Assignment – 2

Maor Ashkenazi – 312498017

Pan Eyal - 208722058

Problem 1 – Section A:

General Iterative Method:

```
import scipy
import scipy.sparse as sparse
import scipy.sparse.linalg as sparselinalg
import numpy as np

def residual_norm(A, b, x):
    return np.linalg.norm(A@x - b)

def general_iter(A, b, x_0, M_inv, max_iter, eps, weight=1, verbose=True):
    x = [x_0]
    r_norms = [residual_norm(A, b, x[0])]
    if verbose:
        print(f"Iter: 0: {x[0]}")
    for k in range(1, max_iter + 1):
        x.append(x[k - 1] + weight*M_inv@(b - A@x[k - 1]))
        r_norms.append(residual_norm(A, b, x[k]))
        if verbose:
            print(f"Iter: {k}: {x[k]}")
        if (r_norms[k] / np.linalg.norm(b)) < eps:
            break
    return x[-1], np.array(r_norms)
```

Jacobi:

```
def jacobi(A, b, x_0, max_iter, eps, weight=1, verbose=True, is_sparse=False):
    diag = sparse.diags(A.diagonal()) if is_sparse else np.diag(A.diagonal())
    M_inv =  sparselinalg.inv(diag) if is_sparse else np.linalg.inv(diag)
    return general_iter(A, b, x_0, M_inv, max_iter, eps, weight, verbose)
```

Gauss-Seidel:

```
def gs(A, b, x_0, max_iter, eps, verbose=True, is_sparse=False):
    tril = sparse.tril(A) if is_sparse else np.tril(A)
    M_inv =  sparselinalg.inv(tril) if is_sparse else np.linalg.inv(tril)
    return general_iter(A, b, x_0, M_inv, max_iter, eps, 1, verbose)
```

Steepest Descent:

```
def sd(A, b, x_0, max_iter, eps, verbose=True):
    x = [x_0]
    r = [b - A@x[0]]
    if verbose:
        print(f"Iter: 0: {x[0]}")
    for k in range(1, max_iter + 1):
        A_r = A@r[k - 1]
        alpha = (r[k - 1] @ r[k - 1]) / (r[k - 1] @ A_r)
        x.append(x[k - 1] + alpha*r[k - 1])
        r.append(r[k - 1] - alpha*A_r)
        if verbose:
            print(f"Iter: {k}: {x[k]}")
        if (np.linalg.norm(r[k]) / np.linalg.norm(b)) < eps:
            break
    return x[-1], np.array([np.linalg.norm(res) for res in r])
```

Conjugate Gradient:

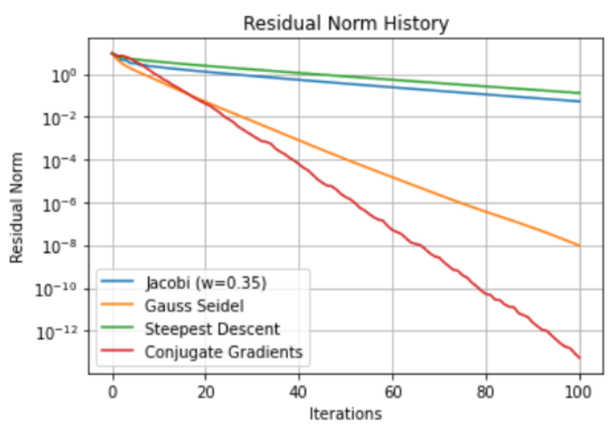
```python
def cg(A, b, x_0, max_iter, eps, verbose=True):
    x = [x_0]
    r = [b - A@x[0]]
    p = [r[0]]
    if verbose:
        print(f"Iter: 0: {x[0]}")
    for k in range(1, max_iter + 1):
        A_p = A@p[k - 1]
        alpha = (r[k - 1] @ p[k - 1]) / (p[k - 1] @ A_p)
        x.append(x[k - 1] + alpha*p[k - 1])
        r.append(r[k - 1] - alpha*A_p)
        if verbose:
            print(f"Iter: {k}: {x[k]}")
        if (np.linalg.norm(r[k]) / np.linalg.norm(b)) < eps:
            break

        beta = (r[k] @ r[k]) / (r[k - 1] @ r[k - 1])
        p.append(r[k] + beta*p[k - 1])
    return x[-1], np.array([np.linalg.norm(res) for res in r])
```

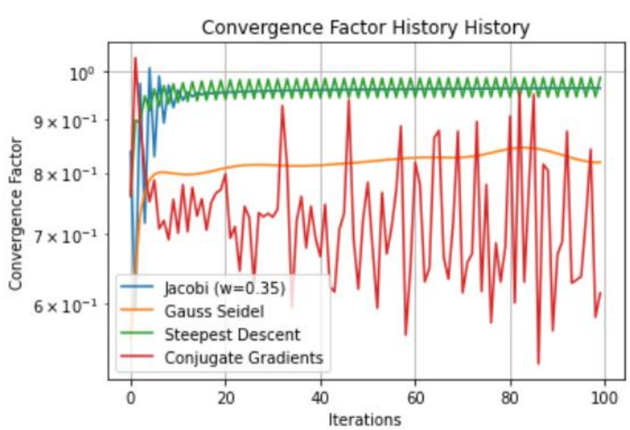
<u>Problem 1</u> – Section B:

Residual Norm History Plot:

```
(Jacobi w=0.35) Residual Norm:       0.05202514826544867
(Gauss Seidel) Residual Norm:        9.475768316373202e-09
(Steepest Descent) Residual Norm:    0.1294911476894247
(Conjugate Gradients) Residual Norm: 5.414264003190013e-14
```



Convergence Factor Plot:



<u>Problem 2</u> – Section A:

We've seen in class (Theorem 16 in chapter 6.1.2) that for A invertible matrix, the general iteration:

$x^{(k+1)} = x^{(k)} + M^{-1}\left(b - Ax^{(k)}\right)$ converges for any starting vector $x^{(0)}$ $if\ and\ only\ if\ \rho(T) = \rho(\mathrm{I} - \mathrm{M}^{-1}A) < 1$.

From the SPD property, we know that A is invertible.

For the Richardson method: $x^{(k+1)} = x^{(k)} + \frac{1}{\|A\|}\left(b - Ax^{(k)}\right)$, thus: $M^{-1} = \frac{1}{\|A\|}I$. In our case, $T = I - \frac{1}{\|A\|}A$.

We will show that if v_i, λ_i are A's eigenpairs, $1 - \frac{1}{\|A\|}\lambda_i$ are T's eigenvalues:

$$Av_i = \lambda_i v_i \Rightarrow Tv_i = \left(\mathrm{I} - \frac{1}{\|A\|}A\right)v_i = v_i - \frac{1}{\|A\|}Av_i = v_i - \frac{1}{\|A\|}\lambda_i v_i = \left(1 - \frac{1}{\|A\|}\lambda_i\right)v_i$$

Also, we know that T's spectral radius is $\rho(T) = \max\left\{\left|1 - \frac{1}{\|A\|}\lambda_{max}\right|, \left|1 - \frac{1}{\|A\|}\lambda_{min}\right|\right\}$.

Using the induced norm definition, we know that $\|A\| > \rho(A)$.

Therefore, for every λ_i: $\frac{1}{\|A\|}\lambda_i < \frac{1}{\rho(A)}\lambda_i < 1$. Also, because A is SPD, we know that $\frac{1}{\|A\|}\lambda_i > 0$.

Using the inequality above, we get $\rho(T) = \rho(\mathrm{I} - M^{-1}A) = \rho\left(\mathrm{I} - I * \frac{1}{\|A\|}A\right) = \max\left\{\left|1 - \frac{1}{\|A\|}\lambda_{max}\right|, \left|1 - \frac{1}{\|A\|}\lambda_{min}\right|\right\} < 1$.

<u>Problem 2</u> – Section B:

For $M^{-1} = \frac{1}{\|A\|}I$ and $T = \mathrm{I} - \frac{1}{\|A\|}A$ as defined in the previous section, we have shown that T's eigenvalues are: $1 - \frac{1}{\|A\|}\lambda_i$ where λ_i are A's eigenvalues.

For the case that A has positive and negative eigenvalues:

$$\rho(T) = \rho\left(\mathrm{I} - \frac{1}{\|A\|}A\right) = \max_{\mathrm{i}}\left|1 - \frac{1}{\|A\|}\lambda_i\right|$$

For any negative λ_i we will get that: $\left|1 - \frac{1}{\|A\|}\lambda_i\right| > 1$, thus, $\rho(T) > 1$ and from Theorem 16 in chapter 6.1.2, we will get that the Richardson method will not converge.

Also, we can see that the method will diverge by looking at the norm of the error vector:

$$\left\|e^{(k+1)}\right\| \underbrace{=}_{\substack{shown\ in \\ class}} \left\|T^{(k+1)}e^0\right\| \underbrace{=}_{\substack{\lambda`,v`\ are\ T's \\ eigenpairs}} \left\|T^{(k+1)}\sum_i \alpha_i v`_i\right\| = \left\|\sum_i \alpha_i \lambda`^{k+1}_i v`_i\right\| \underbrace{\rightarrow}_{\substack{for\ \rho(T)>1 \\ and\ k\rightarrow\infty}} \infty$$

<u>Problem 2</u> – Section C:

i. Using formula (44) in subsection (6.1.7), we set x^{k+1} in the function expression:

$$f\left(x^{(k+1)}\right)=\frac{1}{2}\left(x^{(k+1)}\right)^{T}Ax^{(k+1)}-\left(x^{(k+1)}\right)^{T}b+\frac{1}{2}(x^{*})^{T}b=(*)$$

