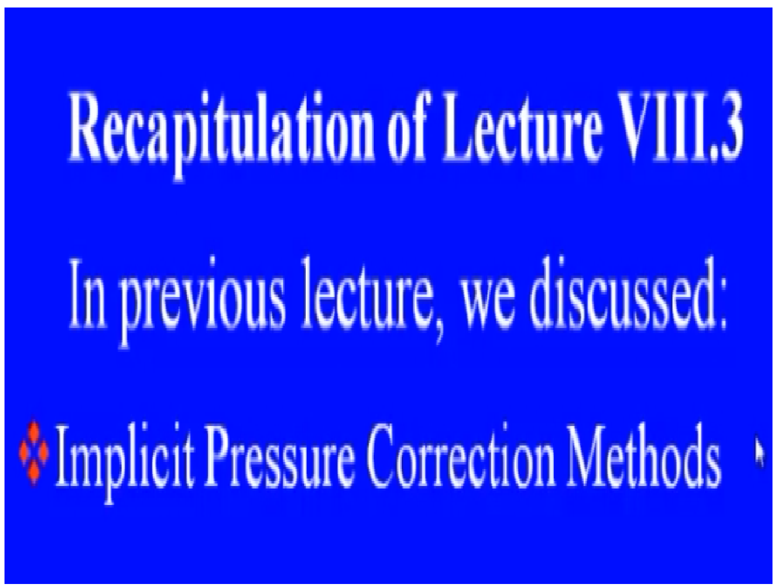
Computational Fluid Dynamics
Dr. Krishna M. Singh
Department of Mechanical and Industrial Engineering
Indian Institute of Technology – Roorkee

Lecture - 38
SIMPLEC, SIMPLER and Fractional Step Methods

Welcome to the fourth lecture in module 8 on numerical solution of Navier Stokes equations. In this module, we have been focusing on the numerical simulation of Navier Stokes equations. We discussed its features and explicit and implicit time integration techniques in earlier lectures. We had also discussed some implicit pressure correction methods and this lecture is a continuation of this category of methods and we will also consider fractional step methods in this lecture.

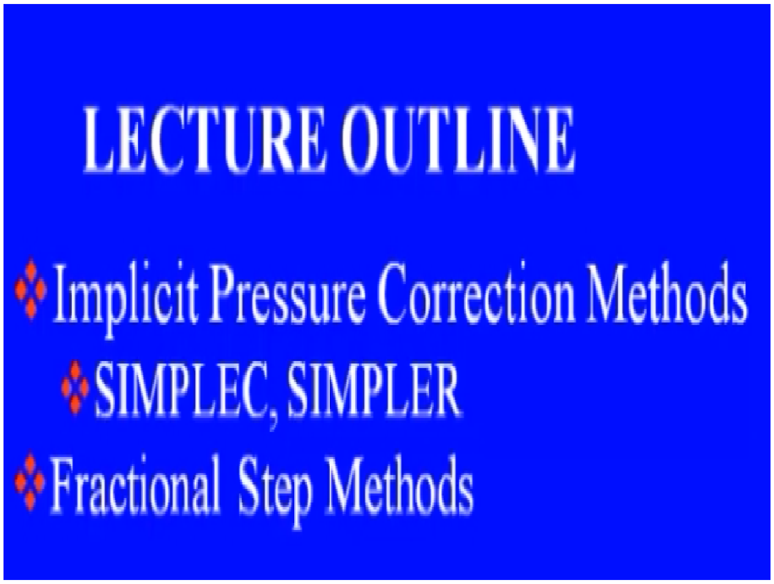
(Refer Slide Time: 01:02)



Before proceeding further, let us have a recap of what we did in the previous lecture. Our main topic was implicit pressure correction methods based on implicit time integration schemes and linearization of non-linear momentum equations. We discussed one of the most widely used algorithms of this family called simple algorithm. Now in this lecture, we would focus on the improved versions of this simple algorithm.

We will take up only two of them, SIMPLEC and SIMPLER and we will look at a family of methods called fractional step methods. So lecture outline, we will have a look at the SIMPLEC and SIMPLER methods of the implicit pressure correction methods family.

(Refer Slide Time: 01:49)



And then we will discuss fractional steps method. So now let us come to SIMPLEC method. SIMPLEC is an acronym. We have already seen what SIMPLE stands for and SIMPLEC precisely called simple consistent because there was a consistent approach to the treatment of vi prime telda, which was ignored by Patankar's scheme SIMPLE and this was proposed by Van Doormal and Raithby in 1984.

So what Van Doormal and Raithby proposed was that now let us approximate the velocity correction term, which was omitted in SIMPLE and this approximation would be obtained by using the simple emerging procedure and what they found is it leads to a much more consistent method, which converges a lot more rapidly and we do not have to worry about the under relaxation factors, which are required by simple algorithm.

So with the main attraction of this SIMPLEC process. Comparison was almost similar set of steps are involved. So now let us derive the equations, which is slightly different from simple. The basic process of course remains the same.

(Refer Slide Time: 03:22)

SIMPLEC Method

Based on concept of velocity correction and pressure correction:

$$v_i^m = v_i^{m*} + v_i' \quad , \quad p^m = p^{m-1} + p' \qquad (8)$$

Substitution of above eqns. in linearized momentum equation gives the relation:

$$v'_{i,P} = \tilde{v}'_{i,P} - \frac{1}{A_P^{v_i}}\left(\frac{\delta p'}{\delta x_i}\right)_P \qquad (9)$$

where

$$\tilde{v}'_{i,P} = -\frac{1}{A_P^{v_i}} \sum_l A_l^{v_i} v'_{i,l} \qquad (10)$$

Let us assume that velocity correction at computational node P can be approximated by a weighted average of velocity corrections at

That we would still use the equations, which we derived earlier for implicit based correction methods and SIMPLEC is also based velocity and pressure connection concepts. So based on the concept of velocity correction and pressure correction, which was introduced by Patankar that is to say we would obtain the velocity field vim as some of where intermittent velocity field vim* and the vim* remember we had obtained it using the linearized version of momentum equation.

Wherein for the right hand side evaluations we have used values of the velocity field and pressure field at the previous equation. So that is how we had obtained this vim prime + a small velocity correction, you would obtain it as a part of this solution process. Similarly, the pressure field had the current equation is obtained as some of the pressure at the previous equation + a small pressure correction.

And as I mentioned earlier here we would not require any under or over relaxation for improvement of the conversions. That is the beauty of this SIMPLEC method. Now what was done in SIMPLE scheme. We can do the same thing. So substitution of these momentum equations. So substitution of above equations in linearized momentum equation gives the relation, which is basically a link between the velocity and pressure corrections.

It is the same equation, which we earlier derived for SIMPLE method and that is vip prime=vip telda-1/apvi delta p prime/delta xi at p and let us continue the same numbering of equations for sake of consistency, which we did earlier. So as continuation we will call this equation as 14. This is our previous equation, which we had numbered as 9 and our vi prime p telda was defined as -1/apl alvi prime.

So this was the previous equation, which we have written earlier for this telda variable. Now in simple scheme we said vi prime telda, it is a weighted summation of vi prime at neighboring nodes and these corrections we do not know, so we ignored it. So instead of ignoring it so let us first try to get its approximation in terms of the neighbouring values. So now let us introduce that approximation.

So let us assume that the velocity correction at central node or at computational node p can be approximated by a weighted average of velocity corrections at neighbouring nodes. That is to say what we claim that vi p prime, we can approximate it as a weighted average. The weight factors are our coefficients alvi, which we have already calculated earlier. So this sigma l alvi vil prime/sigma alvi. So that is what represents our weighted average.

And this is fair enough. We just want to find out correction term, which is fairly small and it is reasonable to assume from the physics that it would be linked or it would be related to the correction values at the neighbouring nodes. So let us call this equation as equation 14. Now if we combine equations 9, 10 and 14, in fact if we combine 10 and 14 in terms of this approximation.

(Refer Slide Time: 10:39)

$$\boxed{v'_{i,P} \approx \frac{\sum_l A_l^{v_i} v'_{i,l}}{\sum A_l^{v_i}}} \qquad (14)$$

Using approximation (14), $\tilde{v}'_{i,P}$ can be written as:

$$\tilde{v}'_{i,P} = -\frac{1}{A_P^{v_i}} \sum_l A_l^{v_i} v'_{i,P} \qquad (15)$$

Thus, from eq. (9):

$$v'_{i,P} = -\frac{1}{A_P^{v_i}} \sum_l A_l^{v_i} v'_{i,P} - \frac{1}{A_P^{v_i}}\left(\frac{\delta p'}{\delta x_i}\right)$$

$$\Rightarrow \left(A_P^{v_i} + \sum_l A_l^{v_i}\right) v'_{i,P} = -\left(\frac{\delta p'}{\delta x_i}\right)$$

$$\Rightarrow \boxed{v'_{i,P} = -\frac{1}{A_P^{v_i} + \sum_l A_l^{v_i}}\left(\frac{\delta p'}{\delta x_i}\right)_P} \qquad (16)$$

Using approximation 14 at v telda prime ip will be written as. Now if you substitute it in our pressure linked equations from equation 9, what we will get that vip prime=-1/apvi sigma l. If I transfer the terms multiply this by apvi and collect them together so we get apvi+sigma l alvi*vip prime=-delta p prime/delta xi or in other words, the modified form for velocity corrections, we get as -1/apvi+sigma l alvi delta p prime/delta xi at point p.

So this is the modified equations for velocity corrections, which is directly linked to the pressure corrections. So now next task is to obtain an equation for the pressure correction and if I look at this equation, what we can clearly realize is there is only one small difference with respect to simple scheme that is denominator on the right hand side has changed in place of ap, we have got an additional contribution coming from the coefficients linked to the neighboring terms and that is what leads to a much better convergence behaviour of the SIMPLEC scheme.

