



Recent Progress on Meshless Methods for Flow Simulations-SPH, MPS, DPD

30 November - 01 December 2016

National Taiwan University

Mixed Lagrangian-Eulerian Particle Method with Mass Preserving for Solving Fluid Flow Problem

黃宇倫 NG Yee-Luon

Scientific Computing and Cardiovascular Simulation Laboratory
Department of Engineering Science and Ocean Engineering
National Taiwan University

Acknowledgement

- ▶ Supervisor

Prof. Tony Wen-Hann Sheu (National Taiwan University, Taiwan)

- ▶ Team members

Prof. Yao-Hsin Hwang (National Kaohsiung Marine University, Taiwan)

A. Prof. Khai Ching Ng (Universiti Tenaga National, Malaysia)

Kuan-Shuo Liu (National Taiwan University, Taiwan)



Outline

- ▶ Introduction
- ▶ Mixed Lagrangian-Eulerian Moving Particle (MLEMP) Method
- ▶ Improvement on MLEMP Method
 - 1) Radial Basis Function (RBF) Interpolation
 - RBF-MQ in polynomial expression
 - Optimum shape parameter ε
 - 2) Mass-Preserving Interpolation
- ▶ Results
 - 1) Validation Studies
 - 2) Verification Studies
- ▶ Conclusion



Introduction

▶ Moving Particle Methods

- Describe fluid motion in Lagrangian Sense.
- Simulation domain is filled with cloud of Lagrangian particles instead of the discretized stationary grid points.

- Advantages:
 - 1) Does not rely on connectivity information between nodal points – Complex geometry can be easily dealt with.
 - 2) The non-linear convection term in governing equation of fluids can be easily modeled by particle movement.



Introduction

▶ Moving Particle Semi-Implicit (MPS) analysis framework

- Solving Incompressible Navier-Stokes equations in a cloud of moving particles.
- Particle interaction models representing differential operators in the governing equation of fluid flow.
- Momentum equation is decomposed into viscous and pressure parts.

$$\text{Viscous part} \left\{ \begin{array}{l} \mathbf{u}^* = \mathbf{u}^n + \Delta t \frac{\mu}{\rho} \nabla^2 \mathbf{u}^n \\ \mathbf{x}^* = \mathbf{x}^n + \Delta t \mathbf{u}^* \end{array} \right. \quad (1)$$

$$\text{Pressure part} \left\{ \begin{array}{l} \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1} \\ \mathbf{x}^{n+1} = \mathbf{x}^* - \frac{(\Delta t)^2}{\rho} \nabla p^{n+1} \end{array} \right. \quad (2)$$

- Viscous term is computed through explicit way, and pressure term is solved implicitly.
-



Introduction

- ▶ **Moving Particle Semi-Implicit (MPS) analysis framework**
- Fluid density: modeled by Particle Number Density (PND)
- Incompressible flow: PND n_i should be constant.
- Source term of pressure Poisson equation (PPE): deviation of the temporal PND n^*_i from a constant n^0

$$\langle \nabla^2 P^{n+1} \rangle_i = - \frac{\rho}{(\Delta t)^2} \frac{n^*_i - n^0}{n^0} \quad (3)$$

- Equation (3) is solved implicitly.
 - The particle velocity and position are corrected using equation (2) after the pressure field is obtained.
-



Introduction

- ▶ **Moving Particle Semi-Implicit (MPS) analysis framework**
 - Major drawback of MPS method: pressure instability.
 - PND source term in PPE results in high-frequency numerical oscillations, and leads to fluctuated pressure solution.
 - Discretization on particle cloud using particle interaction model is not as accurate as the conventional grid based method.



Mixed Lagrangian-Eulerian Moving Particle (MLEMP) Method

- ▶ Hybrid method proposed by Huang in Moving Particles with Pressure Mesh (MPPM model) method [1,2].
- ▶ Improves the pressure instability problem in MPS method by solving pressure on Eulerian grid using the method similar to conventional grid based method.
- ▶ The viscous term is solved on Lagrangian particles as in original MPS method using equation (1) and (2). The guessed velocity \mathbf{u}^* is then interpolated to Eulerian grids to solve for the pressure solution.
- ▶ Particle velocity and position are corrected by interpolating the pressure gradient on Eulerian grids back to Lagrangian particles.

[1] Y.H. Hwang, A Moving Particle Method with Embedded Pressure Mesh (MPPM) for Incompressible Flow Calculations, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 60, pp. 370-398, 2011.

[2] Y.H. Hwang, Assessment of Diffusion Operators in a Novel Moving Particle Method, *Numerical Heat Transfer, Part B*, vol. 61, pp. 329-368, 2012.

Mixed Lagrangian-Eulerian Moving Particle (MLEMP) Method

▶ **Advantages:**

- Stable pressure solution: clustered particles no longer result in large magnitude of PPE source term.
- Addition/removal of particles can be easily done without disturbing the flow: good for inflow/outflow boundary condition implementation.

▶ **Limitation:**

- Hybrid grid system: Data need to be transferred between two different grid systems in every time step.
- Accuracy of interpolation scheme plays an important role on the quality of the pressure solution.



Improvement on Present MLEMP Method

- ▶ The accuracy of the MLEMP methods rely on the accuracy of the interpolation scheme between two grids system.
- ▶ To improve the solution quality of the MLEMP method, new interpolation schemes are proposed:
 - 1) Radial Basis Function (RBF) Interpolation
 - 2) Mass-Preserving Interpolation



Radial Basis Function Interpolation

- ▶ An unknown scalar function $f(\mathbf{x})$ with the discrete value is known at N scattered nodes $\mathbf{x}_i, i = 1, 2 \dots N$. The function $f(\mathbf{x})$ can be approximated by RBF function $s(\mathbf{x})$:

$$f(\mathbf{x}) \approx s(\mathbf{x}) = \sum_{i=1}^N \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (4)$$

α_i : Interpolation coefficients

$\phi(\|\mathbf{x} - \mathbf{x}_i\|)$: RBF kernel chosen



Radial Basis Function Interpolation

- ▶ Finding interpolation coefficients involves solving the linear system of N equations $\mathbf{A}\boldsymbol{\alpha} = \mathbf{f}$, where $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_N]^T$, $\mathbf{f} = [f(\mathbf{x}_1) \ f(\mathbf{x}_2) \ \cdots \ f(\mathbf{x}_N)]^T$, and:

$$\mathbf{A} = \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix} \quad (5)$$



Radial Basis Function Interpolation

- ▶ RBF kernel chosen: Multiquadric (MQ)
- ▶ Known to give higher solution accuracy compared to the other kernels.

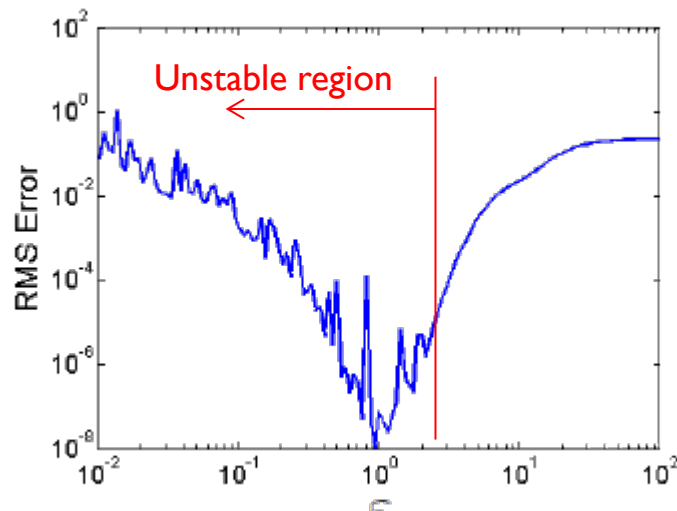
$$\phi(\|\mathbf{x} - \mathbf{x}_i\|) = \sqrt{1 + \varepsilon \|\mathbf{x} - \mathbf{x}_i\|^2}$$

- ▶ The RBF linear system $\mathbf{A}\boldsymbol{\alpha} = \mathbf{f}$ becomes ill-conditioned when shape parameter $\varepsilon \rightarrow 0$.
- ▶ The interpolation coefficients $\boldsymbol{\alpha}$ rapidly as ε decreases.



Radial Basis Function Interpolation

- ▶ Interpolation coefficient α diverged as $\varepsilon \rightarrow 0$.
- ▶ Often however, at any point, the interpolating function $s(\mathbf{x})$ itself (in equation (4)), is well behaved, and converges to finite number even when $\varepsilon \rightarrow 0$.
- ▶ In other words, $\varepsilon = 0$ is a removable singular point.



[3] M. Mongillo, Choosing Basis Functions and Shape Parameters for Radial Basis Function Methods, *SIAM Undergraduate Research Online* 4, 190-209, 2011

RBF-MQ in Polynomial Expression

- ▶ Power series expansion

$$\sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{2 \cdot 4}x^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6}x^3 - + \dots \quad |x| < 1$$

- ▶ Expansion of multiquadric kernel

$$\begin{aligned} \phi(\|r\|) &= \sqrt{1 + \varepsilon\|r\|^2} \\ &= 1 + \frac{1}{2}(\varepsilon r^2) - \frac{1}{2 \cdot 4}(\varepsilon r^2)^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6}(\varepsilon r^2)^3 - + \dots \end{aligned}$$



RBF-MQ in Polynomial Expression

- ▶ From the expansion, the RBF matrix \mathbf{A} in (5) can be decomposed into:

$$\mathbf{A}(\varepsilon) = \mathbf{A}_0 + \varepsilon\mathbf{A}_1 + \varepsilon^2\mathbf{A}_2 + \varepsilon^3\mathbf{A}_3 + \dots$$

$$\mathbf{A}(\varepsilon) = \mathbf{A}_0 + \sum_{k=1}^{\infty} \varepsilon^k \mathbf{A}_k$$

- ▶ Since ε is small, finding inverse of $\mathbf{A}(\varepsilon)$ becomes a problem of finding inverse of small perturbation of a singular matrix. The inverse of matrix \mathbf{A} can be written in Laurent series expansion:

$$\mathbf{A}^{-1}(\varepsilon) = \sum_{k=-s}^{\infty} \varepsilon^k \mathbf{H}_k$$

s is the order of singularity



RBF-MQ in Polynomial Expression

$$\mathbf{A}^{-1}(\varepsilon) = \varepsilon^{-s} \mathbf{H}_{-s} + \varepsilon^{-s+1} \mathbf{H}_{-s+1} + \cdots + \mathbf{H}_0 + \varepsilon \mathbf{H}_1 + \varepsilon^2 \mathbf{H}_2 + \varepsilon^3 \mathbf{H}_3 + \cdots$$

$$\boldsymbol{\alpha}(\varepsilon) = \mathbf{A}^{-1}(\varepsilon) \mathbf{f}$$

$$\boldsymbol{\alpha}(\varepsilon) = \underbrace{\varepsilon^{-s} \mathbf{H}_{-s} \mathbf{f}}_{\alpha_{-s}} + \underbrace{\varepsilon^{-s+1} \mathbf{H}_{-s+1} \mathbf{f}}_{\alpha_{-s+1}} + \cdots + \underbrace{\mathbf{H}_0 \mathbf{f}}_{\alpha_0} + \underbrace{\varepsilon \mathbf{H}_1 \mathbf{f}}_{\alpha_1} + \underbrace{\varepsilon^2 \mathbf{H}_2 \mathbf{f}}_{\alpha_2} + \underbrace{\varepsilon^3 \mathbf{H}_3 \mathbf{f}}_{\alpha_3}$$

- ▶ The interpolating function at any point (x_i, y_i) is:

$$s(x_i, y_i, \varepsilon) = \alpha_1 \phi(r_{i1}, \varepsilon) + \alpha_2 \phi(r_{i2}, \varepsilon) + \cdots + \alpha_n \phi(r_{in}, \varepsilon)$$

$$s(x_i, y_i, \varepsilon) = \boldsymbol{\alpha}(\varepsilon)^T \begin{bmatrix} \phi(r_{i1}, \varepsilon) \\ \phi(r_{i2}, \varepsilon) \\ \vdots \\ \phi(r_{in}, \varepsilon) \end{bmatrix} = \boldsymbol{\alpha}(\varepsilon)^T \begin{bmatrix} 1 + \frac{1}{2} \varepsilon r_{i1}^2 - \frac{1}{2 \cdot 4} \varepsilon^2 r_{i1}^4 + \cdots \\ 1 + \frac{1}{2} \varepsilon r_{i2}^2 - \frac{1}{2 \cdot 4} \varepsilon^2 r_{i2}^4 + \cdots \\ \vdots \\ 1 + \frac{1}{2} \varepsilon r_{in}^2 - \frac{1}{2 \cdot 4} \varepsilon^2 r_{in}^4 + \cdots \end{bmatrix}$$



