The Immersed Interface Method for the axis-symmetric Navier-Stokes equations in cylindrical coordinates

Juan Ruiz[1] Zhilin Li[2]

[1]Department of Mathematics. Universidad de Alcalá (Madrid, Spain).
[2]Department of Mathematics. North Carolina State University (USA).

International Workshop on Fluid-Structure Interaction Problems
Singapore, 2016

Outline

1. Introduction
    - Domain
    - An outline of the solution of the N-S equations using the IIM
    - Level set function
2. The projection method
    - Correction of the derivatives in the z and r directions
3. Numerical solution using a multigrid method
4. Numerical experiments
5. Conclusions and future work

Outline

1. Introduction
   - Domain
   - An outline of the solution of the N-S equations using the IIM
   - Level set function
2. The projection method
   - Correction of the derivatives in the z and r directions
3. Numerical solution using a multigrid method
4. Numerical experiments
5. Conclusions and future work

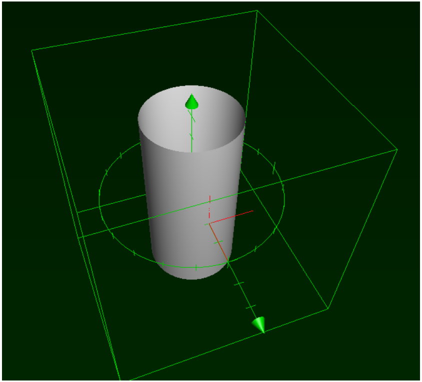
Motivation

- In **The IIM for the Navier-Stokes equations with Singular Forces. Z. Li, Ming-Chih Lai. Journal of Computational Physics. 2001.** The authors present the **Projection-IIM** method in order to solve the N-S equations assuring second order accuracy for the velocity and the pressure in Cartesian coordinates.
- **An Immersed Interface Method for Axisymmetric Electrohydrodynamic Simulations in Stokes flow**. H. Nganguia, Y.N. Young, A. T. Layton, W.F. Hu, and M.C. Lai. Communications in Computational Physics. 2015.

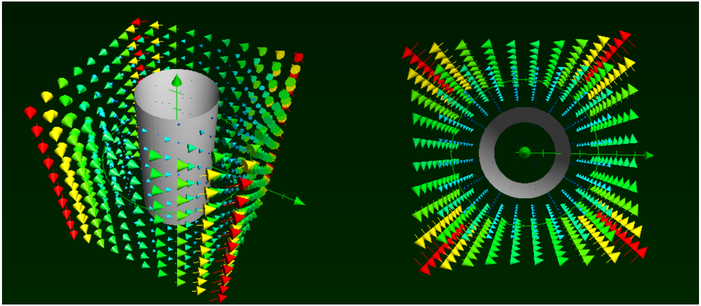
Motivation

- Would it be easy to extend these results to cylindrical problems with axial symmetry?
- The equations will change.
- The jump conditions will change.

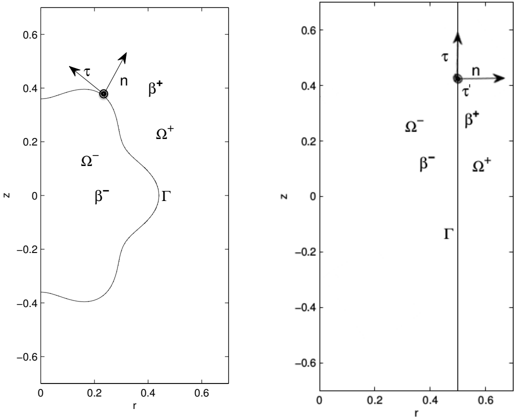
Example of geometry and domain



Example of geometry and domain



Domain



Equations of the problem in Cartesian Coordinates

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p = \mu\Delta\mathbf{u} + \mathbf{G}, \quad \mathbf{x}\in\Omega,$$

$$\nabla\cdot\mathbf{u} = 0,$$

$$\mathbf{u}\big|_{\partial\Omega} = \mathbf{u}_b, BC,$$

$$\mathbf{u}(\mathbf{x},\mathbf{t}=\mathbf{0}) = \mathbf{u}_0, IC.$$

The singular force $\mathbf{G}$ is supposed to have a support only on the immersed interface $\Gamma(t)$, and takes the form,

$$\mathbf{G}(\mathbf{x},t) = \int_{\Gamma(t)} \mathbf{f}(s,t)\delta_2(\mathbf{x}-\mathbf{X}(s,t))ds,$$