As a reminder, the step for steepest descent is:

$$x^{(k+1)}=x^{(k)}+\underbrace{\frac{\langle r^{(k)},r^{(k)}\rangle}{\langle r^{(k)},Ar^{(k)}\rangle}}_{\alpha_{opt}}r^{(k)}$$

We continue developing the expression:

$$(*)=\frac{1}{2}\left(x^{(k)}+\alpha_{opt}r^{(k)}\right)^{T}A\left(x^{(k)}+\alpha_{opt}r^{(k)}\right)-\left(x^{(k)}+\alpha_{opt}r^{(k)}\right)^{T}b+\frac{1}{2}(x^{*})^{T}b$$

$$=\frac{1}{2}\left(\left(x^{(k)}\right)^{T}+\left(\alpha_{opt}r^{(k)}\right)^{T}\right)A\left(x^{(k)}+\alpha_{opt}r^{(k)}\right)-\left(\left(x^{(k)}\right)^{T}+\left(\alpha_{opt}r^{(k)}\right)^{T}\right)b+\frac{1}{2}(x^{*})^{T}b=$$

$$=\frac{1}{2}\left(\left(x^{(k)}\right)^{T}A\left(x^{(k)}+\alpha_{opt}r^{(k)}\right)+\left(\alpha_{opt}r^{(k)}\right)^{T}A\left(x^{(k)}+\alpha_{opt}r^{(k)}\right)\right)-\left(\left(x^{(k)}\right)^{T}b+\left(\alpha_{opt}r^{(k)}\right)^{T}b\right)+\frac{1}{2}(x^{*})^{T}b$$

$$=\frac{1}{2}\left(\left(x^{(k)}\right)^{T}Ax^{(k)}+\left(x^{(k)}\right)^{T}A\alpha_{opt}r^{(k)}+\left(\alpha_{opt}r^{(k)}\right)^{T}Ax^{(k)}+\left(\alpha_{opt}r^{(k)}\right)^{T}A\alpha_{opt}r^{(k)}\right)-\left(x^{(k)}\right)^{T}b-\left(\alpha_{opt}r^{(k)}\right)^{T}b+\frac{1}{2}(x^{*})^{T}b$$

$$=\frac{1}{2}\left(\left(x^{(k)}\right)^{T}Ax^{(k)}\right)-\left(x^{(k)}\right)^{T}b+\frac{1}{2}(x^{*})^{T}b+\frac{1}{2}\left(\left(x^{(k)}\right)^{T}A\alpha_{opt}r^{(k)}+\left(\alpha_{opt}r^{(k)}\right)^{T}Ax^{(k)}+\left(\alpha_{opt}r^{(k)}\right)^{T}A\alpha_{opt}r^{(k)}\right)-\left(\alpha_{opt}r^{(k)}\right)^{T}b$$

$$=f\left(x^{(k)}\right)+\frac{1}{2}\left(\left(x^{(k)}\right)^{T}A\alpha_{opt}r^{(k)}+\left(\alpha_{opt}r^{(k)}\right)^{T}Ax^{(k)}+\left(\alpha_{opt}r^{(k)}\right)^{T}A\alpha_{opt}r^{(k)}\right)-\left(\alpha_{opt}r^{(k)}\right)^{T}b$$

$$=f\left(x^{(k)}\right)+\frac{1}{2}\left(\alpha_{opt}\langle x^{(k)},Ar^{(k)}\rangle+\alpha_{opt}\langle r^{(k)},Ax^{(k)}\rangle+\alpha_{opt}^{2}\langle r^{(k)},Ar^{(k)}\rangle\right)-\alpha_{opt}\langle r^{(k)},\underbrace{Ax^{*}}_{b}\rangle$$

$$\underbrace{=}_{\substack{A\ is\ SPD+\\ Energy\ Norm\ Symmetry}}\quad f\left(x^{(k)}\right)+\frac{1}{2}\left(2\alpha_{opt}\langle x^{(k)},Ar^{(k)}\rangle+\alpha_{opt}^{2}\langle r^{(k)},Ar^{(k)}\rangle\right)-\alpha_{opt}\langle r^{(k)},A\underbrace{\left(e^{(k)}+x^{(k)}\right)}_{x^{*}}\rangle$$

$$=f\left(x^{(k)}\right)+\alpha_{opt}\langle x^{(k)},Ar^{(k)}\rangle+\frac{1}{2}\alpha_{opt}^{2}\langle r^{(k)},Ar^{(k)}\rangle-\alpha_{opt}\langle r^{(k)},Ae^{(k)}\rangle-\alpha_{opt}\langle r^{(k)},Ax^{(k)}\rangle$$

$$=f\left(x^{(k)}\right)+\frac{1}{2}\alpha_{opt}^{2}\langle r^{(k)},Ar^{(k)}\rangle-\alpha_{opt}\langle r^{(k)},Ae^{(k)}\rangle$$