(Refer Slide Time: 14:54)

... SIMPLEC

$$\boxed{v_i^m = v_i^{m*} + v_i' = v_i^{m*} - \frac{1}{\left(A_P^{v_i} + \sum_l A_l^{v_i}\right)}\left(\frac{\delta p'}{\delta x_i}\right)_P} \quad (17)$$

v_i^m must satisfy the continuity eqn.

$$\frac{\delta (v_i^m)}{\delta x_i} = 0 \quad (18)$$

From (17) and (18):

$$\cancel{\frac{\delta (v_i^m)}{\delta x_i}}^{\,0} = \frac{\delta (v_i^{m*})}{\delta x_i} - \frac{\delta}{\delta x_i}\left[\frac{1}{A_P^{v_i} + \sum_l A_l^{v_i}}\left(\frac{\delta p'}{\delta x_i}\right)\right]_P$$

$$\Rightarrow \boxed{\left\{\frac{\delta}{\delta x_i}\left[\frac{1}{A_P^{v_i} + \sum_l A_l^{v_i}}\left(\frac{\delta p'}{\delta x_i}\right)\right]\right\}_P = \left[\frac{\delta v_i^{m*}}{\delta x_i}\right]_P}$$

Now we have got the form for velocity corrections or what would the velocity v, so vim, which we have written as vim*+vi prime substitute for vi prime, so this becomes vim*-1/apvi+sigma alvi delta b/delta xi. Now this velocity filled must satisfy continuity equation, that is what our condition is. We want to find out velocity filled at the current equation level, which would satisfy continuity. So vim must satisfy the continuity equation. This is given as delta vim/delta xi=0.

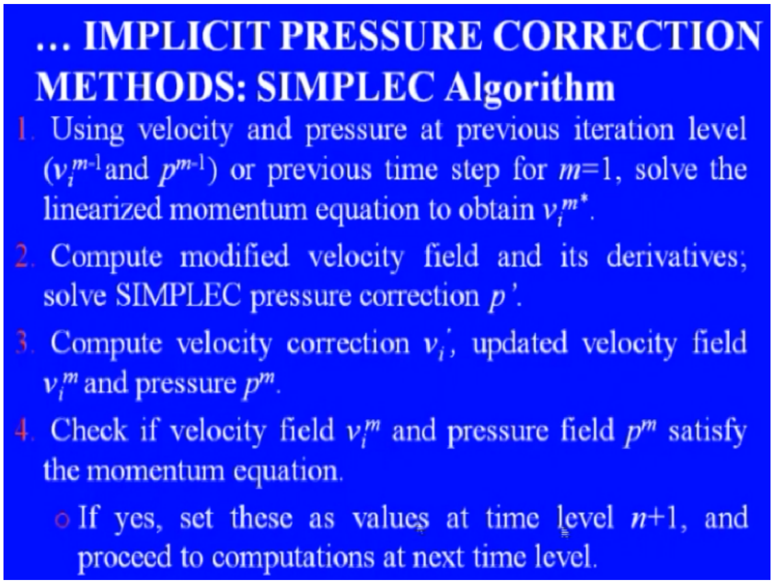
So if you use this, from 17 and 18 what do we get. Left hand side of 17 now would become delta vim/delta xi=delta vim*/delta xi-delta xi 1/apvi+sigma l alvi delta p prime/delta xi at p noting that from continuity equation the left hand side becomes 0 and this gives us delta of delta xi 1/apvi+sigma l alvi delta p prime/delta xi at point p, so it would be =delta/delta xi of vim*. So this is now the discrete Poisson equation for pressure correction.

So now we have got all the equations, which we need to obtain our solution. We have got our linearized momentum equation, which are based on the usage of the values of velocity and pressured field at the previous time equation and from there we can obtain our intermittent velocity field vim*. Now once vim* is available similarly our coefficient apvi and alvi, these would also be based on the values at previous time step. They are also available.

So we can compute all of these. Use it in this discrete Poisson equation for pressure correction and we can solve for p prime. One we know p prime, p prime is directly linked to the velocity corrections v prime. We have derived the formula earlier for that, which was given by the v prime was given by 1/apvi+sigma l alvi delta p prime/delta xi. So that formula directly gives us the velocity correction.

At the velocity correction to the velocity vim* and that will give us the velocity field vim at the current time level, which satisfies the continuity equation. At a pressure correction pm-1 to get the pressure field at their current iteration in pm. We are done with as far the computations are at present iteration or involved. So let us summarize our algorithm.

(Refer Slide Time: 20:18)



And this algorithm looks very, very similar except for a small difference in the coefficients, which are involved in the definition of velocity corrections and the pressure correction equation compared to the simple algorithm. Computations involved are almost steps are identical. So the first step is to use velocity and pressure at previous iteration level that is use vim-1 and pm-1 or if we are at the first iteration use the values to previous time step.

Solve the linearized momentum equation to obtain the velocity field vim*. So that is the first step. Next, once we know vim* find out its derivatives. Use it in SIMPLEC pressure correction equations, I have put this word SIMPLEC to indicate it now. This equation is different from the correction equation for our Poisson equation for pressure correction in simple algorithm. So let us call it SIMPLEC pressure correction Poisson equation.

