JDEP 384H: Numerical Methods in Business

Instructor: Thomas Shores
Department of Mathematics

Lecture 18, February 22, 2007
110 Kaufmann Center

Outline

Outline

1. BT 3.1: Basics of Numerical Analysis
   - Finite Precision Representation
   - Error Analysis
2. BT 3.2: Linear Systems
   - Direct Methods
   - Iterative Methods

Outline

1. BT 3.1: Basics of Numerical Analysis
   - Finite Precision Representation
   - Error Analysis
2. BT 3.2: Linear Systems
   - Direct Methods
   - Iterative Methods

Examples

Example

(Example 7 of NumericalAnalysisNotes) Let $p_n = 1/3^n$, $n = 0, 1, 2, \ldots$. This sequence obeys the rule $p_{n+1} = p_{n-1} - \frac{8}{3}p_n$ with $p_0 = 1$ and $p_1 = 1/3$. Similarly, we see that $p_{n+1} = \frac{1}{3}p_n$ with $p_0 = 1$. Use Matlab to plot the sequence $\{p_n\}_{n=0}^{50}$ directly, and then using the above recursion algorithms with p_0 and p_1 given and overlay the plot of those results. Repeat the plot with the last 11 of the points.

```
>N=50
>p1 = (1/3).^(0:N);
>p2 = p1; p3 = p1;
>for n = 1:N,p2(n+1) = (1/3)*p2(n);end
>for n = 2:N,p3(n+1) = p3(n-1)-8/3*p3(n);end
>plot([p1',p2',p3'])
>plot([p1(N-11:N)',p2(N-11:N)',p3(N-11:N)'])
```

Outline

Direct Methods

The problem: Solve the $n \times n$ linear system $Ax = b$.

Direct Method:

A well defined finite sequence of algebraic operations that produces the solution (accepting that there may be loss of accuracy due to floating point error and system sensitivity.) Most direct methods rely on reducing system to a triangular system of equations, then back solving.

- The condition number of the coefficient matrix, $\operatorname{cond}(A) = \|A\| \left\|A^{-1}\right\|$ is a good indicator of how sensitive the system is to errors in calculation.
- Fact: If the system $A(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$ is solved instead of $A\mathbf{x} = \mathbf{b}$, then $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \operatorname{cond}(A) \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$
- Heuristic: in solving $A\mathbf{x} = \mathbf{b}$ we can lose as many as $\log 10(\operatorname{cond}(A))$ significant digits.

Direct Methods

The problem: Solve the $n \times n$ linear system $Ax = b$.

Direct Method:

A well defined finite sequence of algebraic operations that produces the solution (accepting that there may be loss of accuracy due to floating point error and system sensitivity.) Most direct methods rely on reducing system to a triangular system of equations, then back solving.

- The condition number of the coefficient matrix, $\mathrm{cond}\,(A) = \|A\| \left\|A^{-1}\right\|$ is a good indicator of how sensitive the system is to errors in calculation.
- Fact: If the system $A(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$ is solved instead of $A\mathbf{x} = \mathbf{b}$, then $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \mathrm{cond}\,(A) \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$
- Heuristic: in solving $A\mathbf{x} = \mathbf{b}$ we can lose as many as $\log 10\,(\mathrm{cond}\,(A))$ significant digits.

Direct Methods

The problem: Solve the $n \times n$ linear system $Ax = b$.

Direct Method:

A well defined finite sequence of algebraic operations that produces the solution (accepting that there may be loss of accuracy due to floating point error and system sensitivity.) Most direct methods rely on reducing system to a triangular system of equations, then back solving.

- The condition number of the coefficient matrix, $\text{cond}(A) = \|A\| \left\|A^{-1}\right\|$is a good indicator of how sensitive the system is to errors in calculation.
- Fact: If the system $A(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$ is solved instead of $A\mathbf{x} = \mathbf{b}$, then $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}(A) \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$
- Heuristic: in solving $A\mathbf{x} = \mathbf{b}$ we can lose as many as $\log 10(\text{cond}(A))$significant digits.