$$=f\left(x^{(k)}\right)+\frac{1}{2}\left(\frac{\langle r^{(k)},r^{(k)}\rangle}{\langle r^{(k)},Ar^{(k)}\rangle}\right)^{2}\langle r^{(k)},Ar^{(k)}\rangle-\frac{\langle r^{(k)},r^{(k)}\rangle}{\langle r^{(k)},Ar^{(k)}\rangle}\langle r^{(k)},\underbrace{Ae^{(k)}}_{r^{k}}\rangle$$

$$=f\left(x^{(k)}\right)+\frac{1}{2}\frac{\left(\langle r^{(k)},r^{(k)}\rangle\right)^{2}}{\langle r^{(k)},Ar^{(k)}\rangle}-\frac{\left(\langle r^{(k)},r^{(k)}\rangle\right)^{2}}{\langle r^{(k)},Ar^{(k)}\rangle}$$

$$=f\left(x^{(k)}\right)-\frac{1}{2}\frac{\left(\langle r^{(k)},r^{(k)}\rangle\right)^{2}}{\langle r^{(k)},Ar^{(k)}\rangle}$$

$$=f\left(x^{(k)}\right)-\frac{1}{2}\frac{\left(\langle r^{(k)},Ae^{(k)}\rangle\right)^{2}}{\langle r^{(k)},Ar^{(k)}\rangle}$$

Since $\langle r^{(k)},Ar^{(k)}\rangle$ is the energy norm (and because A is SPD), $\langle r^{(k)},Ar^{(k)}\rangle\geq 0$. Also, $\left(\langle r^{(k)},Ae^{(k)}\rangle\right)^{2}\geq 0$.

Thus, for $r^{(k)} \neq 0$, we get that $\frac{\left(\langle r^{(k)}, Ae^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle} > 0$.

And finally,

$$f\left(x^{(k+1)}\right) = f\left(x^{(k)}\right) - \frac{1}{2}\frac{\left(\langle r^{(k)}, Ae^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle} < f\left(x^{(k)}\right)$$

ii. We'll find an expression for $C^{(k)}$ that does not depend on x^* or $e^{(k)}$:

$$C^{(k)} f\left(x^{(k)}\right) = f\left(x^{(k)}\right) - \frac{1}{2}\frac{\left(\langle r^{(k)}, Ae^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle} \underbrace{\Rightarrow}_{for\ f\left(x^{(k)}\right) \neq 0}$$

$$C^{(k)} = 1 - \frac{1}{2}\frac{\left(\langle r^{(k)}, Ae^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle * f\left(x^{(k)}\right)} = 1 - \frac{1}{2}\frac{\left(\langle r^{(k)}, Ae^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle * \frac{1}{2}\|x^* - x^{(k)}\|_A^2} = 1 - \frac{1}{2}\frac{\left(\langle r^{(k)}, Ae^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle * \frac{1}{2}\|e^{(k)}\|_A^2}$$

$$= 1 - \frac{\left(\langle r^{(k)}, Ae^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\langle e^{(k)}, Ae^{(k)}\rangle} = 1 - \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\langle \underbrace{A^{-1}r^{(k)}}_{e^k}, r^{(k)}\rangle}$$

$$= 1 - \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\left(A^{-1}r^{(k)}\right)^T r^{(k)}} \underbrace{=}_{A^{-1} is\ symmetric} 1 - \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\left(r^{(k)}\right)^T A^{-1} r^{(k)}}$$

$$= 1 - \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\langle r^{(k)}, A^{-1}r^{(k)}\rangle}$$

We see that $C^{(k)} < 1$ because the expression of the right must be positive ($A, A^{-1} \succ 0$ thus the denominator is positive).

iii. We will use the Rayleigh quotient, and the following facts:

- For an $A \succ 0$, SPD matrix, all eigenvalues are positive.
- The eigenvalues of an inverse to $A \succ 0$ are $\frac{1}{\lambda_i}$ where λ_i are A's eigenvalues.

We will show that

$$1 - \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\langle r^{(k)}, A^{-1}r^{(k)}\rangle} \leq 1 - \frac{\lambda_{min}}{\lambda_{max}}$$

$$\rightarrow \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\langle r^{(k)}, A^{-1}r^{(k)}\rangle} \geq \frac{\lambda_{min}}{\lambda_{max}}$$

First, we can see that due to Rayleigh quotient:

$$\frac{\langle r^{(k)}, Ar^{(k)}\rangle}{\langle r^{(k)}, r^{(k)}\rangle} \leq \lambda_{max}$$

Also, because all the eigenvalues and inner products are positive, we can raise every side by -1:

$$\underbrace{\frac{\langle r^{(k)}, r^{(k)}\rangle}{\langle r^{(k)}, Ar^{(k)}\rangle} \geq \frac{1}{\lambda_{max}}}_{(*)}$$

Also, A^{-1} is symmetric and its eigenvalues are $\frac{1}{\lambda} > 0$. In addition, the largest eigenvalue for A^{-1} is $\frac{1}{\lambda_{min}}$.

