Computational Fluid Dynamics

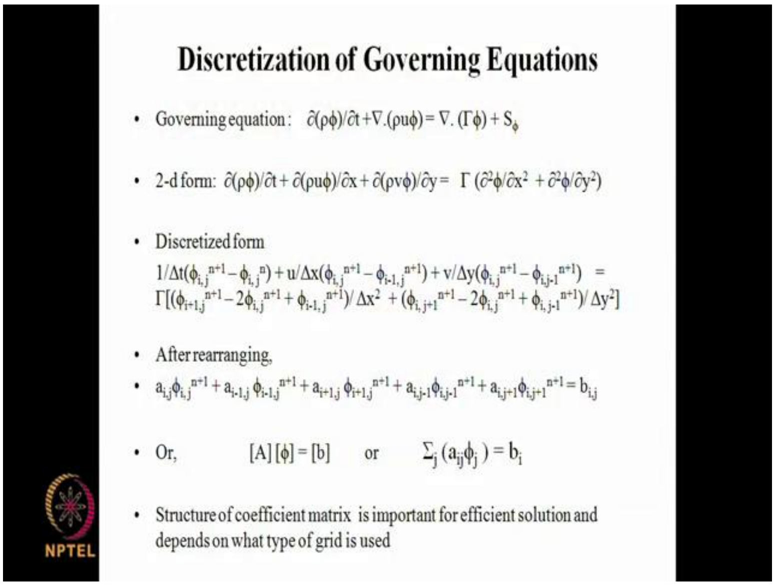
Prof. Dr. Sreenivas Jayanti
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Module - 05
Lecture - 44
Introduction to the basic numerical methods

Today, we are entering a new module. This is module 5 and it deals with the solution of linear algebraic equations. We have seen that as part of the CFD solution we take the partial differential equations, which in govern the flow. And, we use finite difference or finite volume or even finite element methods or other methods, to convert these into a system of algebraic equations.

And we have seen that, for example, the course of solution of compressible flow or incompressible flows we often end up with a system of linearized algebraic equations. And, we would like to focus on how to solve these things properly and efficiently. In the first lecture that we are going to have; we are going to look at the motivation for us to seek some special methods for the solution of these, which is at the heart of the CFD approach.

(Refer Slide Time: 01:18)



Discretization of Governing Equations

- Governing equation: $\partial(\rho\phi)/\partial t+\nabla.(\rho u\phi)=\nabla.(\Gamma\phi)+S_\phi$
- 2-d form: $\partial(\rho\phi)/\partial t+\partial(\rho u\phi)/\partial x+\partial(\rho v\phi)/\partial y= \Gamma\,(\partial^2\phi/\partial x^2+\partial^2\phi/\partial y^2)$
- Discretized form
  $1/\Delta t(\phi_{i,j}^{n+1}-\phi_{i,j}^{n})+u/\Delta x(\phi_{i,j}^{n+1}-\phi_{i-1,j}^{n+1})+v/\Delta y(\phi_{i,j}^{n+1}-\phi_{i,j-1}^{n+1}) =$
  $\Gamma[(\phi_{i+1,j}^{n+1}-2\phi_{i,j}^{n+1}+\phi_{i-1,j}^{n+1})/\Delta x^2+(\phi_{i,j+1}^{n+1}-2\phi_{i,j}^{n+1}+\phi_{i,j-1}^{n+1})/\Delta y^2]$
- After rearranging,
- $a_{i,j}\phi_{i,j}^{n+1}+a_{i-1,j}\,\phi_{i-1,j}^{n+1}+a_{i+1,j}\,\phi_{i+1,j}^{n+1}+a_{i,j-1}\phi_{i,j-1}^{n+1}+a_{i,j+1}\phi_{i,j+1}^{n+1}=b_{i,j}$
- Or, $[A]\,[\phi]=[b]$ or $\Sigma_j\,(a_{ij}\phi_j\,)=b_i$
- Structure of coefficient matrix is important for efficient solution and depends on what type of grid is used

So, we will start with brief recap on the discretization of governing equations. We know that we have a governing equation which is of a this form; dou by dou t of rho phi plus del dot rho u phi equal to del dot gamma delta gradient phi plus s phi. This is a time dependent term, this is the advection advective flux of phi, this is the diffusive flux of phi. And, this is source term. And, we have seen that this equation with different source terms and different diffusivities and different velocities leading to convection can, and different values of a phi here can represent all the basic equations like the mass conservation equation, momentum conservation equation in each directions, energy conservation equation, the species balance equations, all the form, all those equations that we wish to solve are of this particular form.