Direct Methods

The problem: Solve the $n \times n$ linear system $Ax = b$.

Direct Method:

A well defined finite sequence of algebraic operations that produces the solution (accepting that there may be loss of accuracy due to floating point error and system sensitivity.) Most direct methods rely on reducing system to a triangular system of equations, then back solving.

- The condition number of the coefficient matrix, $\text{cond}\,(A) = \|A\| \left\|A^{-1}\right\|$is a good indicator of how sensitive the system is to errors in calculation.
- Fact: If the system $A\left(\mathbf{x} + \delta\mathbf{x}\right) = \mathbf{b} + \delta\mathbf{b}$ is solved instead of $A\mathbf{x} = \mathbf{b}$, then $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}\,(A)\,\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$
- Heuristic: in solving $A\mathbf{x} = \mathbf{b}$ we can lose as many as $\log 10\,(\text{cond}\,(A))$significant digits.

Matlab Knows Direct Methods

Example

Set up this system at the board and solve it directly and using Matlab.

$$\begin{aligned} x_1 + x_2 + x_3 &= 4 \\ 2x_1 + 2x_2 - x_3 &= 5 \\ 4x_1 + 6x_2 + 8x_3 &= 24 \end{aligned}$$

Direct Method:

- If the system has a unique solution (which is the only kind of system we are dealing with), then the coefficient matrix is invertible, which is equivalent to having nonzero determinant. Verify this with the above example.
- A more reliable indicator of potential problems (sensitive matrix, nearly singular matrix) is the condition number. Check this example and various Hilbert matrices in Matlab.

Matlab Knows Direct Methods

Example

Set up this system at the board and solve it directly and using Matlab.

$$\begin{aligned} x_1 + x_2 + x_3 &= 4 \\ 2x_1 + 2x_2 - x_3 &= 5 \\ 4x_1 + 6x_2 + 8x_3 &= 24 \end{aligned}$$

Direct Method:

- If the system has a unique solution (which is the only kind of system we are dealing with), then the coefficient matrix is invertible, which is equivalent to having nonzero determinant. Verify this with the above example.
- A more reliable indicator of potential problems (sensitive matrix, nearly singular matrix) is the condition number. Check this example and various Hilbert matrices in Matlab.

Matlab Knows Direct Methods

Example

Set up this system at the board and solve it directly and using Matlab.

$$\begin{aligned} x_1 + x_2 + x_3 &= 4 \\ 2x_1 + 2x_2 - x_3 &= 5 \\ 4x_1 + 6x_2 + 8x_3 &= 24 \end{aligned}$$

Direct Method:

- If the system has a unique solution (which is the only kind of system we are dealing with), then the coefficient matrix is invertible, which is equivalent to having nonzero determinant. Verify this with the above example.
- A more reliable indicator of potential problems (sensitive matrix, nearly singular matrix) is the condition number. Check this example and various Hilbert matrices in Matlab.

Matlab Knows Direct Methods

Example

Set up this system at the board and solve it directly and using Matlab.

$$\begin{aligned} x_1 + x_2 + x_3 &= 4 \\ 2x_1 + 2x_2 - x_3 &= 5 \\ 4x_1 + 6x_2 + 8x_3 &= 24 \end{aligned}$$

Direct Method:

- If the system has a unique solution (which is the only kind of system we are dealing with), then the coefficient matrix is invertible, which is equivalent to having nonzero determinant. Verify this with the above example.
- A more reliable indicator of potential problems (sensitive matrix, nearly singular matrix) is the condition number. Check this example and various Hilbert matrices in Matlab.

Outline

Iterative Methods

Basic Idea:

Rather than compute a solution $\mathbf{x}$ directly, find an easy to compute iteration scheme that yields a sequence $\mathbf{x}^{(k)}$ of approximations that converge to the solution $\mathbf{x}$.

