Numerical Linear Algebra: iterative methods
Two different approaches

Solve $Ax = b$

Direct methods:

- Deterministic
- Exact up to machine precision
- Expensive (in time and space)

Iterative methods:

- Only approximate
- Cheaper in space and (possibly) time
- Convergence not guaranteed
Iterative methods

Choose any $x_0$ and repeat

$$x^{k+1} = Bx^k + c$$

until $\|x^{k+1} - x^k\|_2 < \epsilon$ or until $\frac{\|x^{k+1} - x^k\|_2}{\|x^k\|} < \epsilon$
Example of iterative solution

Example system

\[
\begin{pmatrix}
10 & 0 & 1 \\
1/2 & 7 & 1 \\
1 & 0 & 6
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
=
\begin{pmatrix}
21 \\
9 \\
8
\end{pmatrix}
\]

with solution (2, 1, 1).

Suppose you know (physics) that solution components are roughly the same size, and observe the dominant size of the diagonal, then

\[
\begin{pmatrix}
10 & 7 \\
7 & 6
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
=
\begin{pmatrix}
21 \\
9 \\
8
\end{pmatrix}
\]

might be a good approximation: solution (2.1, 9/7, 8/6).
Iterative example’

Example system

\[
\begin{pmatrix}
10 & 0 & 1 \\
1/2 & 7 & 1 \\
1 & 0 & 6
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
=
\begin{pmatrix}
21 \\
9 \\
8
\end{pmatrix}
\]

with solution (2, 1, 1).

Also easy to solve:

\[
\begin{pmatrix}
10 \\
1/2 & 7 \\
1 & 0 & 6
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
=
\begin{pmatrix}
21 \\
9 \\
8
\end{pmatrix}
\]

with solution (2.1, 7.95/7, 5.9/6).
Iterative example

Instead of solving $Ax = b$ we solved $L\tilde{x} = b$. Look for the missing part: $\tilde{x} = x + \Delta x$, then $A\Delta x = A\tilde{x} - b \equiv r$. Solve again $L\Delta x = r$ and update $\tilde{x} = \tilde{x} - \Delta x$.

<table>
<thead>
<tr>
<th>iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>2.1000</td>
<td>2.0017</td>
<td>2.000028</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1.1357</td>
<td>1.0023</td>
<td>1.000038</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0.9833</td>
<td>0.9997</td>
<td>0.999995</td>
</tr>
</tbody>
</table>

Two decimals per iteration. This is not typical

Exact system solving: $O(n^3)$ cost; iteration: $O(n^2)$ per iteration. Potentially cheaper if the number of iterations is low.
To solve $Ax = b$; too expensive; suppose $K \approx A$ and solving $Kx = b$ is possible.

Define $Kx_0 = b$, then error correction $x_0 = x + e_0$, and $A(x_0 - e_0) = b$.

so $Ae_0 = Ax_0 - b = r_0$; this is again unsolvable, so

$K\tilde{e}_0$ and $x_1 = x_0 - \tilde{e}_0$.

now iterate: $e_1 = x_1 - x$, $Ae_1 = Ax_1 - b = r_1$ et cetera.
Error analysis

➤ One step

\[ r_1 = Ax_1 - b = A(x_0 - \tilde{e}_0) - b \]  \hspace{1cm} (1)

\[ = r_0 - AK^{-1}r_0 \]  \hspace{1cm} (2)

\[ = (I - AK^{-1})r_0 \]  \hspace{1cm} (3)

➤ Inductively: \( r_n = (I - AK^{-1})^n r_0 \) so \( r_n \downarrow 0 \) if \( |\lambda(I - AK^{-1})| < 1 \)

Geometric reduction (or amplification!)