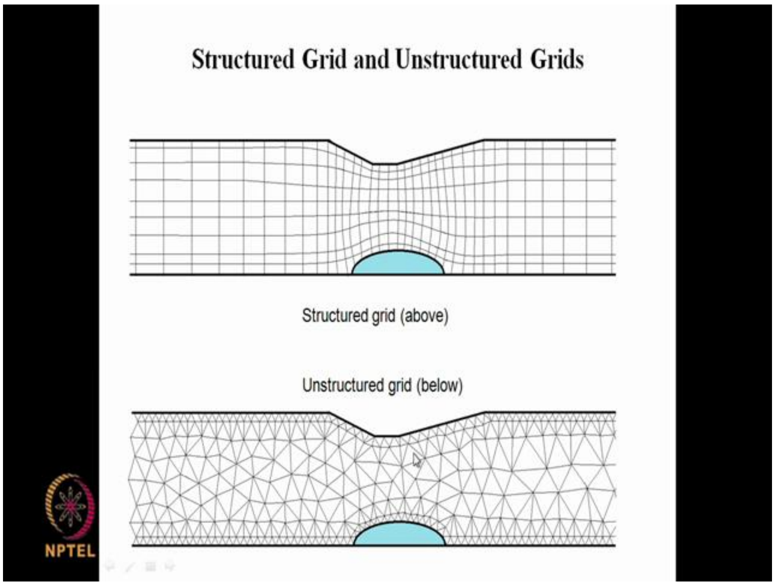
And, if you take the two dimensional form of it for this in Cartesian coordinates, it can be written as dou by dou t of rho phi plus dou by dou x of rho u phi plus dou by dou y of rho v phi equal to gamma, assuming gamma to be constant, dou square phi by dou x square plus dou square phi by dou y square. And, we have seen how to discretize this. And, specially in the latest example, we have seen how this is discretized as part of the simple method for the solution of two dimensional equations.

If you were to discretize this on a simple Cartesian grid of i j. And at point i j, we can write this as using forward differencing for this and then backward differencing for the advection term and again backward differencing for the advection term in the y direction and central differencing for dou square phi by dou x square. And, central differencing for dou square phi by dou y square here. All these things we can put here and we can rearrange this to get an equation of this particular form.

This is our algebraic equation. And, in the process we have linearized the u. And, we have accounted for the involvement of other equations, the coupling with other equations in terms of v here. And, for example, in the x momentum equation we have a source term which is not mentioned here, which is the pressure gradient term. So, all those things are done. And, eventually we have an equation like this. And, this can be put in the form of A phi equal to b, where A, phi and b are matrices. Or, it can also be written as sum over j of a i j phi j equal to b i. And since we are summing over j, we are left with i number of equations for the i number of variables.

So, this is the type of linearized algebraic equation involving these phis which are the solution variables. For example, the velocities at different grid points or the temperature at different grid points or u velocity at i plus half j and v velocity at i j plus half and pressure or pressure correction at i j, temperature at i j and so on. So, the structure of the coefficient matrix A here is important for efficient solution and depends on what type of grid is used. We briefly alluded to a structured grid and unstructured grid.

(Refer Slide Time: 05:12)


Structured Grid and Unstructured Grids

So, let us take; let us now make a distinction between the two because we have seen how to discretize on this kind of rectangular grid that we see here. And, we have also seen how to do the discretization of a governing equation on a triangular grid. And, here we have a slightly more complicated geometry than what we have been dealing with. And, you have a wall here. And, you have a top wall with a converging section, a throat section, which is constant width and then a diverging section here. And, for fun we have also put a blockage here, which is something like ellipsoidal thing. We would like to see the velocity profiles as a flow goes over this and then comes out. We would like to find out what is the drag force acting on this. And, so all those kind of things we can do.

Although we have this kind of angled kind of things, and this kind of curved kind of things, it is still possible to put what we call as a structured grid, where a grid point is located at the intersection of coordinate lines, like this one here and this one here.

And, you can see that this is a non-uniform grid here. And, you can also see that at this point these coordinate lines are curved. So, it is possible to use curvilinear coordinate lines and come up with the structured grid, where the grid points are at the intersection of this curvilinear coordinate lines like this and like this. This particular coordinate line is curving over this part. And then, after it goes up, here it is almost straight. And, this line is like this; this line is curved like this, but far away from it, it is straight.