RBF-MQ in Polynomial Expression

$$s(x_i, y_i, \varepsilon) = \alpha(\varepsilon)^T \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \alpha(\varepsilon)^T \begin{bmatrix} \frac{1}{2} \varepsilon r_{i1}^2 \\ \frac{1}{2} \varepsilon r_{i2}^2 \\ \vdots \\ \frac{1}{2} \varepsilon r_{in}^2 \end{bmatrix} + \alpha(\varepsilon)^T \begin{bmatrix} -\frac{1}{2 \cdot 4} \varepsilon^2 r_{i1}^4 \\ -\frac{1}{2 \cdot 4} \varepsilon^2 r_{i2}^4 \\ \vdots \\ -\frac{1}{2 \cdot 4} \varepsilon^2 r_{in}^4 \end{bmatrix} + \dots$$

Take out ε

$$s(x_i, y_i, \varepsilon) = \alpha(\varepsilon)^T \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\phi_0} + \alpha(\varepsilon)^T \varepsilon \underbrace{\begin{bmatrix} \frac{1}{2} r_{i1}^2 \\ \frac{1}{2} r_{i2}^2 \\ \vdots \\ \frac{1}{2} r_{in}^2 \end{bmatrix}}_{\phi_1} + \alpha(\varepsilon)^T \varepsilon^2 \underbrace{\begin{bmatrix} -\frac{1}{2 \cdot 4} r_{i1}^4 \\ -\frac{1}{2 \cdot 4} r_{i2}^4 \\ \vdots \\ -\frac{1}{2 \cdot 4} r_{in}^4 \end{bmatrix}}_{\phi_2} + \dots$$



RBF-MQ in Polynomial Expression

$$\begin{aligned}
 s(x_i, y_i, \varepsilon) &= \\
 &\varepsilon^{-s}[\alpha_{-s}\phi_0] + \varepsilon^{-s+1}[\alpha_{-s+1}\phi_0] + \dots + \varepsilon^0[\alpha_0\phi_0] + \varepsilon^1[\alpha_1\phi_0] + \varepsilon^2[\alpha_2\phi_0] + \dots \\
 &+ \varepsilon^{-s}\varepsilon^1[\alpha_{-s}\phi_1] + \dots + \varepsilon^{-1}\varepsilon^1[\alpha_{-s+1}\phi_1] + \varepsilon^0\varepsilon^1[\alpha_0\phi_1] + \varepsilon^1\varepsilon^1[\alpha_1\phi_1] + \dots \\
 &+ \varepsilon^{-s}\varepsilon^2[\alpha_{-s}\phi_2] + \dots + \varepsilon^{-2}\varepsilon^2[\alpha_{-s+1}\phi_2] + \varepsilon^{-1}\varepsilon^2[\alpha_0\phi_2] + \varepsilon^0\varepsilon^2[\alpha_1\phi_2] + \dots \\
 &= \varepsilon^{-s}P_{-s}(x_i, y_i) + \varepsilon^{-s+1}P_{-s+1}(x_i, y_i) + \dots + \varepsilon^0P_0(x_i, y_i) + \varepsilon^1P_1(x_i, y_i) + \varepsilon^2P_2(x_i, y_i) + \dots
 \end{aligned}$$

► Where:

$$\begin{aligned}
 P_{-s}(x_i, y_i) &= \alpha_{-s}\phi_0 \\
 P_{-s+1}(x_i, y_i) &= \alpha_{-s}\phi_1 + \alpha_{-s+1}\phi_0 \\
 P_0(x_i, y_i) &= \alpha_{-s}\phi_s + \alpha_{-s+1}\phi_{s+1} + \dots + \alpha_{-1}\phi_1 + \alpha_0\phi_0 \\
 P_1(x_i, y_i) &= \alpha_{-s}\phi_{s+1} + \alpha_{-s+1}\phi_{s+2} + \dots + \alpha_0\phi_1 + \alpha_1\phi_0
 \end{aligned}
 \left. \vphantom{\begin{aligned} P_{-s}(x_i, y_i) \\ P_{-s+1}(x_i, y_i) \end{aligned}} \right\} \text{ Singular part, often } = 0$$



RBF-MQ in Polynomial Expression

- ▶ Previous steps shows that the RBF interpolating function can be simplifies to polynomials
- ▶ The approach of [4,5] is applied to find the inverse of RBF matrix \mathbf{A}^{-1} in Laurent series expansion form, without actually solving the linear system:

$$\mathbf{A}^{-1}(\varepsilon) = \varepsilon^{-s}(\mathbf{Letter } i)^{\#}(\varepsilon) + \dots + \varepsilon^{-2}\mathbf{C}^{\#}(\varepsilon) + \varepsilon^{-1}\mathbf{B}^{\#}(\varepsilon) + \mathbf{A}^{\#}(\varepsilon)$$

$(\mathbf{Letter } i)^{\#}$ is the pseudo inverse of the component of $\mathbf{A}(\varepsilon)$

[4] P. Gonzalez-Rodriguez, M. Moscoso, M. Kindelan, Laurent expansion of the inverse of perturbed, singular matrices, *J. Comput. Phys.*, no. 299, 307-319, 2015

[5] M. Kindelan, M. Moscoso, P. González-Rodríguez , Radial basis function interpolation in the limit of increasingly flat basis function, *J. Comput. Phys.*, no. 307, pp. 225-242, 2016

RBF-MQ in Polynomial Expression

$$\mathbf{A}(\varepsilon) = \mathbf{A}_0 + \varepsilon\mathbf{A}_1 + \varepsilon^2\mathbf{A}_2 + \varepsilon^3\mathbf{A}_3 + \dots$$

Define resolvent of $\mathbf{A}(\varepsilon)$ as:

$$\mathbf{R}(\lambda) = (\mathbf{A} - \lambda\mathbf{I})^{-1}$$

The inverse of \mathbf{A} restricted to the range:

$$\mathbf{A}^\#(\varepsilon) = -\frac{1}{2\pi i} \int_{C_0} \frac{\mathbf{R}(\lambda)}{\lambda} d\lambda$$

C_0 is a positively-oriented contour enclosing $\lambda_0 = 0$ but not other eigenvalue of \mathbf{A}



RBF-MQ in Polynomial Expression

$$\mathbf{A}^\#(\varepsilon) = -\frac{1}{2\pi i} \int_{C_0} \frac{\mathbf{R}(\lambda)}{\lambda} d\lambda$$

The power series is obtained after integrating term by term

$$\mathbf{A}^\#(\varepsilon) = \mathbf{A}_0^\# + \varepsilon \mathbf{A}_1^\# + \varepsilon^2 \mathbf{A}_2^\# + \dots$$

$\mathbf{A}_0^\#$ is the pseudo inverse of \mathbf{A}_0


Other terms, $\mathbf{A}_n^\#$ can be expressed as sum of matrix products:

$$\mathbf{A}_n^\#(\varepsilon) = \sum_{p=1}^n (-1)^p \sum_{\substack{v_1+v_2+\dots+v_p=n, v_i>0 \\ \mu_1+\dots+\mu_{p+1}=p+1, \mu_i\geq 0}} \mathbf{S}_{\mu_1} \mathbf{A}_{v_1} \mathbf{S}_{\mu_2} \mathbf{A}_{v_2} \dots \mathbf{S}_{\mu_p} \mathbf{A}_{v_p} \mathbf{S}_{\mu_{p+1}}$$

$$\mathbf{S}_0 \equiv -\mathbf{P}_0$$

\mathbf{P}_0 = projection of \mathbf{A}_0 onto null space

$$\mathbf{S}_k = (\mathbf{A}_0^\#)^k \text{ for } k > 0$$



RBF-MQ in Polynomial Expression

Extracting $\mathbf{A}^\#$ from \mathbf{A}^{-1}

$$\mathbf{A}^{-1}(\varepsilon) = \mathbf{A}^\#(\varepsilon) + \frac{1}{\varepsilon} \underbrace{\left(\frac{1}{\varepsilon} \mathbf{A}(\varepsilon) \mathbf{P}_{\mathbf{A}0}(\varepsilon) \right)}_{\mathbf{B}(\varepsilon)}^\#$$

$\mathbf{P}_{\mathbf{A}0}(\varepsilon)$ = sum of eigenprojections

$$\mathbf{B}(\varepsilon) = -\frac{1}{2\pi i \varepsilon} \int_{C_0} \lambda \mathbf{R}(\lambda) d\lambda$$

$$\mathbf{B}(\varepsilon) = \mathbf{B}_0 + \sum_{n=1}^{\infty} \varepsilon^n \mathbf{B}_n$$

$$\mathbf{B}_n = - \sum_{p=1}^{n+1} (-1)^p \sum_{\substack{v_1+v_2+\dots+v_p=n+1, v_i>0 \\ \mu_1+\dots+\mu_{p+1}=p-1, \mu_i\geq 0}} \mathbf{S}_{\mu_1} \mathbf{A}_{v_1} \mathbf{S}_{\mu_2} \mathbf{A}_{v_2} \dots \mathbf{S}_{\mu_p} \mathbf{A}_{v_p} \mathbf{S}_{\mu_{p+1}}$$



RBF-MQ in Polynomial Expression

The process continued as:

$$\mathbf{A}^{-1}(\varepsilon) = \mathbf{A}^{\#}(\varepsilon) + \frac{1}{\varepsilon} \mathbf{B}^{\#}(\varepsilon) + \frac{1}{\varepsilon} \left(\frac{1}{\varepsilon} \left(\frac{1}{\varepsilon} \underbrace{\mathbf{B}(\varepsilon) \mathbf{P}_{\mathbf{B0}}(\varepsilon)}_{\mathbf{C}(\varepsilon)} \right) \right)^{\#}$$

Until some stage where the next terms inside the bracket = null matrix

The Laurent series of \mathbf{A}^{-1} can then be express as:

$$\mathbf{A}^{-1}(\varepsilon) = \dots + \varepsilon^{-2} \mathbf{C}^{\#}(\varepsilon) + \varepsilon^{-1} \mathbf{B}^{\#}(\varepsilon) + \mathbf{A}^{\#}(\varepsilon)$$



RBF-MQ in Polynomial Expression

▶ General procedure:

Starts with $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \dots$

- 1) Compute pseudo inverse $\mathbf{A}^\#_0$ of \mathbf{A}_0
- 2) Compute \mathbf{P}_0 , the projection of \mathbf{A}_0 into null space
- 3) Compute $\mathbf{S}_0 = -\mathbf{P}_0$, then $\mathbf{S}_2 = \mathbf{S}_0^2, \mathbf{S}_3 = \mathbf{S}_0^3, \dots$
- 4) Compute $\mathbf{A}^\#_1, \mathbf{A}^\#_2, \mathbf{A}^\#_3 \dots$

- 5) Compute for next letter, $\mathbf{B}_0, \mathbf{B}_1, \mathbf{B}_2, \dots$
- 6) The process (1),(2),(3),(4) are repeated by changing all \mathbf{A}_k with respective \mathbf{B}_k

- 7) Compute for next letter $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2, \dots$ if needed (depends on the order of singularity)



RBF-MQ in Polynomial Expression

- ▶ The inverse can also be written in the form below by grouping the terms with the same power of ε

$$\mathbf{A}^{-1}(\varepsilon) = \sum_{k=-s}^{\infty} \varepsilon^k \mathbf{H}_k$$

- ▶ H_k can be expressed in terms of $A^\#, B^\#, C^\# \dots$. For example, for the singularity order $s=2$, the singular terms:

$$\begin{aligned}\mathbf{H}_{-2} &= \mathbf{C}_0^\# \\ \mathbf{H}_{-1} &= \mathbf{B}_0^\# + \mathbf{C}_1^\# \\ \mathbf{H}_0 &= \mathbf{A}_0^\# + \mathbf{B}_1^\# + \mathbf{C}_2^\#\end{aligned}$$