➤ This is ‘stationary iteration’: every iteration step the same.
Simple analysis, limited applicability.
Computationally

If

\[ A = K - N \]

then

\[ Ax = b \Rightarrow Kx = Nx + b \]

and \( x \) is a fixed point of the iteration

\[ Kx_{i+1} = Nx_i + b \]

This is a stationary iteration:

\[ Kx_{i+1} = Nx_i + b \quad (4) \]
\[ = Kx_i - Ax_i + b \quad (5) \]
\[ = Kx_i - r_i \quad (6) \]

General form of stationary iterative method: \( x_{i+1} = x_i - K^{-1}r_i \)
Choice of $K$

- The closer $K$ is to $A$, the faster convergence.
- Diagonal and lower triangular choice mentioned above: let $A = D_A + L_A + U_A$ be a splitting into diagonal, lower triangular, upper triangular part, then
  - Jacobi method: $K = D_A$ (diagonal part),
  - Gauss-Seidel method: $K = D_A + L_A$ (lower triangle, including diagonal)
  - SOR method: $K = \omega D_A + L_A$
Jacobi

$$K = D_A$$

Algorithm:

for $k = 1, \ldots$ until convergence, do:
  for $i = 1 \ldots n$:
    // $a_{ii} x_i^{(k+1)} = \sum_{j \neq i} a_{ij} x_j^{(k)} + b_i \Rightarrow$
    $x_i^{(k+1)} = a_{ii}^{-1} (\sum_{j \neq i} a_{ij} x_j^{(k)} + b_i)$

Implementation:

for $k = 1, \ldots$ until convergence, do:
  for $i = 1 \ldots n$:
    $t_i = a_{ii}^{-1} (\sum_{j \neq i} a_{ij} x_j + b_i)$
    copy $x \leftarrow t$
Jacobi in pictures:
Gauss-Seidel

\[ K = D_A + L_A \]

Algorithm:

\[ \text{for } k = 1, \ldots \text{ until convergence, do:} \]
\[ \text{for } i = 1 \ldots n: \]
\[ // a_{ii}x_{i}^{(k+1)} + \sum_{j<i} a_{ij}x_{j}^{(k+1)} = \sum_{j>i} a_{ij}x_{j}^{(k)} + b_i \Rightarrow \]
\[ x_{i}^{(k+1)} = a_{ii}^{-1}(- \sum_{j<i} a_{ij}x_{j}^{(k+1)}) - \sum_{j>i} a_{ij}x_{j}^{(k)} + b_i) \]

Implementation:

\[ \text{for } k = 1, \ldots \text{ until convergence, do:} \]
\[ \text{for } i = 1 \ldots n: \]
\[ x_{i} = a_{ii}^{-1}(- \sum_{j\neq i} a_{ij}x_{j} + b_i) \]
GS in pictures:
Choice of $K$ through incomplete LU

- Inspiration from direct methods: let $K = LU \approx A$

Gauss elimination:

for $k, i, j$:

$$a[i, j] = a[i, j] - a[i, k] \times a[k, j] / a[k, k]$$

Incomplete variant:

for $k, i, j$:

if $a[i, j]$ not zero:

$$a[i, j] = a[i, j] - a[i, k] \times a[k, j] / a[k, k]$$

$\Rightarrow$ sparsity of $L + U$ the same as of $A$
Stopping tests

When to stop converging? Can size of the error be guaranteed?

- Direct tests on error $e_n = x - x_n$ impossible; two choices
- Relative change in the computed solution small:

$$\frac{\|x_{n+1} - x_n\|}{\|x_n\|} < \epsilon$$

- Residual small enough:

$$\|r_n\| = \|Ax_n - b\| < \epsilon$$

Without proof: both imply that the error is less than some other $\epsilon'$. 
General form of iterative methods

\[ x_{i+1} = x_i + \sum_{j \leq i} K^{-1} r_j \alpha_{ji}. \]

Compare with stationary iteration, where we only use the last residual, with a coefficient that stays constant.
Residual identities

\[ x_{i+1} = x_i + \sum_{j \leq i} K^{-1} r_j \alpha_{ji}. \]

gives

\[ r_{i+1} = r_i + \sum_{j \leq i} A K^{-1} r_j \alpha_{ji}. \]

Specifically

\[ r_1 = r_0 + A K^{-1} r_0 \alpha_{00}. \]

so \( AK^{-1} r_0 = \alpha_{00}^{-1} (r_1 - r_0) \).