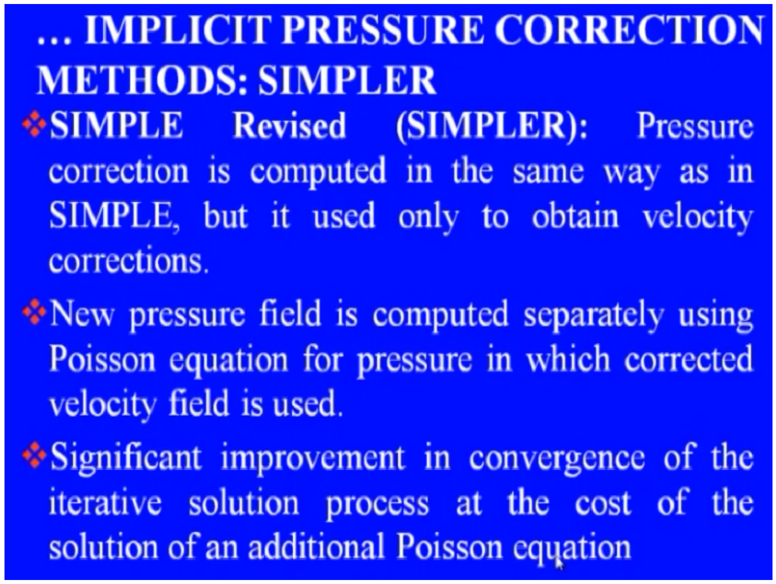
Solve this equation to obtain p prime and once we have got p prime, compute the velocity correction vi prime, then update add this velocity correction to vim* to get the velocity field vim and pressure pm. Next there was checking step. Check if velocity field vim and pressure field pm, satisfy the momentum equation. If yes, we have now got converged velocity and pressure field at the current time level.

So set this vim and pm as the values at time level n+1 and proceed to the comparisons at next time level. If not, we have to repeat the iterations again. So we will increment iteration counter and will be set to n+1. We will go back to step 1 and repeat the whole process until we have obtained converged velocity and pressure field at the current time level tn+1.

Now as I have already mentioned earlier that in SIMPLEC algorithm, we introduced an improvement with reference to simple with no omission of any term that vi prime telda instead of being omitted, it was approximated consistently in terms of the neighbouring values. So SIMPLEC converges a lot better than simple algorithm. So we do not require any under relaxation with SIMPLEC algorithm.

As far as computational requirements are concerned, they are almost identical and SIMPLE and SIMPLEC, both the algorithms require almost same number of computations. So computation time requirements are SIMPLE and SIMPLEC. They are both very similar.

(Refer Slide Time: 23:23)



Next, let us have a brief look at algorithm, which was named SIMPLER. In fact, it is SIMPLER is a shortened form of SIMPLE revised, which was proposed by Patankar. So SIMPLE revised and the acronym became SIMPLER. Now here the Patankar's intention was let us do the computation for pressure correction in the same way as we did in SIMPLE, but this pressure correction is used only to obtain the velocity corrections.

That is to say we have got linking equation and equation, which link the velocity corrections to the pressure corrections. Use those pressure corrections in that equation to obtain our velocity corrections, but do not add the pressure corrections to pm-1 to obtain pm that is to say the pressure at the current outer iteration. In this state, what we will do is that the new pressure field is computed separated using Poisson equation for pressure.

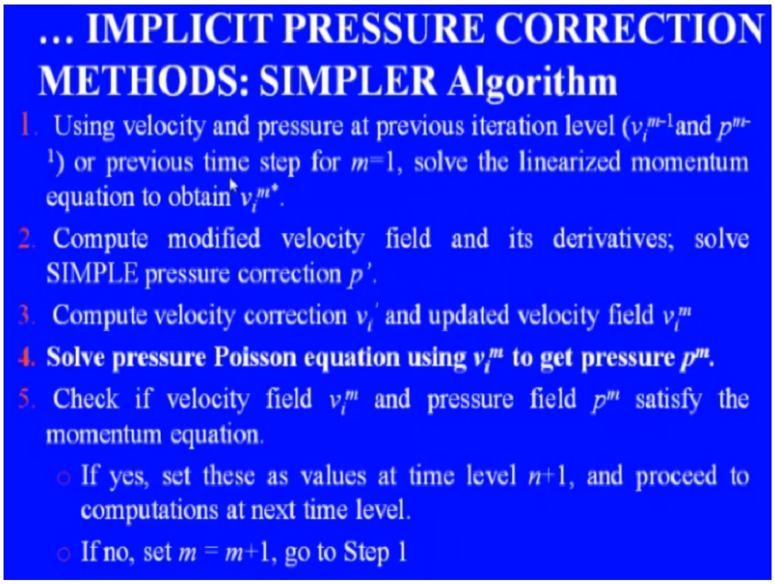
In which corrected velocity filed is used. Remember we have derived this Poisson equation for pressure in our original implicit pressure correction algorithm wherein the right hand side of this pressure Poisson equation was in terms of vim* that is to say in terms of the intermediate velocity field. In SIMPLER algorithm, what we do is, we already know what is the updated velocity at the current outer iteration.

