Chapter 3 : Solving (Square) Systems of Equations

There are 2 kinds of systems : linear and nonlinear systems.

Non-Linear Systems

Non linear systems are systems of equations where there is an exponent to the variables. A solution to a system satisfies all the equations.

In general, a 2x2 system looks like

$f_1(x, y) = 0$

$f_2(x, y) = 0$

with exact solution

(p, q)

a 3x3 system looks like

$f_1(x, y, z) = 0$

$f_2(x, y, z) = 0$

$f_3(x, y, z) = 0$

with exact solution

(p, q, r)

There are three methods we are going to be using to solve nonlinear systems.

1. Newton's Method --> only for 2x2
2. Fixed Point Iteration --> 2x2 or 3x3
3. Gauss-Seidel Iteration --> 2x2 or 3x3

Newtons Method for Non-Linear Systems

We need p_0, q_0 and $f_1(x, y) = 0, f_2(x, y) = 0$

Note : The solution can be written as a vector

The iteration is given by :

$$\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} - J^{-1}\big|_{(p_0, q_0)} \begin{pmatrix} f_1(p_0, q_0) \\ f_2(p_0, 1_0) \end{pmatrix}$$

where J is the **jacobian matrix** of the 2x2 system is given by

$$\begin{pmatrix} \frac{\delta f_1}{dx} & \frac{\delta f_1}{dy} \\ \frac{\delta f_2}{dx} & \frac{\delta f_2}{dy} \end{pmatrix}$$

This can be considered the derivative of the system. Therefore, the above iteration equation is equivalent to $p_{n+1} = p_n - \frac{f(p_)}{f'(p_n)}$

Fixed Point Iteration for Non-Linear Systems

(For **2x2** systems)We need p_0, q_0 and $g_1(x,y) = x, g_2(x,y) = y$

This means that $g_1(p,q) = p$ and $g_2(p,q) = q$

The iteration is given by :

$$p_{n+1} = g_1(p_n, q_n)$$

$$q_{n+1} = g_2(p_n, q_n)$$

Gauss-Seidel Iteration for Non-Linear Systems

It is almost identical to Fixed Point Iteration, with some improvment.

(For **2x2** systems)We need p_0, q_0 and $g_1(x,y) = x, g_2(x,y) = y$

$p_1 = g_1(p_0, q_0)$

$q_1 = g_2(p_1, q_0)$

notice that we are substituting the found value of p_{n+1} to find q_{n+1}. Generally

$$p_{n+1} = g_1(p_n, q_n)$$

$$q_{n+1} = g_2(p_{n+1}, q_n)$$

Linear Systems

Revision : Row Operations

Directly Solving an NxN Linear System

Any *square* linear system is given by :

$A_{nxn} X_{nx1} = b_{bx1}$

We will only deal with linear systems that have a **unique solution.** That is to say, A is **non-singular**(A^{-1} exists). For example, the system

$2x_1 + x_2 - x_3 = 7$

$5x_1 + 5x_3 = 8$

$-3x_1 + x_2 - 2x_3 = -1$

is expressed as

$$A = \begin{pmatrix} 2 & 1 & -1 \\ 5 & 0 & 5 \\ -3 & 1 & -2 \end{pmatrix}$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$b = \begin{pmatrix} 7 \\ 8 \\ -1 \end{pmatrix}$$

There are 5 direct ways to solve a system

1. Cramers rule
2. Gaussian Elimination
3. Gaus-Jordan reduction
4. Inverse Method
5. Lu factorization

The numerical part of what we are learning, we are going to calculate the **cost** of these methods.

The cost, or **complexity** of a method, is the number of operations$(+ - \div x)$ in this method. For example, the cost of solving

$$\frac{5+2*4}{7-3}$$

is 4, because there are 4 operations. Now lets say that we have A_{2x2}. What is the cost of αA ?.

$$\alpha \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

There are 4 operations here, so the cost is 4. Actually, we can say that for any matrix A_{nxn}, the cost of $\alpha A_{nxn} = n^2$

But what about the cost of $A_{nxn} + B_{nxn}$? well, it looks like this :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix}$$

So the cost is 4, or n^2.

And For $A_{nxn} B_{nxn}$, which looks like :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} * \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a*e+b*g & -- \\ -- & -- \end{pmatrix}$$

Each new member of the matrix costs us 3, multiplied by 4, then the cost is 12, or generally, if both matricies are the same size, then the cost is $2n^3 - n^2$.

