Two phase flow simulations based on Immersed boundary method by utilizing MUMPS solver

Yuri Feldman,

MUMPS User Days,

Grenoble, France, June 2017

Ben-Gurion University of the Negev

Outline

1. History of using MUMPS
2. Motivation of the Research
3. Governing Equations
4. Numerical Method
5. Direct Numerical Simulations
    - Rising bubble
    - Droplet Relaxation to Disk
6. Future Optimization Strategies
7. Conclusions and Wish List

Fully Pressure – Velocity Coupling

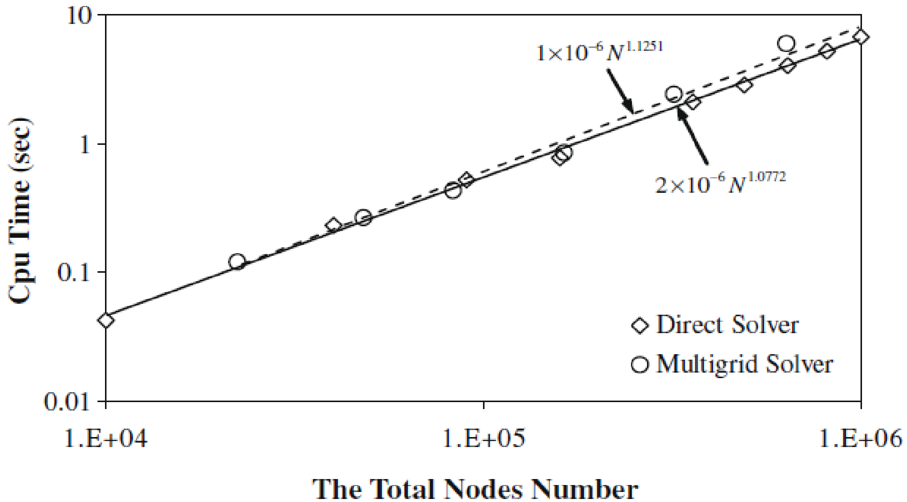
$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \frac{1}{Gr^{0.5}} \nabla^2 \boldsymbol{u} + \theta \overrightarrow{e_z}$$

$$\frac{\partial \theta}{\partial t} + (\boldsymbol{u} \cdot \nabla)\theta = \frac{1}{PrGr^{0.5}} \nabla^2 \theta$$

- ✓ No extra prediction-correction step for pressure is required
- ✓ Large Indefinite ill conditioned matrix

(Feldman, Gelfgat , Computers&Structures, 2009)



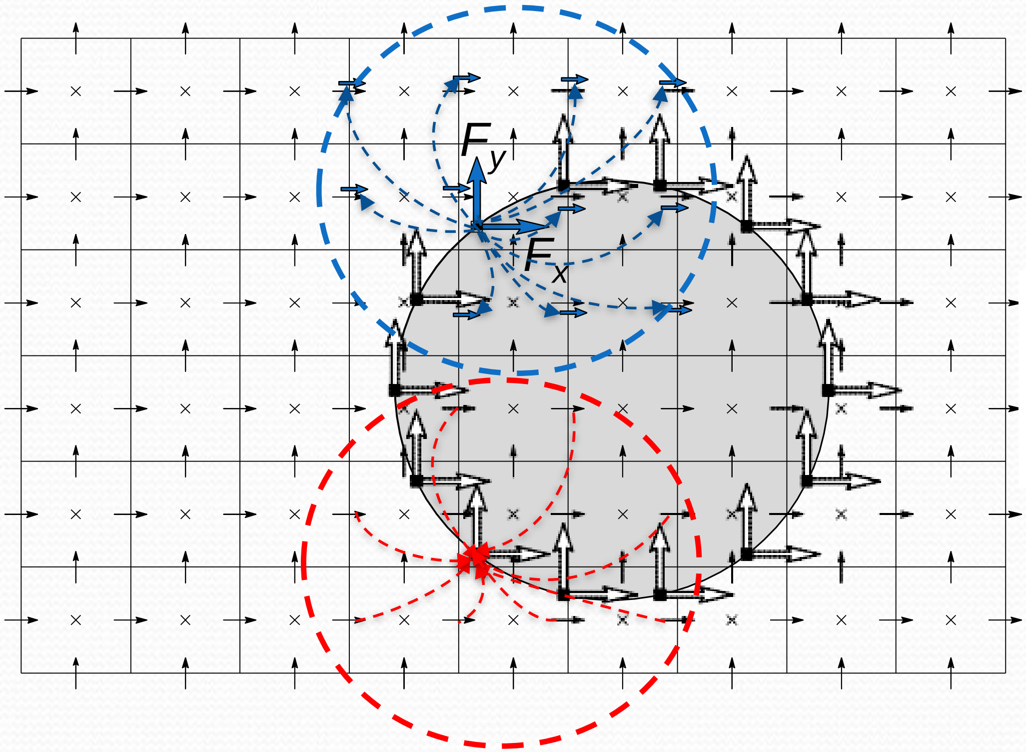
Sequential version of MUMPS

- ✓ Very efficient for 2D problems

Immersed Boundary (IB) Method

- Simulation of flows in the presence of complex geometries and moving boundaries.
- Straight forward computation of the forces/heat fluxes acting on the immersed boundary.
- The implementation of the method requires very limited modifications of the existing time stepping/linear stability codes.

Concept of IB Method



The same discrete Delta function for regularization /interpolation

$$d(r)=\begin{cases}\frac{1}{6\Delta r}\left[5-3\frac{|r|}{\Delta r}-\sqrt{-3\left(1-\frac{|r|}{\Delta r}\right)^2+1}\right] & \text{for } 0.5\Delta r \leqslant |r| \leqslant 1.5\Delta r,\\ \frac{1}{3\Delta r}\left[1+\sqrt{-3\left(\frac{r}{\Delta r}\right)^2+1}\right] & \text{for } |r| \leqslant 0.5\Delta r,\\ 0 & \text{otherwise,}\end{cases}$$

Roma et al. 1999

- Regularization operator to smear force.
- Interpolation operator to interpolate velocity

Ben-Gurion University of the Negev

Implicit IB Method Based on Fully Pressure – Velocity Coupling

- Time marching solver.

- Steady state solver.

- Generalized eigenvalue solver.

Fully Coupled IB Time Marching Solver

$$\begin{bmatrix} H_u & 0 & -\nabla_p^x & R(\overrightarrow{F_x}) & 0 \\ 0 & H_v & -\nabla_p^y & 0 & R(\overrightarrow{F_y}) \\ \nabla_u^x & \nabla_v^y & 0 & 0 & 0 \\ I(u) & 0 & 0 & 0 & 0 \\ 0 & I(v) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^{(n+1)} \\ v^{(n+1)} \\ p^{(n+1)} \\ F_x^{(n+1)} \\ F_y^{(n+1)} \end{bmatrix} = \begin{bmatrix} R_u^{(n)} \\ R_v^{(n)} \\ R_p^{(n)} \\ U_b \\ V_b \end{bmatrix}$$