So, generation of a body fitted grid; this is known as a body fitted grid with curvilinear coordinate systems. It is something that we will see much, will see in module six very briefly. It is not really part of. We are not going to go deeply into that thing, but this is an example of a structured grid.

And, as opposed to that we have an unstructured grid. And, in the case of structured grid, for example, you have this point here and you have a neighbor here and you have a neighbor here and neighbor; all those things are known. So, the neighbors are known by the coordinate line on which they are appearing.

For example, for this point here, we have an immediate right neighbor is the one, which is at the next coordinate line in this direction. And, immediate top neighbor is in the next coordinate line in the j direction. So, immediate left is the left neighbor is the immediate coordinate point intersecting the same j line of correspond to i minus one coordinate line. So, in that sense the surrounding, the immediate neighbor's information is known here. And, each point here has four neighbors, four immediate neighbors and may be four corner points and so on in two dimension.

Whereas, here we have a triangulated domain. And, the same domain is represented like this. And, one could say that the core, at the centroid of each of these triangles is the point, where we would like to get the velocities. And, you can see that we have small triangles, big triangles and so on. And, if necessary we can put more number of triangles wherever we want. And, we can have the velocities determined at the coordinate centers of this grid points. Now, the difference between the structured grid which is at the top and the unstructured grid which is at the bottom is that if you take this point here, the immediate neighbor is here and here and here and may be here. But, for a triangular thing it is only the immediate neighbor, who is sharing a face with you, matters.

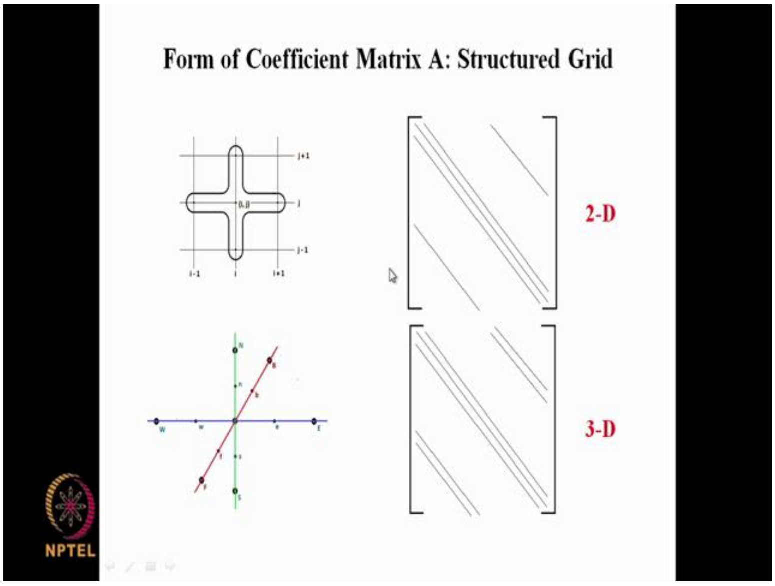
So, in this sense if you are starting to put a numbering from here and here, 1, 2, 3, 4, 5, 6, like this. Then, you can see that the neighbors are not somebody whom you know straight away. You need to have some special storage information of who the neighbor is. You cannot say it is i, i plus 1, i plus 1 j, i minus 1 j, i j plus 1, all that kind of notation is not possible here.

And, in this particular case each cell has three neighbors. And, you could make up a bigger cell here with four neighbors or more number of neighbors. So, in that sense the number of neighbors is not fixed in this unstructured grid. Whether you do the discretization on the structured grid or on the unstructured grid, we still end up with an equation like A phi equal to b, which we have seen.

In the case of structured grid in two dimensions here, we have for the diffusion equation for the Laplace or Poisson's equation; we will have the four immediate neighbors. That is, for point i j we have i plus 1 j, i j plus 1, i j minus 1 and i minus 1 j. In the case of the Poisson's equation on this triangle here, we will have one neighbor here, one neighbor here and one neighbor here.

So, this equation; we will have an equation for this cell will have the value of phi at this centroid, the value of phi at this centroid, the value of phi at this centroid and this centroid. So, it will have; four variables are involved in this. And, in this particular case we will have five variables; the point i j itself and the four neighboring points. So, in that sense we will have an algebraic equation of this particular form with the corresponding coefficients and with the variables that appears here in for each equation for each cell.

