Numerical Methods in Linear Algebra

Dénes Sexty

University of Graz

2020-2021, winter term

UNIVERSITÄT GRAZ
UNIVERSITY OF GRAZ

1. Examples: Bridge, multidim Newton's method, QM, vibrational eigenmodes
2. Linear equations
3. Gauss elimination, Pivoting
4. Numerical errors, condition number
5. LU decomposition, iterative improvement
6. Householder reduction
7. iterative solution: Gauss-Seidel, Successive overrelaxation (SOR), Conjugate gradient (CG) and others

We still want to solve $A\mathbf{x} = b$.
We have used Gauss elimination, LU decomposition, and that pivoting is typically needed.

Householder reduction

using orthogonal transformations, we bring the matrix to a triangular form:

$$OA = R \tag{1}$$

where O is orthogonal: $O^T O = OO^T = \mathbf{1}$,

if we have two orthogonals: O_1, O_2

$$O_1 O_2 (O_1 O_2)^T = O_1 O_2 O_2^T O_1^T = \mathbf{1} \tag{2}$$

$\Longrightarrow$ their product is also orthogonal

R is upper triangular : $R_{ij} = 0 \quad \text{for } i > j$

Can also think of it as factorization: $A = O^T R$

$$A\mathbf{x} = \mathbf{b} \quad \Longrightarrow \quad O^T R\mathbf{x} = \mathbf{b} \quad \Longrightarrow \quad O^T\mathbf{y} = \mathbf{b}, \quad R\mathbf{x} = \mathbf{y} \qquad (3)$$

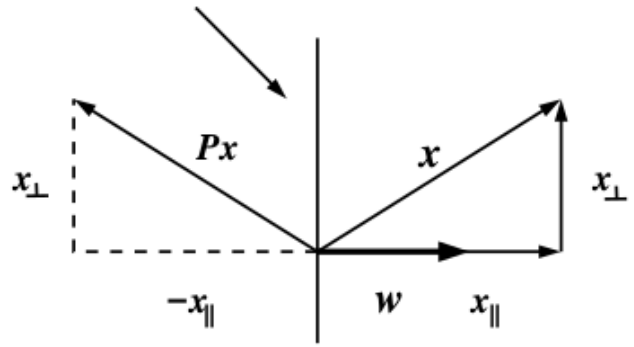
$O^T\mathbf{y} = \mathbf{b}$ is solved simply using $\mathbf{y} = O\mathbf{b}$
$R\mathbf{x} = \mathbf{y}$ is solved using backsubstitution

Replaces Gauss elimination (LU decomposition)

numerically stable since it uses orthogonal transformations

O is a product of reflections

Take a unit vector $\mathbf{w}$, $\mathbf{w}^T\mathbf{w} = 1$
This defines an $n-1$ dimensional hyperplane of vectors orthogonal to $\mathbf{w}$



we want the reflection $\mathbf{x} \to P\mathbf{x}$

$\mathbf{x}_\parallel = (\mathbf{w}^T\mathbf{x})\mathbf{w}$: projection to $\mathbf{w}$
$\mathbf{x} = \mathbf{x}_\parallel + \mathbf{x}_\perp \;\to\; P\mathbf{x} = -\mathbf{x}_\parallel + \mathbf{x}_\perp = \mathbf{x} - 2\mathbf{x}_\parallel$
$P\mathbf{x} = \mathbf{x} - 2\mathbf{w}(\mathbf{w}^T\mathbf{x}) = (\mathbf{1} - 2\mathbf{w}\mathbf{w}^T)\mathbf{x}$
outer product: $\mathbf{w}\mathbf{w}^T = M, \quad M_{ij} = \mathbf{w}_i\mathbf{w}_j$

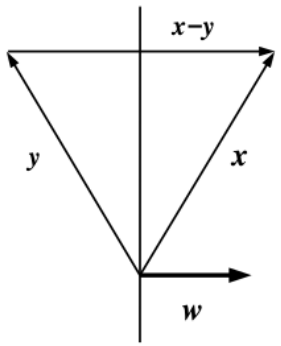
$P = (\mathbf{1} - 2\mathbf{w}\mathbf{w}^T)$
P is symmetric: $P^T = P$

$$P^2 = (1 - 2\mathbf{w}\mathbf{w}^T)(1 - 2\mathbf{w}\mathbf{w}^T) = 1 - 4\mathbf{w}\mathbf{w}^T + 4\mathbf{w}\underbrace{\mathbf{w}^T\mathbf{w}}_{1}\mathbf{w}^T = 1 \tag{4}$$

as $\mathbf{w}$ is a unit vector: $\mathbf{w}_j\mathbf{w}_j = 1 \implies \mathbf{w}_i\mathbf{w}_j\mathbf{w}_j\mathbf{w}_k = \mathbf{w}_i\mathbf{w}_k$