So in our pressure Poisson equation, instead of vim* we would use vim, which has been calculated already. Since it is available, let us use it and thereby obtain the pressure field pm. So thus one important addition, which has been made to SIMPLE algorithm and it leads to significant, very, very significant improvements in SIMPLE. So convergence is very good of the simpler algorithm, but there is a drawback as well.

There is significant improvement in convergence of iterative solution process, but that comes at the cost of this solution of an additional Poisson equation. In SIMPLE, we solved only one Poisson equation that is a Poisson equation for pressure correction. In SIMPLER algorithm, we have to solve for 2 Poisson equations, 1 is for pressure correction and the rest for the pressure field itself.

So in nutshell what we will have to do in SIMPLE, we have to solve for 3 sets of linear equations that is 1 each for each of the velocity components to give us vim* field, so we have to solve 3 systems of linear equations for velocity fields and 1 for the pressure correction. In SIMPLE algorithm, we will have to solve 5 such set of equations, 3 for velocity field and 1 each for pressure and pressure correction.

(Refer Slide Time: 26:48)



Now let us reiterate the algorithm. So here there would be an addition with respect to simple. Few steps are common, like the first step is common that huge velocity and pressure at the previous iteration level vim-1 and pm-1. To solve the linearized momentum equation to obtain the intermediate velocity field vim*. Now compute or use this velocity of vim*, compute its derivatives and solve the SIMPLE pressure correction equation.

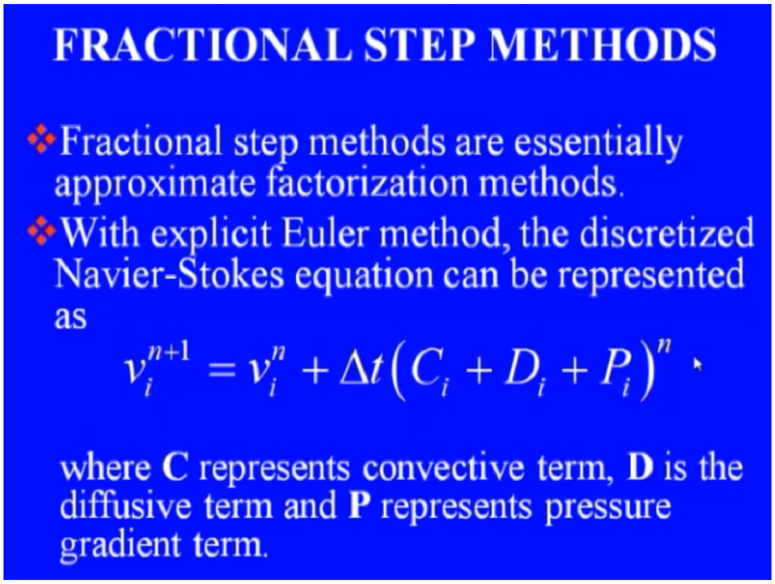
Again, that we have to solve the same pressure correction equation, which we have obtained in SIMPLE, which omitted vi telda primes so that term is still omitted. We would use the same simplified equations for the pressure correction as well as for velocity corrections. So using the same formulae as were used in simple algorithm, obtain the updated velocity field vim. Next what do we do.

Use this vim now in pressure Poisson equation to solve pressure Poisson equation and obtain the pressure at the current outer iteration pm by putting this additional step, now we have got updated velocity field, which satisfied continuity equation and updated pressure field. Next our checking step. Check if vim and pressure field pm satisfy momentum equations. If yes, we are done with the computations at the current time level and we can proceed to the computations at next time level.

If not, we would increment our iteration counter and go to step 1 and repeat the set of iteration once again or this iteration process once again to obtain this solution. So as I remarked earlier SIMPLER has got very good convergence behavior compared to simple. So in fact now it is more popular to use SIMPLEC or SIMPLER, SIMPLEC is computationally more efficient because we have to compute only one pressure Poisson equation or we have to solve for only one of them.

Otherwise their convergence behavior of SIMPLEC or SIMPLER algorithms are almost very similar.

(Refer Slide Time: 29:16)