Immersed boundary formulation

+ Dirichlet point for pressure

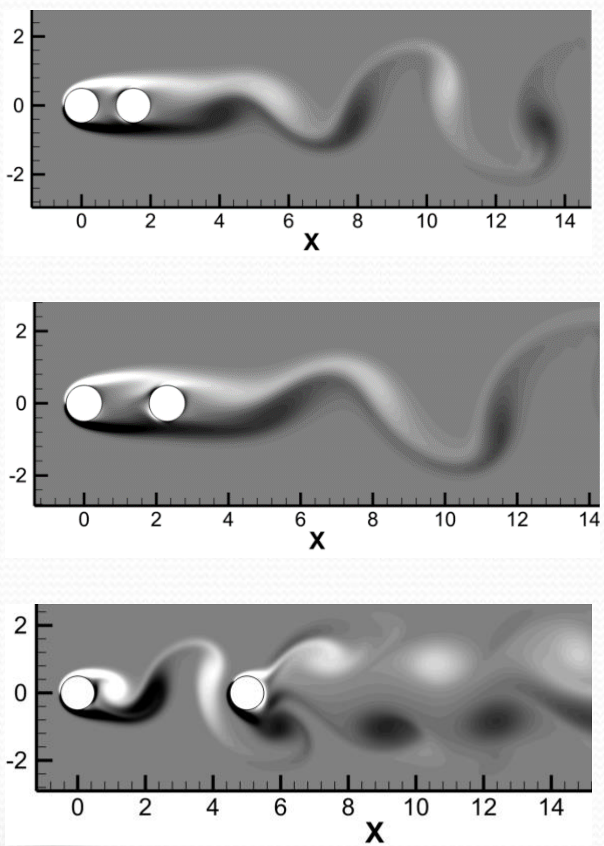
$\overrightarrow{F_x}$ and $\overrightarrow{F_y}$ play a role of Lagrange multipliers

- Extremely efficient for 2-D, axi-symmetric problems.
- Open source MUltifrontal Massively Parallel sparse direct Solver (*MUMPS, http://mumps.enseeiht.fr/) is used for LU decomposition.*

Qualitative verification

Wake characteristics, *Re*=200

Feldman&Gulberg, JCP, 2016

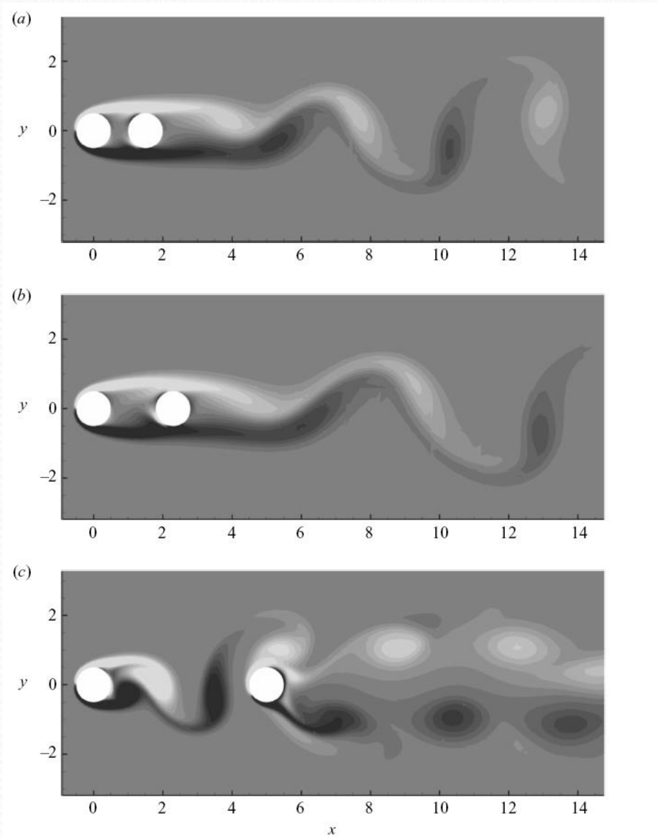


L/d=1.5

L/d=2.3

L/d=5

Carmo et al. JFM (2009)



Fully Coupled IB Steady State Solver

Newton iterations:

$\mathbf{F}(\mathbf{X})=0; \ \mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{X}}$

J – *Jacobian matrix*

$$\mathbf{J}\boldsymbol{\delta}\mathbf{X} = -\mathbf{F}\left(\mathbf{X}^{(n)}\right);$$

$$\boldsymbol{\delta}\mathbf{X} = -\mathbf{J}^{-1}\mathbf{F}\left(\mathbf{X}^{(n)}\right);$$

$$\mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} + \boldsymbol{\delta}\mathbf{X};$$

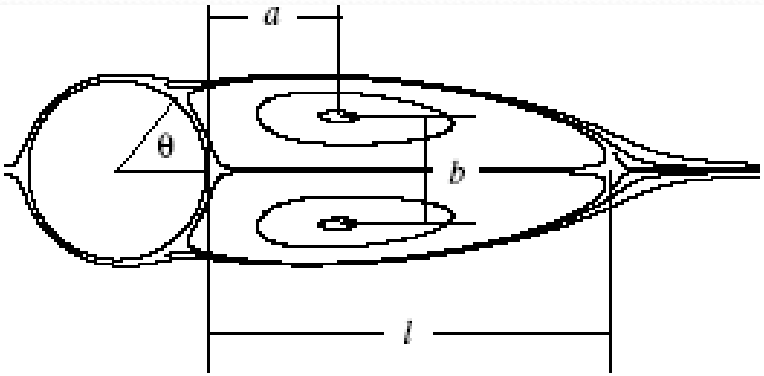
$$\begin{bmatrix} J_x & 0 & 0 & R\left(\overrightarrow{F_x}\right) & 0 \\ 0 & J_y & 0 & 0 & R\left(\overrightarrow{F_y}\right) \\ J_p & J_p & 0 & 0 & 0 \\ I(u) & 0 & 0 & 0 & 0 \\ 0 & I(v) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \\ \delta p \\ \delta F_x \\ \delta F_y \end{bmatrix} = - \begin{bmatrix} F_x + \sum_i \overrightarrow{F_x} \\ F_y + \sum_i \overrightarrow{F_y} \\ F_p \\ \sum_i \overrightarrow{u_x} \\ \sum_i \overrightarrow{u_y} \end{bmatrix}$$

- MUltifrontal Massively Parallel sparse direct Solver (*MUMPS, http://mumps.enseeiht.fr/) is used for LU decomposition.*

Ben-Gurion University of the Negev

Flow around cylinder -verification

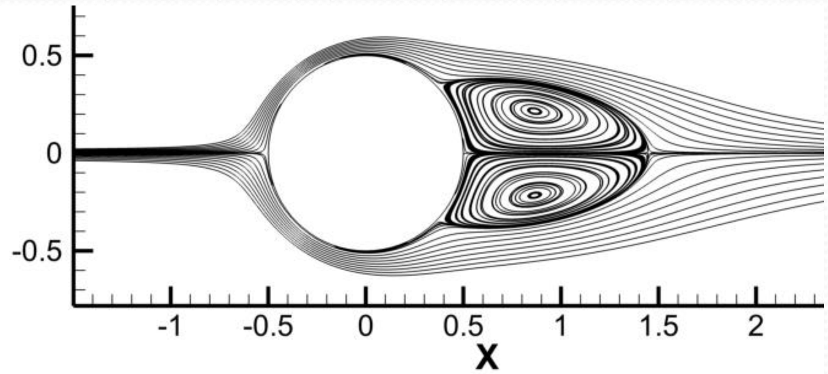
Definition of the characteristic dimensions for the wake structure