Equations of the problem in Cartesian Coordinates

$$
\begin{aligned}
\rho\left(\frac{\partial u_x}{\partial t}+u_x\frac{\partial u_x}{\partial x}+u_y\frac{\partial u_x}{\partial y}\right)+\frac{\partial p}{\partial x} &= \mu\left(\frac{\partial^2 u_x}{\partial x^2}+\frac{\partial^2 u_x}{\partial y^2}\right)+F_x, \quad x\in\Omega,\\
\rho\left(\frac{\partial u_y}{\partial t}+u_x\frac{\partial u_y}{\partial x}+u_y\frac{\partial u_y}{\partial y}\right)+\frac{\partial p}{\partial y} &= \mu\left(\frac{\partial^2 u_y}{\partial x^2}+\frac{\partial^2 u_y}{\partial y^2}\right)+F_y, \quad y\in\Omega,\\
\nabla\cdot\mathbf{u} &= 0,\\
\mathbf{u}\big|_{\partial\Omega} &= \mathbf{u}_b, BC,\\
\mathbf{u}(\mathbf{x},\mathbf{t}=\mathbf{0}) &= \mathbf{u}_0, IC,
\end{aligned}
$$

Equations of the problem in cylindrical Coordinates with axial symmetry

$$
\begin{aligned}
\rho\left(\frac{\partial u_r}{\partial t}+u_r\frac{\partial u_r}{\partial r}+u_z\frac{\partial u_r}{\partial z}\right)+\frac{\partial p}{\partial r} &= \mu\left(\frac{u_r}{r}+\frac{\partial^2 u_r}{\partial r^2}+\frac{\partial^2 u_r}{\partial z^2}-\frac{u_r}{r^2}\right)+F_r, \quad r\in\Omega,\\
\rho\left(\frac{\partial u_\varphi}{\partial t}+u_r\frac{\partial u_\varphi}{\partial r}+\frac{u_r u_\varphi}{r}+u_z\frac{\partial u_\varphi}{\partial z}\right) &= \mu\left(\frac{u_\varphi}{r}+\frac{\partial^2 u_\varphi}{\partial r^2}+\frac{\partial^2 u_\varphi}{\partial z^2}-\frac{u_\varphi}{r^2}\right)+F_\varphi, \quad \varphi\in\Omega,\\
\rho\left(\frac{\partial u_z}{\partial t}+u_r\frac{\partial u_z}{\partial r}+u_z\frac{\partial u_z}{\partial z}\right)+\frac{\partial p}{\partial z} &= \mu\left(\frac{u_z}{r}+\frac{\partial^2 u_z}{\partial r^2}+\frac{\partial^2 u_z}{\partial z^2}\right)+F_z, \quad z\in\Omega,\\
\nabla\cdot\mathbf{u} &= 0,\\
\mathbf{u}\big|_{\partial\Omega} &= \mathbf{u}_b,\ BC,\\
\mathbf{u}(\mathbf{x},\mathbf{t}=\mathbf{0}) &= \mathbf{u}_0,\ IC,
\end{aligned}
$$

Steps to solve the problem:

In order to solve the problem we need to:

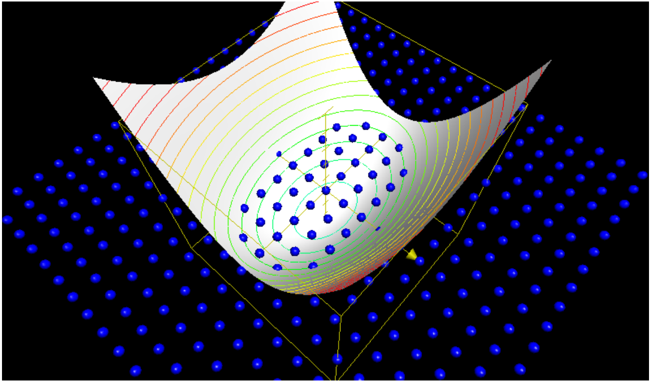
1. Find irregular grid points in the r and z directions.
2. Find the projection of irregular grid-points over the interface in the r and z directions.
3. Find the surface derivatives and interpolate them at the projection in order to obtain the curvature of the interface at the projection and to set a local coordinates system. This is due to the fact that the interface relations are given in a local coordinate system centered on the interface.
4. Discretize the NS equations for the three components of the velocity $\mathbf{u}(r,z,t) = (u_r(r,z,t), u_z(r,z,t), u_\varphi(r,z,t))$ and for the pressure $p(r,z,t)$ using the projection method.
5. Correct the derivatives of the projection method using the IIM.

