Iterative Methods for Systems

1 Iterative Methods for Fixed Points of Systems

The problem of solving a system of nonlinear equations is ubiquitous in scientific computing. It arises in fields from economics (general equilibrium models) to engineering (finite element analysis of nonlinear structures). We often wish to find a vector $\mathbf{x}^* \in \mathbb{R}^n$ such that:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{1}$$

Any system of the form (2) can be rewritten as a **fixed-point problem**:

$$\mathbf{x} = \mathbf{g}(\mathbf{x})$$

where $\mathbf{g} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function. A point $\mathbf{x}^*$ satisfying this equation is called a **fixed point** of the function $\mathbf{g}$.

1.1 Fixed-Point Iteration for Systems

The most straightforward iterative method for finding $\mathbf{x}^*$ is a direct generalization of the one-dimensional case.

Definition 1 (Fixed-Point Iteration)**.** *Given an initial approximation $\mathbf{x}^{(0)} \in \mathbb{R}^n$, generate a sequence of vectors $\{\mathbf{x}^{(k)}\}$ for $k = 0, 1, 2, \ldots$ using the iteration:*

$$\mathbf{x}^{(k+1)} = \mathbf{g}(\mathbf{x}^{(k)})$$

which, in component form, is:

$$\begin{aligned} x_1^{(k+1)} &= g_1(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \\ x_2^{(k+1)} &= g_2(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \\ &\vdots \\ x_n^{(k+1)} &= g_n(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \end{aligned}$$

1.2 Convergence Analysis

The convergence of the fixed-point iteration in $\mathbb{R}^n$ is governed by a generalization of the derivative condition from the one-dimensional case.

Theorem 1 (Banach Fixed-Point Theorem for $\mathbb{R}^n$)**.** *Let $D \subset \mathbb{R}^n$ be a closed set. Suppose $\mathbf{g} : D \to D$ is a function on D and that $\mathbf{g}$ is a **contraction mapping** on D, i.e., there exists a constant L with $0 \leq L < 1$ such that:*

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\| \quad \text{for all } \mathbf{x}, \mathbf{y} \in D.$$

Then:

(i) $\mathbf{g}$ *has a **unique** fixed point $\mathbf{x}^*$ in D.*

(ii) For any initial guess $\mathbf{x}^{(0)} \in D$, the sequence generated by $\mathbf{x}^{(k+1)} = \mathbf{g}(\mathbf{x}^{(k)})$ converges to $\mathbf{x}^*$.

(iii) The following error bounds hold:

$$\left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\| \leq \frac{L^k}{1-L} \left\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right\| \quad \textit{(A priori estimate)}$$

$$\left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\| \leq \frac{L}{1-L} \left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\| \quad \textit{(A posteriori estimate)}$$

1.2.1 Checking the Contraction Condition

In practice, the contraction condition is often verified using the Jacobian matrix.

Definition 2 (Jacobian Matrix)**.** *The Jacobian matrix $J_{\mathbf{g}}(\mathbf{x})$ of the function $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}))^T$ is defined as:*

$$J_{\mathbf{g}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}$$

Theorem 2. *Let $D \subset \mathbb{R}^n$ be a convex, closed set. Suppose $\mathbf{g}$ is continuously differentiable on D and that there exists a constant $L < 1$ such that:*

$$\|J_{\mathbf{g}}(\mathbf{x})\| \leq L \quad \text{for all } \mathbf{x} \in D$$

for some induced matrix norm (e.g., the ∞-norm, or Frobenius norm). Then $\mathbf{g}$ is a contraction mapping on D.

