Assignment – 2

By: Maor Ashkenazi, Pan Eyal

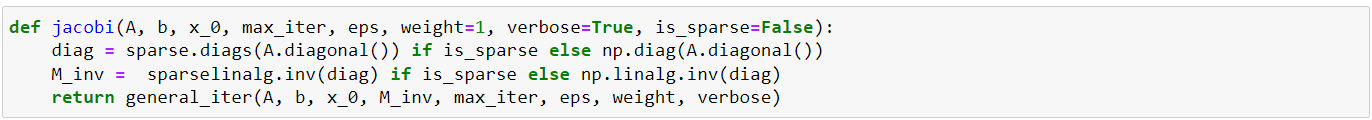
**Problem 1 – Section A:**

**General Iterative Method:**

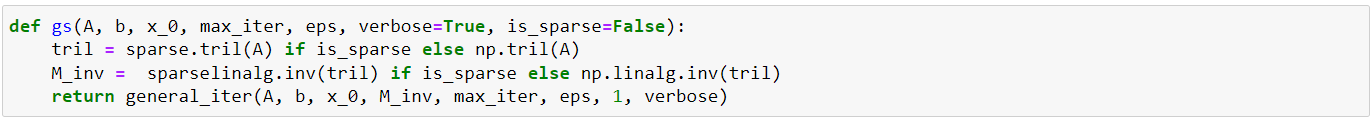
**Graphical user interface, text, application

Description automatically generated**

**Jacobi:**

****

**Gauss-Seidel:**

****

**Steepest Descent:**

**A picture containing text

Description automatically generated**

**Conjugate Gradient:**

**Graphical user interface, text, application

Description automatically generated**

**Problem 1 – Section B:**

**Residual Norm History Plot:**

**Chart, line chart

Description automatically generated**

**Convergence Factor Plot:**

**Chart, line chart

Description automatically generated**

**Problem 2 – Section A:**

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

converges for any starting vector .

From the SPD property, we know that is invertible.

For the Richardson method: , thus: . In our case, .

We will show that if are ’s eigenpairs, are ’s eigenvalues:

Also, we know that ’s spectral radius is .

Using the induced norm definition, we know that .

Therefore, for every . Also, because is SPD, we know that .

Using the inequality above, we get .

**Problem 2 – Section B:**

For and as defined in the previous section, we have shown that ’s eigenvalues are: where are ’s eigenvalues.

For the case that has positive and negative eigenvalues:

For any negative we will get that: , thus, 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:

**Problem 2 – Section C:**

1. Using formula (44) in subsection (6.1.7), we set in the function expression:

As a reminder, the step for steepest descent is:

We continue developing the expression:

Since is the energy norm (and because is SPD), . Also, .

Thus, for , we get that .

And finally,

1. We’ll find an expression for that does not depend on or :

We see that because the expression of the right must be positive ( thus the denominator is positive).

1. We will use the Rayleigh quotient, and the following facts:
   * For an , SPD matrix, all eigenvalues are positive.
   * The eigenvalues of an inverse to are where are ’s eigenvalues.

We will show that

First, we can see that due to Rayleigh quotient:

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

Also, is symmetric and its eigenvalues are . In addition, the largest eigenvalue for is .

We apply the same method as above, for and receive the following:

Combining and we get:

Thus concluding:

1. Using the proofs in the previous sections:

And hence,

**Problem 3 – Section A:**

Let . We want to find

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

This equality holds for:

**Problem 3 – Section B:**

When computing in iteration of the iterative algorithm, we need to calculate (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 , we can calculate and use the previously calculated value once more.

**Problem 3 – Section C:**

Graphical user interface, text, application

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

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, is chosen in such a way that the residual norm is minimal. There can’t be a case where because we can just select and get .

Formally, we will prove by contradiction - let’s assume that there is a s.t . We know that,

Since we need to choose an optimal that minimizes the residual norm (), we get:

In contradiction to the fact that .

**Problem 3 – Section E:**

Let . We want to find

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

This holds when:

**Problem 4 – Section A:**

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

The convergence factor is .

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated | Chart, histogram  Description automatically generated |

Final solution:

**Problem 4 – Section B:**

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

The convergence factor is .

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated | Chart, histogram  Description automatically generated |
|  |  |

Final solution:

**Problem 4 – 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:

Diagram

Description automatically generated

To do so, we modified 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:

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 ).

The convergence factor is .

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated | Chart, histogram  Description automatically generated |

Final solution: