b$ starting with an initial guess $a \leq x_0 \leq b$.
- A method has local convergence if it converges to a given root α for any initial guess that is sufficiently close to α (in the neighborhood of a root).
- A method has global convergence if it converges to the root for any initial guess.
- General rule: Global convergence requires a slower method but is safer.
- It is best to combine a global method to first find a good initial guess close to α and then use a faster local method.

Nonlinear Solvers

- A good initial guess is extremely important in nonlinear solvers!
- Assume we are looking for a unique root $a \leq \alpha \leq b$ starting with an initial guess $a \leq x_0 \leq b$.
- A method has local convergence if it converges to a given root α for any initial guess that is sufficiently close to α (in the neighborhood of a root).
- A method has global convergence if it converges to the root for any initial guess.
- General rule: Global convergence requires a slower method but is safer.
- It is best to combine a global method to first find a good initial guess close to α and then use a faster local method.

Nonlinear Solvers

- A good initial guess is extremely important in nonlinear solvers!
- Assume we are looking for a unique root $a \leq \alpha \leq b$ starting with an initial guess $a \leq x_0 \leq b$.
- A method has local convergence if it converges to a given root α for any initial guess that is sufficiently close to α (in the neighborhood of a root).
- A method has global convergence if it converges to the root for any initial guess.
- General rule: Global convergence requires a slower method but is safer.
- It is best to combine a global method to first find a good initial guess close to α and then use a faster local method.

Nonlinear Solvers

- A good initial guess is extremely important in nonlinear solvers!
- Assume we are looking for a unique root $a \leq \alpha \leq b$ starting with an initial guess $a \leq x_0 \leq b$.
- A method has local convergence if it converges to a given root α for any initial guess that is sufficiently close to α (in the neighborhood of a root).
- A method has global convergence if it converges to the root for any initial guess.
- General rule: Global convergence requires a slower method but is safer.
- It is best to combine a global method to first find a good initial guess close to α and then use a faster local method.

Nonlinear Solvers

- A good initial guess is extremely important in nonlinear solvers!
- Assume we are looking for a unique root $a \leq \alpha \leq b$ starting with an initial guess $a \leq x_0 \leq b$.
- A method has local convergence if it converges to a given root α for any initial guess that is sufficiently close to α (in the neighborhood of a root).
- A method has global convergence if it converges to the root for any initial guess.
- General rule: Global convergence requires a slower method but is safer.
- It is best to combine a global method to first find a good initial guess close to α and then use a faster local method.

Nonlinear Solvers

- Let us start with $f : \mathbb{R} \to \mathbb{R}$, and the interval $[a,b]$, such that $f(a)$ and $f(b)$ have opposite sign. (How?)
- We want to find $f(x) = 0$ (The roots).
- We know how to solve this using Newton's method (At this point you should!)
- Consider the point $m = \frac{a+b}{2}$
- Check the intervals $[a,\ m]$ and $[m,b]$ to see if those have opposite sign.
- Repeat the algorithm with interval that has opposite signs
- This is called the bisection method: Global Convergence but Slow.

Nonlinear Solvers

- Let us start with $f : \mathbb{R} \to \mathbb{R}$, and the interval $[a,b]$, such that $f(a)$ and $f(b)$ have opposite sign. (How?)
- We want to find $f(x) = 0$ (The roots).
- We know how to solve this using Newton's method (At this point you should!)
- Consider the point $m = \frac{a+b}{2}$
- Check the intervals $[a,\ m]$ and $[m,b]$ to see if those have opposite sign.
- Repeat the algorithm with interval that has opposite signs
- This is called the bisection method: Global Convergence but Slow.

Nonlinear Solvers

- Let us start with $f : \mathbb{R} \to \mathbb{R}$, and the interval $[a,b]$, such that $f(a)$ and $f(b)$ have opposite sign. (How?)
- We want to find $f(x) = 0$ (The roots).
- We know how to solve this using Newton's method (At this point you should!)
- Consider the point $m = \frac{a+b}{2}$
- Check the intervals $[a,\ m]$ and $[m,b]$ to see if those have opposite sign.
- Repeat the algorithm with interval that has opposite signs
- This is called the bisection method: Global Convergence but Slow.