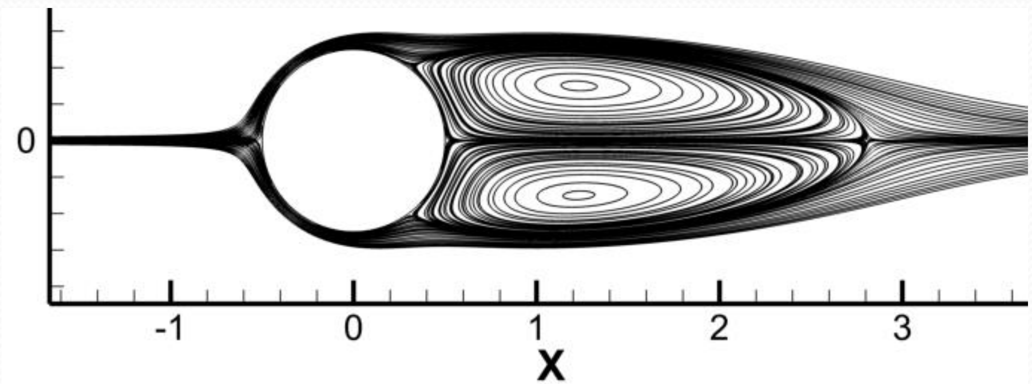
Comparison of the wake characteristics

Re=20		l/d	a/d	b/d	θ	C_D
	Feldman&Gulberg, JCP, 2016	0.95	0.37	0.43	42.9^0	2.09
	Taira and Colonius (2007)	0.94	0.37	0.43	43.3^0	2.06
	Linnick and Fasel (2005)	0.93	0.36	0.43	43.5^0	2.06
	Dennis and Chung (1970)	0.94	-	-	43.7	2.05
Re=40						
	Feldman&Gulberg, JCP, 2016	2.31	0.72	0.60	52.3^0	1.56
	Taira and Colonius (2007)	2.30	0.73	0.60	53.7^0	1.54
	Linnick and Fasel (2005)	2.28	0.72	0.60	53.6^0	1.54
	Dennis and Chung (1970)	2.35	-	-	53.8	1.52

Re=20



Re=40



Ben-Gurion University of the Negev

Eigenvalue Problem

Generalized eigenvalue problem with IB formalism
where **J** is the Jacobian matrix and det**B**=0

$$\lambda \boldsymbol{B}\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{p}\\ \tilde{\theta}\\ \tilde{F}_x\\ \tilde{F}_y\\ \tilde{Q}\end{bmatrix}=\boldsymbol{J}\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{p}\\ \tilde{\theta}\\ \tilde{F}_x\\ \tilde{F}_y\\ \tilde{Q}\end{bmatrix}$$

$$(\boldsymbol{J}-\sigma\boldsymbol{B})^{-1}\boldsymbol{B}\left(\tilde{u},\tilde{v},\tilde{p},\tilde{\theta},\tilde{F}_x,\tilde{F}_y,\tilde{Q}\right)=\mu\left(\tilde{u},\tilde{v},\tilde{p},\tilde{\theta},\tilde{F}_x,\tilde{F}_y,\tilde{Q}\right),\ \mu=\frac{1}{\lambda-\sigma}$$

Inverse of shifted Jacobian matrix

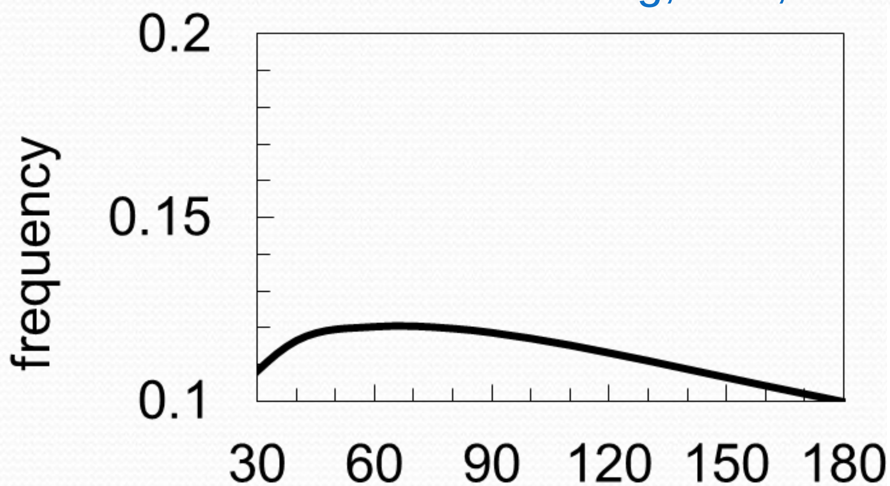
- Arpack library, *http://www.caam.rice.edu/software/ARPACK is used for performing Arnoldi iteration.*

Ben-Gurion University of the Negev

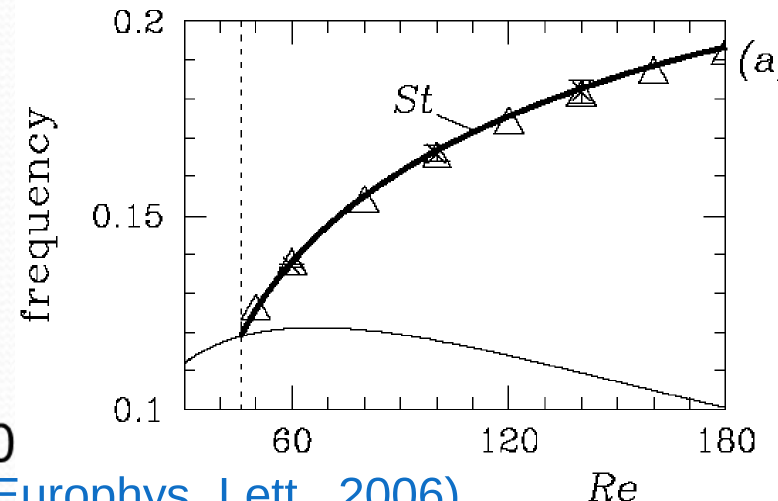
Flow around the Cylinder- Linear Stability Analysis

The most unstable mode of vorticity, *Re*=100

Feldman&Gulberg, JCP, 2016

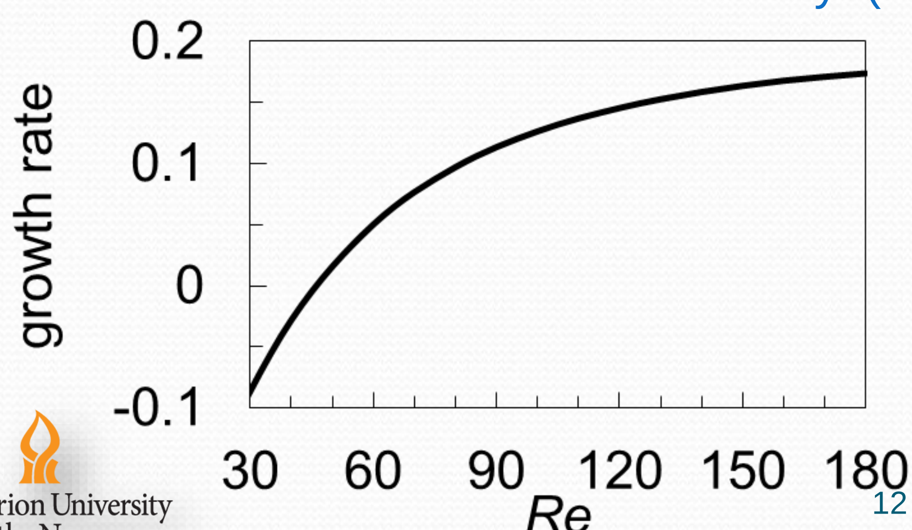


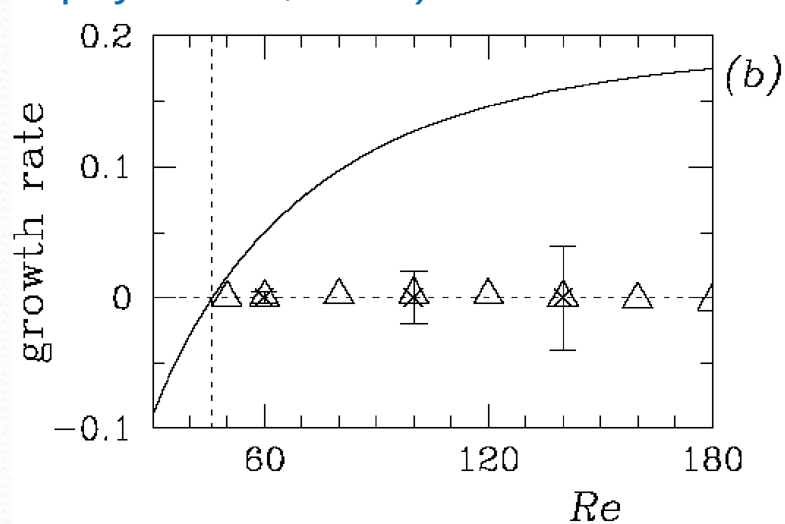
Barkley (Europhys. Ltt., 2006)