RBF-MQ in Polynomial Form

- ▶ The RBF-MQ interpolating function $s(x, y, \varepsilon)$ can be expressed as

$$\begin{aligned} s(x, y, \varepsilon) &= \varepsilon^{-s} P_{-s}(x, y) + \varepsilon^{-s+1} P_{-s+1}(x, y) + \dots \\ &+ \varepsilon^0 P_0(x, y) + \varepsilon^1 P_1(x, y) + \varepsilon^2 P_2(x, y) + \dots \end{aligned}$$

- ▶ Often, the singular parts $\varepsilon^{-ve} P_{-ve}(x, y)$ are zero regardless of (x, y)

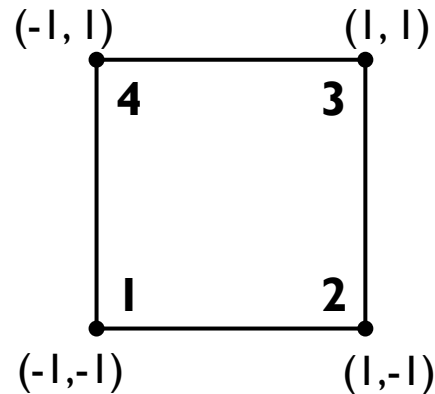
$$s(x, y, \varepsilon) = P_0(x, y) + \varepsilon P_1(x, y) + \varepsilon^2 P_2(x, y) + \dots$$



RBF-MQ in Polynomial Expression

Four nodes RBF-MQ in polynomial expression up to ε^2

$$s(x, y, \varepsilon) = P_0(x, y) + \varepsilon P_1(x, y) + \varepsilon^2 P_2(x, y)$$



$$P_0(x, y) = \frac{1}{4}(f_1 + f_2 + f_3 + f_4) + \frac{1}{4}(-f_1 + f_2 + f_3 - f_4)x + \frac{1}{4}(-f_1 - f_2 + f_3 + f_4)y + \frac{1}{4}(f_1 - f_2 + f_3 - f_4)xy$$


$$P_1(x, y) = \frac{1}{8}(f_1 + f_2 + f_3 + f_4)[(x^2 + y^2) - 2] + \frac{1}{8}(f_1 - f_2 + f_3 - f_4)[6xy - 3(x^3y + xy^3)] + \frac{1}{8}(f_1 - f_3)[(x^3 + y^3) + (x^2y + xy^2) - 2(x + y)] + \frac{1}{8}(f_2 - f_4)[-(x^3 - y^3) + (x^2y - xy^2) + 2(x - y)]$$

RBF-MQ in Polynomial Expression

Four nodes RBF-MQ in polynomial expression up to ε^2

$$\begin{aligned} P_2(x, y) = & \frac{1}{32} (f_1 - f_2 + f_3 - f_4) [15(x^5y + xy^5) + 8(x^3y + xy^3) + 30x^3y^3 - 76xy] \\ & + \frac{1}{32} (f_1 + f_2 + f_3 + f_4) [-(x^4 + y^4) - 16(x^2 + y^2) - 2x^2y^2 + 36] \\ & + \frac{1}{32} (f_1 - f_3) [-3(x^5 + y^5) - 8(x^3 + y^3) + 36(x + y) - 3(x^4y + xy^4) - 6(x^3y^2 + x^2y^3) \\ & - 16(x^2y + xy^2)] \\ & + \frac{1}{32} (f_2 - f_4) [3(x^5 - y^5) + 8(x^3 - y^3) - 36(x - y) - 3(x^4y - xy^4) + 6(x^3y^2 - x^2y^3) \\ & - 16(x^2y - xy^2)] \end{aligned}$$

Four nodes RBF-MQ converged to bilinear function when $\varepsilon \rightarrow 0$



Optimum Shape Parameter ε

- ▶ RBF-MQ in polynomial form allow the interpolation with any value of ε .
- ▶ However the for ε , the optimum value is varies case by case, or problem dependent.
- ▶ It is yet any sound approach to predict optimum ε , although it is found that the optimum value is often small.



Optimum Shape Parameter ε

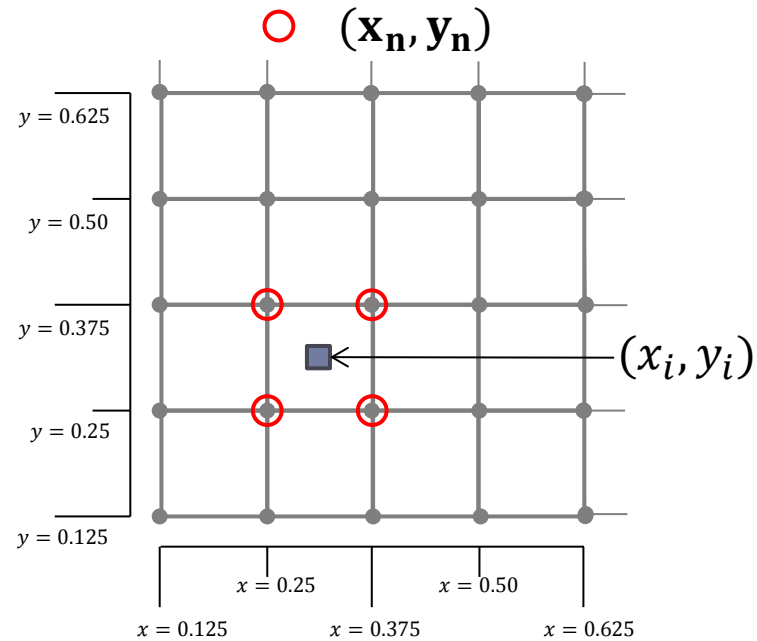
- ▶ Example: test function

$$f(x, y) = \frac{25}{25 + (x - 0.2)^2 + 2y^2} \quad (x, y) \in (0, 1)$$

- ▶ Discretized with 9x9 grid nodes with the discrete value of $f(x, y)$ known at each nodal point.
- ▶ Interpolating $f(x, y)$ to the point of interest $(x_i, y_i) = (0.3125, 0.3125)$
- ▶ The nearest four nodal points are $(\mathbf{x}_n, \mathbf{y}_n) = [(0.25, 0.25), (0.375, 0.25), (0.375, 0.375), (0.25, 0.375)]$, with discrete value of $\mathbf{f}_n = [0.994926, 0.993814, 0.987679, 0.988777]$



Optimum Shape Parameter ε



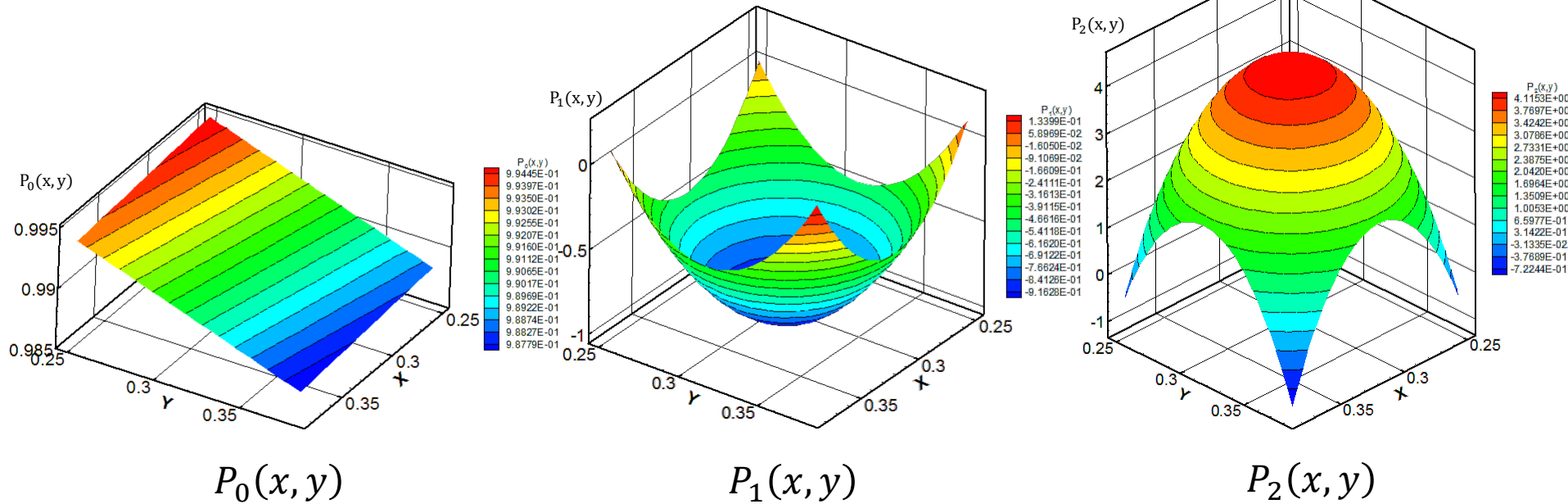
$$\mathbf{f}_n = [0.994926, 0.993814, 0.987679, 0.988777]$$

$$s(x_i, y_i, \varepsilon) = 0.9913 - 0.9913\varepsilon + 4.4608\varepsilon^2$$



Optimum Shape Parameter ε

- Distribution of $P_0(x, y)$, $P_1(x, y)$, $P_2(x, y)$



- The RBF-MQ interpolating function is hence the correction of “base function” $P_0(x, y)$ using the higher order polynomial functions $P_1(x, y)$ and $P_2(x, y)$, with ε as the weighting parameter.

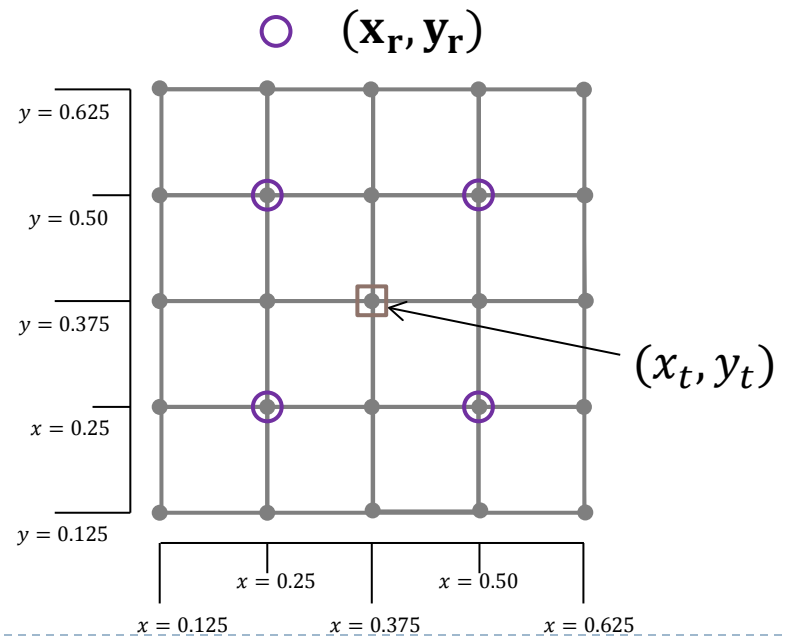
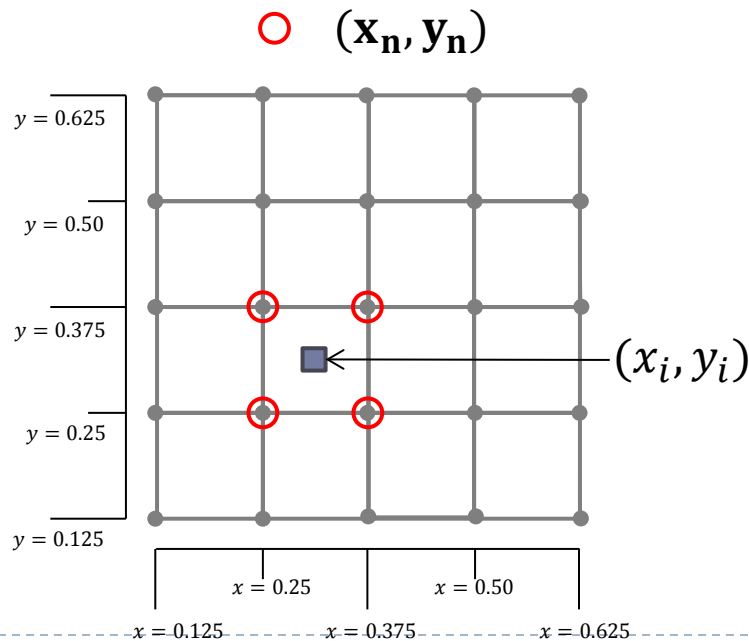
Optimum Shape Parameter ε

- ▶ Since the distribution of $f(x, y)$ is different across the domain, the correction needed at different region is different.
- ▶ In other words, the optimum value of ε across the domain is varies depends on the distribution of $f(x, y)$.
- ▶ To determine the optimum value of ε , the local distribution of $f(x, y)$ is evaluate using nodes at the outer ring of the interpolation domain,



Optimum Shape Parameter ε

- ▶ Taking additional surrounding nodes: $(\mathbf{x}_r, \mathbf{y}_r) = [(0.25, 0.25), (0.5, 0.25), (0.5, 0.5), (0.25, 0.5)]$ to enclose the nodes $(\mathbf{x}_n, \mathbf{y}_n)$ and (x_i, y_i)
- ▶ An enclosed node (x_t, y_t) is use to predict the ε using nodes $(\mathbf{x}_r, \mathbf{y}_r)$



Optimum Shape Parameter ε

- ▶ Interpolating function at (x_t, y_t) using nodes $(\mathbf{x}_r, \mathbf{y}_r)$:

$$s_r(x_t, y_t, \varepsilon) = 0.9859 - 0.9859\varepsilon + 4.4366\varepsilon^2$$

- ▶ Discrete value of at $f(x_t, y_t) = 0.987679$.