Steps to solve the problem:

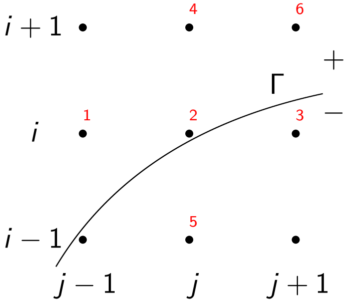
6. Solve the three elliptic equations for the velocity and the poisson equation for the projection using a multigrid method.
7. Update the pressure and the velocity.
8. If we are solving a free boundary problem, update the position of the interface and then iterate again.

The level set function

Using the level set, it is very easy to accomplish the three first steps.



Irregular and regular grid points

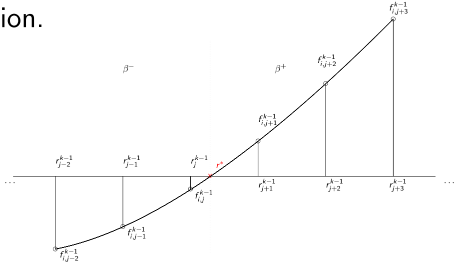


Finite difference discretization

- At regular grid points we use centered finite difference discretization.
- In order to correct the derivatives at irregular grid points, we need to set a local coordinates system. Thus, we need to find the projection of the central point of the stencil over the interface in the r and z directions.

Projection in the r and z directions

- The projection can be easily found using quadratic Lagrange interpolation.



Projection in the r and z directions and local coordinates system

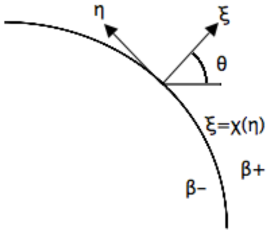
- Once we have the projection $X^* = (r^*, z^*)$ on the interface, the unit normal direction of the interface at X^* is:

$$\xi = \frac{\nabla\varphi}{|\nabla\varphi|} = \frac{(\varphi_r, \varphi_z)}{\sqrt{\varphi_r + \varphi_z}}$$

- The unit tangential direction at $X^*$is:

$$\eta = \frac{(\varphi_z, -\varphi_r)}{\sqrt{\varphi_r + \varphi_z}}$$

Local Coordinates



Local Coordinates

We use a local coordinates system:

$$\begin{aligned} \xi &= (r-R)\cos(\theta) + (z-Z)\sin(\theta), \\ \eta &= -(r-R)\sin(\theta) + (z-Z)\cos(\theta) \end{aligned} \tag{1}$$

At the point (X^*, Y^*), the interface can be written as:

$$\xi = \chi(\eta), \ \textit{with} \ \chi(0) = 0, \ \chi'(0) = 0.$$

Surface derivatives in 2D and Bilinear Interpolation

- In order to obtain the curvature of the interface, we need the surface derivatives of the level set: we can use central differencing.
- But the level set function is only defined at grid points φ_{ij}: maybe not defined exactly on the interface.
- We can use the bilinear interpolation to obtain the interface information at the projections since we assume that the level set function has up to second order continuous partial derivatives in a neighborhood of the interface.

Translating the jump relations to the x and y directions

The interface relations are usually given in the local coordinates, $[u_\xi], [u_\eta], [u_{\xi\xi}], [u_{\xi\eta}], [u_{\eta\eta}]$. They must be translated to the cylindrical coordinates in order to introduce them in the algorithm. It is easy to see that,

$$
\begin{aligned}
[u_r] &= [u_\xi]\cos(\theta) - [u_\eta]\sin(\theta), \\
[u_z] &= [u_\xi]\sin(\theta) + [u_\eta]\cos(\theta), \\
[u_{rr}] &= [u_{\xi\xi}]\cos^2(\theta) - 2[u_{\xi\eta}]\cos(\theta)\sin(\theta) + [u_{\eta\eta}]\sin^2(\theta), \\
[u_{zz}] &= [u_{\xi\xi}]\sin^2(\theta) - 2[u_{\xi\eta}]\cos(\theta)\sin(\theta) + [u_{\eta\eta}]\cos^2(\theta).
\end{aligned}
$$

Using these equations and the interface relations, we can write $[\mathbf{u}], [\mathbf{u}_r], [\mathbf{u}_z], [\mathbf{u}_{rr}], [\mathbf{u}_{zz}], [p], [p_r]$ and $[p_z]$ at the points of the interface in terms of the geometric information of the interface and the force strength and its derivatives.

