```
# Prof. Dr. Serkan Dağ
# ME 310 Numerical Methods
# File 6.1
# Gauss-Seidel Iterative Solver
# Solves an n by n Linear System
# Relaxation not included
# [A] needs to be diagonally-dominant
restart :
with(LinearAlgebra) :
Digits := 16 :
unprotect(sum) :
# Number of significant figures and percent tolerance
t := 3 :
eps_s := 0.5·10^(2 - t) :
# Number of maximum iterations
lmax := 20 :
# Number of Equations
n := 4 :
# Define Coefficient Matrix
A := Matrix(n, n) :
A[1, 1] := 20 :
A[1, 2] := 0.3 :
A[1, 3] := -0.7 :
A[1, 4] := 0.2 :
A[2, 1] := -0.8 :
A[2, 2] := 10. :
A[2, 3] := 0.5 :
A[2, 4] := 0.2 :
A[3, 1] := -0.05 :
A[3, 2] := -0.1 :
A[3, 3] := 12. :
A[3, 4] := 0.6 :
A[4, 1] := 0.2 :
A[4, 2] := 0.5 :
A[4, 3] := -0.7 :
A[4, 4] := -3. :
A;
```

$$\begin{bmatrix} 20 & 0.3 & -0.7 & 0.2 \\ -0.8 & 10. & 0.5 & 0.2 \\ -0.05 & -0.1 & 12. & 0.6 \\ 0.2 & 0.5 & -0.7 & -3. \end{bmatrix} \quad (1)$$

```
# Define Right Hand Side Vector
B := Matrix(n, 1) :
B[1, 1] := 2. :
B[2, 1] := 7. :
B[3, 1] := -2. :
B[4, 1] := 4. :
B;
```

$$\begin{bmatrix} 2. \\ 7. \\ -2. \\ 4. \end{bmatrix} \quad (2)$$

```
> # Invoke LinearSolve command for comparison
> LinearSolve(A, B);
```

$$\begin{bmatrix} 0.09722012769996965 \\ 0.7364429705717713 \\ -0.1010984653223441 \\ -1.180521854482826 \end{bmatrix} \quad (3)$$

```
> # Each equation is divided by its diagonal coefficient
> C := Matrix(n, 1) :
> for i from 1 by 1 to n
  while true do
   C[i, 1] := A[i, i] :
  end do:
> for i from 1 by 1 to n
  while true do
  B[i, 1] := B[i, 1] / C[i, 1] :
  for j from 1 by 1 to n
  while true do
   A[i, j] := A[i, j] / C[i, 1] :
  end do:
  end do:

> # Initialize the unknowns
> X := Matrix(n, 1) :
  Xold := Matrix(n, 1) :
   Xsum := Matrix(n, 1) :
> for i from 1 by 1 to n
  while true do
  X[i, 1] := 0 :
  Xold[i, 1] := 0 :
  end do:
> # Unleash the iterations
> for l from 1 by 1 to lmax
  while true do

  sum := 0 :

  for i from 1 by 1 to n
  while true do

     Xsum[i, 1] := B[i, 1] :
```

for j **from** 1 **by** 1 **to** n
while *true* **do**

if $i \neq j$ **then**

$Xsum[i, 1] := Xsum[i, 1] - A[i, j] \cdot X[j, 1]$:

end if:

end do:

$X[i, 1] := Xsum[i, 1]$:

$$epsa := \text{abs}\left(\frac{(X[i, 1] - Xold[i, 1])}{X[i, 1]} \right) \cdot 100 :$$

if $epsa < eps\_s$ **then**
$sum := sum + 1$:
end if:

end do:

if $sum = n$ **then**

for k **from** 1 **by** 1 **to** n
while *true* **do**

printf("\n %5.1f %5.1f %15.10f", l, k, $X[k, 1]$) :

end do:
break:

else

for k **from** 1 **by** 1 **to** n
while *true* **do**
$Xold[k, 1] := X[k, 1]$:
printf("\n %5.1f %5.1f %15.10f", l, k, $X[k, 1]$) :
end do:

end if:

end do:

```
 1.0   1.0    0.1000000000
 1.0   2.0    0.7080000000
 1.0   3.0   -0.1603500000
 1.0   4.0   -1.1712516667
 2.0   1.0    0.0954802667
 2.0   2.0    0.7390809547
 2.0   3.0   -0.1015472409
 2.0   4.0   -1.1800934669
 3.0   1.0    0.0971605669
 3.0   2.0    0.7364520767
```

```
    3.0   3.0   -0.1011200570
    3.0   4.0   -1.1805192695
    4.0   1.0    0.0972192095
    4.0   2.0    0.7364439250
    4.0   3.0   -0.1010985904
    4.0   4.0   -1.1805217274
    5.0   1.0    0.0972201077
    5.0   2.0    0.7364429727
    5.0   3.0   -0.1010984717
    5.0   4.0   -1.1805218540
> 
```