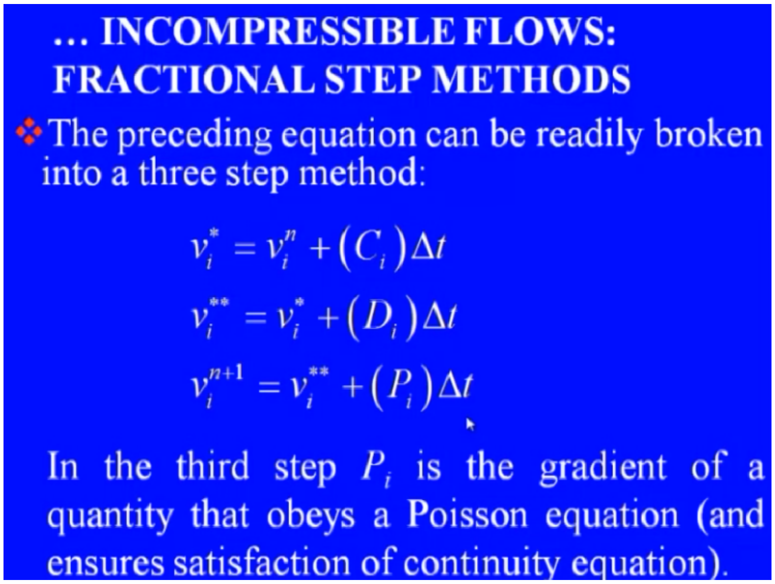


Now we will move on to a next family of methods. There are many, many algorithms which have been suggested in literature to solve incompressible Navier-Stokes equations. We have discussed only few of them. We discussed explicit time integration and implicated schemes. We would have a look at yet another set of methods which are called fractional step methods. Now these fractional step methods are essentially approximate factorizing methods.

This factorizing is fairly similar to what we had seen earlier in the case of ADI method. Say for instance if you use explicit Euler method to integrate our momentum equation, then in that case, the discretized momentum equation can be symbolically represented as vin+1=vin+delta t*Ci+Di*Pi evaluated using the values at time level n. Now here we have used these symbols C, D, and P.

Now C represents the convective term, D represents the diffusive term, which is essentially the terms, which come from our stress field and P represents the pressure gradient term. Now this particular representation opens a way to a next class of methods.

(Refer Slide Time: 30:47)

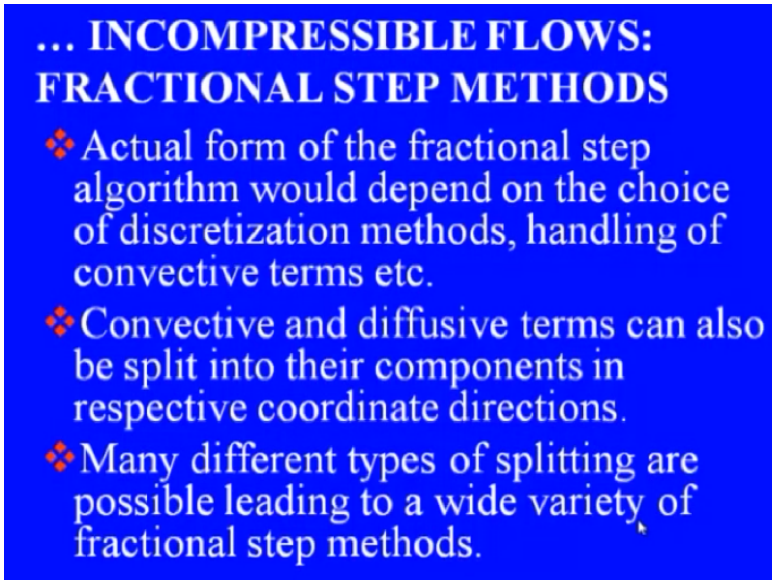


Instead of doing our computations in one step, we can break it down in 3-step method. So first step what do we do. First step, let us use the value at the previous time level and only at the contribution of the convective term. This will give us an intermediate velocity field, let us call it as vi*. So vi*=vin+Ci delta t. In next step, we will add our diffusive term as well as define a new intermediate velocity field, let us call it vi**. So vi**=vi*+Di delta t and finally we obtain the velocity field at new time level n+1 as vin+1=vi**+Pi delta t.

Now this is only one way of breaking the things up. The advantages of this breaking up are not that obvious in the context of explicit Euler method, but it would become a lot more apparent when we use implicit scheme. One thing would still be common here with what we had seen earlier with explicit integration that this Pi would be used only after we have solved our Poisson equation of pressure to ensure the satisfaction of the continuity equation.

So before we proceed for this last step of this computations, we would have solved a pressure Poisson equation using these velocity components vi**.

(Refer Slide Time: 32:27)



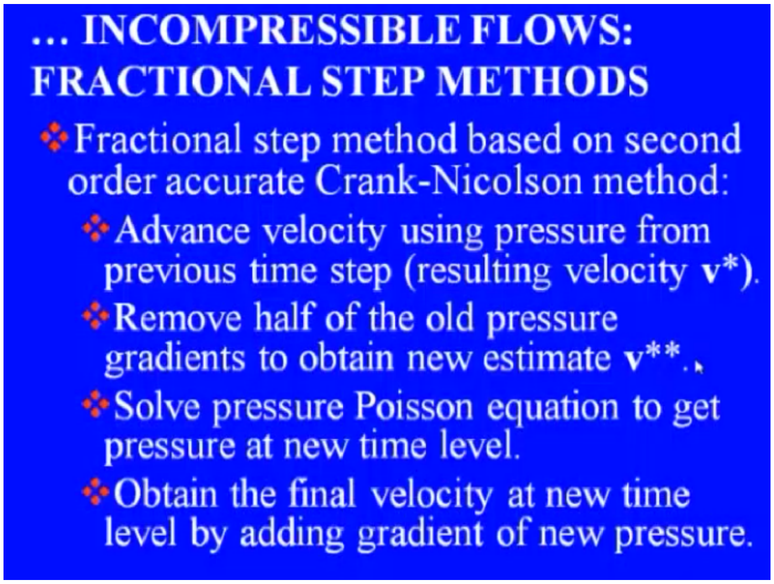
Now what would be actual algorithmic form of a fractional step method that would depend on our choice of discretizating procedure that to say whether we have used finite differences, finite volume, finite element method and in finite difference or finite volume methods, what is our computational molecule or computational stencil. How do you want to handle the convective and diffusive terms and so on. So there are wide varieties of choices.