Outline

1. Introduction
   - Domain
   - An outline of the solution of the N-S equations using the IIM
   - Level set function
2. The projection method
   - Correction of the derivatives in the z and r directions
3. Numerical solution using a multigrid method
4. Numerical experiments
5. Conclusions and future work

The projection method

The method from the time t^n to t^{n+1} can be expressed as,

$$\begin{aligned}\frac{\mathbf{u}^* - \mathbf{u^n}}{\Delta t} + (\mathbf{u}\cdot\nabla_h\mathbf{u})^{n+1} &= \nabla p^{n-\frac{1}{2}} + \frac{\mu}{2}(\Delta_h\mathbf{u}^* + \Delta_h\mathbf{u}^n) + \mathbf{G}^{n+\frac{1}{2}} + \mathbf{C}_1^n, \\ \mathbf{u}^*|_{\partial\Omega} &= \mathbf{u}_b^{n+1},\end{aligned}$$

where $\mathbf{u}^* = (u_r^*, u_z^*)$ is the intermediate velocity field of the projection method. $(\mathbf{u}\cdot\nabla_h\mathbf{u})^{n+1}$ is approximated through,

$$(\mathbf{u}\cdot\nabla_h\mathbf{u})^{n+1} = \frac{3}{2}\left(\mathbf{u}^n\cdot\nabla_h\right)\mathbf{u}^n - \frac{1}{2}\left(\mathbf{u}^{n-1}\cdot\nabla_h\right)\mathbf{u}^{n-1} + \mathbf{C}_2^n,$$

where ∇_h and Δ_h are the finite difference discretizations of the Laplacian and the gradient in cylindrical coordinates.

The projection method

The projection step used is,

$$
\begin{aligned}
\Delta_h \varphi^{n+1} &= \frac{\nabla_h \cdot \mathbf{u}^*}{\Delta t} + C_3^n, \quad \left.\frac{\partial \varphi^{n+1}}{\partial \mathbf{n}}\right|_{\partial\Omega} = 0, \\
\mathbf{u}^{n+1} &= \mathbf{u}^* - \Delta t \nabla_h \varphi^{n+1} + \mathbf{C}_4^n, \\
\nabla_h p^{n+1/2} &= \nabla_h p^{n-1/2} + \nabla_h \varphi^{n+1} + \mathbf{C}_5^n.
\end{aligned}
$$

The projection method

- $$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r^*}{\partial r}\right)+\frac{\partial^2 u_r^*}{\partial z^2}\right)-\frac{2}{\mu}\left(1+\frac{r^2}{\Delta t}\right)\frac{u_r^*}{r^2}=\frac{2}{\mu}\left(\frac{3}{2}\left(u_r\frac{\partial u_r^n}{\partial r}+u_z\frac{\partial u_r^n}{\partial z}\right)-\frac{1}{2}\left(u_r\frac{\partial u_r^{n-1}}{\partial r}+u_z\frac{\partial u_r^{n-1}}{\partial z}\right)\right.$$
$$\left.+C_{r2}^n-\frac{\partial p^{n-\frac{1}{2}}}{\partial r}\right)-\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r^n}{\partial r}\right)+\frac{\partial^2 u_r^n}{\partial z^2}-\frac{u_r^n}{r^2}\right)-\frac{u_r^n}{\Delta t}-G_r^{n+\frac{1}{2}}-C_{r1}^n$$
- $$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\varphi^*}{\partial r}\right)+\frac{\partial^2 u_\varphi^*}{\partial z^2}\right)-\frac{2}{\mu}\left(1+\frac{r^2}{\Delta t}\right)\frac{u_\varphi^*}{r^2}=\frac{2}{\mu}\left(\frac{3}{2}\left(u_r\frac{\partial u_\varphi^n}{\partial r}+u_z\frac{\partial u_\varphi^n}{\partial z}+u_r^n\frac{\partial u_\varphi^n}{\partial r}\right)\right.$$
$$\left.-\frac{1}{2}\left(u_r\frac{\partial u_\varphi^{n-1}}{\partial r}+u_z\frac{\partial u_\varphi^{n-1}}{\partial z}+u_r^n\frac{\partial u_\varphi^n}{\partial r}\right)+C_{\varphi 2}^n\right)-\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\varphi^n}{\partial r}\right)+\frac{\partial^2 u_\varphi^n}{\partial z^2}-\frac{u_\varphi^n}{r^2}\right)-\frac{u_\varphi^n}{\Delta t}-G_\varphi^{n+\frac{1}{2}}-C_{\varphi 1}^n$$
- $$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z^*}{\partial r}\right)+\frac{\partial^2 u_z^*}{\partial z^2}\right)-\frac{2}{\mu}\frac{r^2}{\Delta t}\frac{u_z^*}{r^2}=\frac{2}{\mu}\left(\frac{3}{2}\left(u_r\frac{\partial u_z^n}{\partial r}+u_z\frac{\partial u_z^n}{\partial z}\right)+\frac{1}{2}\left(u_r\frac{\partial u_z^{n-1}}{\partial r}+u_z\frac{\partial u_z^{n-1}}{\partial z}\right)\right.$$
$$\left.+C_{z2}^n-\frac{\partial p^{n-\frac{1}{2}}}{\partial z}\right)-\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z^n}{\partial r}\right)+\frac{\partial^2 u_z^n}{\partial z^2}\right)-\frac{u_z^n}{\Delta t}-G_z^{n+\frac{1}{2}}-C_{z1}^n.$$

