A question about pressure boundary conditions for Velocity Correction Scheme in Nektar++

Krzysztof Michałowski

1 Adding pressure field

In a paper [1] the authors present how to integrate a advection-diffusion equation (with explicit treatment for advection) in physical space. VelocityCorrectionScheme integrates 3 components of velocity in a similar manner but it has an additional pressure field which is treated implicitly (is a part of G). We are interested in solving for cartesian components of velocity thats why for a single component (in this case x component) equation (52) is now:

$$H^{HH}\hat{u}_i^H + H^{HD}\hat{u}_i^D = (B^H)^T W(\Delta t\sum_{j=1}^{i-1} a_{ij}^{IM}\boldsymbol{G}_j + \Delta t\sum_{j=1}^{i-1} a_{ij}^{EX}\boldsymbol{F}_j + \sum_{j=1}^{i-1} u_{ij}\boldsymbol{u_j}) + \upsilon a_{ii}\Delta t\Gamma_i^H - a_{ii}\Delta t(B^H)^T W(\frac{\partial p}{\partial x}) \tag{1}$$

The additional term is an inner product of one component of the pressure gradient with respect to basis functions that are 0 on Dirichlet boundary. This means that for every component of velocity we have to solve a Helmholtz problem, in a cartesian coordinate system this means that the equation for a velocity vector is:

$$\vec{u}_i - \upsilon a_{ii}\Delta t\nabla^2\vec{u}_i = \vec{X} - a_{ii}\Delta t\nabla p \tag{2}$$

Where $\vec{X}$ contains $\Delta t\sum_{j=1}^{i-1} a_{ij}^{IM}\boldsymbol{G}_j + \Delta t\sum_{j=1}^{i-1} a_{ij}^{EX}\boldsymbol{F}_j + \sum_{j=1}^{i-1} u_{ij}\boldsymbol{u_j}$ for each of the velocity components and $\vec{u}_i$ is either a stage value or a solution from the next time step (or both).

2 Solution

To be able to solve this, we need to calculate the pressure field. It is calculated so that it ensures that $\vec{u}$ is divergent free. The equation for pressure is:

$$\nabla^2 p = \frac{1}{a_{ii}\Delta t}\nabla\cdot\vec{X} \tag{3}$$

My question is about the boundary conditions for this equation. If i understood correctly from the code, you are approximating (by extrapolating the terms that are not known yet, when solving for pressure, from the previous time steps) the following boundary condition:

$$\frac{\partial p}{\partial\vec{n}} = \vec{n}\cdot(\frac{\partial\vec{u}}{\partial t} - \nabla\vec{u}\cdot\vec{u} - \nabla\times(\nabla\times\vec{u})) \tag{4}$$

Generaly this boundary condition is used with the incompressible Navier-Stokes equation and the following problem for pressure:

$$\nabla^2 p = -\nabla \cdot (\nabla \vec{u} \cdot u) \tag{5}$$

If velocity and pressure satisfy the incompressible Navier-Stokes equation and (4) and (5), it can be show that the velocity field is divergent free. It can be done by first showing (based on the N-S and pressure equation) that the divergence of velocity (h) satisfies the following equation:

$$\frac{\partial h}{\partial t} = \upsilon \nabla^2 h \tag{6}$$

Than (based on the pressure boundary condition and the N-S equation) it can be show that h satisfies the following boundary condition:

$$\frac{\partial h}{\partial \vec{n}} = 0 \tag{7}$$

If h satisfies (6), (7) and it is equal to zero initially, then the only possible solution is h=const=0. I mentioned this derivation process because i am wondering why are you trying to approximate the pressure boundary conditions for the exact system (so N-S equations + (5)) if we are solving the discrete in time system (equations (2) + (3))? I repeated the derivation procedure, that I described earlier, for the discrete solution and it seams that if we want the solution of the discrete in time N-S system (2) to be incompressible the pressure should satisfy (3) and the following boundary condition:

$$\frac{\partial p}{\partial \vec{n}} = \vec{n} \cdot (\frac{1}{a_{ii}\Delta t}\vec{X} - \frac{1}{a_{ii}\Delta t}\vec{u} - \upsilon \nabla \times (\nabla \times \vec{u})) \tag{8}$$

When solving for pressure (in a segregated solution) this BC also needs to be approximated by extrapolating $\nabla \times (\nabla \times \vec{u})$ and if BC for $\vec{u}$ is not a Dirichlet BC than we also need to extrapolate $\vec{u}$. For a fully initialized (so when we have values and explicit derivatives from appropriate number of previous time steps) IMEX scheme (8) and (4) should be equivalent, but IMEX schemes of order more than 1 have to be initialized with a multistage method (in Nektar++ that would be IMEXdirk) and during this initialization stage (8) and (4) are not equivalent.

3 Summary

It seams to me that you are using a BC for pressure which guarantees incompressibility of a solution of an exact N-S system but you are instead solving a discrete in time N-S system which needs a different BC for pressure to enforce incompressibility. I was wondering why did you make that decision?
I made a slight adjustment for VelocitCorrectionSheme class so that it enforces (8), and for now it seams to work but i need to make it work with all possible BC's for $\vec{u}$, and I was wondering if it makes sense to do that?
I also can't understand why during the initialization, when using a multi stage method, you are extrapolating $\nabla \times (\nabla \times \vec{u})$ and $\nabla\vec{u} \cdot \vec{u}$ using their values from previous stage values and not previous time steps? Of course you can't use previous time steps because we simply don't have them, but i can't understand why we can use stage values instead.
I was thinking that maybe, for the initialization steps, to calculate each of their stage values one could do the following:

1. Solve (3) with (8) BC assuming that the unknown terms are exactly the same as in a previous time step.
2. Using the pressure calculated in the first step solve (2) obtaining some $\vec{\tilde{u}}$ and calculate the unknown terms in (8) based on $\vec{\tilde{u}}$.
3. Solve (3) with (8) again.
4. Solve (2) obtaining $\vec{u}$ which is a stage value.

Of course this method would double the time needed for first few steps.
In general my questions are only about the first few initialization steps when we can't use the one stage IMEX scheme (of order grater than 1).

References

[1] P. E. J. Vos, C. Eskilsson, A. Bolis, S. Chun, R. M. Kirby, and S. J. Sherwin, "A generic framework for time-stepping partial differential equations (PDEs): general linear methods, object-oriented implementation and application to fluid problems," *International Journal of Computational Fluid Dynamics*, vol. 25, no. 3, pp. 107–125, 2011, Taylor & Francis.