Page No.
Date / /

Assignment-2

Ans-1 Given: $L \times K = B$ where $L, K \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$.
We need to device an algorithm to compute the matrix x.

Algorithm is as follows
Given: $L X K = B$ where L, K are lower triangular matrix

1. let XK be equal to A where $A \in \mathbb{R}^{n \times n}$
$XK = A$
$LA = B$

2. now since L is a lower triangular matrix we can compute the value of of A by forward substitution

3. From to previous step we are able to compute the value of A successfully
now $XK = A$ and we need to compute the value of X

$Xk = A$

4. taking transpose on both sides
$(X k)^T = (A)^T$
$k^T X^T = A^T$ $[\because (AB)^T = B^T A^T]$
$\because$ k was a lower triangular matrix
k^T will be upper triangular matrix

now we can compute X^T from eqn
$K^T X^T = A^T$ using back ward substitution

Page No.
Date / /

5. Hence from the step 4 we can get the value of X^T from eq^n $K^T X^T = A^T$ by backward substitution [$\because K^T$ was upper Δ matrix]

Now to compute X from X^T we can again take transpose of X^T [$\because (X^T)^T = X$]

and by computing this we can get the value of X

Using the above 5 mentioned steps we can compute value of X in Eq^n $L X K = B$ where $L, K \in \mathbb{R}^{n\times n}$ and are lower Δ matrices and $B \in \mathbb{R}^{n\times n}$

Answer-2:

Output:

n	matrix type	condition_number	no_partial_pivot_error	no_partial_pivot_residual	partial_pivot_error	partial_pivot_residual	inbuilt_function_error	inbuilt_function_residual
**	***********	****************	**********************	*************************	******************	*********************	*********************	************************
10	random	81.72862839178067	2.011312340136783e-15	2.6259540005203653e-16	2.6380356893428514e-15	1.6761063625462658e-16	1.6161278270545909e-15	9.226661858305929e-17
10	hilbert	16331478489294.637	0.00022377310679871825	6.236675960178107e-17	0.00011306889064265486	1.9294856868982527e-16	0.0002160054136312213	5.774039959701168e-17
10	ones	22.360679774997898	0.0	0.0	0.0	0.0	0.0	0.0
20	random	1570.6057563437437	3.1720248367765567e-13	1.5082096896723206e-15	7.204440719142018e-15	1.1073623911081181e-16	3.479210282559712e-14	2.4449899576579773e-16
20	hilbert	8.24459274245295e+17	23.541742373749674	3.0379314443923253e-16	25.156607499302982	2.4891548829444045e-16	17.290006131414625	2.8757313481144477e-16
20	ones	63.24555320336759	0.0	0.0	0.0	0.0	0.0	0.0
30	random	2688.583475675636	1.949957343086635e-13	6.925126826085552e-16	1.0641923489529531e-13	2.2096234092041378e-16	9.67051096822761e-14	2.026504759812942e-16
30	hilbert	1.1921236022141034e+19	41.67034959290397	3.4283679389372696e-16	16.98646050582484	3.7272342481073856e-16	49.300961392520605	6.419137794290911e-16
30	ones	116.1895003862225	0.0	0.0	0.0	0.0	0.0	0.0
40	random	26036.945020093895	3.831352093795588e-11	8.121527814798785e-15	8.578874543179429e-13	2.856945251387899e-16	7.463895681088446e-13	2.174732645517965e-16
40	hilbert	4.692967988354769e+19	73.95515661546712	5.615562014171272e-16	62.26890299996139	8.096140978918429e-16	484.77808098276324	4.7537230848034455e-15
40	ones	178.88543819998318	0.0	0.0	0.0	0.0	0.0	0.0

1. **Hilbert Matrix:** The Value of the conditional number for the Hilbert matrix is large. Since the value is large the Hilbert matrix becomes highly ill conditioned and hence the value of error calculated becomes significantly large. On increasing the value of n from n=10 to 40 the value of conditional number increases and so is the value of the error
   Order of condition number is 10^{18}
   In the Hilbert matrix when pivoting is used the value of error is reduced because pivoting makes the algorithm backward stable and hence the error value decreases.
   The following are the output of the Hilbert matrix for n =10,20,30 and 40.