- Most common form is fixed point iteration: put the problem in the form $\mathbf{x} = G(\mathbf{x})$ and then iterate with initial guess $\mathbf{x}^{(0)}$ and iterates $\mathbf{x}^{(k)}$ given by $\mathbf{x}^{(k+1)} = G\left(\mathbf{x}^{(k)}\right)$.
- If the scheme works, the iterative scheme is **convergent**, otherwise it is **divergent**.
- "**Convergent**" means that $\lim_{k\to\infty}\mathbf{x}^{(k)} = \mathbf{x}^*$, the desired solution for which $\mathbf{x}^* = G(\mathbf{x}^*)$.

Iterative Methods

Basic Idea:

Rather than compute a solution $\mathbf{x}$ directly, find an easy to compute iteration scheme that yields a sequence $\mathbf{x}^{(k)}$ of approximations that converge to the solution $\mathbf{x}$.

- Most common form is fixed point iteration: put the problem in the form $\mathbf{x} = G(\mathbf{x})$ and then iterate with initial guess $\mathbf{x}^{(0)}$ and iterates $\mathbf{x}^{(k)}$ given by $\mathbf{x}^{(k+1)} = G\left(\mathbf{x}^{(k)}\right)$.
- If the scheme works, the iterative scheme is **convergent**, otherwise it is **divergent**.
- "**Convergent**" means that $\lim_{k\to\infty}\mathbf{x}^{(k)} = \mathbf{x}^*$, the desired solution for which $\mathbf{x}^* = G(\mathbf{x}^*)$.

Iterative Methods

Basic Idea:

Rather than compute a solution $\mathbf{x}$ directly, find an easy to compute iteration scheme that yields a sequence $\mathbf{x}^{(k)}$ of approximations that converge to the solution $\mathbf{x}$.

- Most common form is fixed point iteration: put the problem in the form $\mathbf{x} = G(\mathbf{x})$ and then iterate with initial guess $\mathbf{x}^{(0)}$ and iterates $\mathbf{x}^{(k)}$ given by $\mathbf{x}^{(k+1)} = G\left(\mathbf{x}^{(k)}\right)$.
- If the scheme works, the iterative scheme is **convergent**, otherwise it is **divergent**.
- "**Convergent**" means that $\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}^*$, the desired solution for which $\mathbf{x}^* = G(\mathbf{x}^*)$.

Iterative Methods

Basic Idea:

Rather than compute a solution $\mathbf{x}$ directly, find an easy to compute iteration scheme that yields a sequence $\mathbf{x}^{(k)}$ of approximations that converge to the solution $\mathbf{x}$.

- Most common form is fixed point iteration: put the problem in the form $\mathbf{x} = G(\mathbf{x})$ and then iterate with initial guess $\mathbf{x}^{(0)}$ and iterates $\mathbf{x}^{(k)}$ given by $\mathbf{x}^{(k+1)} = G\left(\mathbf{x}^{(k)}\right)$.
- If the scheme works, the iterative scheme is **convergent**, otherwise it is **divergent**.
- "**Convergent**" means that $\lim_{k\to\infty}\mathbf{x}^{(k)} = \mathbf{x}^*$, the desired solution for which $\mathbf{x}^* = G(\mathbf{x}^*)$.

Example

Find the (unique) real root to the cubic

$$x^3 - x - 6 = 0.$$

Solution. Try two "splittings":

$$x = G(x) \equiv x^3 - 6 \text{ and } x = G(x) \equiv (6 + x)^{1/3}.$$

These yield iterations of the form

$$x^{(k+1)} = \left(x^{(k)}\right)^3 - 6 \text{ and } x^{(k+1)} = \left(6 + x^{(k)}\right)^{1/3}$$

and are easily done by hand in Matlab. Simply cursor up and repeat the second line indefinitely to to the second one, e.g.:

```
> x = 0 % initial guess
> x = (6+x)^(1/3)
```

Iterative Methods for Linear Systems

Splitting:

A general procedure for developing iterations to solve $A\mathbf{x} = \mathbf{b}$:

- First write $A = B - C$, where solving $B\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$ is easy.
- Rewrite the system as $(B - C)\mathbf{x} = \mathbf{b}$, i.e., $B\mathbf{x} = C\mathbf{x} + \mathbf{b}$.
- Or $\mathbf{x} = B^{-1}(C\mathbf{x} + \mathbf{b}) = B^{-1}C\mathbf{x} + B^{-1}\mathbf{b} = G\mathbf{x} + \mathbf{d}$.
- Now iterate on $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$.
- Notation: **spectral radius** of matrix G is $\rho(G)$, the maximum absolute value of any eigenvalue of G.
- **Key Theorem:** If $\rho(G) < 1$, or $\rho(G) = 1$ with exactly one eigenvalue equal 1 and the others smaller than 1, then the iterative method $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$ is guaranteed to converge; however, if $\rho(G) > 1$, method is guaranteed to diverge for nearly all initial $\mathbf{x}^{(0)}$.

Iterative Methods for Linear Systems

Splitting:

A general procedure for developing iterations to solve $A\mathbf{x} = \mathbf{b}$:

- First write $A = B - C$, where solving $B\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$ is easy.
- Rewrite the system as $(B - C)\mathbf{x} = \mathbf{b}$, i.e., $B\mathbf{x} = C\mathbf{x} + \mathbf{b}$.
- Or $\mathbf{x} = B^{-1}(C\mathbf{x} + \mathbf{b}) = B^{-1}C\mathbf{x} + B^{-1}\mathbf{b} = G\mathbf{x} + \mathbf{d}$.
- Now iterate on $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$.
- Notation: **spectral radius** of matrix G is $\rho(G)$, the maximum absolute value of any eigenvalue of G.
- **Key Theorem:** If $\rho(G) < 1$, or $\rho(G) = 1$ with exactly one eigenvalue equal 1 and the others smaller than 1, then the iterative method $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$ is guaranteed to converge; however, if $\rho(G) > 1$, method is guaranteed to diverge for nearly all initial $\mathbf{x}^{(0)}$.

Iterative Methods for Linear Systems

Splitting:

A general procedure for developing iterations to solve $A\mathbf{x} = \mathbf{b}$:

- First write $A = B - C$, where solving $B\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$ is easy.
- Rewrite the system as $(B - C)\mathbf{x} = \mathbf{b}$, i.e., $B\mathbf{x} = C\mathbf{x} + \mathbf{b}$.
- Or $\mathbf{x} = B^{-1}(C\mathbf{x} + \mathbf{b}) = B^{-1}C\mathbf{x} + B^{-1}\mathbf{b} = G\mathbf{x} + \mathbf{d}$.
- Now iterate on $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$.
- Notation: **spectral radius** of matrix G is $\rho(G)$, the maximum absolute value of any eigenvalue of G.
- **Key Theorem:** If $\rho(G) < 1$, or $\rho(G) = 1$ with exactly one eigenvalue equal 1 and the others smaller than 1, then the iterative method $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$ is guaranteed to converge; however, if $\rho(G) > 1$, method is guaranteed to diverge for nearly all initial $\mathbf{x}^{(0)}$.

Iterative Methods for Linear Systems

Splitting:

A general procedure for developing iterations to solve $A\mathbf{x} = \mathbf{b}$:

- First write $A = B - C$, where solving $B\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$ is easy.
- Rewrite the system as $(B - C)\mathbf{x} = \mathbf{b}$, i.e., $B\mathbf{x} = C\mathbf{x} + \mathbf{b}$.
- Or $\mathbf{x} = B^{-1}(C\mathbf{x} + \mathbf{b}) = B^{-1}C\mathbf{x} + B^{-1}\mathbf{b} = G\mathbf{x} + \mathbf{d}$.
- Now iterate on $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$.
- Notation: **spectral radius** of matrix G is $\rho(G)$, the maximum absolute value of any eigenvalue of G.
- **Key Theorem:** If $\rho(G) < 1$, or $\rho(G) = 1$ with exactly one eigenvalue equal 1 and the others smaller than 1, then the iterative method $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$ is guaranteed to converge; however, if $\rho(G) > 1$, method is guaranteed to diverge for nearly all initial $\mathbf{x}^{(0)}$.

Iterative Methods for Linear Systems

Splitting:

A general procedure for developing iterations to solve $A\mathbf{x} = \mathbf{b}$:

- First write $A = B - C$, where solving $B\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$ is easy.
- Rewrite the system as $(B - C)\mathbf{x} = \mathbf{b}$, i.e., $B\mathbf{x} = C\mathbf{x} + \mathbf{b}$.
- Or $\mathbf{x} = B^{-1}(C\mathbf{x} + \mathbf{b}) = B^{-1}C\mathbf{x} + B^{-1}\mathbf{b} = G\mathbf{x} + \mathbf{d}$.
- Now iterate on $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$.
- Notation: **spectral radius** of matrix G is $\rho(G)$, the maximum absolute value of any eigenvalue of G.
- **Key Theorem:** If $\rho(G) < 1$, or $\rho(G) = 1$ with exactly one eigenvalue equal 1 and the others smaller than 1, then the iterative method $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$ is guaranteed to converge; however, if $\rho(G) > 1$, method is guaranteed to diverge for nearly all initial $\mathbf{x}^{(0)}$.

Iterative Methods for Linear Systems

Splitting:

A general procedure for developing iterations to solve $A\mathbf{x} = \mathbf{b}$:

- First write $A = B - C$, where solving $B\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$ is easy.
- Rewrite the system as $(B - C)\mathbf{x} = \mathbf{b}$, i.e., $B\mathbf{x} = C\mathbf{x} + \mathbf{b}$.
- Or $\mathbf{x} = B^{-1}(C\mathbf{x} + \mathbf{b}) = B^{-1}C\mathbf{x} + B^{-1}\mathbf{b} = G\mathbf{x} + \mathbf{d}$.
- Now iterate on $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$.
- Notation: **spectral radius** of matrix G is $\rho(G)$, the maximum absolute value of any eigenvalue of G.
- **Key Theorem:** If $\rho(G) < 1$, or $\rho(G) = 1$ with exactly one eigenvalue equal 1 and the others smaller than 1, then the iterative method $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$ is guaranteed to converge; however, if $\rho(G) > 1$, method is guaranteed to diverge for nearly all initial $\mathbf{x}^{(0)}$.

Iterative Methods for Linear Systems

Splitting:

A general procedure for developing iterations to solve $A\mathbf{x} = \mathbf{b}$:

- First write $A = B - C$, where solving $B\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$ is easy.
- Rewrite the system as $(B - C)\mathbf{x} = \mathbf{b}$, i.e., $B\mathbf{x} = C\mathbf{x} + \mathbf{b}$.
- Or $\mathbf{x} = B^{-1}(C\mathbf{x} + \mathbf{b}) = B^{-1}C\mathbf{x} + B^{-1}\mathbf{b} = G\mathbf{x} + \mathbf{d}$.
- Now iterate on $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$.
- Notation: **spectral radius** of matrix G is $\rho(G)$, the maximum absolute value of any eigenvalue of G.
- **Key Theorem:** If $\rho(G) < 1$, or $\rho(G) = 1$ with exactly one eigenvalue equal 1 and the others smaller than 1, then the iterative method $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$ is guaranteed to converge; however, if $\rho(G) > 1$, method is guaranteed to diverge for nearly all initial $\mathbf{x}^{(0)}$.

