# Assignment 04
# Iterative Methods

Mayank Pathania
204103314

March 15, 2021

# 1 Algorithms

## 1.1 Gauss-Seidel

Algorithm 1: Gauss-Seidel

```
procedure GAUSS_SEIDEL(A[][], B[], ε, Kmax)
                                        ▷ A[][] is system matrix
                                        ▷ B[] is RHS vector
                                        ▷ ε is tolerace/error
                                        ▷ Kmax is maximum number of iterations
    error ← 1000                        ▷ large value(more than ε)
    k ← 0                               ▷ iteration counter
    initialize x with random values
    while k < Kmax and error > ε do
        for i←1:1:n do
            val ← 0
            for j←1:1:n do
                if j≠i then
                    val ← val + A[i][j]*x[j]
                end if
            end for
            x[i] ← (B[i]−val)/A[i][i]
        end for
        error ← 0
        for i←1:1:n do
            residue ← 0
            for j←1:1:n do
                residue ← residue + A[i][j]*x[j]
            end for
            residue ← B[i] - residue
            error ← error + residue^2
```

**end for**
error $\leftarrow \sqrt{error}$
k $\leftarrow k + 1$
**end while**
**return** x
**end procedure**

## 1.2 Conjugate Gradient

Algorithm 2: Conjugate Gradient

**procedure** $conjugate\_gradient(A[][], B[], \epsilon, Kmax)$
▷ A[][] is system matrix
▷ B[] is RHS vector
▷ $\epsilon$ is tolerace/error
▷ Kmax is maximum number of iterations
error $\leftarrow$ 1000 ▷ large value(more than $\epsilon$)
k $\leftarrow$ 0 ▷ iteration counter
initialize x with random values
d $\leftarrow$ compute B - Ax
r $\leftarrow$ d
**while** k $<$ Kmax and error $>$ $\epsilon$ **do**
$v \leftarrow computeA * d$
$\alpha \leftarrow \frac{r^T r}{d^T v}$
**for** i$\leftarrow$1:1:n **do**
x[i] $\leftarrow x[i] + \alpha * d[i]$
**end for**
$error \leftarrow 0$
initialize vector $r_{next}$ of size r
**for** i$\leftarrow$1:1:n **do**
$r_{next}[i] \leftarrow r[i] - \alpha * v[i]$
error $\leftarrow error + r_{next}^2$
**end for**
$\beta \leftarrow \frac{r_{next}^T r_{next}}{d^T r}$
**for** i$\leftarrow$1:1:n **do**
$d[i] \leftarrow r_{next} + \beta * d[i]$
**end for**
r $\leftarrow r_{next}$
error $\leftarrow \sqrt{error}$
k $\leftarrow k + 1$
**end while**
**return** x
**end procedure**

# 2 Problem

$$\begin{bmatrix} 0.2 & 0.1 & 1 & 1 & 0 \\ 0.1 & 47 & -1 & 1 & -1 \\ 1 & -1 & 60 & 0 & -2 \\ 1 & 1 & 0 & 8 & 4 \\ 0 & -1 & -2 & 4 & 700 \end{bmatrix} \begin{Bmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{Bmatrix}$$

# 3 Iterations

## 3.1 Parameters

- $\epsilon = 10^{-10}$
- $Kmax = 1000$

## 3.2 Initial Guess

$$\text{x} = \begin{Bmatrix} -24 \\ -64 \\ 47 \\ 53 \\ -42 \end{Bmatrix}$$

## 3.3 Conjugate Gradient

| iteration | error | x1 | x2 | x3 | x4 | x5 |
|---|---|---|---|---|---|---|
| 1 | 2649.46 | -24.1265 | -63.6939 | 42.7614 | 52.7636 | 0.116514 |
| 2 | 352.151 | -25.5418 | -59.4667 | -1.45776 | 47.1715 | -0.355099 |
| 3 | 135.927 | -31.3974 | -26.4973 | 0.115133 | -1.61659 | -0.0207585 |
| 4 | 2.19193 | -30.2297 | 0.329565 | 0.559426 | 4.28057 | -0.0152505 |
| 5 | 0.00290896 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.010622 |
| 6 | 4.31393e-08 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.0106262 |
| 7 | 1.90982e-12 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.0106262 |

### 3.4 Gauss Seidel

| iteration | error | x1 | x2 | x3 | x4 | x5 |
|---|---|---|---|---|---|---|
| 1 | 187.81 | -463 | 0.075 | 6.36792 | 79.3656 | -0.428074 |
| 10 | 2.07507 | -20.4058 | 0.0210206 | 0.389874 | 3.05671 | -0.00918013 |
| 11 | 1.47584 | -12.2435 | 0.137081 | 0.256036 | 2.01789 | -0.00346056 |
| 20 | 0.0687228 | 6.92361 | 0.409616 | -0.058243 | -0.421505 | 0.00997021 |
| 30 | 0.00227598 | 7.82871 | 0.422486 | -0.0730839 | -0.536697 | 0.0106044 |
| 31 | 0.00161873 | 7.83766 | 0.422613 | -0.0732307 | -0.537837 | 0.0106107 |
| 40 | 7.53767e-05 | 7.85869 | 0.422912 | -0.0735754 | -0.540512 | 0.0106254 |
| 41 | 5.36098e-05 | 7.85898 | 0.422916 | -0.0735803 | -0.54055 | 0.0106257 |
| 50 | 2.49635e-06 | 7.85968 | 0.422926 | -0.0735917 | -0.540639 | 0.0106261 |
| 51 | 1.77546e-06 | 7.85969 | 0.422926 | -0.0735918 | -0.54064 | 0.0106261 |
| 60 | 8.26748e-08 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.0106262 |
| 61 | 5.88004e-08 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.0106262 |
| 70 | 2.73805e-09 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.0106262 |
| 71 | 1.94737e-09 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.0106262 |
| 80 | 9.06798e-11 | 7.85971 | 0.422926 | -0.0735922 | -0.540643 | 0.0106262 |

# 4 Solution

Solution by both gauss seidel and conjugate gradient method is same

$$\text{x} = \begin{Bmatrix} 7.85971 \\ 0.422926 \\ -0.0735922 \\ -0.540643 \\ 0.0106262 \end{Bmatrix}$$

# 5 Conclusion

- For same parameters $\epsilon$ and initial guess Gauss Seidel Method tool more iterations than Conjugate Gradient.
- Various initial guess were taken in range (-100, 100) and it was noticed that conjugate gradient took 7 iterations to converge to solution while gauss-seidel method took around 75 to 80 iterations. $\epsilon$ was kept constant at $\epsilon = 10^{-10}$.