n	matrix type	condition_number	no_partial_pivot_error	no_partial_pivot_residual	partial_pivot_error	partial_pivot_residual	inbuilt_function_error	inbuilt_function_residual
**	***********	****************	**********************	*************************	******************	*********************	*********************	************************
10	hilbert	16331478489294.637	0.00022377310679871825	6.236675960178107e-17	0.00011306889064265486	1.9294856868982527e-16	0.0002160054136312213	5.774039959701168e-17
20	hilbert	8.24459274245295e+17	23.541742373749674	3.0379314443923253e-16	25.156607499302982	2.4891548829444045e-16	17.290006131414625	2.8757313481144477e-16
30	hilbert	1.1921236022141034e+19	41.67034959290397	3.4283679389372696e-16	16.98646050582484	3.7272342481073856e-16	49.300961392520605	6.419137794290911e-16
40	hilbert	4.692967988354769e+19	73.95515661546712	5.615562014171272e-16	62.26890299996139	8.096140978918429e-16	484.77808098276324	4.7537230848034455e-15

2. **Random Matrix**: For the random matrix the order of conditional number is 10^1 - 10^3 which is comparatively less than that of Hilbert matrix and hence is the magnitude of error for random matrix

   For random matrices pivoting decreased the magnitude of error that was being produced by the no pivoting algorithm as the algorithm becomes backward stable.

n	matrix type	condition_number	no_partial_pivot_error	no_partial_pivot_residual	partial_pivot_error	partial_pivot_residual	inbuilt_function_error	inbuilt_function_residual
**	***********	****************	**********************	*************************	*******************	**********************	**********************	*************************
10	random	81.72862839178067	2.011312340136783e-15	2.6259540005203653e-16	2.6380356893428514e-15	1.6761063625462658e-16	1.6161278270545909e-15	9.226661858305929e-17
20	random	1570.6057563437437	3.1720248367765567e-13	1.5082096896723206e-15	7.204440719142018e-15	1.1073623911081181e-16	3.479210282559712e-14	2.4449899576579773e-16
30	random	2688.583475675636	1.949957343086635e-13	6.925126826085552e-16	1.0641923489529531e-13	2.2096234092041378e-16	9.67051096822761e-14	2.026504759812942e-16
40	random	26036.945020093895	3.831352093795588e-11	8.121527814798785e-15	8.578874543179429e-13	2.856945251387899e-16	7.463895681088446e-13	2.174732645517965e-16

3. **Ones Matrix:** For the ones matrix the order of magnitude 10^1 which is much less than that of hilbert matrix and random matrix and hence the problem is well conditioned. So the value of error is 0 in all cases. The following is the result for ones matrix for n = 10,20,30 and 40. For ones matrix the algorithm, Gaussian Elimination with partial pivoting and without partial pivoting produces 0 error and hence no difference is observed even if we use pivoting.

n	matrix type	condition_number	no_partial_pivot_error	no_partial_pivot_residual	partial_pivot_error	partial_pivot_residual	inbuilt_function_error	inbuilt_function_residual
**	***********	****************	**********************	*************************	*******************	**********************	**********************	*************************
10	ones	22.360679774997898	0.0	0.0	0.0	0.0	0.0	0.0
20	ones	63.24555320336759	0.0	0.0	0.0	0.0	0.0	0.0
30	ones	116.1895003862225	0.0	0.0	0.0	0.0	0.0	0.0
40	ones	178.88543819998318	0.0	0.0	0.0	0.0	0.0	0.0

NOTE: The values computed here are relative errors and residuals. Also the conditional number is relative

Answer-3:

OUTPUT:

```
Question-3
**********



Part-A
[[5.55555556e-01]
 [2.22222222e-01]
 [1.11111111e-10]]


Part-B
Row Equilibrated Matrices
(array([[ 5.e-10,  5.e-10,  1.e+00],
       [ 2.e-09, -1.e-09,  1.e+00],
       [ 5.e-01,  1.e+00,  0.e+00]]), array([[5.e-10],
       [1.e-09],
       [5.e-01]]))

Row Equilibrated and solved using Gaussian elimination without pivoting
[[5.55555556e-01]
 [2.22222222e-01]
 [1.11111111e-10]]
```

The results of computation in both ways are the same. The result is the same because we are performing the scaling operations when we are row equilibrating the matrix A.

The scaling transformation is basically multiplying the matrix A with a matrix which has only diagonal entries(diagonal matrix). This is called the scaling matrix. By performing the scaling operation the solution of the equation Ax=b remains the same. Hence the result is the same while calculating using row equilibrated way and gaussian elimination with partial pivoting

Keshav Gambhir
2019249