We apply the same method as above, for A^{-1} and receive the following:

$$\underbrace{\frac{\langle r^{(k)}, r^{(k)}\rangle}{\langle r^{(k)}, A^{-1}r^{(k)}\rangle} \geq \frac{1}{\frac{1}{\lambda_{min}}} = \lambda_{min}}_{(**)}$$

Combining $(*)$ and $(**)$ we get:

$$\frac{\langle r^{(k)}, r^{(k)}\rangle}{\langle r^{(k)}, Ar^{(k)}\rangle}\frac{\langle r^{(k)}, r^{(k)}\rangle}{\langle r^{(k)}, A^{-1}r^{(k)}\rangle} \geq \frac{1}{\lambda_{max}}\lambda_{min}$$

$$\Longrightarrow \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\langle r^{(k)}, A^{-1}r^{(k)}\rangle} \geq \frac{\lambda_{min}}{\lambda_{max}}$$

Thus concluding:

$$1 - \frac{\left(\langle r^{(k)}, r^{(k)}\rangle\right)^2}{\langle r^{(k)}, Ar^{(k)}\rangle\langle r^{(k)}, A^{-1}r^{(k)}\rangle} \leq 1 - \frac{\lambda_{min}}{\lambda_{max}} \underbrace{<}_{\substack{A>0\to \\ \lambda_i>0 \text{ for every i}}} 1$$

iv. Using the proofs in the previous sections:

$$f\left(x^{(k)}\right) = C^{(k-1)}f\left(x^{(k-1)}\right) = \prod_{i=1}^{k-1} C^{(i)}\, f\left(x^{(0)}\right) \underbrace{\rightarrow}_{\substack{k\to\infty \\ C^{(i)}<1}} 0$$

And hence,

$$\left\|x^* - x^{(k)}\right\|_A^2 \underbrace{\rightarrow}_{k\to\infty} 0 \Rightarrow x^{(k)} \underbrace{\rightarrow}_{k\to\infty} x^*$$

Problem 3 – **Section A:**

Let $g(\alpha^{(k)}) = \|b - Ax^{(k+1)}\|_2$. We want to find $\alpha = \underset{\alpha^{(k)}}{\operatorname{argmin}}\, g(\alpha^{(k)})$

$$g(\alpha^{(k)}) = \|b - A(x^{(k)} - \alpha^{(k)}r^{(k)})\|_2 = \|b - Ax^{(k)} - \alpha^{(k)}Ar^{(k)}\|_2 = \|r^{(k)} - \alpha^{(k)}Ar^{(k)}\|_2$$

$$= \sqrt{\langle r^{(k)} - \alpha^{(k)}Ar^{(k)}, r^{(k)} - \alpha^{(k)}Ar^{(k)}\rangle}$$

$$= \sqrt{(r^{(k)} - \alpha^{(k)}Ar^{(k)})^T(r^{(k)} - \alpha^{(k)}Ar^{(k)})}$$

$$= \sqrt{(r^{(k)})^T r^{(k)} - (\alpha^{(k)}Ar^{(k)})^T r^{(k)} - (r^{(k)})^T\alpha^{(k)}Ar^{(k)} + (\alpha^{(k)}Ar^{(k)})^T\alpha^{(k)}Ar^{(k)}}$$

$$= \sqrt{(r^{(k)})^T r^{(k)} - \alpha^{(k)}(r^{(k)})^T A^T r^{(k)} - \alpha^{(k)}(r^{(k)})^T Ar^{(k)} + (\alpha^{(k)})^2(r^{(k)})^T A^T Ar^{(k)}}$$

$$= \sqrt{\langle r^{(k)}, r^{(k)}\rangle - \alpha^{(k)}\langle Ar^{(k)}, r^{(k)}\rangle - \alpha^{(k)}\langle r^{(k)}, Ar^{(k)}\rangle + (\alpha^{(k)})^2\langle Ar^{(k)}, Ar^{(k)}\rangle}$$

$$\underbrace{=}_{\substack{dot\ product \\ symmetry}} \sqrt{\langle r^{(k)}, r^{(k)}\rangle - 2\alpha^{(k)}\langle r^{(k)}, Ar^{(k)}\rangle + (\alpha^{(k)})^2\langle Ar^{(k)}, Ar^{(k)}\rangle}$$

To find the minimum, we will derive the expression and look for the root:

$$\frac{d}{d\alpha^{(k)}} g(\alpha^{(k)}) = \frac{-2\langle r^{(k)}, Ar^{(k)}\rangle + 2\alpha^{(k)}\langle Ar^{(k)}, Ar^{(k)}\rangle}{2g(\alpha^{(k)})} = 0$$

This equality holds for:

$$2\langle r^{(k)}, Ar^{(k)}\rangle \ = 2\alpha^{(k)}\langle Ar^{(k)}, Ar^{(k)}\rangle \Longrightarrow \alpha^{(k)} = \frac{\langle r^{(k)}, Ar^{(k)}\rangle}{\langle Ar^{(k)}, Ar^{(k)}\rangle} = \frac{\left(r^{(k)}\right)^T Ar^{(k)}}{(r^{(k)})^T A^T Ar^{(k)}}$$

Problem 3 – **Section B:**

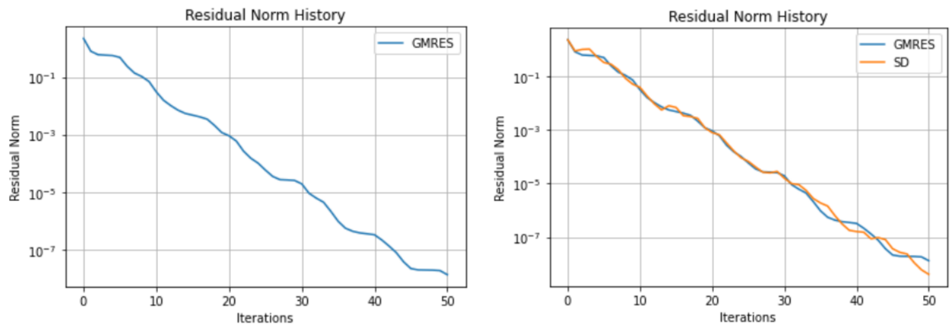
When computing $\alpha^{(k)}$ in iteration $k+1$ of the iterative algorithm, we need to calculate $Ar^{(k)}$ (let's say for the numerator). This value can be saved aside and used to calculate the denominator (for both expressions of the dot product).

Next, when updating the residual, instead of calculating $r^{(k+1)} = b - Ax^{(k)}$, we can calculate $r^{(k+1)} = r^{(k)} - \alpha^{(k)}Ar^{(k)}$ and use the previously calculated value once more.

Problem 3 – **Section C:**

```
def gmres(A, b, x_0, max_iter, eps, verbose=True):
    x = [x_0]
    r = [b - A@x[0]]
    if verbose:
        print(f"Iter: 0: {x[0]}")
    for k in range(1, max_iter + 1):
        A_r = A@r[k - 1]
        alpha = (r[k - 1] @ A_r) / (A_r @ A_r)
        x.append(x[k - 1] + alpha*r[k - 1])
        r.append(r[k - 1] - alpha*A_r)
        if verbose:
            print(f"Iter: {k}: {x[k]}")
        if (np.linalg.norm(r[k]) / np.linalg.norm(b)) < eps:
            break
    return x[-1], np.array([np.linalg.norm(res) for res in r])
```

```
x_gmres, res_gmres = gmres(A, b, np.zeros(5), 50, 1e-10, verbose=False)
```



We've also added a comparison to SD. In the residual norm history plot for SD, the graph is not monotone while it is monotone for GMRES.

Problem 3 – Section D:

The reason the residual history is monotone is because in each step, α is chosen in such a way that the residual norm is minimal. There can't be a case where $\|r^{(k+1)}\|_2 > \|r^{(k)}\|_2$ because we can just select $\alpha^{(k)} = 0$ and get $\|r^{(k+1)}\|_2 = \|r^{(k)}\|_2$.

Formally, we will prove by contradiction - let's assume that there is a $k \in N$ s.t $\|r^{(k+1)}\|_2 > \|r^{(k)}\|_2$. We know that,

$$r^{(k+1)} = Ax^{(k+1)} - b = A\left(x^{(k)} + \alpha^{(k)} r^{(k)}\right) - b = r^{(k)} + \alpha^{(k)} A r^{(k)}$$

$$\Longrightarrow \|r^{(k+1)}\|_2 = \|r^{(k)} + \alpha^{(k)} A r^{(k)}\|_2$$

Since we need to choose an optimal α that minimizes the residual norm ($\alpha_{opt}^{(k)} = argmin\|r^{(k+1)}\|_2$), we get:

$$\|r^{(k+1)}\|_2 = \|r^{(k)} + \alpha_{opt}^{(k)} A r^{(k)}\|_2 \underbrace{\leqq}_{\substack{for\ any \\ other\ \alpha}} \|r^{(k)} + 0 * A r^{(k)}\|_2 = \|r^{(k)}\|_2$$

In contradiction to the fact that $\|r^{(k+1)}\|_2 > \|r^{(k)}\|_2$.

Problem 3 – Section E:

Let $g(\vec{\alpha}^{(k)}) = \|b - Ax^{(k+1)}\|_2$. We want to find $\vec{\alpha} = \underset{\vec{\alpha}^{(k)}}{\operatorname{argmin}}\, g(\vec{\alpha}^{(k)})$

$$g(\vec{\alpha}^{(k)}) = \|b - A(x^{(k)} + R^{(k)}\vec{\alpha}^{(k)})\|_2 = \|r^{(k)} - AR^{(k)}\vec{\alpha}^{(k)}\|_2$$

$$= \sqrt{\langle r^{(k)} - AR^{(k)}\vec{\alpha}^{(k)}, r^{(k)} - AR^{(k)}\vec{\alpha}^{(k)}\rangle}$$