Re

Barkley (Europhys. Lett., 2006)

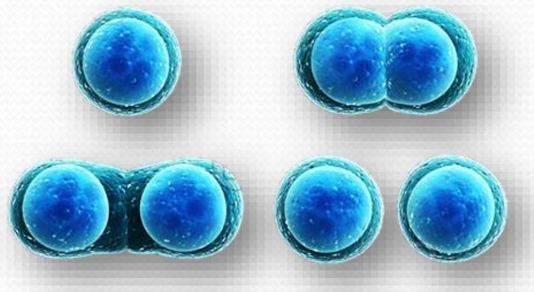




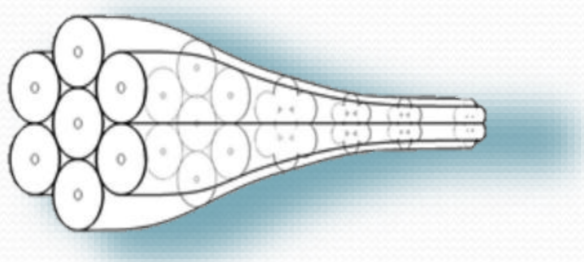
Ben-Gurion University of the Negev

Two Phase Flows

- Biomechanical engineering
  - Modeling cells motility in human body
  - Modeling tumor growth
- Optical wave guides
  - Tapered fiber bundle



[1] Cells division



[2] Seven fiber structure which allows to achieve good quality beam

[1] http://www.wisegeekhealth.com/what-is-the-connection-between-the-cell-cycle-and-cancer.htm
[2] http://soreq.gov.il/mmg/Pages/

Purpose of the Present Research

- Major Objective:

  To develop a **robust and efficient** numeric tool which represents the **underlying physical processes** inherent to two phase flow.

- Challenges:
  - To consistently provide preservation of initial volume of both phases.
  - To implicitly couple between the interface curvature and the velocity field.

Governing Equations

- Mass conservation

$$\nabla \cdot \tilde{\boldsymbol{u}} = 0$$

- Navier Stokes

$$\rho_r(I)\left(\frac{\partial \tilde{\boldsymbol{u}}}{\partial \tilde{t}} + \tilde{\boldsymbol{u}} \cdot \boldsymbol{\nabla} \tilde{\boldsymbol{u}}\right) =$$

$$= -\nabla \tilde{p} + \frac{\rho_r}{Re\mu_r} \nabla \cdot \left(\mu_r(I)(\nabla \tilde{\boldsymbol{u}} + (\nabla \tilde{\boldsymbol{u}})^T)\right) + \frac{\rho_r}{We} \int_\Gamma \frac{\partial^2 \tilde{\boldsymbol{X}}_f}{\partial \tilde{S}^2} - \rho_r(I)$$

$$Re = \frac{\rho_H U_g D}{\mu_H} \qquad We = \frac{\rho_H U_g^2 D}{\sigma}$$

Ben-Gurion University of the Negev

Fully Coupled IB Time Marching Solver

$$\frac{\rho_r \mu_r(I)}{Re\,\mu_r} \nabla^2 \widetilde{\boldsymbol{u}}^{\boldsymbol{n+1}} - \frac{3\rho_r(I)\widetilde{\boldsymbol{u}}^{n+1}}{2\Delta\tilde{t}} - \nabla p + \frac{\rho_r}{We}\int_\Gamma \frac{\partial^2 \widetilde{\boldsymbol{X}}_{\boldsymbol{f}}}{\partial \tilde{S}^2} = \rho_r(I)\left(\frac{-4\widetilde{\boldsymbol{u}}^n + \widetilde{\boldsymbol{u}}^{n-1}}{2\Delta\tilde{t}} + \widetilde{\boldsymbol{u}}^{\boldsymbol{n}} \cdot \boldsymbol{\nabla}\widetilde{\boldsymbol{u}}^{\boldsymbol{n}}\right) + \begin{bmatrix}0\\1\end{bmatrix}\rho_r(I)$$

- The fully pressure velocity coupled direct (FPCD) solver *(Feldman & Gelfgat 2009).*

$$\begin{bmatrix} H_u & 0 & -\nabla_p^x & \frac{\rho_r}{We} R\left(\frac{\partial^2 \tilde{X}_f}{\partial \tilde{S}^2}\right) & 0 \\ 0 & H_v & -\nabla_p^y & 0 & \frac{\rho_r}{We} R\left(\frac{\partial^2 \tilde{Y}_f}{\partial \tilde{S}^2}\right) \\ \nabla_u^x & \nabla_v^y & 0 & 0 & 0 \\ Int(\tilde{u}) & 0 & 0 & -1 & 0 \\ 0 & Int(\tilde{v}) & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \tilde{u}^{(n+1)} \\ \tilde{v}^{(n+1)} \\ \tilde{p} \\ \frac{\partial^2 \tilde{X}_f}{\partial \tilde{S}^2} \\ \frac{\partial^2 \tilde{Y}_f}{\partial \tilde{S}^2} \end{bmatrix} = \begin{bmatrix} R_u^{(n)} \\ R_v^{(n)} \\ R_p \\ R_{fx} \\ R_{fy} \end{bmatrix}$$

Immersed boundary formulation

Ben-Gurion University of the Negev

Curvature as Lagrange Multiplier

$$\begin{bmatrix} H_u & 0 & -\nabla_p^x & \frac{\rho_r}{We} R\left(\frac{\partial^2 \tilde{X}_f}{\partial \tilde{S}^2}\right) & 0 \\ 0 & H_v & -\nabla_p^y & 0 & \frac{\rho_r}{We} R\left(\frac{\partial^2 \tilde{Y}_f}{\partial \tilde{S}^2}\right) \\ \nabla_u^x & \nabla_v^y & 0 & 0 & 0 \\ Int(\tilde{u}) & 0 & 0 & -1 & 0 \\ 0 & Int(\tilde{v}) & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \tilde{u}^{(n+1)} \\ \tilde{v}^{(n+1)} \\ \tilde{p} \\ \frac{\partial^2 \tilde{X}_f}{\partial \tilde{S}^2} \\ \frac{\partial^2 \tilde{Y}_f}{\partial \tilde{S}^2} \end{bmatrix} = \begin{bmatrix} R_u^{(n)} \\ R_v^{(n)} \\ R_p \\ R_{fx} \\ R_{fy} \end{bmatrix}$$