(Refer Slide Time: 11:46)



Form of Coefficient Matrix A: Structured Grid

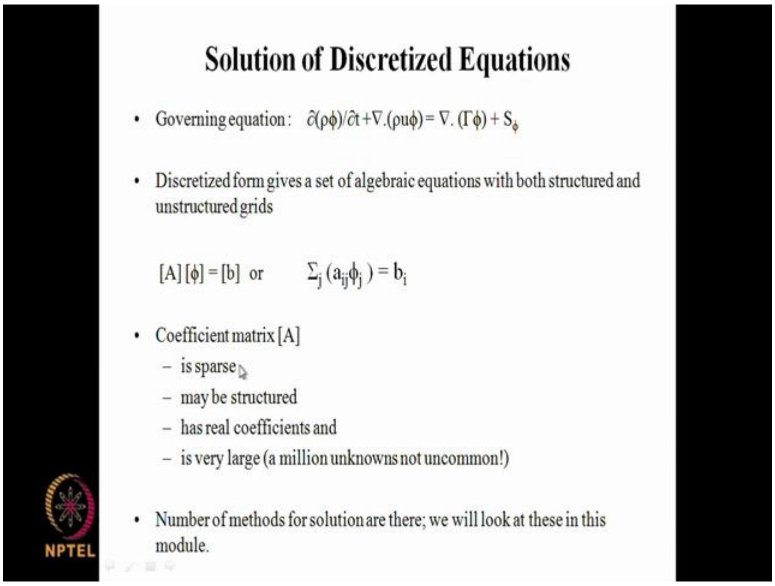
So, in the case of structured grid, the coefficient matrix takes a specific structure. For example, in the case of two dimension where you have i j here and i j plus one, j i j minus 1, i minus 1 j, i j plus 1, i plus 1 j, we have a (Refer Time: 12:10) molecule like this. If you put all the unknowns in the lexicographic ordering, then it is possible to get a pentadiagonal matrix, which is like this; where we have the main diagonal which is non-zero. Usually, in the case of delta x and delta y being equal to, being equal.

And if you have, for example, the heat conduction equation or the Poisson equation or the Laplace equation with constant coefficient and so on, this will be minus four, 1, 1, 1, 1. So, that is a type of thing that we have for all interior points. So, they all; the non-zero points lie along the main diagonal and the immediate higher diagonal and the immediate lower diagonal. And then, there will be lots of zeros here and then you have a non-zero diagonal and then zero diagonals, like this.

In the case of three dimension, where you are looking at a point here and you have an east neighbor capital E, west neighbor capital W, north neighbor capital N, south neighbor capital S, a back neighbor capital B and a front neighbor capital F, you can make up a control volume around this, which extends half grid distance in each direction, so that you have a north face here, east face here, south face, west face, front face and back face.

So, on this control volume you can, for example, discretize the governing equation. And then, end up with the discretized equation, which will have seven non-zero coefficients. And, these seven lie in this form; with zeros in between. I think that is not clearly marked here. So, you have the central one, the two immediate ones and then zeros diagonal, again zeros, a non-zero diagonal, again zeros like this. So, we will have a seven diagonal matrix, but you have a diagonal structure to the matrix A. This is what you would have with the structured grid.

(Refer Slide Time: 14:30).



Solution of Discretized Equations

- Governing equation: $\partial(\rho\phi)/\partial t+\nabla.(\rho u\phi)=\nabla.(\Gamma\phi)+S_\phi$
- Discretized form gives a set of algebraic equations with both structured and unstructured grids

$[A][\phi]=[b]$ or $\Sigma_j(a_{ij}\phi_j)=b_i$

- Coefficient matrix [A]
  - is sparse
  - may be structured
  - has real coefficients and
  - is very large (a million unknowns not uncommon!)
- Number of methods for solution are there; we will look at these in this module.

In an unstructured grid, you do not have this diagonal structure, but still the coefficient matrix has certain common features with a structured grid kind of thing. And, these common features are important when we consider the solution. And, the most distinct common feature of coefficient matrix A is that it is sparse. In the sense that most of the coefficients, most of the constants that appear in the matrix A are zero. Very few of them are non-zero. And, how many of them are non-zero?