Next:

\[ r_2 = r_1 + A K^{-1} r_1 \alpha_{11} + A K^{-1} r_0 \alpha_{01} \]
\[ = r_1 + A K^{-1} r_1 \alpha_{11} + \alpha_{00}^{-1} \alpha_{01} (r_1 - r_0) \]
\[ \Rightarrow AK^{-1} r_1 = \alpha_{11}^{-1} (r_2 - (1 + \alpha_{00}^{-1} \alpha_{01}) r_1 + \alpha_{00}^{-1} \alpha_{01} r_0) \]

so \( AK^{-1} r_1 = r_2 \beta_2 + r_1 \beta_1 + r_0 \beta_0 \), and that \( \sum_i \beta_i = 0 \).
Generalization

Inductively:

\[ r_{i+1} = r_i + AK^{-1}r_i \delta_i + \sum_{j \leq i+1} r_j \alpha_{ji} \]

\[ r_{i+1}(1 - \alpha_{i+1,i}) = AK^{-1}r_i \delta_i + r_i(1 + \alpha_{ii}) + \sum_{j < i} r_j \alpha_{ji} \]

\[ r_{i+1} \alpha_{i+1,i} = AK^{-1}r_i \delta_i + \sum_{j \leq i} r_j \alpha_{ji} \]

substituting

\[ \alpha_{ii} := 1 + \alpha_{ii} \]

\[ \alpha_{i+1,i} := 1 - \alpha_{i+1,i} \]

note that

\[ \alpha_{i+1,i} = \sum_{j \leq i} \alpha_{ji} \]

\[ r_{i+1} \alpha_{i+1,i} \delta_i^{-1} = AK^{-1}r_i + \sum_{j \leq i} r_j \alpha_{ji} \delta_i^{-1} \]

\[ r_{i+1} \gamma_{i+1,i} = AK^{-1}r_i + \sum_{j \leq i} r_j \gamma_{ji} \]

substituting

\[ \gamma_{ij} = \alpha_{ij} \delta_j^{-1} \]

and we have that

\[ \gamma_{i+1,i} = \sum_{j \leq i} \gamma_{ji}. \]
General form of iterative methods

\[ r_{i+1, \gamma_{i+1}, i} = AK^{-1} r_i + \sum_{j \leq i} r_j \gamma_{ji} \]

and \( \gamma_{i+1, i} = \sum_{j \leq i} \gamma_{ji} \).

Write this as \( AK^{-1} R = RH \) where

\[
H = \begin{pmatrix}
-\gamma_{11} & -\gamma_{12} & \cdots \\
\gamma_{21} & -\gamma_{22} & -\gamma_{23} & \cdots \\
0 & \gamma_{32} & -\gamma_{33} & -\gamma_{34} \\
\emptyset & \cdots & \cdots & \cdots
\end{pmatrix}
\]

\( H \) is a Hessenberg matrix, and note zero column sums.

Divide \( A \) out:

\[ x_{i+1, \gamma_{i+1}, i} = K^{-1} r_i + \sum_{j \leq i} x_j \gamma_{ji} \]
General form of iterative methods

\[
\begin{align*}
  r_i &= Ax_i - b \\
  x_{i+1} \gamma_{i+1,i} &= K^{-1} r_i + \sum_{j \leq i} x_j \gamma_{ji} \\
  r_{i+1} \gamma_{i+1,i} &= AK^{-1} r_i + \sum_{j \leq i} r_j \gamma_{ji}
\end{align*}
\]

where \( \gamma_{i+1,i} = \sum_{j \leq i} \gamma_{ji} \).

Choice of \( \gamma_{ji} \) coefficients?
Orthogonality

Idea one:

If you can make all your residuals orthogonal to each other, and the matrix is of dimension $n$, then after $n$ iterations you have to have converged: it is not possible to have an $n + 1$-st residuals that is orthogonal and nonzero.