The projection method

- $\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r^*}{\partial r}\right)+\frac{\partial^2 u_r^*}{\partial z^2}\right)-\frac{2}{\mu}\left(1+\frac{r^2}{\Delta t}\right)\frac{u_r^*}{r^2}=f_r,$
- $\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\varphi^*}{\partial r}\right)+\frac{\partial^2 u_\varphi^*}{\partial z^2}\right)-\frac{2}{\mu}\left(1+\frac{r^2}{\Delta t}\right)\frac{u_\varphi^*}{r^2}=f_\varphi,$
- $\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z^*}{\partial r}\right)+\frac{\partial^2 u_z^*}{\partial z^2}\right)-\frac{2}{\mu}\frac{r^2}{\Delta t}\frac{u_z^*}{r^2}=f_z.$

Discretization of differential operators at regular points

$$\Delta_h f = \left(\frac{1}{r_i} \frac{r_{i-\frac{1}{2}} f_{i-1,j} - (r_{i-\frac{1}{2}} + r_{i+\frac{1}{2}}) f_{i,j} + r_{i+\frac{1}{2}} f_{i+1,j}}{(\Delta r)^2} + \frac{f_{i,j-1} - 2 f_{i,j} + f_{i,j+1}}{(\Delta z)^2} \right)$$

$$\nabla_h f = \left(\frac{f_{i+1,j} - f_{i-1,j}}{\Delta r}, \frac{f_{i,j+1} - f_{i,j-1}}{\Delta z} \right)$$

$$\nabla_h \cdot f = \left(\frac{1}{r_i} f_{i,j} + \frac{f_{i+1,j} - f_{i-1,j}}{\Delta r} + \frac{f_{i,j+1} - f_{i,j-1}}{\Delta z} \right)$$

In order to deal with the pole singularity introduced by the Laplacian and the divergence operators, we use a staggered grid. That means that the r coordinate of every grid point takes the value

$$r_i = (i - 1/2)\,\Delta r, \quad \Delta r = \frac{2R_1}{M}, \quad i = 1, 2, \cdots, \frac{M+1}{2}.$$

Note that $r_1 = \Delta r/2 = h/2$ and $r_{\frac{M+1}{2}} = R_1$, being M and odd number.

This configuration makes it easy to deal with the immersed boundary condition at the left side of the boundary. As shown in Figure 1, if we set a ghost point at $r = -\Delta r/2$, then the expression of the Laplacian at $r = \Delta r/2$ takes the form,

$$\Delta u = \left(2\frac{u_{2,j} - u_{1,j}}{(\Delta r)^2} + \frac{u_{1,j-1} - 2u_{1,j} + u_{1,j+1}}{(\Delta z)^2} \right).$$

Ghost points for the immersed boundary condition

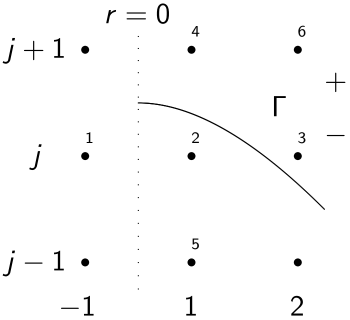


Figure: Example of an irregular point close to the left boundary of the domain and the points chosen for the stencil. The first column is composed by ghost points.