Nonlinear Solvers

- Let us start with $f : \mathbb{R} \rightarrow \mathbb{R}$, and the interval $[a,b]$, such that $f(a)$ and $f(b)$ have opposite sign. (How?)
- We want to find $f(x) = 0$ (The roots).
- We know how to solve this using Newton's method (At this point you should!)
- Consider the point $m = \frac{a+b}{2}$
- Check the intervals $[a,\ m]$ and $[m,b]$ to see if those have opposite sign.
- Repeat the algorithm with interval that has opposite signs
- This is called the bisection method: Global Convergence but Slow.

Nonlinear Solvers

- Let us start with $f : \mathbb{R} \to \mathbb{R}$, and the interval $[a,b]$, such that $f(a)$ and $f(b)$ have opposite sign. (How?)
- We want to find $f(x) = 0$ (The roots).
- We know how to solve this using Newton's method (At this point you should!)
- Consider the point $m = \frac{a+b}{2}$
- Check the intervals $[a,\ m]$ and $[m,b]$ to see if those have opposite sign.
- Repeat the algorithm with interval that has opposite signs
- This is called the bisection method: Global Convergence but Slow.

Nonlinear Solvers

- Let us start with $f : \mathbb{R} \to \mathbb{R}$, and the interval $[a,b]$, such that $f(a)$ and $f(b)$ have opposite sign. (How?)
- We want to find $f(x) = 0$ (The roots).
- We know how to solve this using Newton's method (At this point you should!)
- Consider the point $m = \frac{a+b}{2}$
- Check the intervals $[a,\ m]$ and $[m,b]$ to see if those have opposite sign.
- Repeat the algorithm with interval that has opposite signs
- This is called the bisection method: Global Convergence but Slow.

Nonlinear Solvers

- Let us start with $f : \mathbb{R} \to \mathbb{R}$, and the interval $[a,b]$, such that $f(a)$ and $f(b)$ have opposite sign. (How?)
- We want to find $f(x) = 0$ (The roots).
- We know how to solve this using Newton's method (At this point you should!)
- Consider the point $m = \frac{a+b}{2}$
- Check the intervals $[a,\ m]$ and $[m,b]$ to see if those have opposite sign.
- Repeat the algorithm with interval that has opposite signs
- This is called the bisection method: Global Convergence but Slow.

Nonlinear Solvers

- In practice, a robust but fast algorithm for root finding would combine bisection with Newton's method.
- Specifically, a method like Newton's that can easily take huge steps in the wrong direction and lead far from the current point must be safeguarded by a method that ensures one does not leave the search interval and that the zero is not missed.
- Once x_k is close to α, the safeguard will not be used and quadratic or faster convergence will be achieved.
- Newton's method requires first-order derivatives so often other methods are preferred that require function evaluation only (or approximations of the derivatives).

Nonlinear Solvers

- In practice, a robust but fast algorithm for root finding would combine bisection with Newton's method.
- Specifically, a method like Newton's that can easily take huge steps in the wrong direction and lead far from the current point must be safeguarded by a method that ensures one does not leave the search interval and that the zero is not missed.
- Once x_k is close to α, the safeguard will not be used and quadratic or faster convergence will be achieved.
- Newton's method requires first-order derivatives so often other methods are preferred that require function evaluation only (or approximations of the derivatives).

Nonlinear Solvers

- In practice, a robust but fast algorithm for root finding would combine bisection with Newton's method.
- Specifically, a method like Newton's that can easily take huge steps in the wrong direction and lead far from the current point must be safeguarded by a method that ensures one does not leave the search interval and that the zero is not missed.
- Once x_k is close to α, the safeguard will not be used and quadratic or faster convergence will be achieved.
- Newton's method requires first-order derivatives so often other methods are preferred that require function evaluation only (or approximations of the derivatives).

Nonlinear Solvers

- In practice, a robust but fast algorithm for root finding would combine bisection with Newton's method.
- Specifically, a method like Newton's that can easily take huge steps in the wrong direction and lead far from the current point must be safeguarded by a method that ensures one does not leave the search interval and that the zero is not missed.
- Once x_k is close to α, the safeguard will not be used and quadratic or faster convergence will be achieved.
- Newton's method requires first-order derivatives so often other methods are preferred that require function evaluation only (or approximations of the derivatives).

Nonlinear Solvers

- What happens if we want to solve a square system of nonlinear equations?