Immersed boundary formulation

$$\left(\frac{\partial^2 \widetilde{\boldsymbol{X}}_f}{\partial S^2}\right) = \frac{\left(\widetilde{\boldsymbol{X}}_{l+1}^{n+1} - 2\widetilde{\boldsymbol{X}}_l^{n+1} + \widetilde{\boldsymbol{X}}_{l-1}^{n+1}\right)}{(\Delta S)^2}$$

$$\frac{\Delta T\left(Int(u)_{l+1}^{n+1} - 2Int(u)_l^{n+1} + Int(u)_{l-1}^{n+1}\right)}{2(\Delta S)^2} - \left(\frac{\partial^2 X}{\partial S^2}\right) =$$

$$\frac{-X_{l+1}^n + 2X_l^n - X_{l-1}^n}{(\Delta S)^2} + \frac{\Delta T\left(-Int(u)_{l+1}^n + 2Int(u)_l^n - Int(u)_{l-1}^n\right)}{2(\Delta S)^2}$$

Runge Kutta 2

$$X_l^{n+1} = X_l^n + (K_1 + K_2)/2$$

$$K_1 = \Delta T \cdot Int(u)_{l(X_l, Y_l)}^n$$

$$K_2 = \Delta T \cdot Int(u)_{l(X_l^n + K_1, Y_l^n + K_1)}^{n+1}$$

Ben-Gurion University of the Negev

Indicator Function

- Far from the interface:

$$\boldsymbol{I}(\boldsymbol{x}) = \begin{cases} 1, & if\ \boldsymbol{x}\ is\ in\ fluid\ A \\ 0, & if\ \boldsymbol{x}\ is\ in\ fluid\ \mathrm{B} \end{cases}$$

- Near the interface

$$(\nabla I)_{i,j} = \sum_l \boldsymbol{n}_l \delta^2(r_x, r_y)\, \Delta s_l\,,$$

$$\nabla^2 \boldsymbol{I} = \nabla \cdot \sum_l \boldsymbol{n}_l \delta^2(r_x, r_y)\, \Delta s_l$$ Solved with MUMPS

$$\big(\rho_r(\boldsymbol{I}), \mu_r(\boldsymbol{I})\big) = (\rho_r, \mu_r - 1) \cdot \boldsymbol{I} + 1$$

Indicator Function, Normal

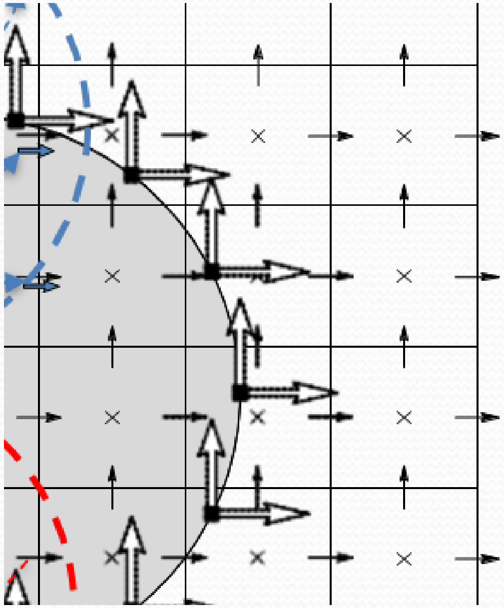
$$(\nabla I)_{i,j} = \sum_l \boldsymbol{n}_l \delta^2(r_x, r_y)\,\Delta s_l$$

$$\begin{bmatrix} x \\ y \end{bmatrix}(t) = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} t^2 + \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} t + \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}$$

$$\boldsymbol{X}_{(t=0)} = \boldsymbol{X}_{l-1}\ ,\boldsymbol{X}_{(t=\Delta s_l)} = \boldsymbol{X}_l\ ,\boldsymbol{X}_{(t=\Delta s_{l+1}+\Delta s_l)} = \boldsymbol{X}_{l+1}$$

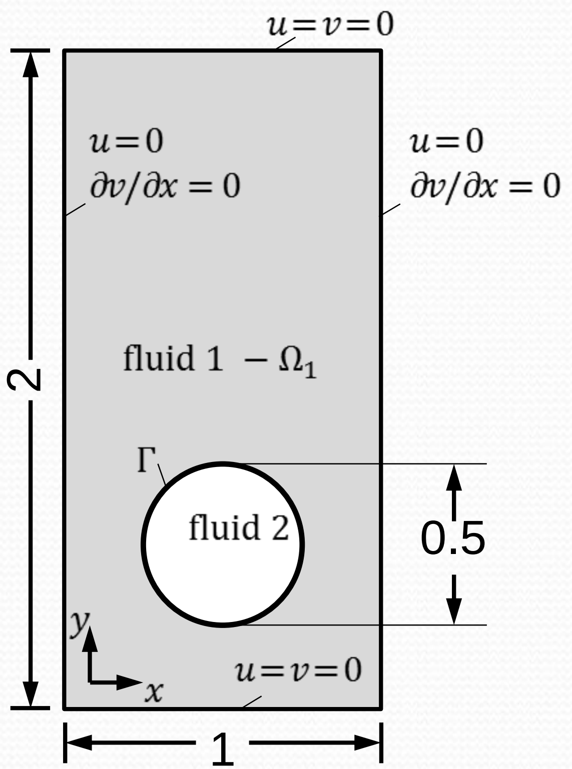
$$\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \Delta s_l^2 & \Delta s_l & 1 \\ (\Delta s_l + \Delta s_{l+1})^2 & \Delta s_l + \Delta s_{l+1} & 1 \end{bmatrix}^{-1} \begin{bmatrix} X_{l-1} \\ X_l \\ X_{l+1} \end{bmatrix}$$

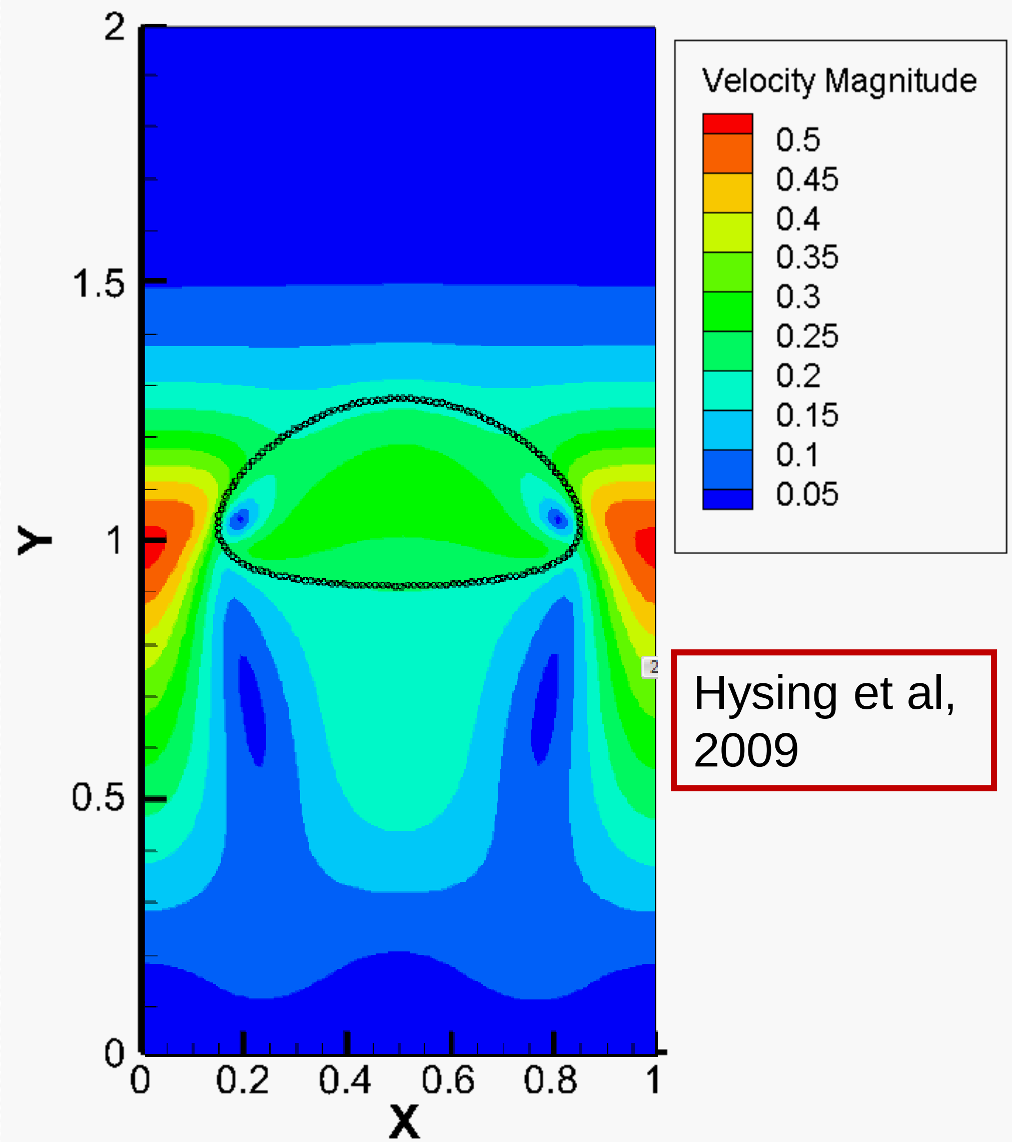
$$\boldsymbol{n}_l = \left(\frac{\frac{dy(\Delta s_l)}{dt}}{\sqrt{\left(\frac{dx(\Delta s_l)}{dt}\right)^2 + \left(\frac{dy(\Delta s_l)}{dt}\right)^2}}, \frac{-\frac{dx(\Delta s_l)}{dt}}{\sqrt{\left(\frac{dx(\Delta s_l)}{dt}\right)^2 + \left(\frac{dy(\Delta s_l)}{dt}\right)^2}} \right)$$