Idea two:

The sequence of residuals spans a series of subspaces of increasing dimension, and by orthogonalizing the initial residual is projected on these spaces. This means that the errors will have decreasing sizes.
Let $r_0$ be given

For $i \geq 0$:

- let $s \leftarrow K^{-1}r_i$
- let $t \leftarrow AK^{-1}r_i$
- for $j \leq i$:
  - let $\gamma_j$ be the coefficient so that $t - \gamma_j r_j \perp r_j$
- for $j \leq i$:
  - form $s \leftarrow s - \gamma_j x_j$
  - and $t \leftarrow t - \gamma_j r_j$

let $x_{i+1} = (\sum_j \gamma_j)^{-1}s$, $r_{i+1} = (\sum_j \gamma_j)^{-1}t$. 

Full Orthogonalization Method
Let $r_0$ be given

For $i \geq 0$:
- let $s \leftarrow K^{-1}r_i$
- let $t \leftarrow AK^{-1}r_i$

for $j \leq i$:
- let $\gamma_j$ be the coefficient so that $t - \gamma_j r_j \perp r_j$
- form $s \leftarrow s - \gamma_j x_j$
- and $t \leftarrow t - \gamma_j r_j$

let $x_{i+1} = (\sum_j \gamma_j)^{-1} s$, $r_{i+1} = (\sum_j \gamma_j)^{-1} t$. 

Modified Gramm-Schmidt
Practical differences

- Modified GS more stable
- Inner products are global operations: costly
Coupled recurrences form

\[ x_{i+1} = x_i - \sum_{j \leq i} \alpha_{ji} K^{-1} r_j \]  \hspace{1cm} (7)

This equation is often split as

- Update iterate with search direction: direction:
  \[ x_{i+1} = x_i - \delta_i \rho_i, \]

- Construct search direction from residuals:
  \[ \rho_i = K^{-1} r_i + \sum_{j<i} \beta_{ij} K^{-1} r_j. \]

Inductively:

\[ \rho_i = K^{-1} r_i + \sum_{j<i} \gamma_{ij} \rho_j, \]
Conjugate Gradients

Basic idea:

\[ r_i^t K^{-1} r_j = 0 \quad \text{if } i \neq j. \]

Split recurrences:

\[
\begin{align*}
    x_{i+1} &= x_i - \delta_i p_i \\
    r_{i+1} &= r_i - \delta_i A p_i \\
    p_i &= K^{-1} r_i + \sum_{j<i} \gamma_{ij} p_j,
\end{align*}
\]
Symmetric Positive Definite case

Three term recurrence is enough:

\[
\begin{align*}
    x_{i+1} &= x_i - \delta_i p_i \\
    r_{i+1} &= r_i - \delta_i A p_i \\
    p_{i+1} &= K^{-1} r_{i+1} + \gamma_i p_i
\end{align*}
\]
Preconditioned Conjugate Gradients

Compute $r^{(0)} = b - Ax^{(0)}$ for some initial guess $x^{(0)}$

for $i = 1, 2, \ldots$

solve $Mz^{(i-1)} = r^{(i-1)}$

$\rho_{i-1} = r^{(i-1)^T}z^{(i-1)}$

if $i = 1$

$p^{(1)} = z^{(0)}$

else

$\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$

$p^{(i)} = z^{(i-1)} + \beta_{i-1}p^{(i-1)}$

endif

$q^{(i)} = Ap^{(i)}$

$\alpha_i = \rho_{i-1}/p^{(i)^T}q^{(i)}$

$x^{(i)} = x^{(i-1)} + \alpha_i p^{(i)}$

$r^{(i)} = r^{(i-1)} - \alpha_i q^{(i)}$

check convergence; continue if necessary

end
Observations on iterative methods

- Conjugate gradients: constant storage and inner products; works only for symmetric systems
- GMRES (like FOM): growing storage and inner products: restarting and numerical cleverness
- BiCGstab and QMR: relax the orthogonality
CG derived from minimization

Special case of SPD:

For which vector $x$ with $\|x\| = 1$ is $f(x) = 1/2x^tAx - b^tx$ minimal? (8)

Taking derivative:

$$f'(x) = Ax - b.$$ 

Update

$$x_{i+1} = x_i + p_i\delta_i$$

optimal value:

$$\delta_i = \arg\min_\delta \|f(x_i + p_i\delta)\| = \frac{r_i^tp_i}{p_i^tAp_i}$$

Other constants follow from orthogonality.