$$= \sqrt{(r^{(k)} - AR^{(k)}\vec{\alpha}^{(k)})^T (r^{(k)} - AR^{(k)}\vec{\alpha}^{(k)})}$$

$$= \sqrt{((r^{(k)})^T - (\vec{\alpha}^{(k)})^T (R^{(k)})^T A^T)(r^{(k)} - AR^{(k)}\vec{\alpha}^{(k)})}$$

$$= \sqrt{(r^{(k)})^T r^{(k)} - (\vec{\alpha}^{(k)})^T (R^{(k)})^T A^T r^{(k)} - (r^{(k)})^T AR^{(k)}\vec{\alpha}^{(k)} + (\vec{\alpha}^{(k)})^T (R^{(k)})^T A^T AR^{(k)}\vec{\alpha}^{(k)}}$$

$$= \sqrt{\langle r^{(k)}, r^{(k)}\rangle - \langle AR^{(k)}\vec{\alpha}^{(k)}, r^{(k)}\rangle - \langle r^{(k)}, AR^{(k)}\vec{\alpha}^{(k)}\rangle + \langle AR^{(k)}\vec{\alpha}^{(k)}, AR^{(k)}\vec{\alpha}^{(k)}\rangle}$$

$$\underbrace{=}_{\substack{dot\ product \\ symmetry}} \sqrt{\langle r^{(k)}, r^{(k)}\rangle - 2\langle r^{(k)}, AR^{(k)}\vec{\alpha}^{(k)}\rangle + \langle AR^{(k)}\vec{\alpha}^{(k)}, AR^{(k)}\vec{\alpha}^{(k)}\rangle}$$

To find the minimum, we will derive the expression and look for the root:

$$\nabla g\left(\vec{\alpha}^{(k)}\right) = \frac{-2\left(r^{(k)}\right)^T AR^{(k)} + 2\left(AR^{(k)}\right)^T AR^{(k)}\vec{\alpha}^{(k)}}{2g(\vec{\alpha}^{(k)})} = 0$$

This holds when:

$$-2\left(r^{(k)}\right)^T AR^{(k)} + 2\left(AR^{(k)}\right)^T AR^{(k)}\vec{\alpha}^{(k)} = \vec{0} \Longrightarrow$$

$$\left(AR^{(k)}\right)^T AR^{(k)}\vec{\alpha}^{(k)} = \left(r^{(k)}\right)^T AR^{(k)}$$

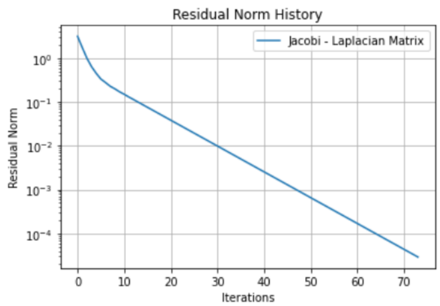
$$\vec{\alpha}^{(k)} = \left(\left(AR^{(k)}\right)^T AR^{(k)}\right)^{-1} \left(r^{(k)}\right)^T AR^{(k)}$$

<u>Problem 4</u> – Section A:

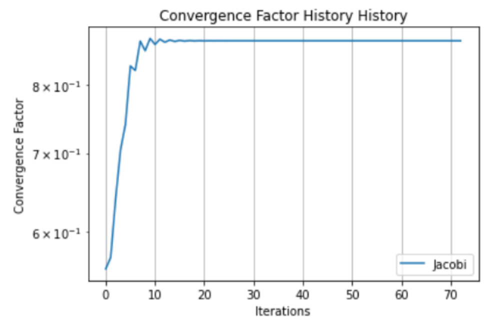
Standard Jacobi needed **74 iterations** to converge.

The convergence factor is ~ 0.87.

```
Final Residual: 2.9189073430572584e-05
Number of Iterations: 74
```




```
(Jacobi) Convergence Factor: 0.8732094848469012
```



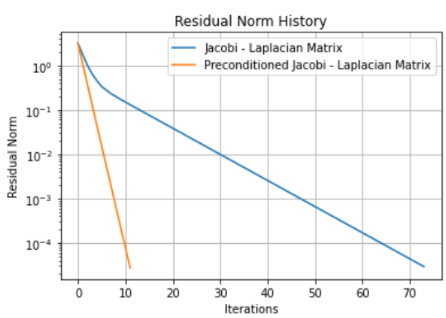
Final solution: $x = [1.24961, 0.58295, 0.91629, -0.08366, -0.09123, -0.59312, -0.13858, -0.54956, 0.2705\,, -0.3753]$

<u>Problem 4</u> – Section B:

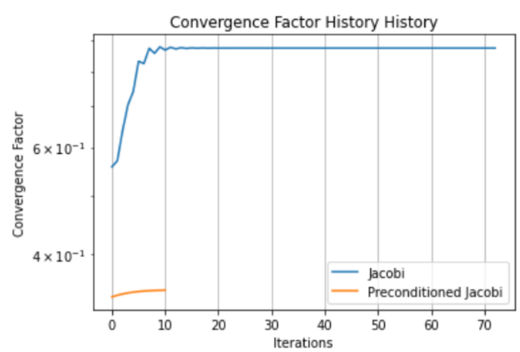
Yes, the method converged in fewer iteration than the standard Jacobi. It took **12 iterations** to converge (with $\omega = 0.65$).

