Why don't we do it on the lattice

From particles to lattice and back

Daniel Duque [1] and Pep Español [2]

[1]CEHINAV (Hydrodynamic Research Basin), ETSIN (Naval Engineers), UPM (Technical University of Madrid, Spain)
[2]Dep. Física Fundamental, UNED (National Spanish Distance Learning University)

Iberian SPH Meeting 2015

Table of Contents

Motivation

Summary

- In CFD, particle-based methods take care of convection
- The price to pay is that a mesh is hard to define
- So, can't we somehow project onto a lattice, do our things there, then back?
- Numerics: much numerical work (e.g. decomposition) can be done at the beginning of the simulation, then used all over, perhaps even save it for future simulations
- Attribution: Dr. Monaghan, SPH meeting 2015, who called it "embedded particle". Then pFEM-2 actually follows this idea
- Results are relevant for any remeshing: particle splitting and merging, field smoothing . . .

Motivation

Summary

- In CFD, particle-based methods take care of convection
- The price to pay is that a mesh is hard to define
- So, can't we somehow project onto a lattice, do our things there, then back?
- Numerics: much numerical work (e.g. decomposition) can be done at the beginning of the simulation, then used all over, perhaps even save it for future simulations
- Attribution: Dr. Monaghan, SPH meeting 2015, who called it "embedded particle". Then pFEM-2 actually follows this idea
- Results are relevant for any remeshing: particle splitting and merging, field smoothing . . .

Motivation

Summary

- In CFD, particle-based methods take care of convection
- The price to pay is that a mesh is hard to define
- So, can't we somehow project onto a lattice, do our things there, then back?
- Numerics: much numerical work (e.g. decomposition) can be done at the beginning of the simulation, then used all over, perhaps even save it for future simulations
- Attribution: Dr. Monaghan, SPH meeting 2015, who called it "embedded particle". Then pFEM-2 actually follows this idea
- Results are relevant for any remeshing: particle splitting and merging, field smoothing ...

Motivation

Summary

- In CFD, particle-based methods take care of convection
- The price to pay is that a mesh is hard to define
- So, can't we somehow project onto a lattice, do our things there, then back?
- Numerics: much numerical work (e.g. decomposition) can be done at the beginning of the simulation, then used all over, perhaps even save it for future simulations
- Attribution: Dr. Monaghan, SPH meeting 2015, who called it "embedded particle". Then pFEM-2 actually follows this idea
- Results are relevant for any remeshing: particle splitting and merging, field smoothing . . .

Motivation

Summary

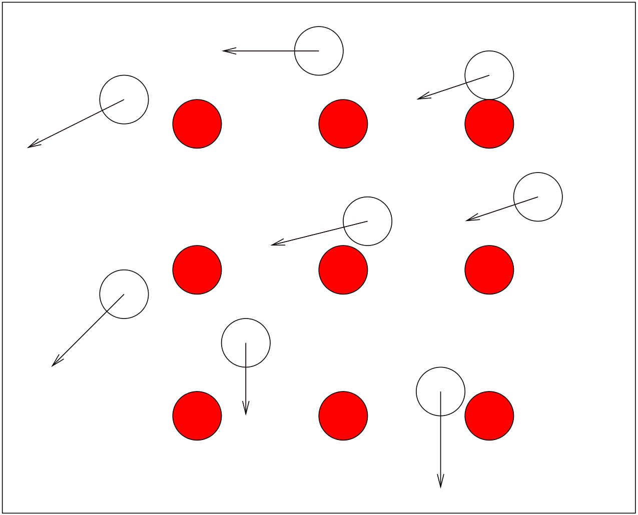
- In CFD, particle-based methods take care of convection
- The price to pay is that a mesh is hard to define
- So, can't we somehow project onto a lattice, do our things there, then back?
- Numerics: much numerical work (e.g. decomposition) can be done at the beginning of the simulation, then used all over, perhaps even save it for future simulations
- Attribution: Dr. Monaghan, SPH meeting 2015, who called it "embedded particle". Then pFEM-2 actually follows this idea
- Results are relevant for any remeshing: particle splitting and merging, field smoothing ...