Ben-Gurion University of the Negev

Rising Bubble



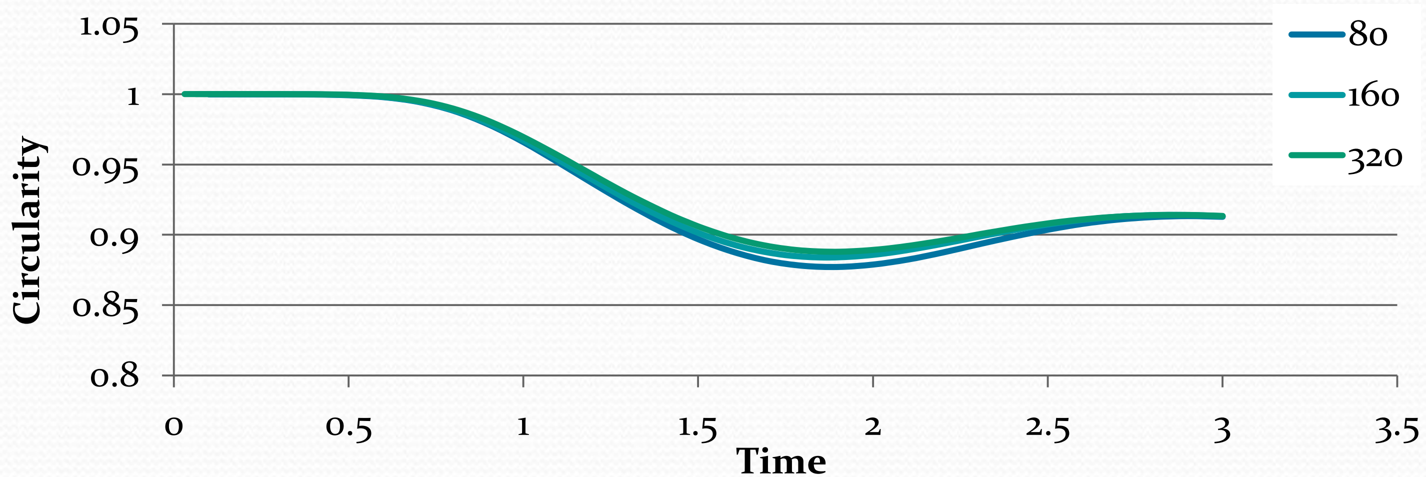


$$\rho_r = \frac{\rho_1}{\rho_2} = 10, \qquad \mu_r = 10$$

$$Re = 40, \qquad We = 10$$

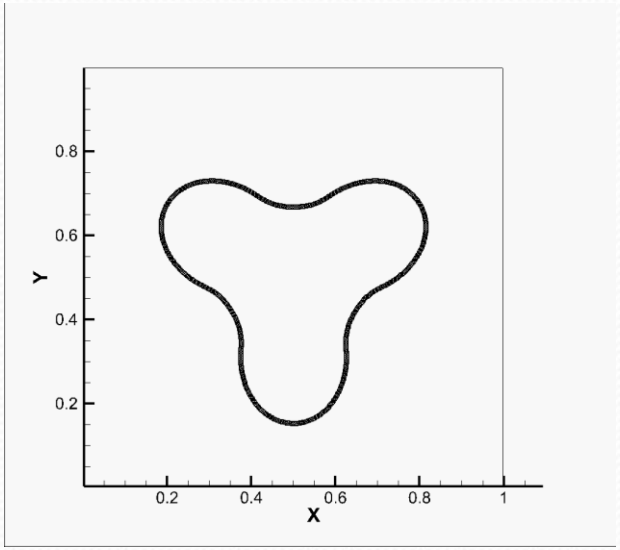
$$1/h: 80,\ 160,\ 320 \quad , h - grid\ step$$

Rising Bubble Mesh Convergence

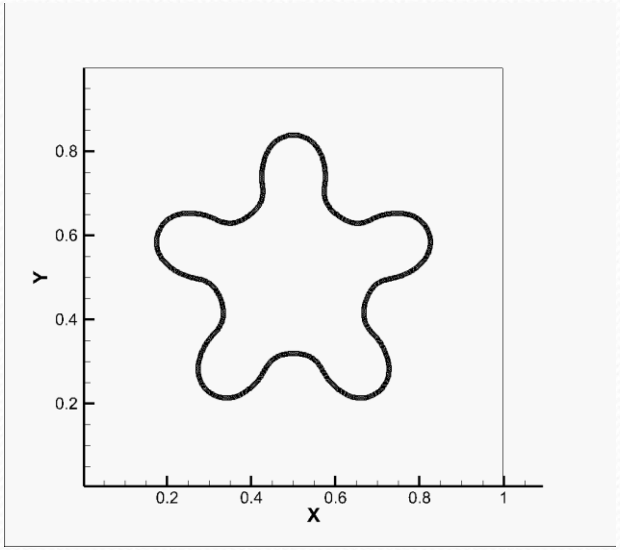


	320	80	160	320
8	0.888	0.8771	0.8838	0.888
	1.91	1.9	1.88	1.91
	1.08	1.08	1.08	1.08
Area Max Discrepancy		2.04e-3	5.09e-4	1e-5