However, if we are multiplying $A_{3x3} B_{3x1}$, the cost is $15 = 3(3+2)$. generally, for $A_{nxn} B_{nx1}$, the cost is $2n^2 - n$.

The cost of $det(A_{2x2}) = 3$. The cost of $det(A_{3x3}) = 14$, because we are calculating the determinant of three $3x3$ matricies, doing 3 multiplications, 1 addition, and 1 subtraction. For a $4x4$ matrix, the cost of $det(A_{4x4}) = 4 * 14 + 4 + 3 = 63$ in the same way.

Consider the following :

```
for p = 1 : n

    a = (p+s)/(2p+1)
```

The cost of this segment of code is $4 * n$

Cramers Rule

Let there be a $2x2$ system given by :

$2x_1 + x_2 = 5$

$x_1 - x_2 = 1$

Let us solve this system using cramers rule, and the cost of solving it.

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix}, b = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$$

Cramers rule :

$$x_n = \frac{det(A_n)}{det(A)}$$

Where A_n is A with column n replaced with b, so

$$A_1 = \begin{pmatrix} 5 & 1 \\ 1 & -1 \end{pmatrix}, A_2 = \begin{pmatrix} 2 & 5 \\ 1 & 1 \end{pmatrix}, det(A_1) = -6, det(A_2) = -3, det(A) = -3$$

The cost so far is $3 * 3 = 9$

$x_1 = \frac{-6}{-3} = 2$

$x_2 = \frac{-3}{-3} = 1$

The total cost is $3 * 3 + 2 = 11$

Generally speaking, the cost of solving using cramers rule is

$$(n + 1) * (cost(det(A_{nxn})) + n$$

Special Case : Strictly Upper Triangular System of NxN

An Upper triangular matrix is a matrix where all members beneath the diagonal are `0` and none of the members of the diagonal are `0`, for example :

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 0 & 6 & 2 \\ 0 & 0 & 3 \end{pmatrix}, b = \begin{pmatrix} 3 \\ 8 \\ 3 \end{pmatrix}$$

The cost of solving such a system (by back substitution) **is always** n^2

The cost of converting a system to upper triangular is $\sum_1^{n-1} 2(n-1)^2 + (n-1)$

Row Operations

1. Row Operation I : Switch two rows.
2. Row Operation II : Multiply a row with a nonzero constante
3. Row Operation III : Replace a row by adding it to a multiple of another row, that is $R_k \to R_k - M_{kp}R_p$

Gaussian Elimination

To do gaussian elimination, we must

1. Convert $[A|b]$ into an upper triangular system $[\mu|c]$.
2. Solve by back substitution.

So the cost of the gaussian elimination

$$\sum_{t=n-1}^{1} (2(t)(t+1) + (t)) + n^2$$

Where $t = n - k$. Which when simplified further

$$\frac{4n^3 + 3n^2 - 7n}{6} + n^2$$

Simplifying again

$$\frac{4n^3 + 9n^2 - 7n}{6}$$

Partial Pivoting

Choose a pivot so that all multipliers have magnitude less than 1.

Switch Rows so that the pivot element is the largest in magnitude

Becuase the magnitude of multipliers is less than 1, That means that the error will decrease and the result is closer to the real value.

LU Factorisation

The cost is

$$\sum_{k=1}^{n-1} (2(n-k)^2 + (n+k)) + (n^2 - n) + n^2$$

Which when simplified, gives us

$$\frac{4n^3 + 9n^2 - 7n}{6}$$

again.

Gauss-Jordan Reduction

the cost is

$$\frac{2n^3 + n^2 + n}{2}$$

Inverse Method

the cost is

$$\frac{6n^3 - n^2 - n}{2}$$

Summary Of Costs

Operation	Solving Cost
Solve by Cramers Rule	$(n+1) * (cost(det(A_{nxn})) + n$
Converting a System to Upper Triangular	$\sum_1^{n-1} 2(n-1)^2 + (n-1)$
Cost of Solving an Upper Triangular System	n^2
Solve by Gaussian Elimination	$\frac{4n^3+9n^2-7n}{6}$
Solve by LU Factorization	$\frac{4n^3+9n^2-7n}{6}$
Solve by Gauss-Jordan Reduction	$\frac{2n^3+n^2+n}{2}$
Solve by Inverse Method	$\frac{6n^3-n^2-n}{2}$