In the general case, that is for an interior point, in case of three dimension for a Laplace or Poisson equation, you will have seven non-zero values. And, in the case two dimension you will have 5 non-zero values; in the case of one dimension you will have only three non-zero values. So, we will have in the case of one dimension, you will have a tridiagonal matrix with these also as zero.

And, in the case of general Navier-Strokes equations, you may get more number of points because you may have more derivatives; you have up ending and those kinds of things. But, despite all these things that number of non-zero coefficients for any cell is far less than the total number of points that are available. So, we have, what we can call as a compact (Refer Time: 16:10) molecule.

In the sense that, the value of the variable phi at a point is influenced only by a small number of the neighboring cell values, the influence or the dependence does not go too far into the overall computational domain. So because of this we will have a few non-zero components. And therefore, the matrices pass.

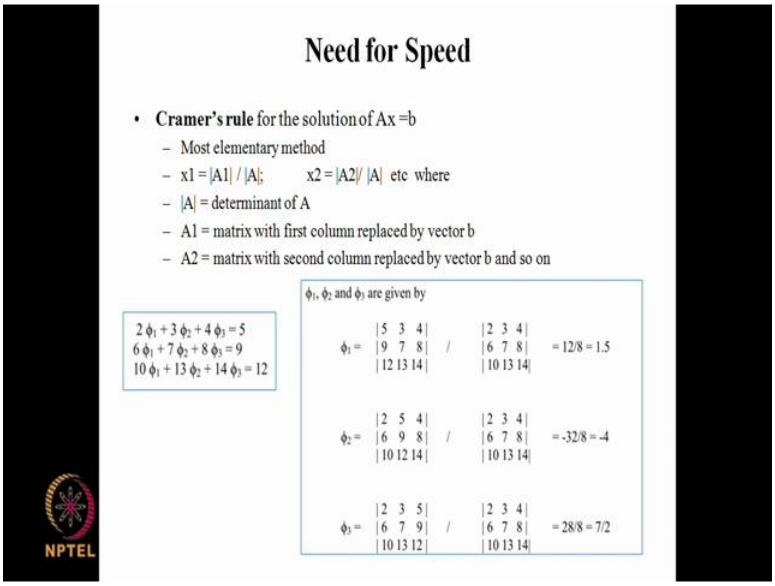
So, most of them non-zero and if you we have say thousand equations. So out of the thousand equations, each equation will have only about 7 non-zero coefficients. So, along each row we have 1000 possible points and only 7 are non-zero. And, all the 900 and 93 are zero. If you have million grid points, then you have million elements in each row. And out of these million, again you have only 7 which are non-zero, and you have whatever, million minus 7 is the number of zeros.

So, in that sense if you only were to put the non-zero points, it will be like stars in the sky. You will see only some of them are non-zero and most of them are zero. And, so this is a sparseness; is a very characteristic feature of a coefficient matrix of A phi equal to b; that is obtained from a CFD solution.

And as we have seen, it may be structured in the sense that them, the non-zero coefficients may lie along certain diagonals. And also importantly, the coefficients are real, they are not imaginary because these are essentially the velocities and the properties, physical, some of physical properties of the fluids and the grid spacing and so on. So, there is no reason why these coefficients are non-real. So, these are real coefficients. And, in general the coefficient matrix is very large. For example, a million unknowns, million grid points are something that you these days, people are doing on their laptops.

So, that means that if you have million grid points, you have, the coefficient matrix has million unknowns. So, it will be million by million; is a size of this matrix. So, we have to deal with these kind of equations in which the coefficient matrix is sparse, may be structured, has real coefficients, and is large. So, when you look at this type of A phi equal to b, it is something that is very well known to us.

(Refer Slide Time: 18:58).



Need for Speed

- **Cramer's rule** for the solution of Ax =b
  - Most elementary method
  - x1 = |A1| / |A|; x2 = |A2|/ |A| etc where
  - |A| = determinant of A
  - A1 = matrix with first column replaced by vector b
  - A2 = matrix with second column replaced by vector b and so on

$$2\phi_1 + 3\phi_2 + 4\phi_3 = 5$$
$$6\phi_1 + 7\phi_2 + 8\phi_3 = 9$$
$$10\phi_1 + 13\phi_2 + 14\phi_3 = 12$$