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- We can use Newton's method where $f : \mathbb{R}^n \to \mathbb{R}^n$.
- This requires solving many linear systems, which can become complicated when there are many variables.
- It also requires computing a whole matrix of derivatives, which can be expensive or hard to do (differentiation by hand?)
- Newton's method converges fast if the Jacobian is well-conditioned, otherwise it can "blow up"
- Normally use quasi-Newton methods to approximate the Jacobian.

Nonlinear Solvers

- What happens if we want to solve a square system of nonlinear equations?

$$\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- We can use Newton's method where $f : \mathbb{R}^n \to \mathbb{R}^n$.
- This requires solving many linear systems, which can become complicated when there are many variables.
- It also requires computing a whole matrix of derivatives, which can be expensive or hard to do (differentiation by hand?)
- Newton's method converges fast if the Jacobian is well-conditioned, otherwise it can "blow up"
- Normally use quasi-Newton methods to approximate the Jacobian.

Nonlinear Solvers

- What happens if we want to solve a square system of nonlinear equations?

$$\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- We can use Newton's method where $f : \mathbb{R}^n \to \mathbb{R}^n$.
- This requires solving many linear systems, which can become complicated when there are many variables.
- It also requires computing a whole matrix of derivatives, which can be expensive or hard to do (differentiation by hand?)
- Newton's method converges fast if the Jacobian is well-conditioned, otherwise it can "blow up"
- Normally use quasi-Newton methods to approximate the Jacobian.

Nonlinear Solvers

- What happens if we want to solve a square system of nonlinear equations?

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- We can use Newton's method where $f : \mathbb{R}^n \to \mathbb{R}^n$.
- This requires solving many linear systems, which can become complicated when there are many variables.
- It also requires computing a whole matrix of derivatives, which can be expensive or hard to do (differentiation by hand?)
- Newton's method converges fast if the Jacobian is well-conditioned, otherwise it can "blow up"
- Normally use quasi-Newton methods to approximate the Jacobian.

Nonlinear Solvers

- What happens if we want to solve a square system of nonlinear equations?

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \implies f_i(x_1, x_2, \cdots, x_n) = 0 \text{ for } i = 1, \cdots, n$$

- We can use Newton's method where $f : \mathbb{R}^n \to \mathbb{R}^n$.
- This requires solving many linear systems, which can become complicated when there are many variables.
- It also requires computing a whole matrix of derivatives, which can be expensive or hard to do (differentiation by hand?)
- Newton's method converges fast if the Jacobian is well-conditioned, otherwise it can "blow up"
- Normally use quasi-Newton methods to approximate the Jacobian.

On the Agenda

In Practice

- It is much harder to construct general robust solvers in higher dimensions and some problem-specific knowledge is required.
- In python we can use fsolve function from scipy.optimize
- In many practical situations there is some continuity of the problem so that a previous solution can be used as an initial guess.
- For large problems specialized sparse-matrix solvers need to be used.
- We can think of this as an optimization problem! (How?)

In Practice

- It is much harder to construct general robust solvers in higher dimensions and some problem-specific knowledge is required.
- In python we can use fsolve function from scipy.optimize
- In many practical situations there is some continuity of the problem so that a previous solution can be used as an initial guess.
- For large problems specialized sparse-matrix solvers need to be used.
- We can think of this as an optimization problem! (How?)

In Practice

- It is much harder to construct general robust solvers in higher dimensions and some problem-specific knowledge is required.
- In python we can use fsolve function from scipy.optimize
- In many practical situations there is some continuity of the problem so that a previous solution can be used as an initial guess.
- For large problems specialized sparse-matrix solvers need to be used.
- We can think of this as an optimization problem! (How?)

In Practice

- It is much harder to construct general robust solvers in higher dimensions and some problem-specific knowledge is required.
- In python we can use fsolve function from scipy.optimize
- In many practical situations there is some continuity of the problem so that a previous solution can be used as an initial guess.
- For large problems specialized sparse-matrix solvers need to be used.
- We can think of this as an optimization problem! (How?)

In Practice

- It is much harder to construct general robust solvers in higher dimensions and some problem-specific knowledge is required.
- In python we can use fsolve function from scipy.optimize
- In many practical situations there is some continuity of the problem so that a previous solution can be used as an initial guess.
- For large problems specialized sparse-matrix solvers need to be used.
- We can think of this as an optimization problem! (How?)