The convergence factor is ~ 0.35.

```
(Jacobi) Final Residual: 2.9189073430572584e-05
(Jacobi) Number of Iterations: 74
(Preconditioned Jacobi) Final Residual: 2.742292947714525e-05
(Preconditioned Jacobi) Number of Iterations: 12
```




```
(Jacobi) Convergence Factor: 0.8732094848469012
(Preconditioned Jacobi) Convergence Factor: 0.34947375687
```



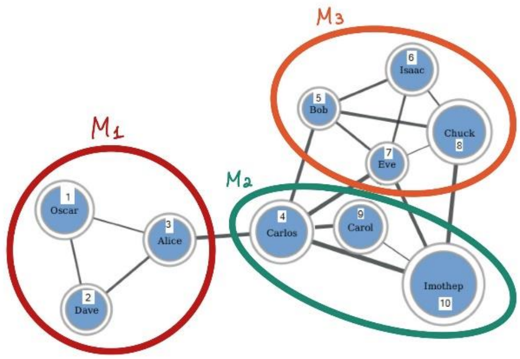
Final solution: $x = \; [0.83333, 0.16667, 0.5, -0.65909, -0.19887, -0.6534\,, -0.28409, -0.5, -0.0625, -0.625]$

<u>Problem 4</u> – Section C:

After some experimentation, we have noticed that the relaxation applied in section (b) was, in some sense, splitting the graph into two separate connected components.

Following this logic, we would like to split the graph into three separate connected components, such that the minimum number of edges will be lost.

We have chosen the following structure:



To do so, we modified L such that "Carlos" is 8th, "Bob" is 4th, "Isaac" is 5th, "Eve" is 6th & "Chuck" is 7th.

Now we can split our matrix into three separate sub-matrices and compute their inverse:

$$M_1 = L_{modified}[1:3,1:3],\ \ M_2 = L_{modified}[4:7,4:7],\ \ M_3 = L_{modified}[8:10,8:10]$$

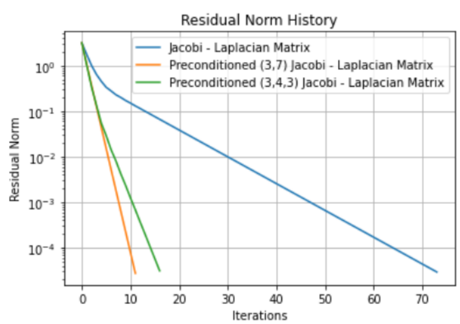
$$M = \begin{pmatrix} M_1 & 0 & 0 \\ 0 & M_2 & 0 \\ 0 & 0 & M_3 \end{pmatrix}, \qquad M^{-1} = \begin{pmatrix} M_1^{-1} & 0 & 0 \\ 0 & M_2^{-1} & 0 \\ 0 & 0 & M_3^{-1} \end{pmatrix}$$

This preconditioner was less powerful than the one used in section 4(b), but as stated in the assignment – is easier to compute.

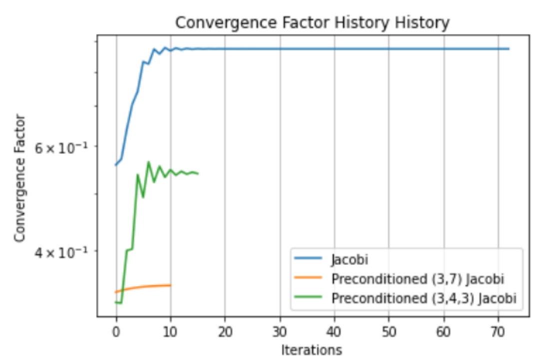
Still, the method converged in fewer iteration than the standard Jacobi. It took **17 iterations** to converge (with $\omega = 0.69$).

The convergence factor is ~ 0.54.

```
(Jacobi) Final Residual: 2.9189073430572584e-05
(Jacobi) Number of Iterations: 74
(Preconditioned (3,7) Jacobi) Final Residual: 2.7422929477145
(Preconditioned (3,7) Jacobi) Number of Iterations: 12
(Preconditioned (3,4,3) Jacobi) Final Residual: 3.10673598314
(Preconditioned (3,4,3) Jacobi) Number of Iterations: 17
```




```
(Jacobi) Convergence Factor: 0.8732094848469012
(Preconditioned (3,7) Jacobi) Convergence Factor: 0.349473756
(Preconditioned (3,4,3) Jacobi) Convergence Factor: 0.5389014
```



Final solution: $x = [1.28332,\ 0.61665,\ 0.94999, -0.20909,\ 0.25114, -0.2034,\ 0.16592, -0.05,\ 0.3875, -0.175]$