Similarly, it will also depend on our choice of the time integration schemes. Furthermore, we can split the convective and diffusive terms into their components in respective coordinate directions and this can facilitate our computations specifically when we use an implicit time integration scheme wherein we have to solve a set of linear equations or linearized equations at each step. So if you can break our diffusion and convective terms in respective coordinate directions.

And if you use a certain type of stencils, it is possible for us break each of these steps in fractional steps into system of fragment of equations, which can be solved very, very efficiently and this many different type of splitting are possible, which lead to a wide variety of fractional step methods. So try and look into the literature, you can find at least 100 variants and let us discuss only one of these. We have already seen one based on explicit time integration.

Let us see one based on implicit time integration scheme and this would be based on a second order accurate Crank-Nicolson method.

(Refer Slide Time: 34:01)



The conceptual steps are, what we will first do is we will advance velocity using pressure from previous step and we would obtain the resulting velocity v*. Now next step what we will do is we would remove half of the old pressure gradient to obtain a new estimate v**. Next use this v** to solve a pressure Poisson equation to get pressure at new time level and then obtain the final velocity at new time level by adding the gradient of new pressure, which will ensure the satisfaction of the continuity equation.

So now let us have a look at the equations, which are involved and this fractional step method based on Crank-Nicolson scale. So let us get to our board to derive the appropriate equations.

(Refer Slide Time: 34:59)

Fractional Step Method

based Crank-Nicolson Method

Step 1 Use pressure at previous time level for advancement of velocity field.

Notation: Semi-discrete Navier-Stokes Eqn.

$$\frac{\partial(\rho v_i)}{\delta t} = H_i - \frac{\delta p}{\delta x} \qquad (1)$$

(represent contribution from convective and diffusive terms)

$$(\rho v_i^*) - \rho v_i^n = \Delta t \left[\frac{1}{2} H_i(v_i^n) + H_i(v_i^*) \right] - \Delta t \frac{\delta p^n}{\delta x_i} \qquad (2)$$

It is fractional step method based on Crank-Nicolson method. Now this is for the sake of illustration purposes. This is only one way. The similar approach can be utilized to obtain a wide variety of fractional step method as I have mentioned earlier. So our step 1 what do we do in step 1. We would use pressure at previous time level for advancement of velocity field and remember the use of notation earlier.

The earlier notation which I have used was short hand notation, which have adopted for semi discrete in Navier-Stokes equation. We can write this as del of rho vi/del t=hi-delta p/delta xi where this hi represents contribution from convective and diffusive terms. So this was our initial equation. Now we are using Crank-Nicolson scheme. So Crank-Nicolson scheme involves the values at the new time level as well as the value the old time level and simple average of 2.

That is what the basis of Crank-Nicolson method. Pressure we do not know the new value, so we will use only the old values here. So our step 1 becomes rho vi*-rho vin=delta t*1/2 hivi at n+hi at vi*-delta t*delta pn/delta xi. Let us call this as equation 2. Now remember here, we have got the terms in vi* on both the sides. So this would represent a system of equations, which have to be solved to obtain this solution vi*.

In step 2, what we would do is, we would remove half the contribution, which we have incorporated in equation 2 from the previous time level. So remove half of the old pressure gradient from vi* to obtain vi**.

(Refer Slide Time: 40:05)

$$\boxed{\rho v_i^{**} = \rho v_i^{*} + \frac{\Delta t}{2}\frac{\delta p^{n}}{\delta x_i}} \qquad (3)$$

Final velocity field is obtained by introducing the gradient pressure field at t^{n+1}:

$$(\rho v_i)^{n+1} = (\rho v_i^{**}) - \frac{\Delta t}{2}\left(\frac{\delta p^{n+1}}{\delta x_i}\right) \qquad (4)$$

v_i^{n+1} must satisfy the continuity eqn. Hence

$$\cancel{\frac{\delta}{\delta x_i}(\rho v_i)^{n+1}}^{\,0} = \frac{\delta}{\delta x_i}(\rho v_i^{**}) - \frac{\Delta t}{2}\frac{\delta}{\delta x_i}\left(\frac{\delta p^{n+1}}{\delta x_i}\right)$$

$$\Rightarrow \boxed{\frac{\delta}{\delta x_i}\left(\frac{\delta p^{n+1}}{\delta x_i}\right) = \frac{2}{\Delta t}\frac{\delta}{\delta x_i}(\rho v_i^{**})}$$

↑ Poisson eqn. for pressure p^{n+1}

That is rho vi**=rho vi*+delta t/2 delta pn/delta xi. So this is our second step. Now final velocity field is obtained by introducing the gradient of pressure field at tn+1, but that is still unknown. So let us write down that in theoretical format, what we would like to do is just write that rho vi at n+1=rho vi**-delta t/2 delta p at n+1/delta xi. Let us call this as equation 4. Before we can use this formula to compute vin+1, we ought to know what is pn+1. So how do we do that.

