Iterative Methods

Simple Iteration

Numerical Analysis
Direct Methods

We have so far focused on solving $Ax = b$ using *direct* methods.

- Gaussian Elimination
- LU Decomposition
- Variants of LU, including Crout and Doolittle
- Other decomposition methods including QR

Advantages

- Algorithm terminates in a fixed number of iterations
- Can take advantage of sparsity (to some extent)
Possible disadvantages?

- We have to store the matrix, or at least know what the entries are
- No obvious way to get a solution that is “close enough” to the exact solution for whatever purpose we have in mind
- No way to reduce the operation count unless we have a special sparse structure.
Iterative methods - scalar case

Can we apply what we know about root-finding to the problem of finding the solution to a linear system of equations?

Newton’s method, when applied to a scalar linear equation converges in one step:

\[ r = b - ax \]

\[ f(x) = b - ax \]

Newton: \[ x_1 = x_0 - \frac{b - ax}{-a} = \frac{b}{a} \]
Iterative methods - scalar case

We can formulate a fixed point problem with the function $g(x)$ defined as

$$g(x) = \frac{1}{m} (b - ax) + x = \left(1 - \frac{a}{m}\right)x + \frac{b}{m}$$

where we choose $m$ so that $|g'(x)| < 1$ (choose $m = 2a$, for example).

$$g(x) = \frac{1}{2}x + \frac{5}{6}$$

$f(x) = 5 - 3x$
We can apply Newton’s Method to the function

\[ F(x) = b - Ax \]

where \( x, b \in \mathbb{R}^n \) and \( A \in \mathbb{R}^{n \times n} \).

We get

\[
x_{k+1} = x_k - [F'(x_k)]^{-1} F(x_k)
\]

\[
= x_k + A^{-1} (b - Ax_k)
\]

This also gives us the answer in exactly one step!
Appling the fixed point iteration to $F(x) = b - Ax$, we get

$$g(x) = x + M^{-1} (b - Ax)$$

where now $M \in \mathbb{R}^{n \times n}$.

The fixed point iteration then looks like

$$x_{k+1} = x_k + M^{-1} (b - Ax_k)$$

The choice of $M$ will affect the convergence of the scheme.
Comparison

Newton step: \[ x_{k+1} = x_k + A^{-1}(b - Ax_k) \]

Fixed point step: \[ x_{k+1} = x_k + M^{-1}(b - Ax_k) \]

☐ The Newton step is no different than using a direct method, since we have to invert \( A \) at each step.

☑ The fixed point approach is promising, if we can understand how different choices for \( M \) affect convergence of the scheme.
A simple iteration

Some classic schemes. Note that for each scheme, $M$ approximates $A$.

$$x_{k+1} = x_k + M^{-1} (b - A x_k)$$

$M = D$ The Jacobi iteration

$M = L$ The Gauss-Seidel iteration

$M = \omega^{-1} D - L$ The Successive Over Relaxation or SOR Method

where $D$ is the diagonal of $A$ and $L$ is the lower triangular portion of $A$ (including diagonal).

The matrix $M$ for each of these choices is easy to invert!
A simple iteration scheme

For the scalar case, we have that

\[ g(x) = \frac{1}{m} (b - ax) + x = \left(1 - \frac{a}{m}\right)x + \frac{b}{m} \]

which will converge to a root of \( f(x) = b - ax \) if

\[ |g'(x)| = \left|1 - \frac{a}{m}\right| < 1 \]

and the closer \( m \) is to \( a \), the faster the convergence. In fact, if \( m = a \), we get convergence in one step.
A simple iteration scheme

In the matrix case, we have

\[
g(x) = x + M^{-1} (b - Ax) \\
= (I - M^{-1}A) x + M^{-1}b
\]

we also want \( M \) to be as close to \( A \) as is feasible, while still being easily invertible. If \( M = A^{-1} \), we converge in one step.

Do we also have to have a condition like \( |g'(x)| < 1 \)?
Let $A$ be an $n \times n$ matrix. The *spectral radius* of a matrix is defined as

$$\rho(A) = \max \{|\lambda| : \lambda \text{ is an eigenvalue of } A\}$$

The following are equivalent

- $\rho(A) < 1$
- $A^k \to 0$ as $k \to \infty$, and
- $Ax \to 0$ as $k \to \infty$ for any vector $x$.

*If one of the above conditions is true, they are all true.*
Let $e_k = A^{-1}b - x_k$. Then

$$e_k = (I - M^{-1}A)e_{k-1} = \ldots = (I - M^{-1}A)^k e_0$$

Then

$$\|e_k\| \leq \|(I - M^{-1})^k\| \|e_0\|$$

And the iteration converges for every initial guess if and only if

$$\rho(I - M^{-1}A) < 1$$
In a general *splitting method*, we write

\[ A = P - Q \]

This leads to the iterative scheme

\[ Px_{k+1} = Qx_k + b \]

or

\[ x_{k+1} = P^{-1}Qx_k + P^{-1}b \]
A simple iteration

Our method

\[ \mathbf{x}_{k+1} = (I - M^{-1}A) \mathbf{x}_k + M^{-1}\mathbf{b} \]

can be viewed as a \textit{simple splitting method} with splitting

\[ M^{-1}A = I - (I - M^{-1}A) \]

\[ P \quad Q \]

applied to the \textit{preconditioned system}

\[ M^{-1}A\mathbf{x} = M^{-1}\mathbf{b} \]

where \( M \) is a \textit{preconditioner}. 
In the scalar case, we could have chosen \( m \) directly:

\[
\frac{1}{m} \ (b - ax) = 0
\]

to ensure convergence of the fixed point method.

This led to the requirement that

\[
\left| 1 - \frac{a}{m} \right| < 1
\]

Of course, there were many other ways that we could have designed the fixed point scheme.
Simple Iteration Algorithm

Given an initial guess $x_0$, compute $r_0 = b - Ax_0$, and solve $Mz_0 = r_0$.

For $k = 1, 2, \ldots$,

Set $x_k = x_{k-1} + z_{k-1}$.

Compute $r_k = b - Ax_k$.

Solve $Mz_k = r_k$.