$\implies$ P is orthogonal: $P^T P = PP^T = P^2 = 1$

Can we bring any two vectors in to each other with a reflection?
given $\mathbf{x} \neq \mathbf{y}$ two vectors with the same length $\mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2 = \|\mathbf{y}\|^2 = \mathbf{y}^T\mathbf{y}$ $\implies$ There exists a reflection such that $P\mathbf{x} = \mathbf{y},\ P\mathbf{y} = \mathbf{x}$



Proof: $\mathbf{w} = \frac{\mathbf{x}-\mathbf{y}}{\|\mathbf{x}-\mathbf{y}\|}$

$(\mathbf{w}\mathbf{w}^T)\mathbf{x} = (\mathbf{x}-\mathbf{y})\frac{(\mathbf{x}^T-\mathbf{y}^T)\mathbf{x}}{\|\mathbf{x}-\mathbf{y}\|^2} = (\mathbf{x}-\mathbf{y})\frac{(\mathbf{x}^T\mathbf{x}+\mathbf{y}^T\mathbf{y}-\mathbf{y}^T\mathbf{x}-\mathbf{x}^T\mathbf{y})}{2\|\mathbf{x}-\mathbf{y}\|^2} =$

$= (\mathbf{x}-\mathbf{y})\frac{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}{2\|\mathbf{x}-\mathbf{y}\|^2} = \frac{\mathbf{x}-\mathbf{y}}{2}$

$$\implies P\mathbf{x} = (1 - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} = \mathbf{x} - 2\frac{\mathbf{x}-\mathbf{y}}{2} = \mathbf{y} \tag{5}$$

construct reflection such that

$$a^1 = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \xrightarrow{P_1 a^1} \begin{pmatrix} -\sigma_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad P_1 \underbrace{a^1}_{=x} = \underbrace{-\sigma_1 \mathbf{e_1}}_{=y} \text{ with } \sigma_1 = \pm \left\| a^1 \right\| \tag{6}$$

$$\mathbf{u}^T = (\mathbf{x} - \mathbf{y})^T = (a_{11} + \sigma_1, a_{21}, \ldots, a_{n1}), \qquad P_1 = 1 - \mathbf{u}\mathbf{u}^T / H \tag{7}$$

$$H = \frac{1}{2} u^T u = \frac{1}{2}(a^1 + \sigma_1 \mathbf{e_1})(a^1 + \sigma_1 \mathbf{e_1}) = \frac{1}{2}(\left\| a^1 \right\|^2 + 2\sigma_1 a_{11} + \sigma_1^2) = \sigma_1(\sigma_1 + a_{11}) \tag{8}$$

avoiding precision loss if $\text{sign}\sigma_1 = \text{sign}a_{11} \quad \Longrightarrow \quad \sigma_1 = \text{sign}(a_{11}) \left\| a^1 \right\|$
No pivoting required, the diagonal always becomes $\left\| a^1 \right\|$. If that is zero, the matrix is singular

After first step $a_{ij} \to a'_{ij} = (P_1 A)_{ij}$
2nd step: $\mathbf{u}^T = (0, a'_{22} + \sigma_2, a'_{32}, \ldots, a'_{n2}), \qquad \sigma_2 = \mathrm{sign}(a'_{22})\sqrt{\sum_{i=2}^{n}(a'_{i2})^2}$

$$P_2 = 1 - \mathbf{u}\mathbf{u}^T/H = \left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & [X] & \\ 0 & & & \end{array}\right), \qquad H = \sigma_2(\sigma_2 + a'_{22}) \tag{9}$$

$$P_2 P_1 A = \begin{pmatrix} -\sigma_1 & a'_{12} & \cdots & a'_{1n} \\ 0 & -\sigma_2 & & \\ \vdots & 0 & [A''] & \\ \vdots & \vdots & & \\ 0 & 0 & & \end{pmatrix} \tag{10}$$