For that, let us derive a pressure Poisson equation so vin+1 must satisfy the continuity equation. Hence let us take divergence of 4, so what do we get, del/del xi of rho vin+1=del/del xi of rho vi**-delta t/2 del/del xi of del pn+1/delta xi. LHS must be 0 because vin+1 must satisfy our continuity equation, so let us rearrange and this will give us the discrete Poisson equation for pressure that is del/del xi of delta pn+1/delta xi=2/delta t delta/delta xi rho vi**.

So this is Poisson equation for pressure at the time level n+1, rather pn+1. So now we have got all the equations with us. So in fact our steps would be step 1, we have already seen that advance the velocity field using the known pressure value that was the step 1. In step 2, we removed half the old pressure value, so thereby we obtained this vi**. Next we use this vi** in pressure Poisson equation to solve for pn+1.

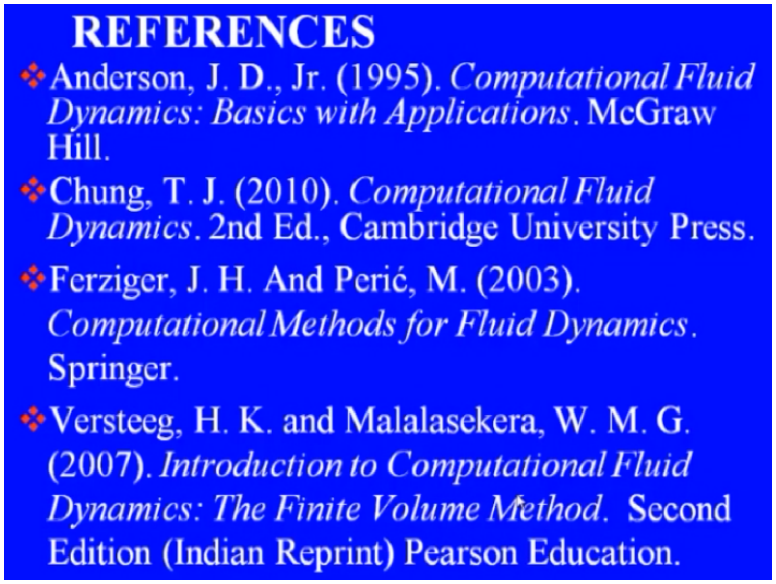
And then add its effect to vi** to get the velocity field at time level n+1, so that is whatever actual fractional step algorithm. So these were the formal steps that is advance velocity using pressure from the previous time step. We obtained the resulting velocity which we called v* and then from this velocity field v*, we removed half the old pressure gradients to obtain a new estimate v**. Now this was used in the derivation of the pressure Poisson equation.

Take divergence of this, use it as a source term in the pressure Poisson equation and solve the pressure Poisson equation to get the pressure at new time level then, to obtain the final velocity at new time level, we have to add the gradient of new pressure into v** and that completes our solution process at the given time step. This way we would put a full stop to our discussions on the fractional step methods.

There are quite a few other interesting algorithms for the integration of incompressible Navier-Stokes equations. When we had our discussions we saw that the conceptual solution of compressible form is very simple, because we had the appropriate number of equations. Pressure was obtained from the equation of this state. We had one equation continuity for density, 3 momentum equations for the velocity components or the related fluxes rho u, rho v, and rho w, then we had energy equation and we solved and we got our solution field.

Convey the similar thing, so the same type of technique, which is called artificial compressibility method wherein the continuity equation is modified to incorporate the artificial compressibility for incompressible flows and we can use a similar solution process such we had used in the case of compressible flows. So that is the same time that we are solving incompressible Navier-Stokes equations.

(Refer Slide Time: 46:43)



So I would encourage you to look into these nice books. The first one is Anderson Jerry, Computational Fluid Dynamics Basics and Applications. This book deals particularly with aerodynamic application, so you will find very beautiful discourse on very simple time integration schemes for compressible flows in the context of finite difference especially discretizing schemes.

Chung's book on Computational Fluid Dynamics that is the compendium of almost all known schemes till 2010. It has got discussions on solution of Navier-Stokes equation using finite differences, finite volume, finite element techniques and plethora of the time integration schemes for both compressible as well as incompressible flows and their applications. So you can refer to this book.

Ferziger and Peric's book is yet another one, which provides a very nice introduction to computational method for fluid dynamics and if you are interested only in finite volume method, Versteeg and Malalasekara's book gives you a very good introduction to the application of finite volume method in solution of CFD specifically in solution of Navier-Stokes equations and the solution of advection and diffusion problems.

So with timing, we would stop as far this course is concerned in connection with Navier-Stokes equations. In the next module, we will take up application of what we have learnt so far for solution of turbulent flows.