ϕ_1, ϕ_2 and ϕ_3 are given by

$$\phi_1 = \begin{vmatrix} 5 & 3 & 4 \\ 9 & 7 & 8 \\ 12 & 13 & 14 \end{vmatrix} / \begin{vmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ 10 & 13 & 14 \end{vmatrix} = 12/8 = 1.5$$

$$\phi_2 = \begin{vmatrix} 2 & 5 & 4 \\ 6 & 9 & 8 \\ 10 & 12 & 14 \end{vmatrix} / \begin{vmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ 10 & 13 & 14 \end{vmatrix} = -32/8 = -4$$

$$\phi_3 = \begin{vmatrix} 2 & 3 & 5 \\ 6 & 7 & 9 \\ 10 & 13 & 12 \end{vmatrix} / \begin{vmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ 10 & 13 & 14 \end{vmatrix} = 28/8 = 7/2$$

And, the one method that has been taught to us from our primary school days or higher secondary school days is the Cramer's rule. And, if you take an equation like A x equals to B, Cramer's rule says that the solution for A 1 x1, x 1 is a variable is obtained by A 1. Determinant of A 1 by determinant of A and x 2 is determinant A 2 by determinant of A, where determinant of A is the determinant of A, matrix, the whole matrix A. And, A 1 matrix. A is the matrix with the first column of the coefficient matrix, say replaced by the vector b. And, A 2 is A for the case where the coefficient, the second column of the coefficient matrix is replaced by vector b like this.

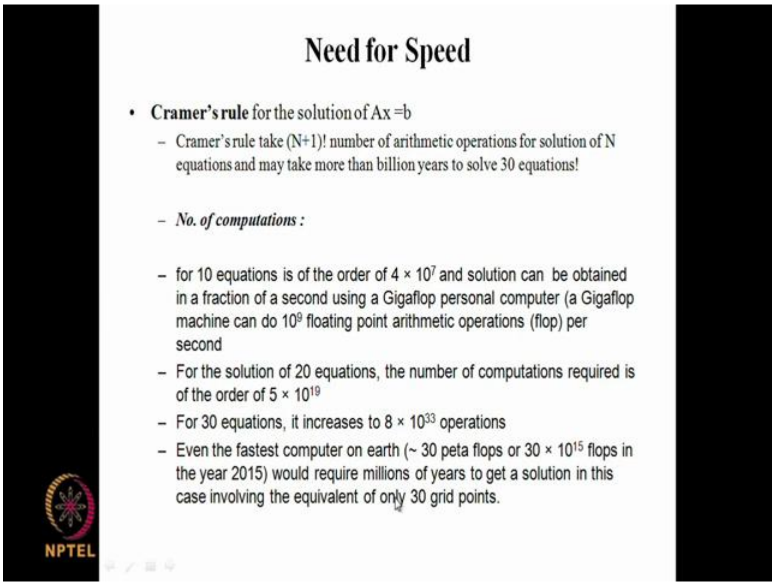
So, let us just take a simple three by three examples, where you have three unknowns; phi 1, phi 2, phi 3. And, these are such that the three equations which give this are 2 phi 1 plus 3 phi 2 plus 4 phi 3 equal to 5, 6 phi 1 plus 7 phi 2 plus 8 phi 3 equal to 9 and 10 phi 1 plus 13 phi 2 plus 14 phi 3 equal to 12. We can see it is a made up kind of thing. And, so this equation as per the Cramer's rule, phi 1 is given by, now the coefficient matrix is 2 3 4; 6 7 8; 10 13 14. And, out of that for the phi 1, we replace the first column with the right hand side column; 5 9 12. So, we have 5 9 12. And then, 3 4 here; 3 4; 7 8; 7 8; 13 14; 13 14.

So, the determinant of this divided by the determinant of the coefficient matrix, which is 2 3 4, 6 7 8, 10 13 14. And, you can go through an evaluation of this; the two determinants. And, you will get 1.5.

If you want to get the solution for phi 2, then it is given by the original members of the coefficient matrix; that is 2 6 10, and all that in which we replace the second column; that is this 3 7 13 by 5 9 12. So, that is why we have 2 6 7, 4 8 14. These are unchanged. And, this matrix, this column here is replaced by the right hand side on 5 9 12 here. Again, you evaluate this determinant, evaluate this determinant. And then by dividing by this, you get minus 4. And, for phi 3 you take the coefficient matrix, replace the third column by the right hand side column 5 9 12. So, that is what you have done here. Determinant of this divided by determinant of this will give you the solution here.