Motivation

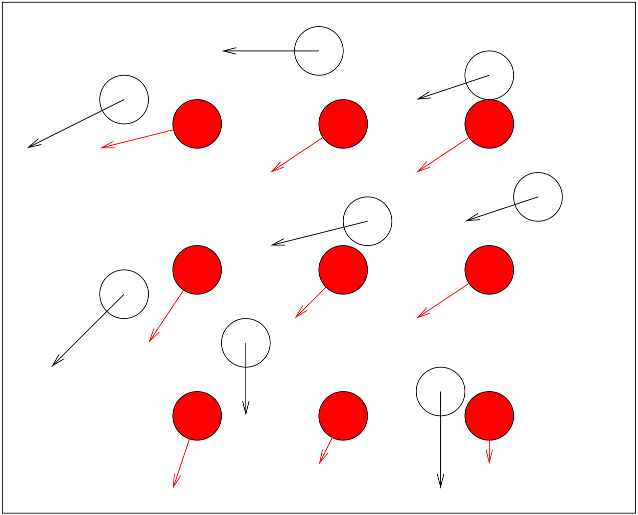
Summary

- In CFD, particle-based methods take care of convection
- The price to pay is that a mesh is hard to define
- So, can't we somehow project onto a lattice, do our things there, then back?
- Numerics: much numerical work (e.g. decomposition) can be done at the beginning of the simulation, then used all over, perhaps even save it for future simulations
- Attribution: Dr. Monaghan, SPH meeting 2015, who called it "embedded particle". Then pFEM-2 actually follows this idea
- Results are relevant for any remeshing: particle splitting and merging, field smoothing ...