- ▶ Optimum ε : $s_r(x_t, y_t, \varepsilon) = f(x_t, y_t)$

- ▶ Since the interpolating function $s_r(x_t, y_t, \varepsilon)$ is a second order polynomial of ε , the value of optimum ε can be computed by solving for root of polynomial.

- ▶ $\varepsilon_1 = 0.224, \varepsilon_2 = -1.78 \times 10^{-3}$
-



Optimum Shape Parameter ε

- ▶ The ε with smaller absolute value is chosen in order to reduce the error arisen from the numerical cancellation between the correction functions $\varepsilon P_1(x, y)$ and $\varepsilon^2 P_2(x, y)$.
- ▶ In present implementation, the use of negative value of ε is permissible as ε is merely treated as parameter to control the weights of $P_1(x, y)$ and $P_2(x, y)$.
- ▶ Hence, $\varepsilon_{opt} = -1.78 \times 10^{-3}$ for interpolation from $(\mathbf{x}_r, \mathbf{y}_r)$ to (x_t, y_t) .



Optimum Shape Parameter ε

- ▶ Previous steps provided some information regarding the discrepancy between the base function P_0 (bilinear for four nodes interpolation) and the function to be interpolated $f(x, y)$.
- ▶ Since $f(x, y)$ is a smooth function, the predictions obtained by using nodal points $(\mathbf{x}_n, \mathbf{y}_n)$ and $(\mathbf{x}_r, \mathbf{y}_r)$ have continuous and similar distributions.
- ▶ However, the magnitude of correction in $(\mathbf{x}_n, \mathbf{y}_n)$ is less than that of $(\mathbf{x}_r, \mathbf{y}_r)$ due to the smaller nodal distance, which would result in a smaller difference between the maximum and the minimum values of $f(x, y)$ among the nodal points $(\mathbf{x}_n, \mathbf{y}_n)$



Optimum Shape Parameter ε

- ▶ The value of ε_{opt} found using $(\mathbf{x}_r, \mathbf{y}_r)$ is scaled by $(r_{(x_n, y_n)} / r_{(x_r, y_r)})^2$, in which is equivalent radius of the area bounded by the nodal point set $(\mathbf{x}_n, \mathbf{y}_n)$ and $(\mathbf{x}_r, \mathbf{y}_r)$.
- ▶ In this example, nodes $(\mathbf{x}_n, \mathbf{y}_n)$ and $(\mathbf{x}_r, \mathbf{y}_r)$ are lying on uniform grid, the scaling factor = 0.25
- ▶ Predicted shape parameter for nodes $(\mathbf{x}_n, \mathbf{y}_n)$ at (x_i, y_i) is $\varepsilon = -4.5404 \times 10^{-4}$.



Optimum Shape Parameter ε

- ▶ Interpolating function at (x_i, y_i) :

$$s(x_i, y_i, \varepsilon) = 0.9913 - 0.9913\varepsilon + 4.4608\varepsilon^2$$

- ▶ With $\varepsilon = -4.5404 \times 10^{-4}$, predicted solution $f(x_i, y_i) \approx s(x_i, y_i, \varepsilon) = 0.991741$. (Error = 9.02875×10^{-6})

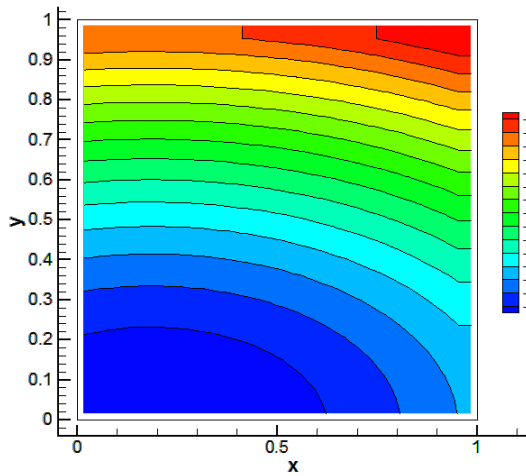
- ▶ Exact ε that makes $f(x_i, y_i) = s(x_i, y_i, \varepsilon)$:

$$\varepsilon = -4.4497 \times 10^{-4}$$

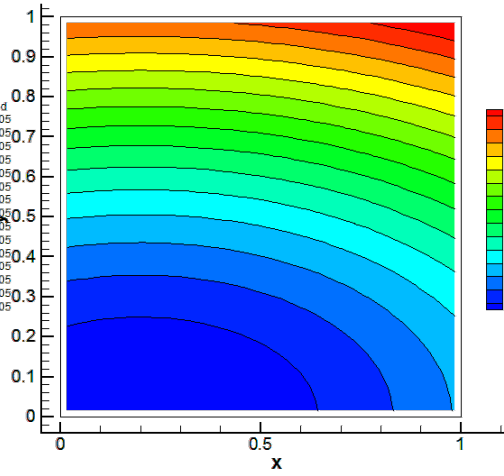
- ▶ Discrepancy with predicted ε is less than 0.1%
-



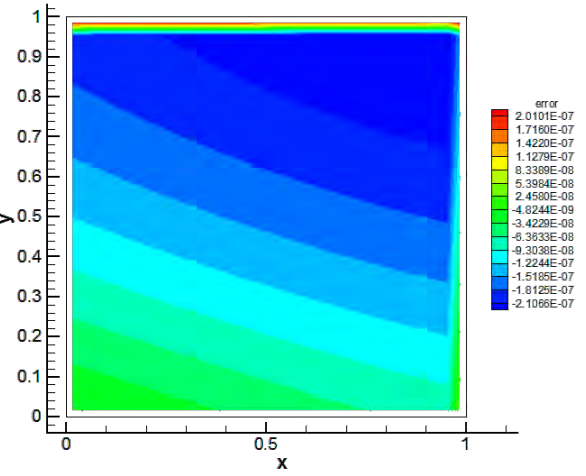
Optimum Shape Parameter ε



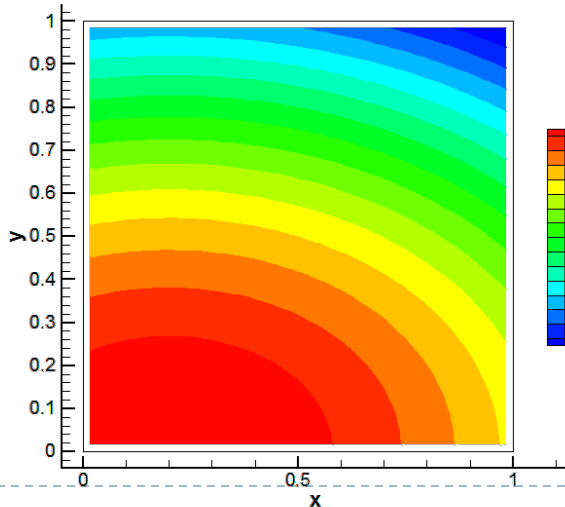
Predicted ε



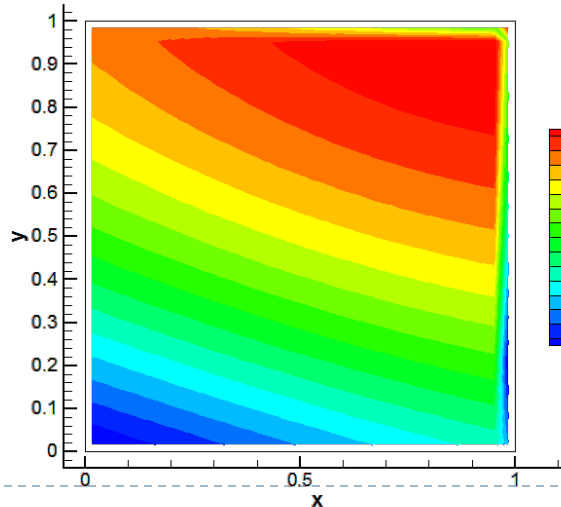
Exact ε



Exact $\varepsilon - \text{predicted } \varepsilon$



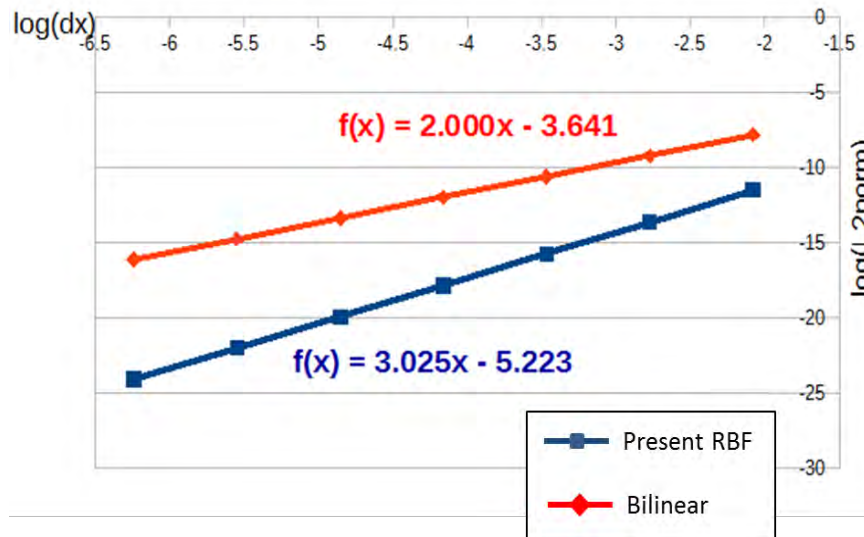
Predicted $f(x, y)$



$\|f_{exact} - \text{predicted } f(x, y)\|$

Optimum Shape Parameter ε

- ▶ Improvements from base function (bilinear)



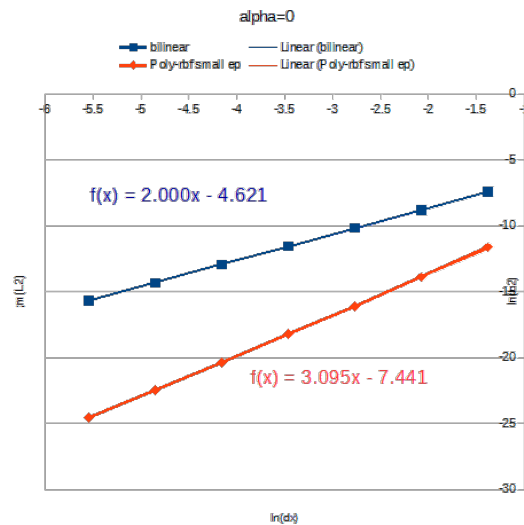
$$L^2 \text{ norm} = \sqrt{\frac{\sum_{i=1}^N (f_{\text{exact}} - f_{\text{calculate}})^2}{N}}$$