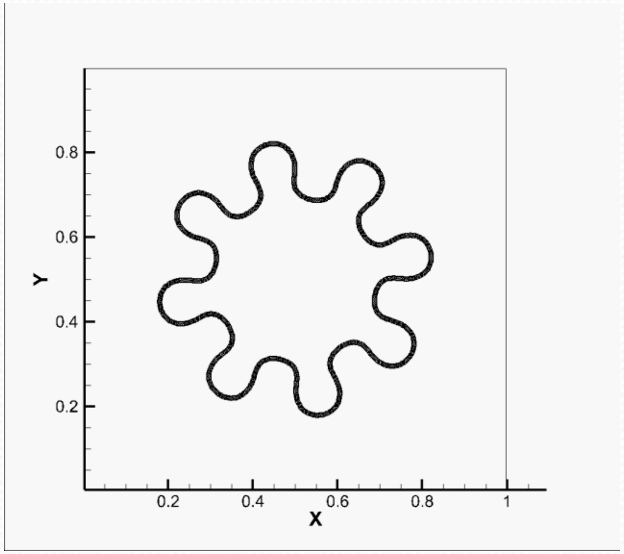
Droplet Relaxation to Disk



$n = 3$



$n = 5$



$n = 8$

Max area discrepancy

$\rho_r = 1, \mu_r = 1, We = 0.02, \mathrm{Re} = 11$

	8	**3**	**5**	**8**
-3	1.61e-3	2.36e-3	2.22e-3	1.61e-3
-3	1.75e-3	1.61e-3	1.70e-3	1.75e-3

Ben-Gurion University of the Negev

Toward Fully 3D Flows

- Implicit coupling between Lagrangian forces and heat fluxes and the velocity and temperature fields.
- Diminishing of flow leakages through the boundaries of immersed bodies.
- Modularity and compatibility with any generic solver of the N-S equations based on the segregated pressure-velocity coupling (e.g. SIMPLE algorithm , fractional step, projection method) .

Implicit Direct Forcing: General Concepts

Discrete 2nd order accuracy N-S equations with imbedded IB functionality (the SIMPLE method)

$$\frac{3\theta^{n+1}}{2\Delta t} - \frac{1}{\sqrt{PrRa}}\boldsymbol{L}(\theta^{n+1}) - \boldsymbol{R}(Q^k(\boldsymbol{X}^k)) = \frac{4\theta^n - \theta^{n-1}}{2\Delta t} - \boldsymbol{N}(\theta^n, \boldsymbol{u}^n) + \theta^n \vec{e}_z$$

$$\frac{3\boldsymbol{u}^*}{2\Delta t} - \sqrt{\frac{Pr}{Ra}}\boldsymbol{L}(\boldsymbol{u}^*) - \boldsymbol{R}(\boldsymbol{F}^k(\boldsymbol{X}^k)) = \frac{4\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{2\Delta t} - \boldsymbol{N}(\boldsymbol{u}^n, \boldsymbol{u}^n) - \nabla p^n$$

$$\triangle(\delta p) = \frac{3}{2\Delta t}\nabla \cdot \boldsymbol{u}^*$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{2\Delta t}{3}\nabla(\delta p), \qquad p^{n+1} = p^n + \delta p,$$

Ben-Gurion University of the Negev

Implicit Direct Forcing: General Concepts, Cont.

Square matrix, *n×n*

Rectangular, extremely sparse matrix, *n×m, m<<n*

$$\begin{bmatrix} \boldsymbol{H}_{\theta,\boldsymbol{u}} & \boldsymbol{R}(Q^k(\boldsymbol{X}^k), \boldsymbol{F}^k(\boldsymbol{X}^k)) \\ \boldsymbol{I}(\theta(\boldsymbol{x}_i), \boldsymbol{u}^*(\boldsymbol{x}_i)) & 0 \end{bmatrix} \begin{bmatrix} \theta^{n+1}, \boldsymbol{u}^* \\ Q, \boldsymbol{F} \end{bmatrix} = \begin{bmatrix} RHS^{n-1,n}_{\theta,u} \\ \theta_b, \boldsymbol{U}_b \end{bmatrix}$$

Rectangular, extremely sparse matrix, *m×n, m<<n*

$$[\boldsymbol{H}_{\theta,\boldsymbol{u}}][\theta^{n+1}, \boldsymbol{u}^*] = RHS^{\,n-1,n}_{\,\theta,\boldsymbol{u}} \longrightarrow [\boldsymbol{H}_{\theta,\boldsymbol{u}}]^{-1}\,[any\ vector]$$

Efficient solver, originally developed for direct inversion of Helmholtz operator is utilized, *Vitoshkin H., Gelfgat A. Yu., Commun. Comput. Phys., 2013*

Schur Complement Approach

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{R} \\ \boldsymbol{I} & 0 \end{bmatrix} \begin{bmatrix} \theta, \boldsymbol{u} \\ Q, \boldsymbol{F} \end{bmatrix} = \begin{bmatrix} RHS_{\theta,\boldsymbol{u}}^{n-1,n} \\ \theta_b, U_b \end{bmatrix}$$

$$[Q, \boldsymbol{F}] = (\boldsymbol{I}\boldsymbol{H}^{-1}\boldsymbol{R})^{-1}(\boldsymbol{I}\boldsymbol{H}^{-1}RHS_{\theta,\boldsymbol{u}}^{n-1,n} - \theta_b, U_b)$$

$$[\theta, \boldsymbol{u}] = \boldsymbol{H}^{-1}(RHS_{\theta,\boldsymbol{u}}^{n-1,n} - \boldsymbol{R}[Q, \boldsymbol{F}])$$

$\boldsymbol{H}^{-1}RHS_{\theta,\boldsymbol{u}}^{n-1,n}$ - Original existing solver

$\boldsymbol{H}^{-1}\boldsymbol{R}$ - Action of original solver on each column of ***R*** *(can be precomputed)*

$\boldsymbol{I}\boldsymbol{H}^{-1}\boldsymbol{R}$, $\boldsymbol{I}\boldsymbol{H}^{-1}RHS_{\theta,\boldsymbol{u}}^{n-1,n}$ - Sparse matrix-matrix and matrix-vector multiplications (performed by existing MKL routines)

$(\boldsymbol{I}\boldsymbol{H}^{-1}\boldsymbol{R})^{-1}$ - Small, *m*×*m* indefinite matrix, is factorized at the beginning by a direct method *(MUMPS)* or solved by GMRES or AMG iterative methods

$\boldsymbol{H}^{-1}(RHS_{\theta,\boldsymbol{u}}^{n-1,n} - \boldsymbol{R}[Q, \boldsymbol{F}])$ - Computed by original existing solver

Ben-Gurion University of the Negev

Verification Study 1 – Sphere in Cube

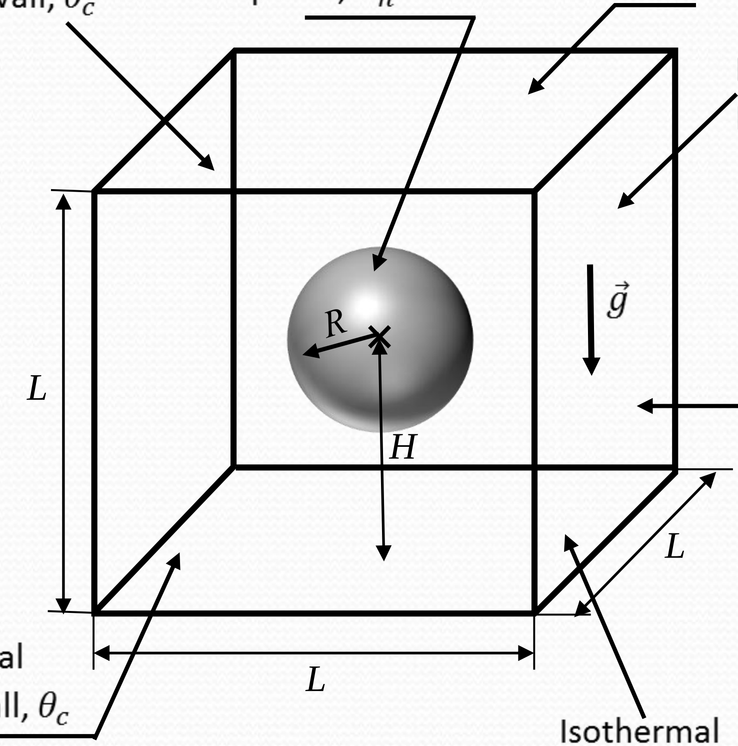
$$Ra = \frac{g\beta(\theta_h - \theta_c)L^3}{\nu\alpha}$$