So, in that sense it is pretty straight forward to apply. And, it is a very good method in the sense that if there is a unique solution, then you can get it. There is no problem with this method. This method can be applied. And, the only condition is that there should be a solution. And, you are solving the solutions, you are putting together these equations because you feel that there is a solution. If it is a (Refer Time: 22: 36) problem, there will be a unique solution. So, if you got a unique solution, Cramer's rule will give it to you but, at what cost?

(Refer Slide Time: 22:47)



Need for Speed

- **Cramer's rule** for the solution of Ax =b
  - Cramer's rule take (N+1)! number of arithmetic operations for solution of N equations and may take more than billion years to solve 30 equations!
  - ***No. of computations :***
  - for 10 equations is of the order of 4 × 10^7 and solution can be obtained in a fraction of a second using a Gigaflop personal computer (a Gigaflop machine can do 10^9 floating point arithmetic operations (flop) per second
  - For the solution of 20 equations, the number of computations required is of the order of 5 × 10^{19}
  - For 30 equations, it increases to 8 × 10^{33} operations
  - Even the fastest computer on earth (~ 30 peta flops or 30 × 10^{15} flops in the year 2015) would require millions of years to get a solution in this case involving the equivalent of only 30 grid points.

So, it can be shown; you can go to many elementary books. You can show that Cramer's rule takes N plus one factorial number of arithmetic operations for the solution of n equations. Now, what does this actually mean? If you have ten equations, then N plus one factorial is of the order four times 10 to the power 7. It looks like a large number.

But, these days we have computers which are very fast. For example, you can have a Gigaflop personal computer; a giga, one Gigaflop is a machine which can do ten to the power nine floating point operations per second. Floating point operations are like this; division; arithmetic operations like division and multiplication.

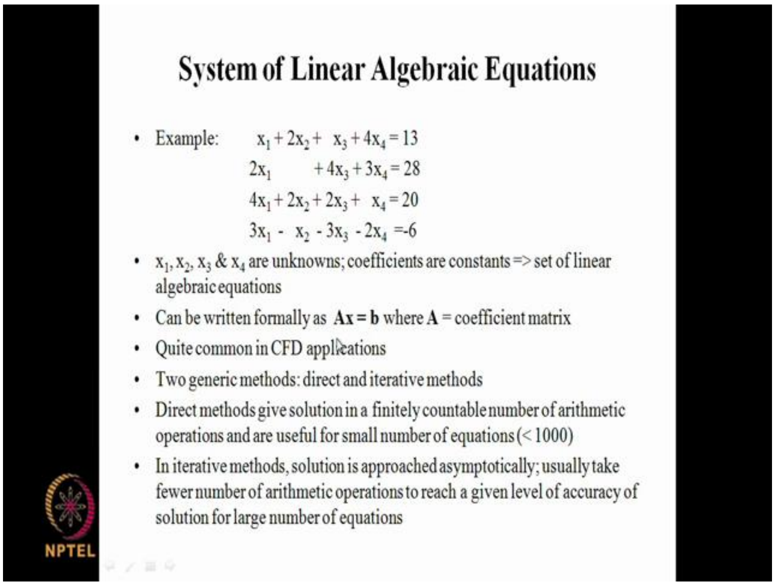
And, it can do 10 to the power 9, 1 billion of them per second. And here for 10 equations, this method will take about four times 10 to the power 7 operations. So, that means that you can get to the solution in less than 1, 0.1 second because you can you have machine which can do 10 to the power 9 operations per second, and you need only ten to the power of seven operations.

So, you can get that in 0.04of a second. So, looks pretty simple. But, if you have twenty equations, then 21 factorial is 5 times 10 to the power 19. So, if you now have the same Gigaflop kind of machine, the time it takes is the number of operations divided by the number of operations it can do per second; which is 10 to the power 9, so that 5 times 10 to the power 10 seconds. And, if you go to 30 equations, the number of operations required will be 8 times 10 to the power 33. And, if you divide that by the fastest machine that is available that which can do 30 peta flop operations. So, 30 times 10 to the power 15, which is 30,000 times faster than or 30 billion times faster than your Gigaflop machine.