Examples

Some Classical Splittings:

- Write $A = L(ower) + D(iagonal) + U(pper)$
- Jacobi: $D\mathbf{x} = -(L+U)\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -D^{-1}(L+U)\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$.
- Gauss-Seidel: $(L+D)\mathbf{x} = -U\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -(L+D)^{-1}U\mathbf{x}^{(k)} + (L+D)^{-1}\mathbf{b}$.
- SOR: Given any iteration scheme $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$, speed it up by $\mathbf{x}^{(k+1)} = \omega\left(G\mathbf{x}^{(k)} + \mathbf{d}\right) + (1-\omega)\mathbf{d}$, with $0 < \omega < 2$. (What does $\omega = 1$ give?)

Let's try these on the example system given earlier, then check spectral radius of each iteration matrix. Smaller spectral radius means faster convergence.

Examples

Some Classical Splittings:

- Write $A = L(ower) + D(iagonal) + U(pper)$
- Jacobi: $D\mathbf{x} = -(L+U)\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -D^{-1}(L+U)\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$.
- Gauss-Seidel: $(L+D)\mathbf{x} = -U\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -(L+D)^{-1}U\mathbf{x}^{(k)} + (L+D)^{-1}\mathbf{b}$.
- SOR: Given any iteration scheme $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$, speed it up by $\mathbf{x}^{(k+1)} = \omega\left(G\mathbf{x}^{(k)} + \mathbf{d}\right) + (1-\omega)\mathbf{d}$, with $0 < \omega < 2$. (What does $\omega = 1$ give?)

Let's try these on the example system given earlier, then check spectral radius of each iteration matrix. Smaller spectral radius means faster convergence.

Examples

Some Classical Splittings:

- Write $A = L(ower) + D(iagonal) + U(pper)$
- Jacobi: $D\mathbf{x} = -(L+U)\,\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -D^{-1}(L+U)\,\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$.
- Gauss-Seidel: $(L+D)\,\mathbf{x} = -U\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -(L+D)^{-1}\,U\mathbf{x}^{(k)} + (L+D)^{-1}\,\mathbf{b}$.
- SOR: Given any iteration scheme $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$, speed it up by $\mathbf{x}^{(k+1)} = \omega\left(G\mathbf{x}^{(k)} + \mathbf{d}\right) + (1-\omega)\,\mathbf{d}$, with $0 < \omega < 2$. (What does $\omega = 1$ give?)

Let's try these on the example system given earlier, then check spectral radius of each iteration matrix. Smaller spectral radius means faster convergence.

Examples

Some Classical Splittings:

- Write $A = L(ower) + D(iagonal) + U(pper)$
- Jacobi: $D\mathbf{x} = -(L+U)\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -D^{-1}(L+U)\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$.
- Gauss-Seidel: $(L+D)\mathbf{x} = -U\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -(L+D)^{-1}U\mathbf{x}^{(k)} + (L+D)^{-1}\mathbf{b}$.
- SOR: Given any iteration scheme $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$, speed it up by $\mathbf{x}^{(k+1)} = \omega\left(G\mathbf{x}^{(k)} + \mathbf{d}\right) + (1-\omega)\mathbf{d}$, with $0 < \omega < 2$. (What does $\omega = 1$ give?)

Let's try these on the example system given earlier, then check spectral radius of each iteration matrix. Smaller spectral radius means faster convergence.

Examples

Some Classical Splittings:

- Write $A = L(ower) + D(iagonal) + U(pper)$
- Jacobi: $D\mathbf{x} = -(L+U)\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -D^{-1}(L+U)\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$.
- Gauss-Seidel: $(L+D)\mathbf{x} = -U\mathbf{x} + \mathbf{b}$, so $\mathbf{x}^{(k+1)} = -(L+D)^{-1}U\mathbf{x}^{(k)} + (L+D)^{-1}\mathbf{b}$.
- SOR: Given any iteration scheme $\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{d}$, speed it up by $\mathbf{x}^{(k+1)} = \omega\left(G\mathbf{x}^{(k)} + \mathbf{d}\right) + (1-\omega)\mathbf{d}$, with $0 < \omega < 2$. (What does $\omega = 1$ give?)

Let's try these on the example system given earlier, then check spectral radius of each iteration matrix. Smaller spectral radius means faster convergence.