β-coefficient of thermal expansion

ν-kinematic viscosity

α-thermal conductivity

$R = 0.2L\,;\delta = 0.5L \quad - H$

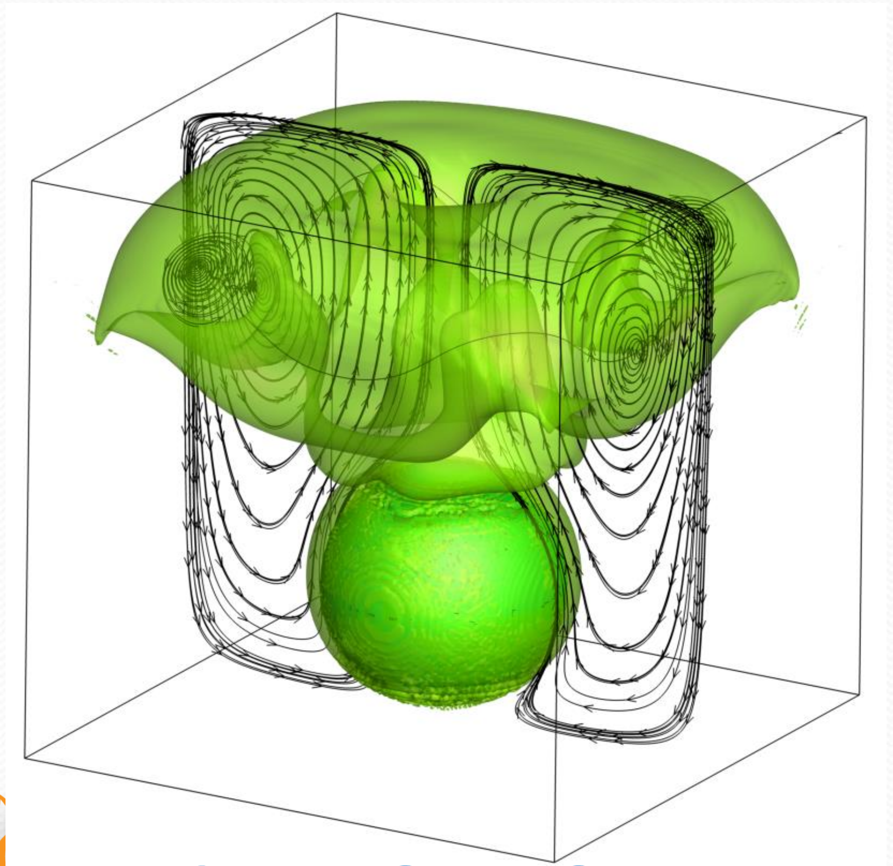


- 200^3 structured grid

The Flow Visualization

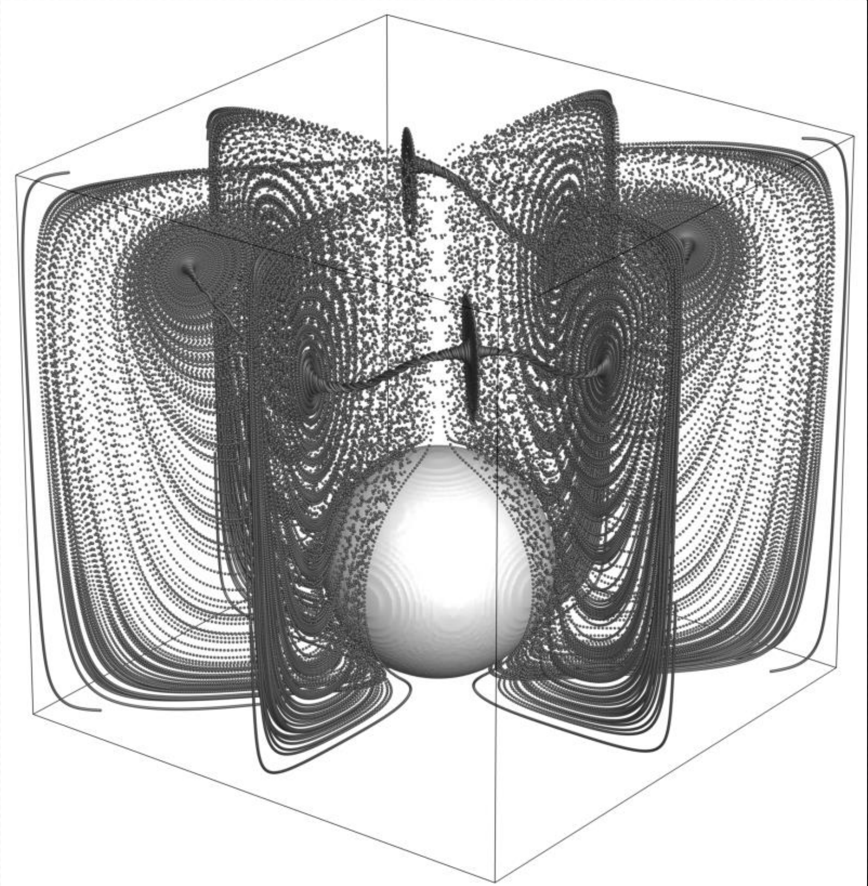
- Steady flow around the hot sphere, $Ra = 10^6, \delta = 0.25$

Vortical structures based on λ_2 criterion



Iso-surface of $\lambda_2 = -0.1$

Particle traces



Nu number Comparison

Steady flow results

δ	Ra=10^5			Ra=10^6		
	Ref[1]	Ref[2]	**Present**	Ref[1]	Ref[2]	**Present**
-0.25	13.665	13.774	**13.489**	20.890	21.993	**20.611**
-0.2	12.931	13.058	**12.768**	20.631	21.862	**20.517**
-0.1	12.729	13.105	**12.819**	20.772	22.164	**21.216**
0	12.658	13.415	**13.160**	20.701	23.344	**21.589**
0.1	12.351	13.446	**13.230**	20.367	22.525	**21.674**
0.2	12.254	13.635	**13.462**	19.664	22.208	**21.487**
0.25	12.944	14.426	**14.277**	19.721	22.393	**21.757**

[1] H.Yoon, D. Yu. M. Ha, Y. Park, Three-dimensional natural convection in an enclosure with a sphere at different locations. Int. J. Heat Mass Transfer 53(2010) 3143-3155.

[2] Y. Gulberg, Y. Feldman, On laminar natural convection inside multi-layered spherical shells. Int. J. Heat Mass Transfer 91(2015) 908-921.

Conclusions and Wish List

- Over the years of research activity MUMPS has been found as a very convenient tool for analysis of a broad spectrum of CFD problems.
- MUMPS usage allows for a high flexibility in choosing and manipulating well established numerical techniques to address applications requiring transient, steady and linear stability analysis.
- It will be helpful if the MUMPS developers provide users with an automatic script for installation of the whole package with all pre-requirements, or with the prebuilt libraries for the standard Linux platforms.
- It will be very convenient to have a capability to save the factorized LU matrix on hard disk for its further utilizing.