Lemma

Let $u(x)$ be a piecewise twice differentiable function. Assume that $u(x)$ and its derivatives have finite jumps $[u], [u_x]$ and $[u_{xx}]$ at $x^* = x + \alpha h$, $-1 \leq \alpha \leq 1$, then the following relation holds,

$$\frac{u(x+h) - u(x-h)}{2h} = \begin{cases} u'(x) + \frac{C(x,\alpha)}{2h} + O(h^2), & \text{if } 0 \leq \alpha \leq 1, \\ u'(x) - \frac{C(x,\alpha)}{2h} + O(h^2), & \text{if } -1 \leq \alpha < 0, \end{cases}$$

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + \frac{C(x,\alpha)}{h^2} + O(h),$$

where,

$$C(x,\alpha) = [u] + [u_x](1 - |\alpha|)h + [u_{xx}]\frac{(1-|\alpha|)^2 h^2}{2},$$

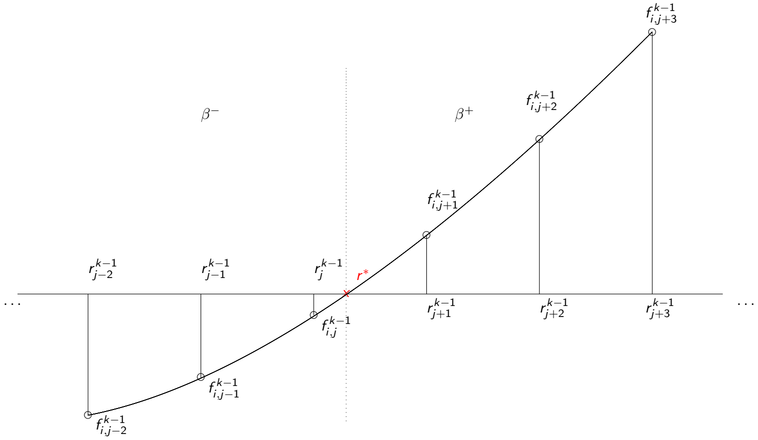
The jumps are defined as

$$[u] = \begin{cases} \lim_{x\to x^{*+}} u(x) - \lim_{x\to x^{*-}} u(x) \text{ if } 0 \leq \alpha \leq 1, \\ \lim_{x\to x^{*-}} u(x) - \lim_{x\to x^{*+}} u(x) \text{ if } -1 \leq \alpha < 0, \end{cases}$$

$$[u_x] = \begin{cases} \lim_{x\to x^{*+}} u_x(x) - \lim_{x\to x^{*-}} u_x(x) \text{ if } 0 \leq \alpha \leq 1, \\ \lim_{x\to x^{*-}} u_x(x) - \lim_{x\to x^{*+}} u_x(x) \text{ if } -1 \leq \alpha < 0, \end{cases}$$

$$[u_{xx}] = \begin{cases} \lim_{x\to x^{*+}} u_{xx}(x) - \lim_{x\to x^{*-}} u_{xx}(x) \text{ if } 0 \leq \alpha \leq 1, \\ \lim_{x\to x^{*-}} u_{xx}(x) - \lim_{x\to x^{*+}} u_{xx}(x) \text{ if } -1 \leq \alpha < 0, \end{cases}$$

Correction of the derivatives in the z and r directions



Examples of correction terms

- The correction term for $u^n D_{r,h} u^n$ from the material derivative $(\mathbf{u}^n \cdot \nabla_h)\mathbf{u}^n$:

$$-\frac{3}{4h}\left([u_r^n](r_{i+1}-r^*)+[u_{rr}^n]\frac{(r_{i+1}-r^*)^2}{2}\right)u^n(r^*,y_j).$$

- The correction term for $D_{r,h}p^{n-\frac{1}{2}}$ from the gradient of the pressure $\nabla_h p^{n-\frac{1}{2}}$:

$$-\frac{1}{2h}\left([p^{n-\frac{1}{2}}]+[p_r^{n-\frac{1}{2}}](r_{i+1}-r^*)\right)$$

- The correction term for $\frac{\mu}{2}D_{rr,h}u^n$ from the Laplacian $\frac{\mu}{2}\Delta u^n$:

$$-\frac{\mu}{2h^2}\left([u_r^n](r_{i+1}-r^*)+[u_{rr}^n]\frac{(r_{i+1}-r^*)^2}{2}\right).$$

Outline

1. Introduction
   - Domain
   - An outline of the solution of the N-S equations using the IIM
   - Level set function
2. The projection method
   - Correction of the derivatives in the z and r directions
3. Numerical solution using a multigrid method
4. Numerical experiments
5. Conclusions and future work