1.3 Example

Example 1. *Consider the nonlinear system:*

$$\begin{aligned} 3x_1 - \cos(x_2 x_3) - \frac{1}{2} &= 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 &= 0 \\ e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0 \end{aligned}$$

A common fixed-point form for this system is:

$$\begin{aligned} x_1 &= \frac{1}{3}\cos(x_2 x_3) + \frac{1}{6} = g_1(x_1, x_2, x_3) \\ x_2 &= \frac{1}{9}\sqrt{x_1^2 + \sin(x_3) + 1.06} - 0.1 = g_2(x_1, x_2, x_3) \\ x_3 &= -\frac{1}{20}e^{-x_1 x_2} - \frac{10\pi - 3}{60} = g_3(x_1, x_2, x_3) \end{aligned}$$

Starting from an initial guess $\mathbf{x}^{(0)} = (0.1, 0.1, -0.1)^T$*, the fixed-point iteration can be applied. The Jacobian should be analyzed to check for a contraction on a suitable domain* D*.*

1.4 Stopping Criteria

A robust implementation must have a stopping criterion. Common choices for systems include:

- **Absolute Error:** $\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| < \epsilon$
- **Relative Error:** $\dfrac{\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|}{\|\mathbf{x}^{(k)}\| + \delta} < \epsilon$ (δ is a small number to avoid division by zero)
- **Residual:** $\left\|\mathbf{g}(\mathbf{x}^{(k)}) - \mathbf{x}^{(k)}\right\| < \epsilon$ or $\left\|\mathbf{f}(\mathbf{x}^{(k)})\right\| < \epsilon$

1.5 Advantages and Disadvantages

- **Advantages:**
  - Conceptually simple and easy to implement. Low computational cost per iteration (no linear system to solve).
- **Disadvantages:**
  - Often requires a careful reformulation $\mathbf{f}(\mathbf{x}) = \mathbf{0} \Rightarrow \mathbf{x} = \mathbf{g}(\mathbf{x})$ to ensure convergence.
  - Convergence can be very slow (linear convergence, L close to 1).
  - The convergence region might be small.

1.6 Conclusion

Fixed-point iteration for systems is a fundamental method. While its simplicity is attractive, its convergence is not guaranteed unless the function $\mathbf{g}$ is a contraction. This often requires problem-specific knowledge to choose a good reformulation and initial guess. Its linear convergence rate makes it suitable only for problems where low accuracy is sufficient or where the contraction constant L is very small. In practice, it is often superseded by faster methods like Newton's method for systems, which converges quadratically but requires solving a linear system at each step.

2 The Newton-Raphson method

The Newton-Raphson method is one of the most powerful and widely used algorithms for solving systems of nonlinear equations. It is the natural generalization of the Newton method for scalar equations to the n-dimensional case. Given a system of n nonlinear equations with n unknowns:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{2}$$

we can write this system in vector form as:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$ and $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function:

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix}$$

2.1 Derivation of the Method

The derivation follows from the first-order Taylor expansion of $\mathbf{F}(\mathbf{x})$ around a point $\mathbf{x}^{(k)}$:

$$\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x}^{(k)}) + \mathbf{J}(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)})$$

where $\mathbf{J}(\mathbf{x})$ is the **Jacobian matrix** of $\mathbf{F}$ at $\mathbf{x}$.

We want to find $\mathbf{x}$ such that $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. Setting the approximation to zero:

$$\mathbf{F}(\mathbf{x}^{(k)}) + \mathbf{J}(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) = \mathbf{0}$$

Solving for $\mathbf{x}$ gives the next approximation $\mathbf{x}^{(k+1)}$:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [\mathbf{J}(\mathbf{x}^{(k)})]^{-1}\mathbf{F}(\mathbf{x}^{(k)})$$

In practice, we avoid explicitly calculating the inverse. Instead, we solve the linear system:

$$\mathbf{J}(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$$

for the **step** or **update vector** $\mathbf{p}^{(k)}$, and then update:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{p}^{(k)}$$

Definition 3 (Jacobian Matrix)**.** *The Jacobian matrix* $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{n\times n}$ *is defined as:*

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

2.1.1 Mathematical Formulation

1. Choose an initial guess $\mathbf{x}^{(0)} \in \mathbb{R}^n$.
2. For $k = 0, 1, 2, \ldots$ until convergence:
   a) Evaluate the function vector $\mathbf{F}(\mathbf{x}^{(k)})$ and the Jacobian matrix $\mathbf{J}(\mathbf{x}^{(k)})$.
   b) Solve the linear system $\mathbf{J}(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$ for $\mathbf{p}^{(k)}$.
   c) Update the solution: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{p}^{(k)}$.
   d) Check for convergence.