First row and first column of $P_1 A$ is unchanged

Do it $n-1$ times: $P_{n-1}\dots P_1 A = R$ with R upper triangular
$Q = (P_{n-1}P_{n-2}\dots P_2P_1)^T = P_1P_2\dots P_{n-2}P_{n-1}$
$A = QR \quad \Longrightarrow$ "QR decomposition"

Complex QR
$\mathbf{x} \in \mathbb{C}^n$, $x_1 = re^{i\Theta}$ with $r, \Theta \in \mathbb{R}$
$\mathbf{u} = \mathbf{x} \pm e^{i\Theta}\|x\|\,\mathbf{e_1}$ and $P = 1 - \frac{\mathbf{u}\mathbf{u}^\dagger}{H}, \quad H = \mathbf{u}^\dagger\mathbf{u}/2$
$P\mathbf{x} = \mp e^{i\theta}\|x\|\,\mathbf{e_1}$
orthogonality $\rightarrow$ unitarity

$$P^\dagger P = \left(1 - \frac{\mathbf{u}\mathbf{u}^\dagger}{H}\right)\left(1 - \frac{\mathbf{u}\mathbf{u}^\dagger}{H}\right) = 1 - 2\frac{\mathbf{u}\mathbf{u}^\dagger}{H} + \frac{\mathbf{u}\mathbf{u}^\dagger\mathbf{u}\mathbf{u}^\dagger}{(\frac{1}{2}\mathbf{u}^\dagger\mathbf{u})^2} \tag{11}$$

$$= 1 - 2\frac{\mathbf{u}\mathbf{u}^\dagger}{H} + 2\frac{\mathbf{u}\mathbf{u}^\dagger}{H} = 1 \tag{12}$$

sign choosen to maximize H

cost of LU decomposition: $O(n^3)$
matrix multiplication: $O(n^3)$, need $n-1$ reflections: $O(n^4)$?

Actually multiplication with reflector is cheaper.
$\mathbf{w}\mathbf{w}^T = w_i w_j$: calculating $w_i w_j A_{jk}$
First calculate $v_k = w_j A_{jk}$, takes $O(n^2)$ operations Than $a'_{ij} = a_{ij} - C w_i v_j$ takes $O(n^2)$ operations $\implies$ multiplication with reflector takes $O(n^2)$
$\implies$ QR reduction needs $O(n^3)$

LU $\sim n^3/3$, QR reduction $\sim 2n^3/3$
but QR needs no pivoting and is numerically stable.

Often one can have an iterative procedure that delivers a solution to some problem. We are interested here in the solution to $A\mathbf{x} = \mathbf{b}$
An example (not particularly efficient)
let M be a matrix. The geometric series is defined as:

$$S_n = \sum_{i=0}^{n} M^i = 1 + M + M^2 + \ldots + M^n \tag{13}$$

now we can write $S_{n+1} = S_n + M^{n+1} = 1 + S_n M$
$\implies S_n(1 - M) = 1 - M^{n+1}$

$$S_n = (1 - M^{n+1})(1 - M)^{-1} \tag{14}$$

if $M^n \xrightarrow[n\to\infty]{} 0$, than $S_\infty = (1 - M)^{-1}$ This happens if M is "small" (e.g. all eigenvalues are < 1)

Suppose M is diagonalizable: $M = U\mathrm{diag}(\lambda_1, \ldots, \lambda_n)U^{-1}$, with U orthogonal (unitary)

$$M^k = U\mathrm{diag}(\lambda_1, \ldots, \lambda_n)\underbrace{U^{-1}U}_{=1}\mathrm{diag}(\lambda_1, \ldots, \lambda_n)U^{-1}\ldots = \tag{15}$$

$$= U(\mathrm{diag}(\lambda_1, \ldots, \lambda_n))^k U^{-1} = U\mathrm{diag}(\lambda_1^k, \ldots, \lambda_n^k)U^{-1}$$