Optimum Shape Parameter ε

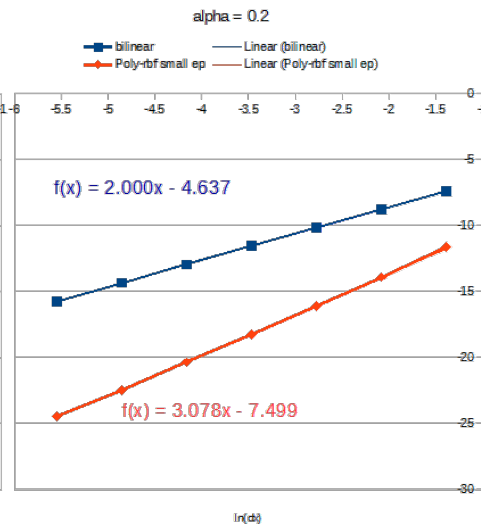
► Other test function

$$f_1(x, y) = \sqrt{1 + 0.04((x - 0.5)^2 + (y - 1)^2)}$$

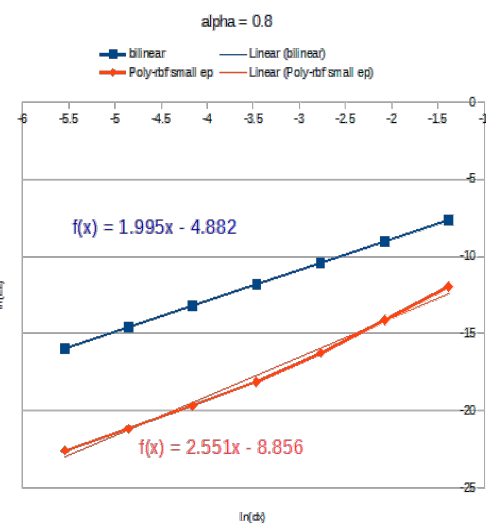
$\alpha = 0$



$\alpha = 0.2$



$\alpha = 0.8$

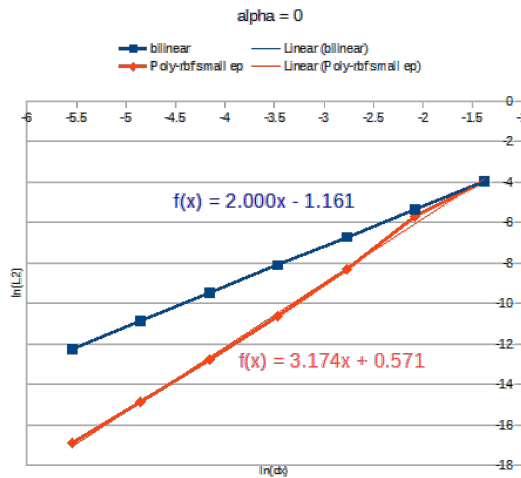


Optimum Shape Parameter ε

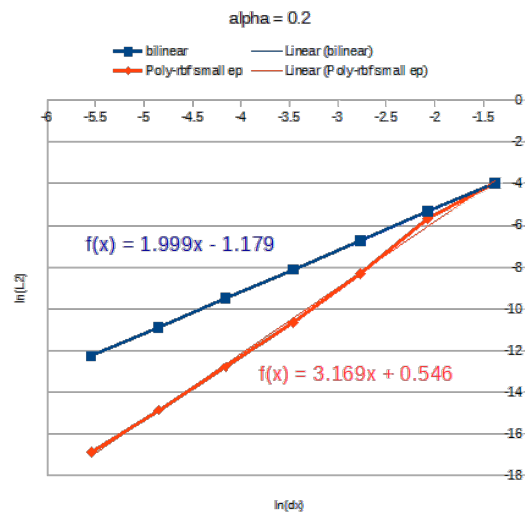
► Other test function

$$f_3(x, y) = 0.25(x^3 + y^3) + 0.1(x^2y) + 0.2(xy^2) + 0.3(x^2)$$

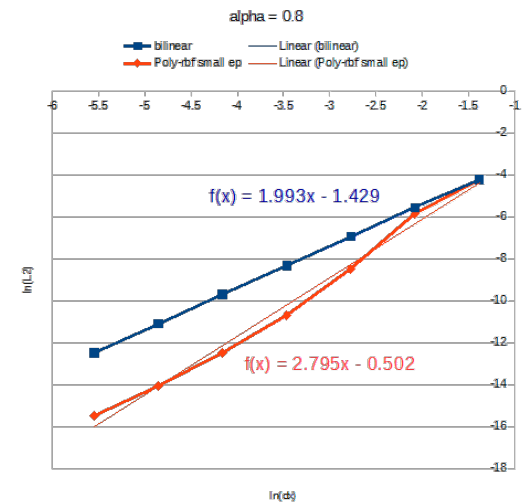
$\alpha = 0$



$\alpha = 0.2$



$\alpha = 0.8$



Mass-Preserving Interpolation

► **Motivation:**

- The continuity constraint satisfied on Eulerian grids may be lost after the velocity update on Lagrangian particles.
- Conventional scalar interpolation scheme does not preserve continuity constraint.
- The interpolation scheme with mass/continuity preserving property is applied to transfer the velocity from Eulerian grids to Lagrangian particles.



Mass-Preserving Interpolation

- ▶ A divergence-free RBF kernel can be defined from scalar RBF via:

$$\boldsymbol{\varphi}(\mathbf{x}) = (-\nabla^2 \mathbf{I} + \nabla \nabla^T) \emptyset(\mathbf{x}) \quad (7)$$

$\emptyset(\mathbf{x})$: Scalar RBF kernel chosen

\mathbf{I} : Identity matrix

- ▶ The operator $(-\nabla^2 \mathbf{I} + \nabla \nabla^T)$ transforms the scalar BRF kernel $\emptyset(\mathbf{x})$ to the vector RBF kernel $\boldsymbol{\varphi}(\mathbf{x})$.



Mass-Preserving Interpolation

- ▶ For a multiquadric RBF kernel, the transformation to divergence-free vector RBF is:

$$\boldsymbol{\varphi}(\mathbf{x}) = \begin{pmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{pmatrix} = \begin{pmatrix} \frac{\varepsilon^2 y^2}{(\sqrt{1 + \varepsilon r^2})^3} - \frac{\varepsilon}{\sqrt{1 + \varepsilon r^2}} & -\frac{\varepsilon^2 xy}{(\sqrt{1 + \varepsilon r^2})^3} \\ -\frac{\varepsilon^2 xy}{(\sqrt{1 + \varepsilon r^2})^3} & \frac{\varepsilon^2 x^2}{(\sqrt{1 + \varepsilon r^2})^3} - \frac{\varepsilon}{\sqrt{1 + \varepsilon r^2}} \end{pmatrix} \quad (8)$$

- ▶ The interpolating function $\mathbf{s}(\mathbf{x})$ is an interpolating vector. Vector $\mathbf{s}(\mathbf{x})$ is always divergence-free regardless of particle position \mathbf{x} .

$$\mathbf{s}(\mathbf{x}) = \begin{Bmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{Bmatrix} = \sum_{i=1}^N \alpha_i \boldsymbol{\varphi}(\mathbf{x} - \mathbf{x}_i) \quad (9)$$



Mass-Preserving Interpolation

- ▶ Same procedure is applied to change the interpolating functions into polynomial form

$$\mathbf{s}(\mathbf{x}) = \begin{Bmatrix} u(\mathbf{x}, \varepsilon) \\ v(\mathbf{x}, \varepsilon) \end{Bmatrix} = \begin{Bmatrix} P_0(\mathbf{x}) + \varepsilon P_1(\mathbf{x}) + \varepsilon^2 P_2(\mathbf{x}) \\ Q_0(\mathbf{x}) + \varepsilon Q_1(\mathbf{x}) + \varepsilon^2 Q_2(\mathbf{x}) \end{Bmatrix}$$

- ▶ $\frac{dP_0(\mathbf{x})}{dx} + \frac{dQ_0(\mathbf{x})}{dy} = 0, \frac{dP_1(\mathbf{x})}{dx} + \frac{dQ_1(\mathbf{x})}{dy} = 0, \frac{dP_2(\mathbf{x})}{dx} + \frac{dQ_2(\mathbf{x})}{dy} = 0$
 - ▶ Each vector component may have their own optimum ε .
 - ▶ Present implementation: ε with smaller magnitude is chosen.
-



Results

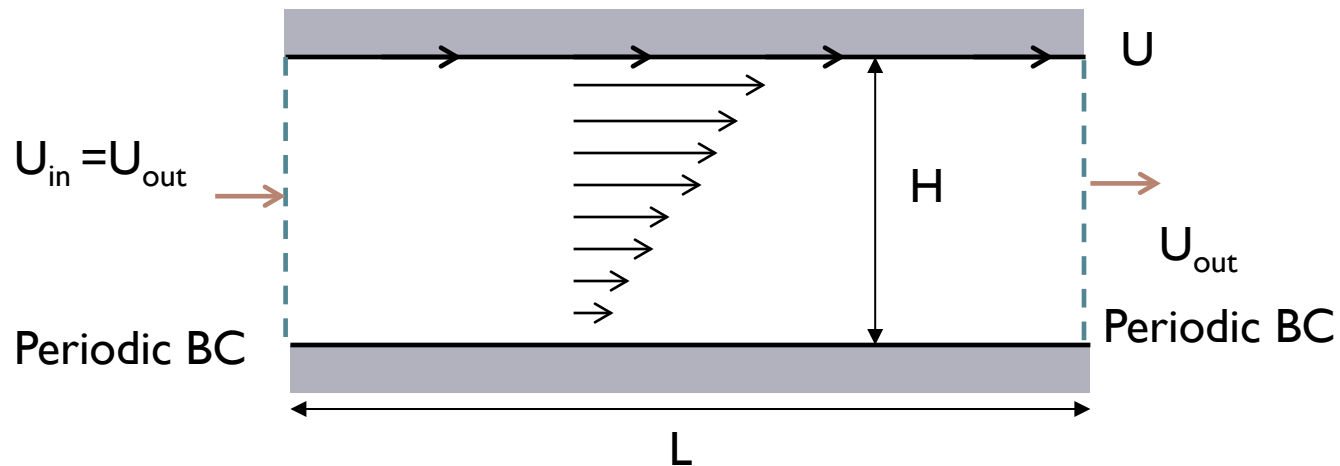
- ▶ Validation Studies
 - Couette Flow
 - Planar Poiseuille Flow
 - Developing Laminar Flow

- ▶ Verification Studies
 - Lid Driven Cavity Flow
 - Backward Facing Step Flow



Results: Validation Studies

► Couette Flow

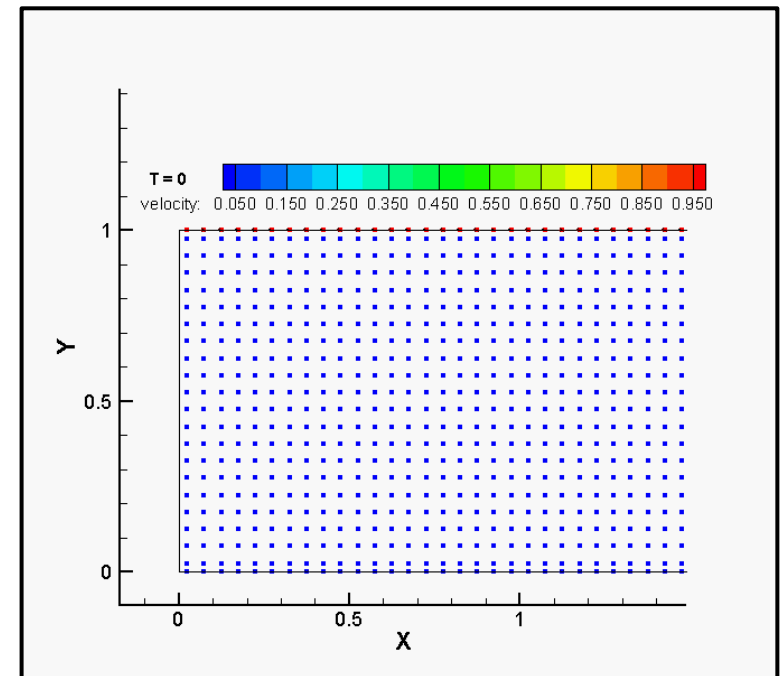
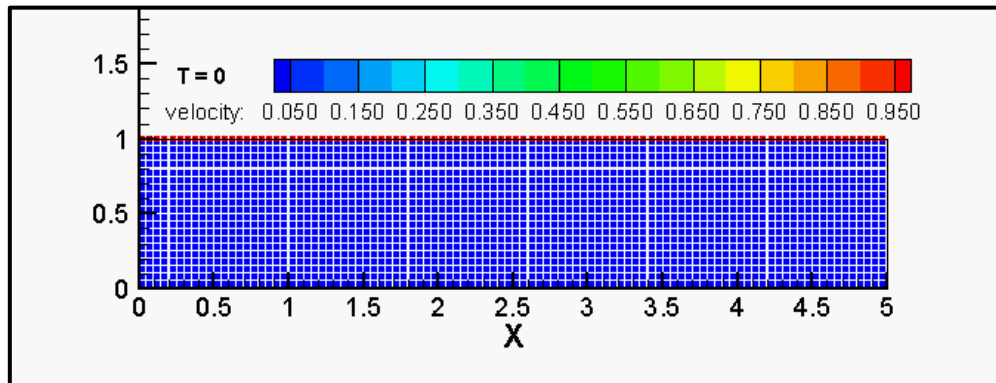