We need to solve three elliptic equations for the three components of u^*:

$$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r^*}{\partial r}\right)+\frac{\partial^2 u_r^*}{\partial z^2}\right)-\left(1+\frac{2}{\mu}\frac{r^2}{\Delta t}\right)\frac{u_r^*}{r^2}=f_r$$

$$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\varphi^*}{\partial r}\right)+\frac{\partial^2 u_\varphi^*}{\partial z^2}\right)-\left(1+\frac{2}{\mu}\frac{r^2}{\Delta t}\right)\frac{u_\varphi^*}{r^2}=f_\varphi$$

$$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z^*}{\partial r}\right)+\frac{\partial^2 u_z^*}{\partial z^2}\right)-\left(\frac{2}{\mu}\frac{r^2}{\Delta t}\right)\frac{u_z^*}{r^2}=f_z.$$

and one Poisson equation for the projection step:

$$\begin{aligned}\Delta_h\varphi^{n+1} &= \frac{\nabla_h\cdot\mathbf{u}^*}{\Delta t}+C_3^n, \quad \left.\frac{\partial\varphi^{n+1}}{\partial\mathbf{n}}\right|_{\partial\Omega}=0,\\ \mathbf{u}^{n+1} &= \mathbf{u}^*-\Delta t\nabla_h\varphi^{n+1}+\mathbf{C}_4^n,\\ \nabla_h p^{n+1/2} &= \nabla_h p^{n-1/2}+\nabla_h\varphi^{n+1}+\mathbf{C}_5^n.\end{aligned}$$

The first three equations do not have the form of a Helmholtz equation in cylindrical coordinates,

$$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{\partial^2 u}{\partial z^2}\right)+\lambda\frac{u}{r^2}=f(r,z),$$

as λ is not a constant anymore, due to its dependence on r. Thus, the linear system of algebraic equations resulting from the FD discretization can not be solved using the subroutines for Helmholtz equations in cylindrical coordinates of Fishpack package. Even so, a general multigrid method for elliptic equations, such as DMGD9V can be used to solve this part of the problem.

Outline

1. Introduction
   - Domain
   - An outline of the solution of the N-S equations using the IIM
   - Level set function
2. The projection method
   - Correction of the derivatives in the z and r directions
3. Numerical solution using a multigrid method
4. Numerical experiments
5. Conclusions and future work

Validation with a fixed interface 1

- Domain in $z = [-1, 1]$, $r = [0, 1]$.
- The interface is the straight line $r = 0.5$

$$
\begin{aligned}
p(r,z,t) &= 0, \quad u(r,z,t) = 0 \quad v(r,z,t) = 0 \\
w(r,z,t) &= \begin{cases} h(t)(2r-1) & \text{if} \quad r > \frac{1}{2}, \\ 0 & \text{if} \quad r \leq \frac{1}{2}, \end{cases}
\end{aligned}
$$

- $\mu = 2$, $\rho = 1$, $\Delta t = \frac{h}{2}$, final time $t = 2$.
- The velocity verifies the incompressibility constraint and is continuous, but presents a jump in the normal derivative, $[u_n] = [u_r] = 2h(t)$, $[v_n] = [u_r] = 0$, $[w_n] = [u_r] = 0$.
- **G** is determined from the exact solution. There is a finite jump in **G**.
- We take $h(t) = 1 - e^{-t}$.

Jump Conditions

In *Immersed interface methods for stokes flow with elastic boundaries or surface tension. R. J. Leveque, Z. Li. SIAM J. Sci. Comput. 1997.*, the authors show that the jump conditions across the interface for the Navier Stokes equations for the 2D problem can be expressed as,

$$\begin{aligned} [\mathbf{u}] &= 0, \quad [\mu \mathbf{u_n}] = -\hat{f}_2 \tau, \\ [p] &= \hat{f}_1, \quad [p_n] = \frac{\partial \hat{f}_2}{\partial s} + [\mathbf{G}] \cdot \mathbf{n}, \end{aligned}$$

where $\tau = (-\sin(\theta), \cos(\theta))$ is the unit tangent direction. $\hat{f}_1$ and $\hat{f}_2$ are the force strengths in the normal and tangential directions,

$$\begin{aligned} \hat{f}_1(s,t) &= f_1(s,t)\cos(\theta) + f_2(s,t)\sin(\theta) \\ \hat{f}_2(s,t) &= -f_1(s,t)\sin(\theta) + f_2(s,t)\cos(\theta). \end{aligned}$$