2.2 Convergence Analysis

Theorem 3 (Convergence of Newton's Method)**.** *Let* $\mathbf{F} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ *be continuously differentiable in an open convex set* D*. Suppose there exists a solution* $\mathbf{x}^* \in D$ *such that* $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$ *and that* $\mathbf{J}(\mathbf{x}^*)$ *is nonsingular. Then, if the initial guess* $\mathbf{x}^{(0)}$ *is sufficiently close to* $\mathbf{x}^*$*, the Newton-Raphson iterates converge to* $\mathbf{x}^*$ *and satisfy:*

$$\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\right\| \leq C \left\|\mathbf{x}^{(k)} - \mathbf{x}^*\right\|^2$$

for some constant $C > 0$*. That is, the method converges* ***quadratically****.*

2.3 Practical Considerations

2.3.1 Advantages

- **Quadratic convergence**: Very fast when near the root.
- **Direct generalization**: Works for any number of dimensions.

2.3.2 Disadvantages and Challenges

- **Jacobian required**: Requires analytical derivation and coding of the Jacobian matrix.
- **Computational cost**: Each iteration requires:
  - $O(n^2)$ evaluations of derivatives (for the Jacobian).
  - $O(n^3)$ operations to solve the linear system (using Gaussian elimination).
- **Initial guess sensitivity**: Convergence is only guaranteed if the initial guess is sufficiently close to the solution.
- **Singular Jacobian**: The method fails if $\mathbf{J}(\mathbf{x}^{(k)})$ becomes singular at any iteration.

2.3.3 Modifications

To address these challenges, several modifications exist:

- **Inaccurate Jacobian**: Use a finite difference approximation (Broyden's method, etc.).
- **Line search**: Damp the update $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)}\mathbf{p}^{(k)}$ with a step length $0 < \lambda^{(k)} \leq 1$ to ensure $\left\|\mathbf{F}(\mathbf{x}^{(k+1)})\right\| < \left\|\mathbf{F}(\mathbf{x}^{(k)})\right\|$.
- **Hybrid methods**: Combine with a slower but globally convergent method (e.g., steepest descent).

3 Example

Example 2. *Solve the nonlinear system:*

$$x_1^2 + x_2^2 - 4 = 0$$
$$e^{x_1} + x_2 - 1 = 0$$

Here, $\mathbf{F}(\mathbf{x}) = \begin{pmatrix} x_1^2 + x_2^2 - 4 \\ e^{x_1} + x_2 - 1 \end{pmatrix}$.

The Jacobian matrix is:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 & 2x_2 \\ e^{x_1} & 1 \end{bmatrix}$$

Choose initial guess $\mathbf{x}^{(0)} = (1.0, 1.0)^T$.

Iteration 1:

$$\mathbf{F}(\mathbf{x}^{(0)}) = \begin{pmatrix} (1)^2 + (1)^2 - 4 \\ e^1 + 1 - 1 \end{pmatrix} = \begin{pmatrix} -2 \\ e \end{pmatrix} \approx \begin{pmatrix} -2 \\ 2.7183 \end{pmatrix}$$

$$\mathbf{J}(\mathbf{x}^{(0)}) = \begin{bmatrix} 2(1) & 2(1) \\ e^1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2.7183 & 1 \end{bmatrix}$$

Solve $\begin{bmatrix} 2 & 2 \\ 2.7183 & 1 \end{bmatrix} \mathbf{p}^{(0)} = - \begin{pmatrix} -2 \\ 2.7183 \end{pmatrix}$*:*

$$\mathbf{p}^{(0)} \approx \begin{pmatrix} 0.5 \\ -1.5 \end{pmatrix}, \quad \mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{p}^{(0)} = \begin{pmatrix} 1.5 \\ -0.5 \end{pmatrix}$$

Continue iterating until $\left\|\mathbf{p}^{(k)}\right\|$ *is sufficiently small.*

3.1 Conclusion

The Newton-Raphson method for systems is a cornerstone of computational mathematics. Its quadratic convergence makes it exceptionally powerful, but its effectiveness depends critically on a good initial guess and the ability to compute and invert the Jacobian matrix. Understanding its derivation, implementation, and limitations is essential for solving complex nonlinear problems in science and engineering.