Diney Anand	DibyaJyoti Sarkar	Medha Jain	Shiffin N
2022aa05030	2022aa05005	2022aa05083	2022aa05055

Q.2 To solve systems like $Ax=b$ where A is a invertible $n\times n$ matrix we write a program Solved (A, b) that makes a matrix A and right-hand b as input & computes the solution to $Ax=b$ Suppose that algo. used by Solve (A, b) is the augmented matrix method. Let us say we need to solve k system of type $Ax=b$, where the right hand side changes, but the left hand side stays the same. We can do this by making k invocations to the procedure Solve (A, b) Can you come up with a better way of solving such systems. & characterize the improvement in operation count compared with making k calls to Solve (A, b)?

Answer:

Observation	Inference
1. A is an invertible $n\times n$ (square) matrix.	This means the below property holds true for $A_{n\times n}$ – $A^{-1}A = I_n$ And in this case, this property can also be applied to simplify a linear system of equations as follows $Ax=b$ $A^{-1}Ax = A^{-1}b$ $I_n x = A^{-1}b$ $x = A^{-1}b$
2) The function Solve (A, b) uses the augmented matrix method to solve $Ax=b$.	This means the function Solve (A, b) uses Gaussian Elimination with back-substitution (row-echelon form) to solve $Ax=b$ We can also solve it with Gauss Jordan Elimination.

Diney Anand 2022aa05030 | DibyaJyoti Sarkar 2022aa05005 | Medha Jain 2022aa05083 | Shiffin N 2022aa05055

Observation	Inference
3. We are attempting to solve k systems of the type $Ax=b$. With constraint right-hand side changes while the left hand side does not.	This means that we are attempting to solve a linear system of equations k times. $a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$ $a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$ ... $a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$ where over k iterations: ⓐ the coefficients a_{ij} $(1 \geq i, j \leq n)$ (i.e. matrix A) are not changing ⓑ the variable x_i $(1 \geq i, j \leq n)$ are changing. ⓒ and as a result, the values b_i $(1 \geq i \leq n)$ are also changing
4. We have to suggest an optimal implementation for Solve (Ax, b) in terms of operations performed. When Solve (A, b) is called k times under the above mentioned constraint	This means that we have to show that when Solve (Ax, b) is called k times, the gauss Jordan elimination method does a lot fewer operations.

Case 1: Calculating the number of operations performed when Solve (A, b) is called k times using augmented matrix method.

↳ To solve a linear equation $Ax=b$, using augmented matrix or guassian elimination with back-substitution method our goal is to reduce the augmented matrix $[A|b]$ using elementary operations to the row-echelon form.

↳ A matrix is said to be in row-echelon form when:

* All rows containing at least one non-zero element is on top of rows containing zero.

Diney Anand   Dibya Jyoti Sarkar   Medha Jain   Shiffin N
2022aa05030   2022aa05005   2022aa05083   2022aa05055

↳ looking at the non-zero rows only, the first non-zero element is on to the left (i.e. pivot or leading coefficient) is always strictly on the right of the pivot above it.

↳ The steps of Gaussian elimination are:

1. Write $Ax = b$ as a augmented matrix $[A \mid b]$
2. Get a 1 in the i^{th} row of the i^{th} column
3) Use row i to get 0's in the i^{th} column of rows $i+1$ to n
4) Repeat steps 2 & 3 from $i = 1$ to n.
5/ Change the augmented matrix back into a linear system of equation
6/ Use back substitution to solve for the variables.

Augmented Matrix

$$\left[\begin{array}{cccc|c} * & * & \cdots & * & * \\ * & * & \cdots & * & * \\ * & * & \cdots & * & * \\ \vdots & \vdots & & \vdots & \vdots \\ * & * & \cdots & * & * \end{array}\right] \longrightarrow \left[\begin{array}{cccc|c} 1 & * & \cdots & * & * \\ 0 & * & \cdots & * & * \\ 0 & * & \cdots & * & * \\ 0 & \vdots & & & \vdots \\ \vdots & & & & \\ 0 & * & \cdots & * & * \end{array}\right]$$

Get 1 in the 1st row of 1st column
Use row 1 to get 0 in the 1st column of rows 2 to n
Total operation performed: n.

$$x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$x_2 + \cdots \quad a_{2n}x_n = b_2$$
$$x_n = b_n$$

Use $x_n = b_n$ to back substitute the values of x_1 to x_{n-1}
Total operation performed n.

Therefore, we can conclude that to solve $Ax = b$ just once the Gaussian elimination we have to perform:

$(n + (n-1) + 1 \cdots + 1$ operations to reduce to the row-echelon form) + (n operations to back-substitute the values of x

$$\Rightarrow \frac{n(n+1)}{2} + n \text{ operations}$$

Now, since we intend to solve $Ax = b$ k times where the right hand side changes, but the left hand side stays the same, we must perform the complete Gaussian elimination process from start to finish k numbers of time.

Divey Anand
2022 ad 05030

Dibyajyoti Sarkar
2022 aa 05005

Medha Jain
2022 aa 05083

Shiffin N
2022 aa 05055

We need to do this as the augmented matrix $[A|b_i]$ $(1 \geq i \leq k)$ used in Gaussian elimination process from start to finish k number of times.
We need to do this as the augmented matrix $[A|b_i]$ $(1 \geq i \leq k)$ used in Gaussian elimination is depended on b_i for being reduced to the row-echelon form. ∴ Our total number of operation $= K\left(\frac{n(n+1)}{2} + n\right)$

Case 2: Calculating the number of operations performed when Solve (A, b) is called k times using the Gauss-Jordan elimination method to solve $Ax = b$ under the given constraint.

We know that, since the matrix A is invertible, we can solve the linear system of equations $Ax = b$ as $x = A^{-1}b$
We also know that we can find A^{-1} using the Gauss-Jordan elimination method. To do the same, we have to start with the augmented matrix, $[A|I_n]$ and by using elementary operations reduce it to the reduced-row-echelon form to get $[I_n, A^{-1}]$
A matrix is said to be in reduced-row echelon form when:

* if it is already in row-echelon.
* Every pivot it is 1.
* The pivot is the only non-zero entry in the column.

The steps for Gaussian Jordan elimination are:

1. Write the augmented matrix as $[A|I_n]$.
2. Get a 1 in the i^{th} row of the i^{th} column.
3. Use row i to get 0's in the i^{th} column of rows (1 to i-1) and (i+1 to n).
4) Repeat steps 2 & 3 from (i=1 to n)
5) When augmented matrix becomes $[I_n|A^{-1}]$ get A^{-1}!

Diney Anand 2022aa05030
DibyaJyoti Sarkar 2022aa05005
Medha Jain 2022aa05083
Shiffin N 2022aa05055

For example, Augmented Matrix.

$$\left[\begin{array}{cccc|cccc} * & * & \cdots & * & 1 & 0 & \cdots & 0 \\ * & * & \cdots & * & 0 & 1 & \cdots & 0 \\ \vdots & & & & 0 & 0 & 1 \cdots & 0 \\ * & * & \cdots & * & 0 & 0 & \cdots & 1 \end{array}\right]$$

$$\left[\begin{array}{cccc|cccc} 1 & * & \cdots & * & 1 & 0 & \cdots & 0 \\ 0 & * & \cdots & * & * & * & \cdots & * \\ 0 & * & \cdots & * & * & * & \cdots & * \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & * & \cdots & * & * & * & \cdots & * \end{array}\right]$$

Get 1 in the 1st row of the 1st column
Use row 1 to get 0's in the 1st column of rows 2 to n
Total operation performed : n.

Finally doing all this information once we have A^{-1} we can compute A^{-1} by performing n operation to get each row of the matrix.

Therefore, we can conclude that to solve $Ax = b$ using Gauss Jordan elimination we have to perform:

$\Rightarrow$ ($n + n + \cdots + n$ operations to find A^{-1}) + (n operations to compute $A^{-1}b$)

$\Rightarrow$ $n^2 + n$ operations.

Now, since we are intending to solve $Ax = b$ k times where the right hand side changes, but the left hand side stays the same, i.e, matrix A remains constant & matrix b changes, we can:

* Calculate A^{-1} just once using the Gauss Jordan elimination method & store the value of A^{-1} for subsequent $(k-1)$ calls.
* Calculate $A^{-1}b_i$ $(1 \geq i \leq k)$ k times to get values of x_i

Therefore, our total number of operations would be:

$$n^2 + kn \text{ operations}$$

Thus, in conclusion, we can see that CASE 2 requires a significantly lesser number of operations to solve $Ax = b$.

Answer