$\implies$ $M^k \to 0$ if $\lambda_i < 1$

This leads to the following procedure:
Given a matrix A, and righthandside b,

1. calculate the matrix $M = 1 - A$
2. $\mathbf{v}_0 = \mathbf{b}$
3. $\mathbf{v}_{i+1} = \mathbf{b} + M\mathbf{v}_i$
4. exit if converged
5. goto 3

we have calculated the geometric series using $S_{n+1} = 1 + MS_n$, applied to the vector b
by convergence we have

$$\mathbf{v} = (1 + M + \ldots M^n + \ldots)\mathbf{b} = (1 - M)^{-1}\mathbf{b} = A^{-1}\mathbf{b} \tag{16}$$

converges only if A is "close" to the unit matrix

We want to solve $a_{ij}x_j = b_i$
solve for x_i formally:

$$x_i = -\frac{1}{a_{ii}}\left[\sum_{j\neq i} a_{ij}x_j - b_i\right] \tag{17}$$

Make this into an iteration:

$$x_i^{(i+1)} = -\frac{1}{a_{ii}}\left[\sum_{j\neq i} a_{ij}x_j^{(i)} - b_i\right] \tag{18}$$

if we get $x^{(i+1)} = x^{(i)}$ we have found the solution.
$\Longrightarrow$ iterate until convergence

We rewrite $A\mathbf{x} = \mathbf{b}$ in the form: $\mathbf{x} = G(\omega)\mathbf{x} + \mathbf{g}(\omega)$, where ω is a relaxation parameter
We can thus define an iteration: $\mathbf{x}_{i+1} = G(\omega)\mathbf{x}_i + \mathbf{g}(\omega)$
In case of convergence

$$\mathbf{x} = \lim_{i \to \infty} \mathbf{x}_i \tag{19}$$

Procedure is not unique, depends on the choice of $G(\omega)$ and $\mathbf{g}(\omega)$
e.g. $A = N(\omega) - P(\omega)$ with $N(\omega)$ non-singular

$$(N(\omega) - P(\omega))\mathbf{x} = \mathbf{b} \quad \Longrightarrow \quad N(\omega)\mathbf{x}_{i+1} = P(\omega)\mathbf{x}_i + \mathbf{g}(\omega) \tag{20}$$

We choose $N(\omega)$ such that one can easily solve $N(\omega)\mathbf{x} = \mathbf{y}$.

$$\mathbf{x} = N^{-1}(\omega)P(\omega)\mathbf{x} + N^{-1}(\omega)\mathbf{b} \tag{21}$$

$$\Longrightarrow \; G(\omega) = N^{-1}(\omega)P(\omega), \qquad \mathbf{g}(\omega) = N^{-1}(\omega)\mathbf{b}$$

Can be shown to converge if the spectral radius ρ of $G(\omega)$ is smaller than 1 (i.e. eigenvalues have absolute value smaller than 1)

$$\rho(G) = \max|\lambda_k(G)| \tag{22}$$

(In case $G(\omega)$ is not diagonalizable, diagonals < 1 in the Jordan normal form)

Convergence rate is given by $\lambda_{max}(G(\omega))$
The inverse of $N(\omega)$ must be easily calculable.

$N(\omega)$ is completely arbitrary ($P(\omega) = N(\omega) - A$), but it makes sense if it is choosen to be related to A

$$\begin{aligned} \mathbf{e}_k &= \mathbf{x}_k - \mathbf{x} = N^{-1}P\mathbf{x}_{k-1} + N^{-1}\mathbf{b} - (N^{-1}P\mathbf{x} + N^{-1}\mathbf{b}) = \\ &= N^{-1}P(\mathbf{x}_{k-1} - \mathbf{x}) = N^{-1}P\mathbf{e}_{k-1} \end{aligned} \tag{23}$$

So the error vector is: $\mathbf{e}_k = (N^{-1}P)^k \mathbf{e}_0$

$\lim_{k\to\infty} M^k = 0$ is equivalent to

- $\rho(M) < 1$
  as seen above
- There is a norm for which $\|M\| < 1$
  because $\left\|M^i\right\| \leq \|M\|^i \to 0$

$$A = D - L - U = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix} - \begin{pmatrix} 0 & 0 & \dots & 0 \\ -a_{21} & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \dots & 0 \end{pmatrix} - \begin{pmatrix} 0 & -a_{12} & \dots & -a_{1n} \\ 0 & 0 & \dots & -a_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad (24)$$

Assuming no zeroes on the diagonal, $N(\omega) = \frac{1}{\omega} D$

$$\implies P(\omega) = \frac{1-\omega}{\omega} D + L + U \quad \text{in} \quad A = N(\omega) - P(\omega) \quad (25)$$

$$G(\omega) = (1-\omega) + \omega D^{-1}(L+U), \quad \mathbf{g}(\omega) = \omega D^{-1}\mathbf{b} \quad (26)$$

for $\omega = 1$ we get the Jacobi method, for $\omega \neq 1$ we have Jacobi overrelaxation (JOR)