Interface relations for the Axis-Symmetric case

It is easy to prove that the interface relations for the Navier-Stokes equations with constant density ρ and viscosity μ are,

$$[p] = \hat{f}_1, \quad [p_\xi] = \frac{\partial \hat{f}_2}{\partial \eta}$$

$$[\mathbf{u}] = 0, \quad [\mu \mathbf{u}_\xi] = -\hat{f}_2 \tau \quad [\mathbf{u}_\eta] = 0,$$

$$[\mu \mathbf{u}_{\eta\eta}] = \kappa \hat{f}_2 \hat{\tau}, \quad [\mu \mathbf{u}_{\xi\eta}] = -\frac{\partial \hat{f}_2}{\partial \eta}\hat{\tau} - \kappa \hat{f}_2 \hat{n},$$

$$[\mu \mathbf{u}_{\xi\xi}] = -[\mu \mathbf{u}_{\eta\eta}] + [p_\xi]\hat{n} + [p_\eta]\hat{\tau} + \rho(\mathbf{u} \cdot \hat{n} - \frac{\mu}{r}\cos(\theta))[\mathbf{u}_\xi] - [\mathbf{G}].$$

N	$\|\|E_N(\mathbf{u})\|\|_\infty$	Order
16×32	3.1705e-3	-
32×64	9.0961e-4	1.804
64×128	2.444e-4	1.8962
128×256	6.3383e-5	1.9469

Validation with a fixed interface 2

- Domain in $z = [-1, 1]$, $r = [0, 1]$.
- The interface is the straight line $r = 0.5$

$$
\begin{aligned}
p(r,z,t) &= \begin{cases} 3, & \text{if} \quad r > \frac{1}{2}, \\ 0 & \text{if} \quad r \leq \frac{1}{2}, \end{cases} \\
u(r,z,t) &= 0, \quad v(r,z,t) = 0 \\
w(r,z,t) &= \begin{cases} h(t)(2r-1) & \text{if} \quad r > \frac{1}{2}, \\ 0 & \text{if} \quad r \leq \frac{1}{2}, \end{cases}
\end{aligned}
$$

- $\mu = 2$, $\rho = 1$, $\Delta t = \frac{h}{2}$, final time $t = 2$.
- The velocity verifies the incompressibility constraint and is continuous, but presents a jump in the normal derivative, $[u_n] = [u_r] = 2h(t)$, $[v_n] = [u_r] = 0$, $[w_n] = [u_r] = 0$.
- $\mathbf{G}$ is determined from the exact solution. There is a finite

N	$\|\|E_N(\mathbf{u})\|\|_\infty$	Order
16×32	3.1704e-3	-
32×64	9.096e-4	1.8014
64×128	2.4437e-4	1.8962
128×256	6.3383e-5	1.9469

Validation with a fixed interface 3

- Domain in $z = [-1, 1]$, $r = [0, 1]$.
- The interface is the straight line $r = 0.5$

$$
\begin{aligned}
p(r,z,t) &= \begin{cases} 3r-1, & \text{if} \quad r > \frac{1}{2}, \\ 0 & \text{if} \quad r \leq \frac{1}{2}, \end{cases} \\
u(r,z,t) &= 0, \quad v(r,z,t) = 0 \\
w(r,z,t) &= \begin{cases} h(t)(2r-1) & \text{if} \quad r > \frac{1}{2}, \\ 0 & \text{if} \quad r \leq \frac{1}{2}, \end{cases}
\end{aligned}
$$

- $\mu = 2$, $\rho = 1$, $\Delta t = \frac{h}{2}$, final time $t = 2$.
- We take $h(t) = 1 - e^{-t}$.

N	$\|\|E_N(\mathbf{u})\|\|_\infty$	Order
16×32	3.497e-3	-
32×64	1.003e-3	1.8012
64×128	2.6959e-4	1.8961
128×256	6.9928e-5	1.9468

Outline

1. Introduction
   - Domain
   - An outline of the solution of the N-S equations using the IIM
   - Level set function
2. The projection method
   - Correction of the derivatives in the z and r directions
3. Numerical solution using a multigrid method
4. Numerical experiments
5. Conclusions and future work

Conclusions

- The solution of the NS equations using the IIM in cylindrical coordinates presents substantial differences with the solution of the problem in Cartesian coordinates: the equations are different, the interface relations are different, a staggered grid is needed due to the pole condition near the origin and we can not use Fishpack to solve the linear system of algebraic equations resulting from the FD discretization.
- A software package for the NS equations with axial symmetry and fixed interface in cylindrical coordinates has been developed.
- From the experiments, second order accuracy is observed for the velocity.

Future work

- We are working to include a moving interface in order to reach the objective of solving free boundary problems.

Thank you very much for your attention!