$H =$	5	$\Delta x =$	5/50, 5/100
$L =$	1	$\Delta y =$	1/10, 1/20
$U =$	1.0	$\Delta t =$	0.05
$Re =$	100, 1000		

Results: Validation Studies

► Couette Flow

- Re 100: Velocity contour and particles distribution



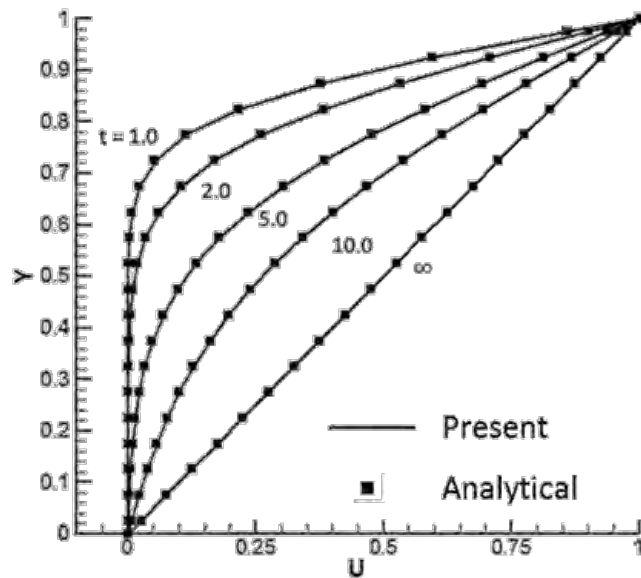
Near inflow



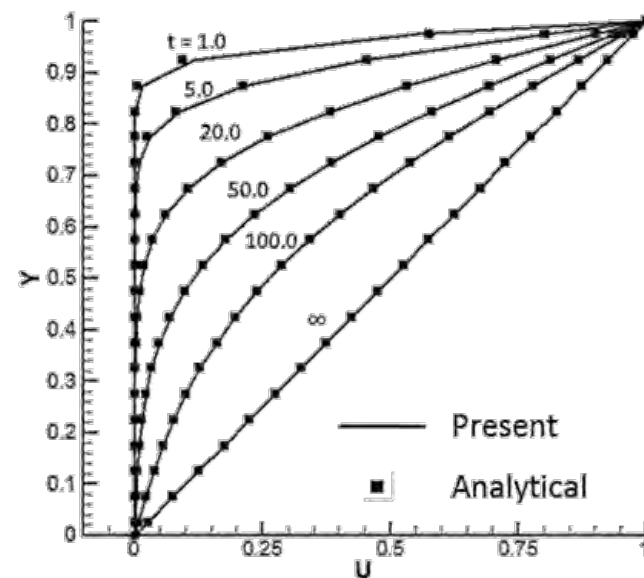
Results: Validation Studies

► Couette Flow

- Normalized velocity profile



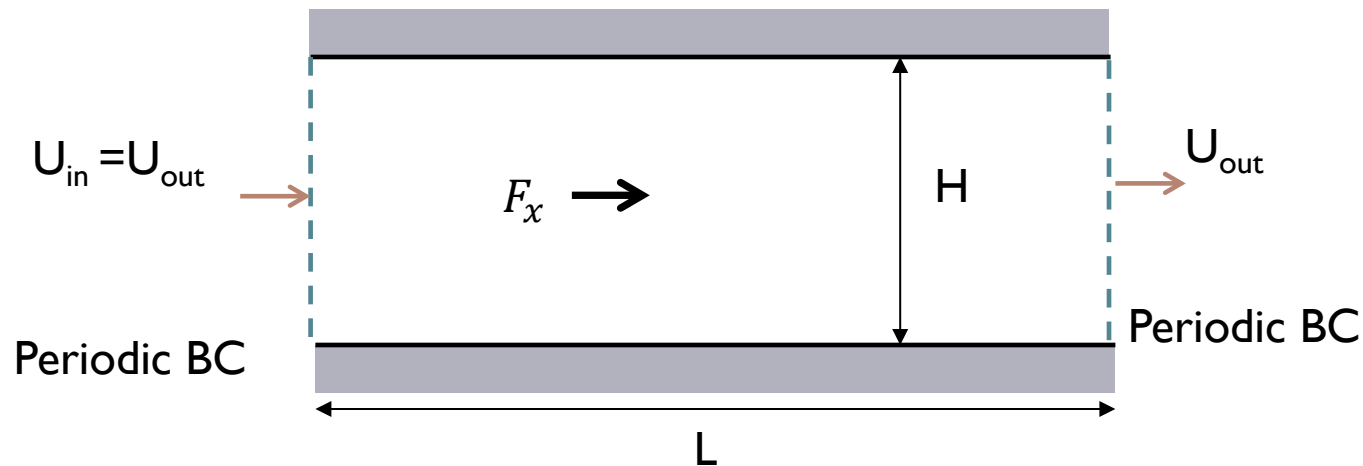
(a) $Re = 100$



(b) $Re = 1000$

Results: Validation Studies

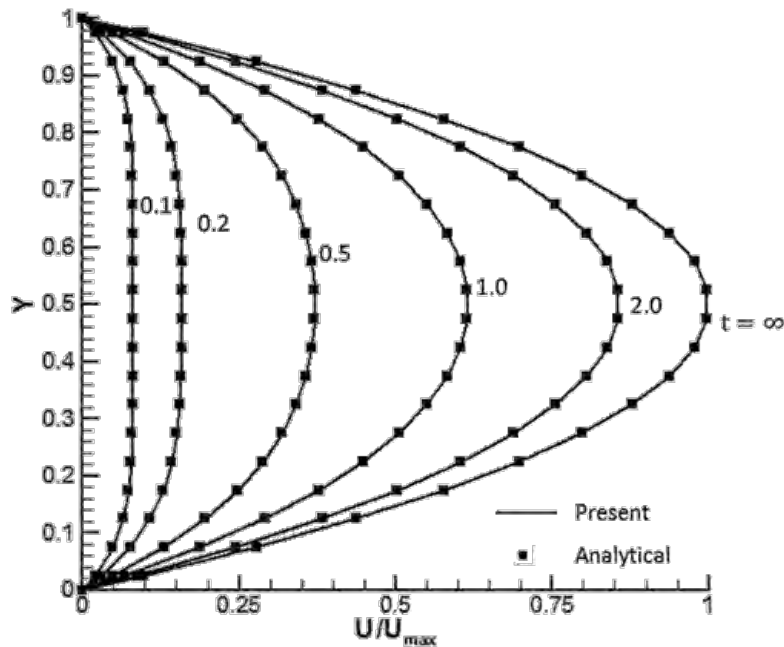
► Planar Poiseuille Flow



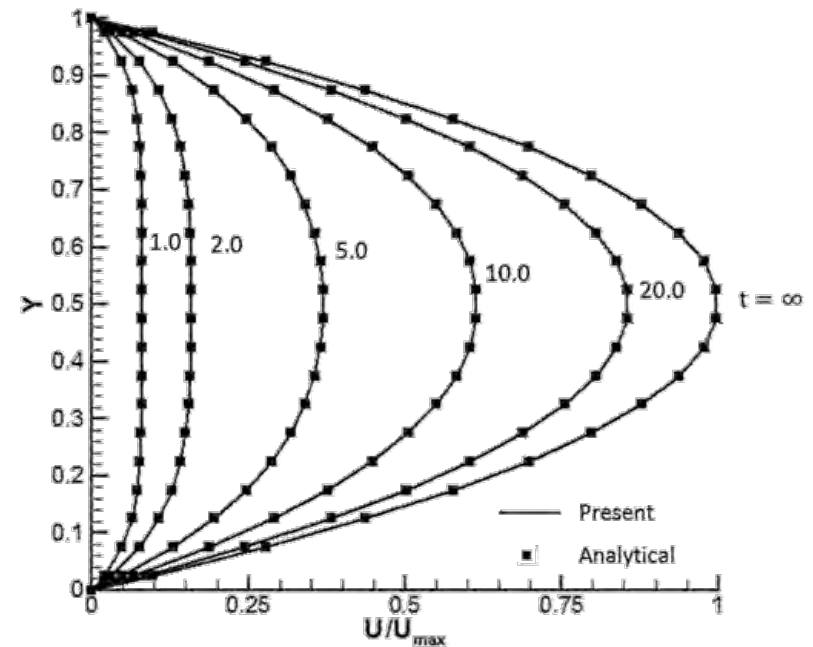
$H =$	5	$\Delta x =$	5/100
$L =$	1	$\Delta y =$	1/20
$U =$	1.0	$\Delta t =$	0.05
$\rho =$	1.0		
$\mu =$	0.1, 0.01		
	(Re 12.5, 1250)		