Iterate until convergence
Condition for exit if it's non-convergent is also needed

Start again from $A = D - L - U$, choose $N(\omega) = \frac{1}{\omega}D - L$
$\implies G(\omega) = (D - \omega L)^{-1}[(1-\omega)D + \omega U], \quad \mathbf{g}(\omega) = \omega(D - \omega L)^{-1}\mathbf{b}$
Iteration equation:(after multiplication with $(D - \omega L)$

$$(D - \omega L)\mathbf{x}_{i+1} = (1-\omega)D\mathbf{x}_i + \omega U\mathbf{x}_i + \omega\mathbf{b} \tag{27}$$

equivalent to:

$$\mathbf{x}_{i+1} = (1-\omega)\mathbf{x}_i + \omega D^{-1}(L\mathbf{x}_{i+1} + U\mathbf{x}_i + \mathbf{b}) \tag{28}$$

Can be solved using forward substitution. Inverse of $(D - \omega L)$ not needed

$$x_j^{(i+1)} = (1-\omega)x_j^{(i)} - \frac{\omega}{a_{ii}}\left(\sum_{k=1}^{j-1} a_{jk}x_k^{(i+1)} + \sum_{k=j+1}^{n} a_{jk}x_k^{(i)} - b_j\right) \tag{29}$$

$\omega = 1$ is called Gauss seidel method. $\omega \neq 1$ is Successive over-relaxation
ω is called relaxation parameter
weighted average of previous vector and Gauss-Seidel iterate

Jacobi (JOR):

$$\mathbf{x}_{i+1} = (1-\omega)\mathbf{x}_i + \omega D^{-1}(L\mathbf{x}_i + U\mathbf{x}_i + \mathbf{b}) \tag{30}$$

$$x_j^{(i+1)} = (1-\omega)x_j^{(i)} - \frac{\omega}{a_{ii}}\left[\sum_{k=1}^{j-1} a_{jk}x_k^{(i)} + \sum_{k=j+1}^{n} a_{jk}x_k^{(i)} - b_j\right] \tag{31}$$

Gauss-Seidel (SOR):

$$\mathbf{x}_{i+1} = (1-\omega)\mathbf{x}_i + \omega D^{-1}(L\mathbf{x}_{i+1} + U\mathbf{x}_i + \mathbf{b}) \tag{32}$$

$$x_j^{(i+1)} = (1-\omega)x_j^{(i)} - \frac{\omega}{a_{ii}}\left[\sum_{k=1}^{j-1} a_{jk}x_k^{(i+1)} + \sum_{k=j+1}^{n} a_{jk}x_k^{(i)} - b_j\right] \tag{33}$$

Same, except already known elements of $\mathbf{x}_{i+1}$ are used instead of $\mathbf{x}_i$.

if A is symmetric and positive definite than SOR process can be shown to converge for $0 < \omega < 2$
Typically one chooses $\omega = 1.5 - 2$ to speed up convergence.
$\omega < 1$ is sometimes useful if it otherwise does not converge.

Jacobi iteration can be shown to converge for a strictly diagonally dominant matrix, as well as JOR for $0 < \omega \leq 1$
for symmetric and positive definit A, JOR can be shown to converge for

$$0 < \omega < \frac{2}{\mu_{max}}, \tag{34}$$

where μ_{max} is the maximal eigenvalue of $D^{-1}A$ Typically one takes $\omega = 2/3$.

a matrix A is strictly diagonally dominan if for every i

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \tag{35}$$

What should be the convergence criterion?
we need $\|\mathbf{x}_k - \mathbf{x}\| < \epsilon$ but we don't know $\mathbf{x}$.

