Workshop 6 - HANDIN 3 - 517 28018

EXERCISE 2

i) We know one step of the Jacobi method is:

$$Dx_k = (b - (L+U)x_{k-1})$$

we can re-arange the above equation

$$Dx_k = (b - (L+U)x_{k-1})$$
$$x_k = D^{-1}(b-(Lx_{k-1} + Ux_{k-1}))$$
$$x_k = D^{-1}(b - Lx_{k-1} - Ux_{k-1})$$

Note : $D^{-1} = \begin{pmatrix} 1/a_{11} & 0 & \cdots & 0 \\ 0 & 1/a_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/a_{nn} \end{pmatrix}$ due to the fact that D is a diagonal matrix.

Also $L = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ a_{21} & 0 & & \vdots \\ a_{31} & & 0 & \vdots \\ \vdots & & \ddots & 0 \\ a_{n1} & \cdots & a_{nn-1} & 0 \end{pmatrix}$, $U = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & 0 & & \vdots \\ \vdots & & \ddots & a_{n-1n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix}$ due to definition of how we split.

For the iteration step, we want to iterate over the rows of x_k, $\therefore$ we want i from 1 to n

$$x_k = \underset{①}{D^{-1}} (b - \underset{②}{Lx_{k-1}} - \underset{③}{Ux_{k-1}})$$

① Since we have D^{-1} the value of x_k will have the value of row i times column of $(b - Lx_{k-1} - Ux_{k-1})$
$\therefore$ for $(x_k)_i = (D^{-1})_i (b - Lx_{k-1} - Ux_{k-1})$

But since the only value in row $(D^{-1})_i$ which is non zero is $1/a_{ii}$ $\therefore$ for element $(x_k)_i$ we multiply by $\frac{1}{a_{ii}}$.

(2) Lx_{k-1} for element $(x_k)_i$ will be the ith row of L times x_{k-1}

However note that L is strictly <u>lower</u> triangular $L_{ij} = 0$ for $j \geq i$, as a result we have that $(Lx_{k-1})_i = \sum_{j=1}^{n} L_{ij} x_{j_{k-1}} = \sum_{j=1}^{i-1} L_{ij} x_{j_{k-1}}$

This is how we derive the expression for (2)

(3) Ux_{k-1} for element $(x_k)_i$ we will multiply the i^{th} row of U with the x_{k-1}

However U is strictly <u>upper</u> triangular as a result $U_{ij} = 0$ for $j \leq i$ as a result when we evaluate $(Ux_{k-1})_i = \sum_{j=1}^{n} U_{ij}(x_{k-1})_j = \sum_{j=i+1}^{n} U_{ij}(x_{k-1})_j$

Putting (1), (2), (3) together we can get an expression for the iteration of Jacobi method for each row of $(x_k)_i$ where b_i is the value of b at row i

for $i = 1, \ldots n$ do

$$(x_k)_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}(x_{k-1})_j - \sum_{j=i+1}^{n} a_{ij}(x_{k-1})_j\right)$$

end for

ii) Similarly we can rearrange the one step of the Gauss-Seidel method:

$$(L+D)x_k = (b - Ux_{k-1})$$
$$Lx_k + Dx_k = (b - Ux_{k-1})$$
$$Dx_k = b - Ux_{k-1} - Lx_k$$
$$x_k = D^{-1}(b - Lx_k - Ux_{k-1})$$

Where U, L, D are defined as before. $(A = L + U + D)$

$$x_k = \underset{①}{D^{-1}} (b - \underset{②}{L x_k} - \underset{③}{U x_{k-1}})$$

Again we can analyse ①, ② and ③ for each value of the row $(x_k)_i$

① D^{-1} is defined [redacted] $(D^{-1})_{ij} = \begin{cases} 1/a_{ii} & i=j \\ 0 & \text{otherwise} \end{cases}$

hence the value which is non zero for row i will be $1/a_{ii}$ so we only need to multiply $(b - Lx_k - Ux_{k-1})$ by $1/a_{ii}$ for the value of $(x_k)_i$.

② Lx_k ~~to~~ evaluate it for row i of $(x_k)_i$. We need to multiply ~~row by~~ row i of L by x_k, where L is strictly lower diagonal.

$$\therefore \quad (L x_k)_i = \sum_{j=1}^{n} L_{ij} (x_k)_j = \sum_{j=1}^{i-1} L_{ij} (x_k)_j \quad \text{this is}$$

because for values $(L)_{ij}$ where $i \leq j$ we have that [redacted] $(L)_{ij} = 0$ so we do not need to compute them in the equation above, $i-1$ (stops us from calculating them)

③ $U x_{k-1}$ to evaluate it for row i of $(x_{k-1})_i$ we need to multiply row i of U with (x_{k-1}), where U is strictly upper triangular

$$(U x_{k-1})_i = \sum_{j=1}^{n} U_{ij} (x_{k-1})_j = \sum_{j=i+1}^{n} U_{ij} (x_{k-1})_j$$

This is because since U is strictly upper triangular $U_{ij} = 0$ for $i \geq j$, so for row i we need to 'skip' the first $i+1$ zeros and evaluate it from $i+1 \longrightarrow n$.

Again putting ①, ② and ③ together we can get a value for iteration step for each value of (x_k), where b_i is the value of b at row i.

for $i = 1, \dots n$ do

$$(x_k)_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} (x_k)_j - \sum_{j=i+1}^{n} a_{ij} (x_{k-1})_j \right)$$

end for $\square$

iii) We can see that from (i) and (ii) that the iteration steps for both of them are the same, expect that they differ by one element.

The element they differ in is $(x_k)_j$ and $(x_{k-1})_j$ used in the first summation.
Intuitively we can see that Gauss-Seidel will produce more accurate results due to the fact that itc uses information from x_k when calculating x_k, not exclusively x_{k-1}. Since it computes the entries of x_k sequentially and gaining ~~if~~ information at each iteration step.

As a result we can also see that value for $\|R\|_p$ of the 2 algorithms we will have: $\|R_J\|_p > \|R_{G-S}\|_p$ and as a result converges slower and thus has lower accuracy at the k-th iteration; (Jacobi)