Results: Validation Studies

- ▶ **Planar Poiseuille Flow**
 - Normalized velocity profile



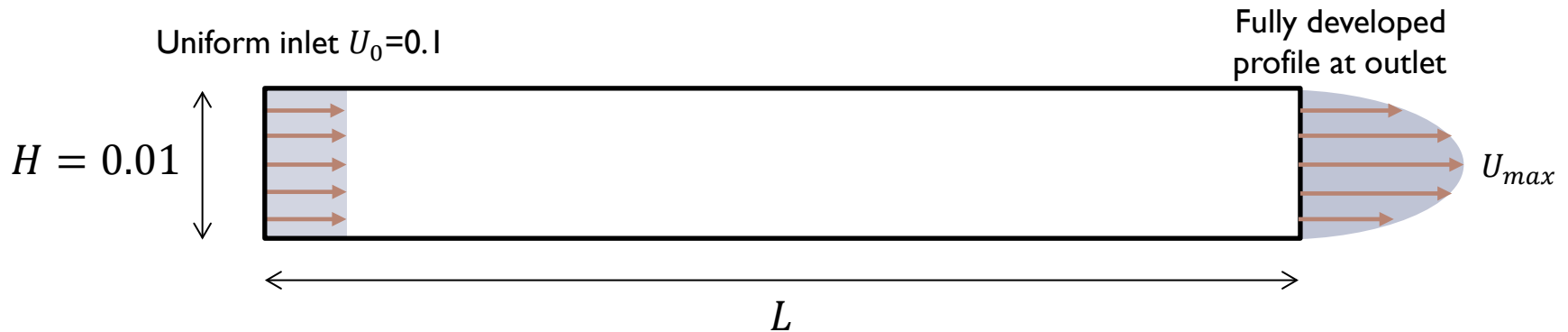
(a) Re 12.5



(b) Re 1250

Results: Validation Studies

▶ Developing Laminar Flow



For Laminar flow, the entry length, is $L_h = 0.05 Re H$

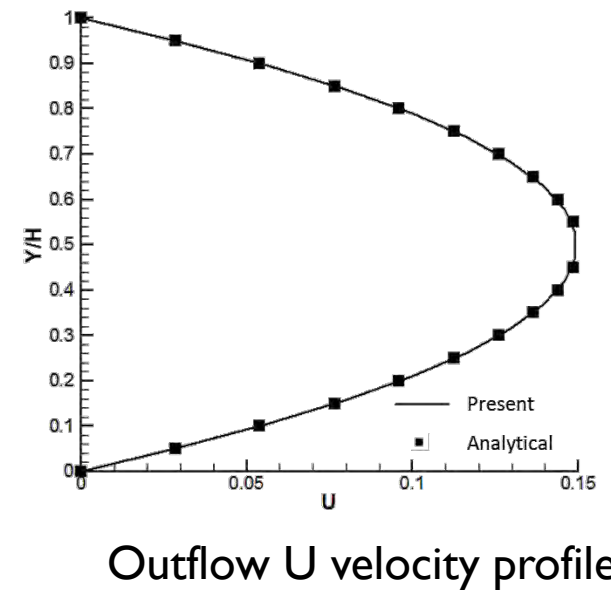
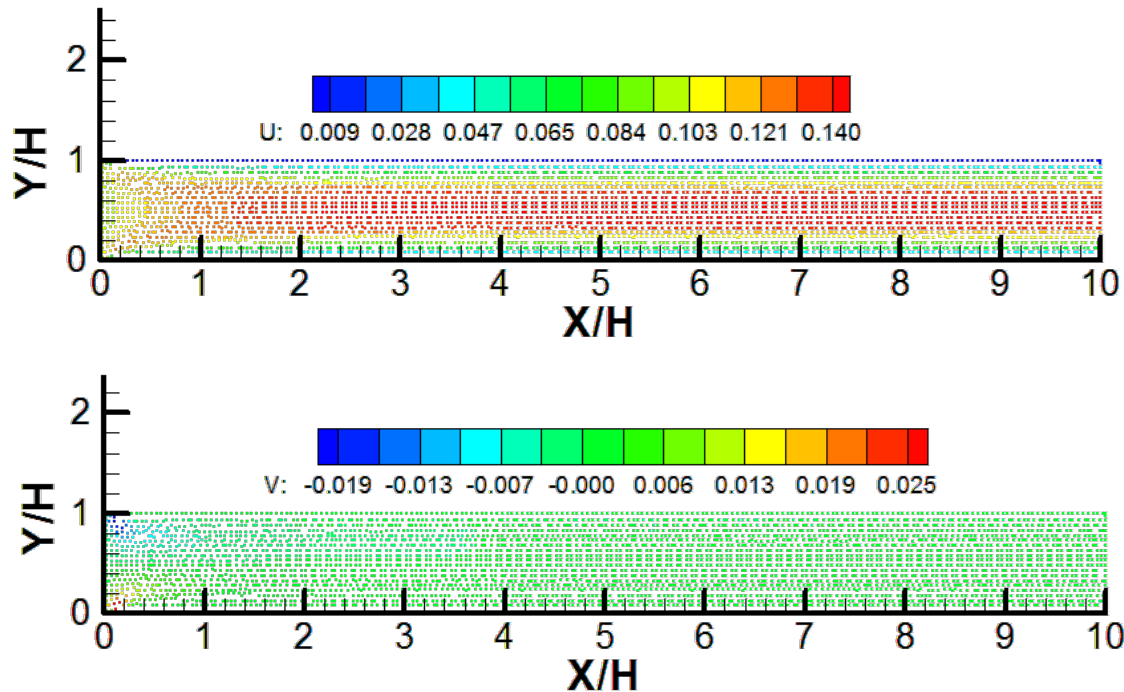
To ensure flow to be fully developed, $L = 2L_h$ is used



Results: Validation Studies

► Developing Laminar Flow

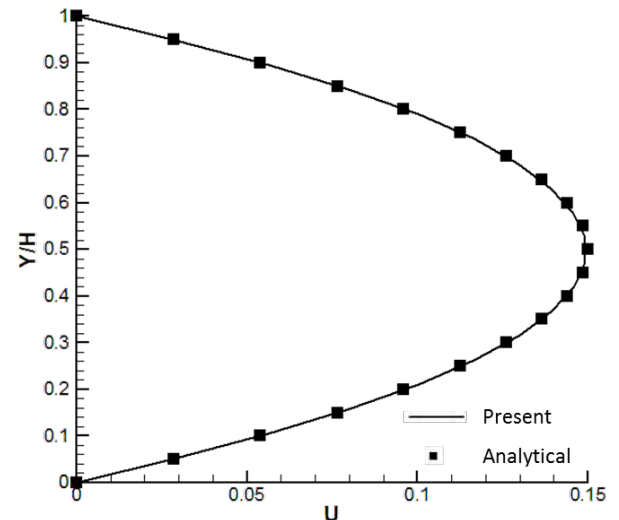
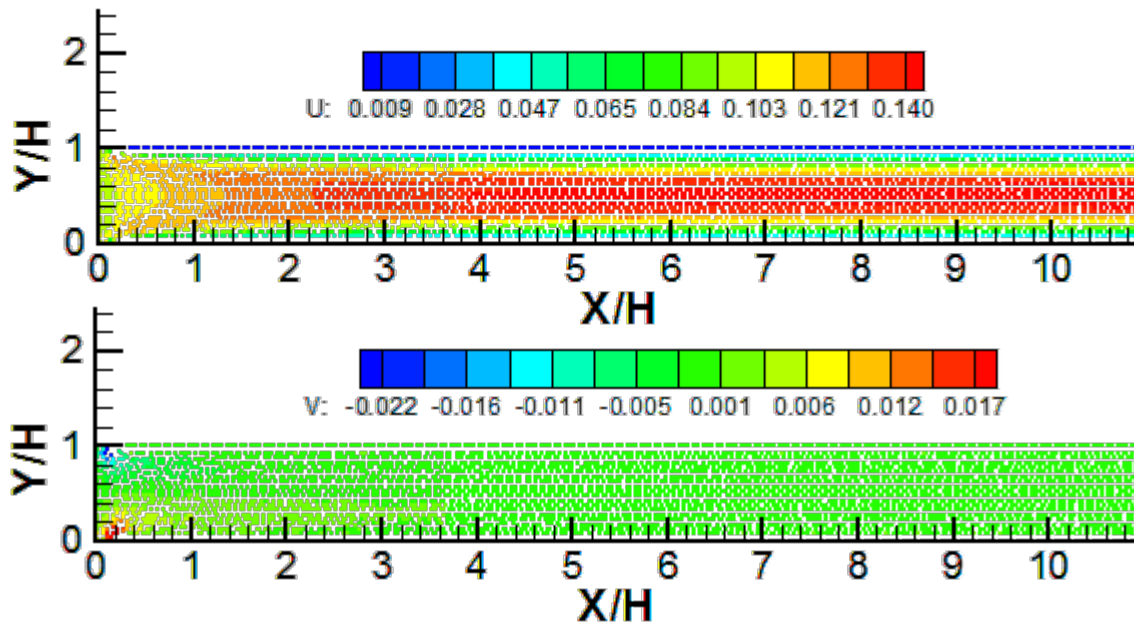
Re 100: U,V velocity contours and particle distribution



Results: Validation Studies

► Developing Laminar Flow

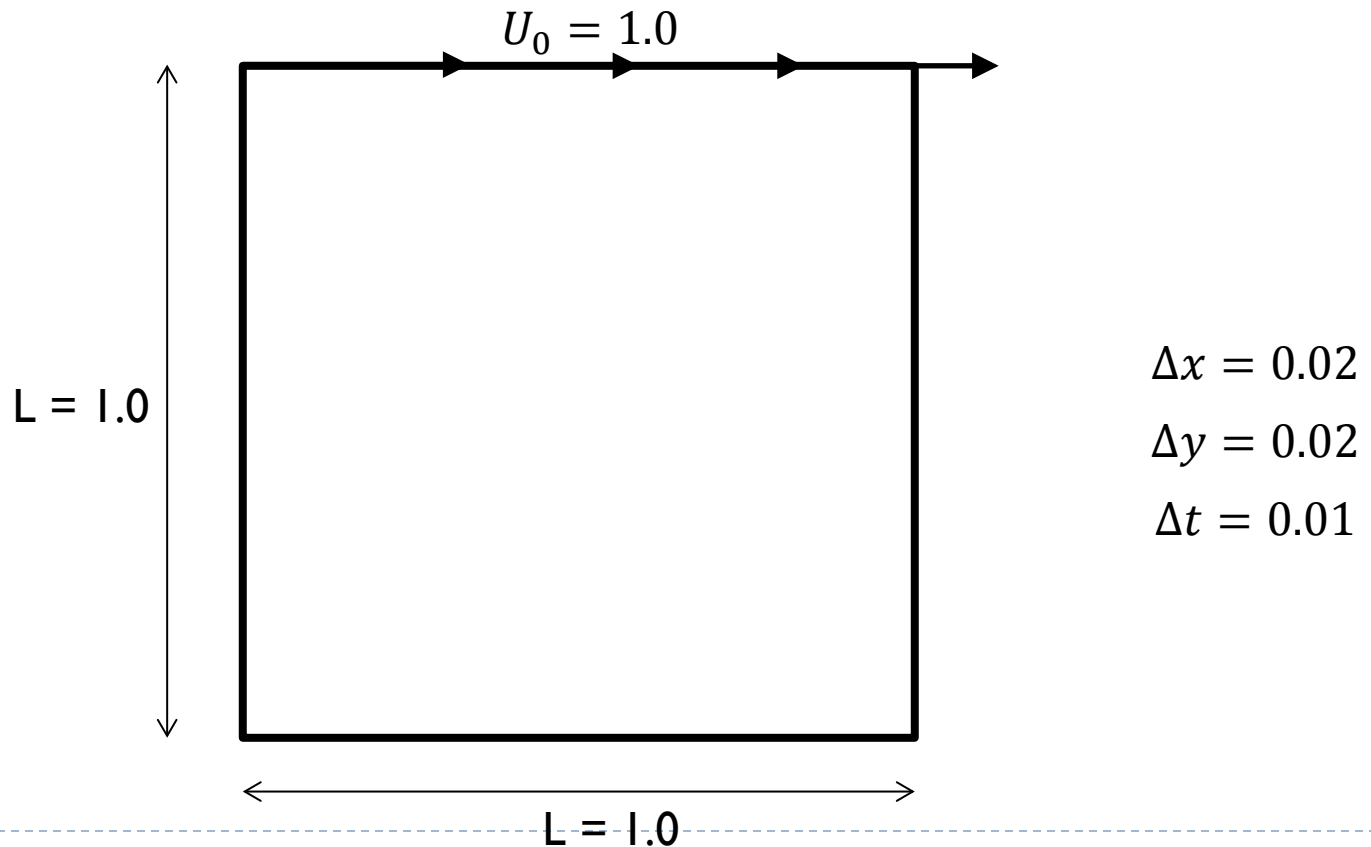
Re 200: U,V velocity contours and particle distribution