So, this machine is 30 million times faster than this. And yet, even this would take 10 to the power 18 number of seconds. And, that is more than the total age of the earth. So, that means that if you were to use Cramer’s rule, and then if you had 30 grid points, then it would take a large number of computational seconds, in order to get a solution. So, thirty grid points is not much; because as part of this course, in your first assignment you would have seen that we had grid of 10 by 10, 20 by 20, 30 by 30, that is, a number of points. So, for a 30 by 30 you would have 900 equations. And, here you are looking at only 30 equations.

So, Cramer’s rule is not something that can be used very effectively because of the number of computations that is required to do the solution. And, the (Refer Time: 26:15) is finding the determinants. Determining, finding the determinant is really a number of is number crunching operation. And, it is not such a good method. So, we do need to have efficient methods.

(Refer Slide Time: 26:24)



System of Linear Algebraic Equations

- Example:

$$\begin{aligned} x_1 + 2x_2 + x_3 + 4x_4 &= 13 \\ 2x_1 \quad + 4x_3 + 3x_4 &= 28 \\ 4x_1 + 2x_2 + 2x_3 + x_4 &= 20 \\ 3x_1 - x_2 - 3x_3 - 2x_4 &= -6 \end{aligned}$$

- x_1, x_2, x_3 & x_4 are unknowns; coefficients are constants => set of linear algebraic equations
- Can be written formally as **Ax = b** where **A** = coefficient matrix
- Quite common in CFD applications
- Two generic methods: direct and iterative methods
- Direct methods give solution in a finitely countable number of arithmetic operations and are useful for small number of equations (< 1000)
- In iterative methods, solution is approached asymptotically; usually take fewer number of arithmetic operations to reach a given level of accuracy of solution for large number of equations

And, here we can see that there are number of methods. And, this is what we are going to explore in the rest of this particular module. For example, if you have a four by four equation like this, we have four equations and four unknowns; x 1, x 2, x 3, x 4 are the unknowns. And, coefficients are constants. And, so we have a set of linear algebraic equation, linear algebraic equations which we can write as A x equal to b. And, for the solution of this A x equal to b, we have essentially two types of methods.

One is direct type of methods. Direct method for the solution of A x equal to b. And, the other is the class of problems known as the iterative methods for the solution of A x equal to b. Direct method gives solution in a finitely countable number of arithmetic operations. And, are useful typically for small number of equations. When you have less than about thousand, you can use this. And, one such direct method is a Cramer’s rule. It gives you in about N plus one factorial number of operations, I t gives you the solution.

The new problem is that N plus one factorial is too many operations, when we are looking at larger number of equations. Otherwise, it is a method which is quite good. It would work. And, it is an example of a direct method. Other examples are methods that you might have heard like the Gaussian elimination method or the tridiagonal matrix algorithm. These are direct methods, which will give you a solution in countably finite number of arithmetic operations; pluses and minuses.

And, mostly what we are interested in from a computational point of view are divisions and multiplications. Iterative methods; on the contrary, on the other hand are methods which will approach the solution asymptotically. So, you start with some guess solution. And then, you get an improved solution. You put back the improved solution in this iterative method, you get a more improved solution. And, Gauss-Seidel method that we have seen in the first module is an example of an iterative method. And, we saw there that with every iteration it was approaching the true solution. And, after some ten or twenty iterations for that simple problem, we got a solution which was correct up to so many decimal places.

So, that is a characteristic feature of a iterative method. You can stop at a certain point when you feel that you are got sufficiently accurate solution. And, if you want more and more accurate solution, you have to go more and more number of iterations. So to that extent, iterative methods never stop, if you want the absolute truth, absolutely correct solution. Then, you have infinite number of mathematical operations, arithmetic operations. So, you are distinguishing between arithmetic operations and general mathematical operations, gradient operatives and mathematical operations.

Whereas, here in computer solution that we are looking at for the solution of linear algebraic equations, we are looking at arithmetic operations; that is addition, subtraction, division and multiplication of real numbers. And, these are what we are interested in. And, usually addition and subtraction take a small fraction of a time required for multiplication and division. So, when we look at the number of arithmetic operations that are required for a solution of A x equal to b, we are usually interested in the number of multiplications or divisions. And, it is these things that matter to us in terms of estimating the total computational time. And, we have these different classes of methods.

And, nowadays we have combinational methods. And, some of the methods that are relevant to CFD, which have gained popularity, which have gained acceptancy in CFD are what we are going to discuss in the rest of the module. So, we are going to start with the direct methods and then we are going to some basic iterative methods and then we will go into more advance methods in the second week of this particular module.

Thank you.