$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$ is necessary, but not sufficient. (look at the sum $\sum_i 1/i$)

We can look at the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$.
First try: $\|\mathbf{b} - A\mathbf{x}_k\| < \epsilon$ This might also be misleading as

$$\|\mathbf{x} - \mathbf{x}_k\| = \left\|A^{-1}(\mathbf{b} - A\mathbf{x}_k)\right\| \leq \left\|A^{-1}\right\| \|\mathbf{b} - A\mathbf{x}_k\| \leq \left\|A^{-1}\right\| \epsilon \tag{36}$$

So the error norm could be large if $\left\|A^{-1}\right\|$ is large

Safest choice for unknown A:

$$\frac{\|\mathbf{b} - A\mathbf{x}_k\|}{\|\mathbf{b} - A\mathbf{x}_0\|} < \epsilon \tag{37}$$

The iteration eq.: $\mathbf{x}_{k+1} = N^{-1}P\mathbf{x}_k + N^{-1}\mathbf{b}$ can be written as:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_k - N^{-1}N\mathbf{x}_k + N^{-1}P\mathbf{x}_k + N^{-1}\mathbf{b} = \mathbf{x}_k + N^{-1}(-(N-P)\mathbf{x}_k + \mathbf{b}) = \\
&= \mathbf{x}_k + N^{-1}(\mathbf{b} - A\mathbf{x}_k) = \mathbf{x}_k + N^{-1}\mathbf{r}_k \qquad (38) \\
\mathbf{r}_{k+1} &= \mathbf{b} - A(\mathbf{x}_k + N^{-1}\mathbf{r}_k) = \mathbf{r}_k - AN^{-1}\mathbf{r}_k \qquad (39)
\end{aligned}$$

So the pseudocode could be:

1. Initialize: $\mathbf{x} = 0,\ \mathbf{r} = \mathbf{b}$
2. While $\|\mathbf{r}\| > \epsilon \|\mathbf{b}\|$
3. Solve $N\mathbf{y} = \mathbf{r}$ for $\mathbf{y}$
4. update solution: $\mathbf{x} = \mathbf{x} + \mathbf{y}$
5. update residual: $\mathbf{r} = \mathbf{r} - A\mathbf{y}$

Costs per iteration:

1. Calculating the norm costs n operations
2. Computing y costs n operations for Jacobi, $n^2/2$ for Gauss-Seidel
3. Computing $A\mathbf{v}$ costs n^2 operations for a dense matrix (for sparse matrix with $O(n)$ nonzero elements this costs is only $O(n)$)

One iteration takes therefore $O(n^2)$ operations in contrast with $O(n^3)$ of the direct methods.

$\Longrightarrow$ do we need $O(n)$ iterations or less? The answer depends on many things... (what is A, what is ω?) Often one finds out by experimenting

We want to get to the conjugate gradient method, let's first discuss the Gradient method (a.k.a Steepest descent, Richardson's method)
Take

$$N = \frac{1}{\alpha}, \quad P = \frac{1}{\alpha} - A \tag{40}$$

with $\alpha > 0$. This leads to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha(\mathbf{b} - A\mathbf{x}_k) \tag{41}$$

This converges for $0 < \alpha < 2/\lambda_{max}(A)$ for a positive definite A and might not converge for a non positive definite A

Suppose A is symmetric (hermitic) and positive definit ($\forall \mathbf{x}: \ \mathbf{x}^T A\mathbf{x} > 0$) define

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b} \tag{42}$$

Since A is positive definit, this has a minimum. There we have $\nabla f(\mathbf{x}) = 0$

$$\nabla f(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = 0 \tag{43}$$

Minumum of $f(\mathbf{x})$ solves the eq. $A\mathbf{x} = \mathbf{b}$

The iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha(\mathbf{b} - A\mathbf{x}_k) \tag{44}$$

steps along the negative gradient of $f(\mathbf{x})$.
If α is small enough ($0 < \alpha < 2/\rho(A)$) than $f(\mathbf{x})$ decreases $\rightarrow$ we might get to the minimum

if we take small steps it's going to take long to get to the minimum.
Idea: choose α such that $f(\mathbf{x} + \alpha(\mathbf{b} - A\mathbf{x}))$ is minimized $\rightarrow$ optimal stepsize

$$f(\mathbf{x} + \alpha\mathbf{r}) = \frac{1}{2}(\mathbf{x} + \alpha\mathbf{r})^T A(\mathbf{x} + \alpha\mathbf{r}) - b^T(x + \alpha\mathbf{r}) \tag{45}$$

We get minimum from $\partial_\alpha f(\mathbf{x} + \alpha\mathbf{r}) = 0$

$$\implies \alpha = \frac{r^T r}{r^T A r} \tag{46}$$

Green: Gradient method with optimal stepsize

Can still take many steps to converge

Teaser: Red corresponds to Conjugate Gradient, which converges in two steps (in 2d)