Outflow U velocity profile

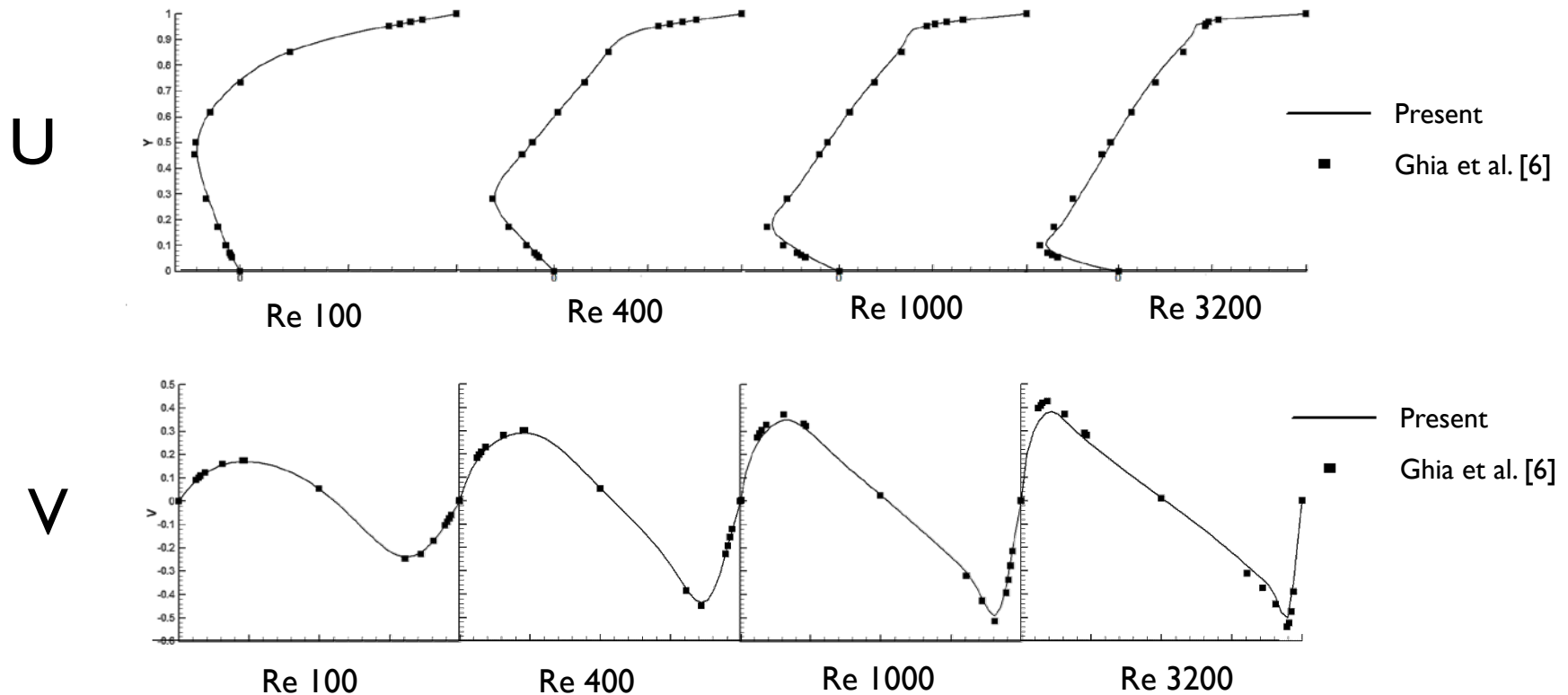
Results: Verification Studies

► Lid Driven Cavity Flow



Results: Verification Studies

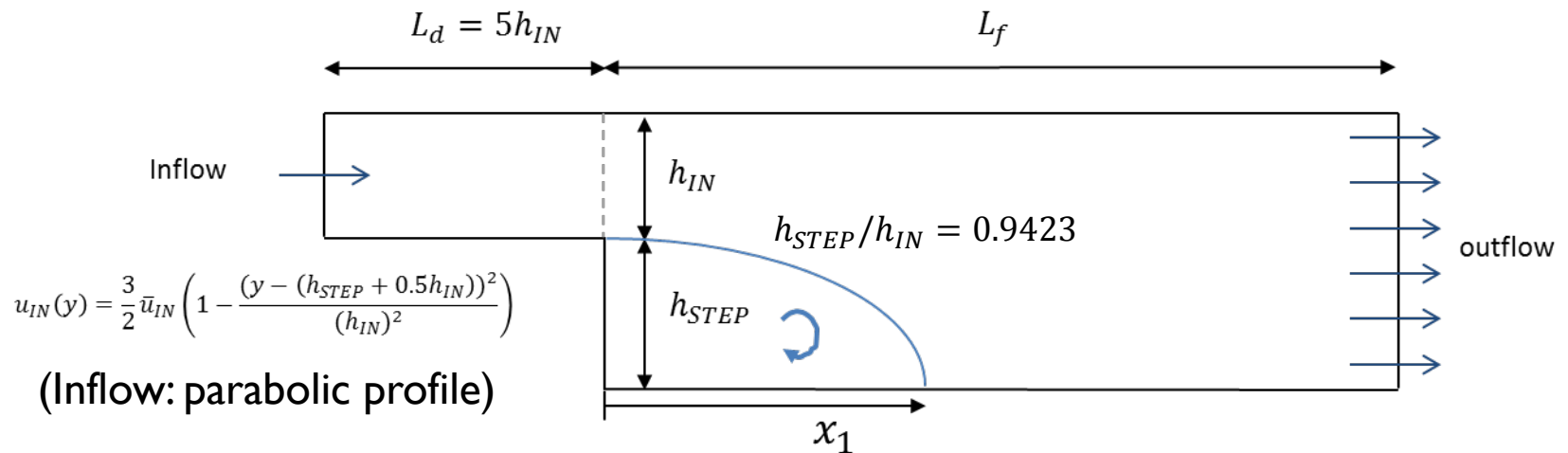
► Lid Driven Cavity Flow



[6] U. Ghia, K.N. Ghia, and C.T. Shin, High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, *J. Comput. Phys.*, vol. 48, pp. 387-411, 1982.

Results: Verification Studies

▶ Backward Facing Step Flow

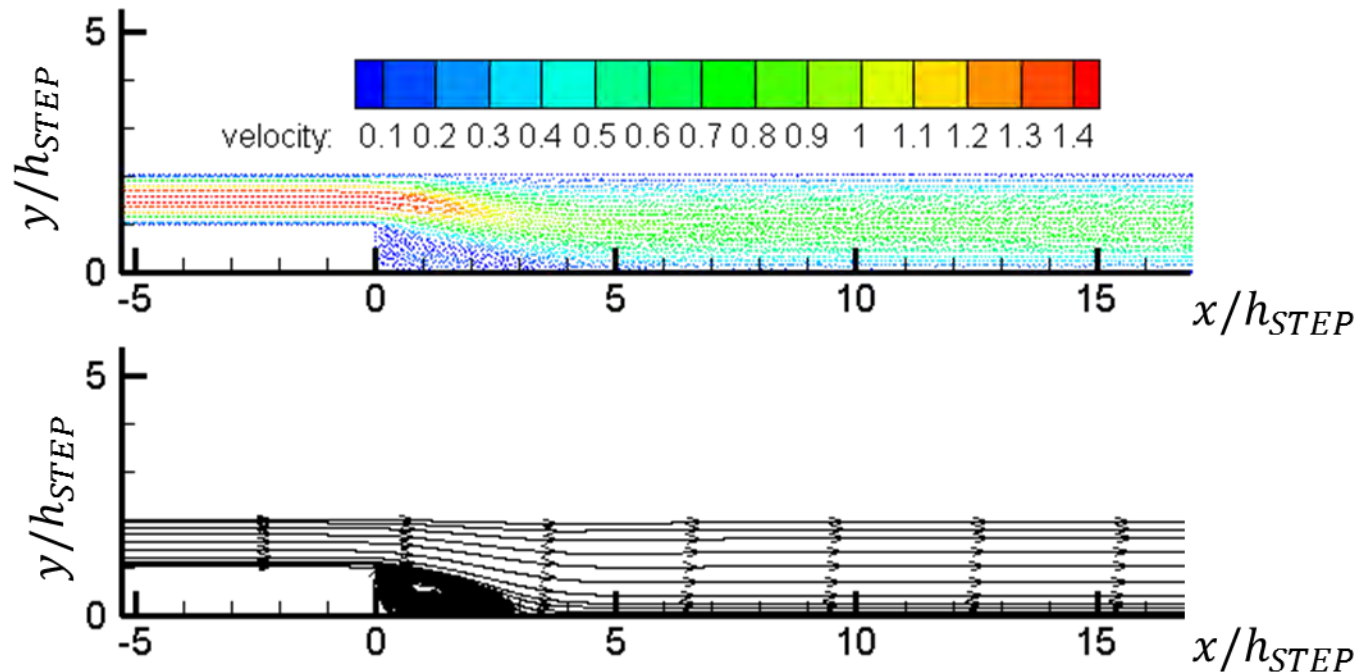


Re	100	389
L_f	$16h_{IN}$	$25h_{IN}$
N_x	210	300
N_y	20	20
Δt	0.02	0.02

Results: Verification Studies

▶ Backward Facing Step Flow

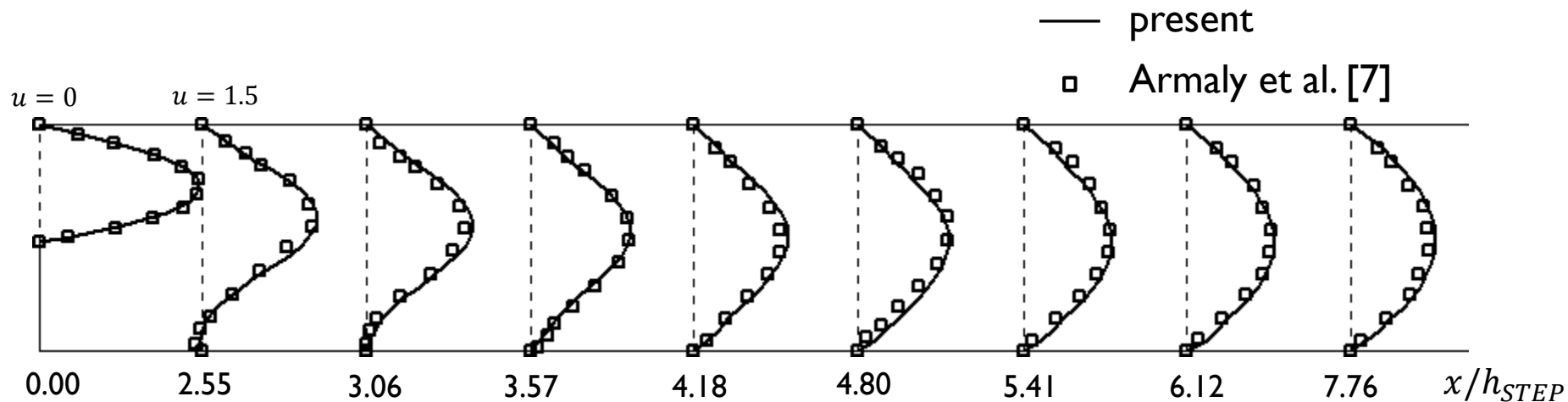
Re 100



Results: Verification Studies

► Backward Facing Step Flow

Re 100

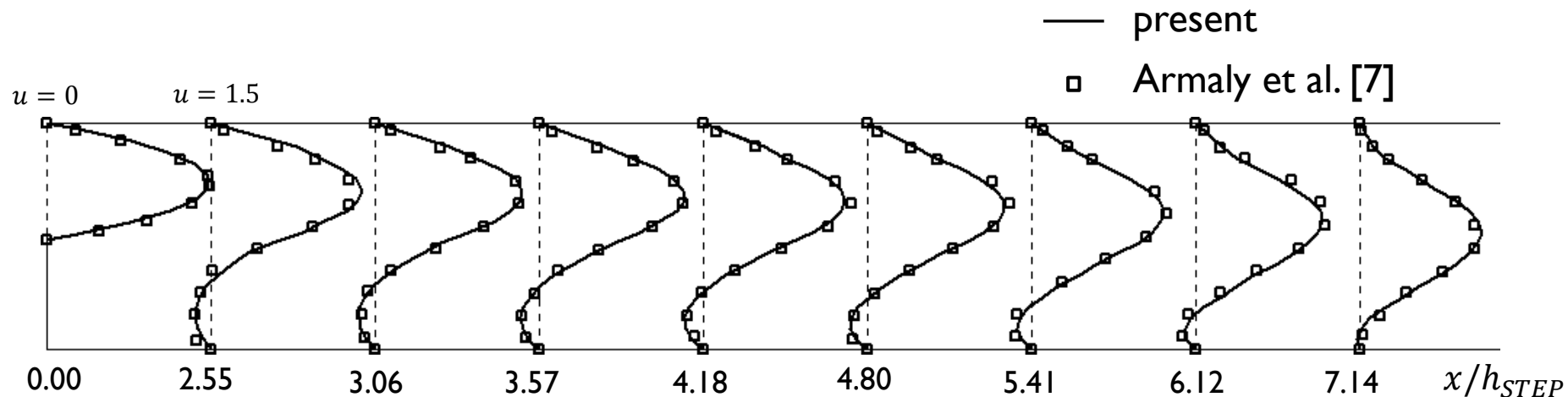


[7] B.F. Armaly, F. Durst, J.C. F. Pereira, B. Schoung, Experimental and theoretical investigation of backward-facing step flow, *J. Fluid. Mech.*, vol. 127, 473-496, 1983

Results: Verification Studies

► Backward Facing Step Flow

Re 389



[7] B.F. Armaly, F. Durst, J.C. F. Pereira, B. Schoung, Experimental and theoretical investigation of backward-facing step flow, *J. Fluid. Mech.*, vol. 127, 473-496, 1983

Conclusion

- ▶ The shape parameter ε of RBF-MQ is to control the correction to the base function.
- ▶ The optimum value of ε is problem dependent, hence it is subjected to the local distribution of the function to be interpolated.
- ▶ In present implementation, the value of ε is predicted using the surrounding information of the interpolation domain to achieves accurate approximation. It is found that the interpolation accuracy, as well as rate of convergence is improved with the appropriate ε .





Thank you