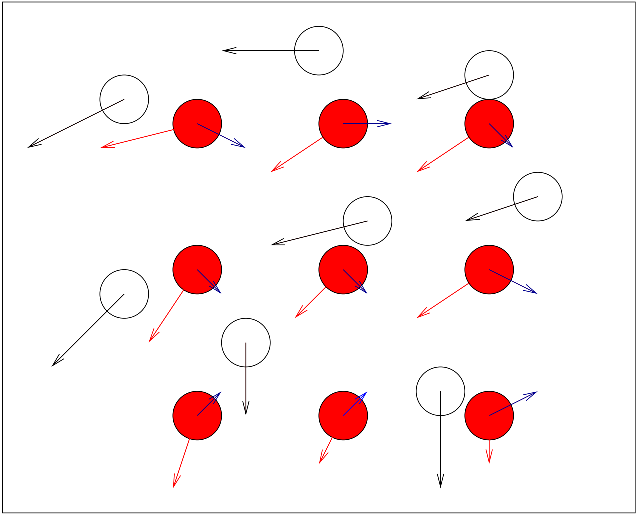
The idea



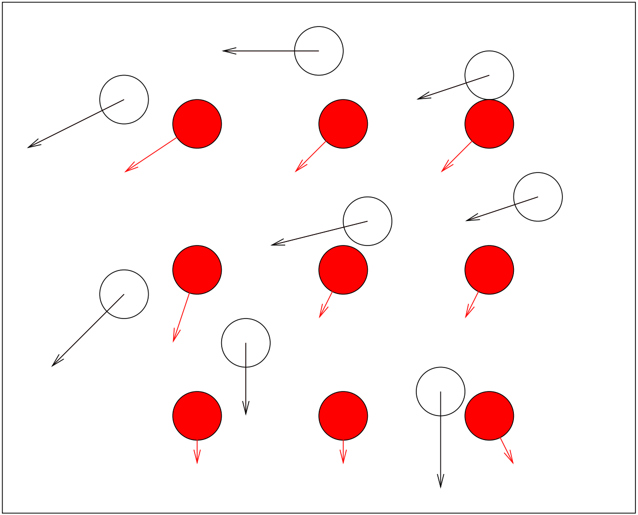
The idea



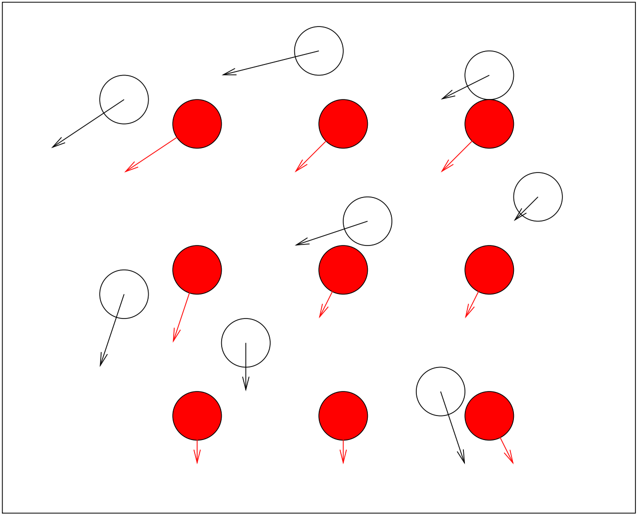
The idea



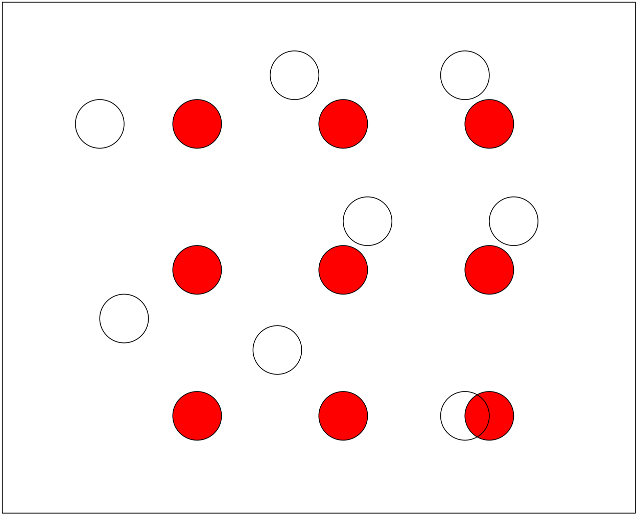
The idea



The idea



The idea



Projecting from the particles

Definition

The particles move about, so we want to interpolate values of fields onto the lattice nodes

This may be achieved with particle basis functions (I know, this usually still requires a mesh)

SPH shape functions may be tried (they must!), but for this talk I'm using p-FEM functions, and an "quad" extension of them

Projecting from the particles

Definition

The particles move about, so we want to interpolate values of fields onto the lattice nodes

This may be achieved with particle basis functions (I know, this usually still requires a mesh)

SPH shape functions may be tried (they must!), but for this talk I'm using p-FEM functions, and an "quad" extension of them

Projecting from the particles

Definition

The particles move about, so we want to interpolate values of fields onto the lattice nodes
This may be achieved with particle basis functions (I know, this usually still requires a mesh)
SPH shape functions may be tried (they must!), but for this talk I'm using p-FEM functions, and an "quad" extension of them

FEM functions

FEM shape functions can be built on particle arrangements at every time step, on the Delaunay triangulation (the dual of the Voronoi diagram)

These will interpolate “linearly” between nodes

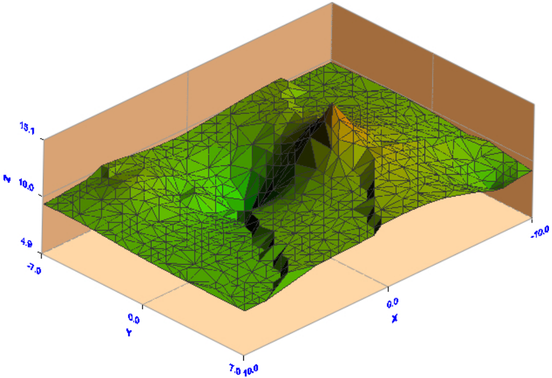


Figure : Taken from graphnow

FEM functions

FEM shape functions can be built on particle arrangements at every time step, on the Delaunay triangulation (the dual of the Voronoi diagram)
These will interpolate “linearly” between nodes

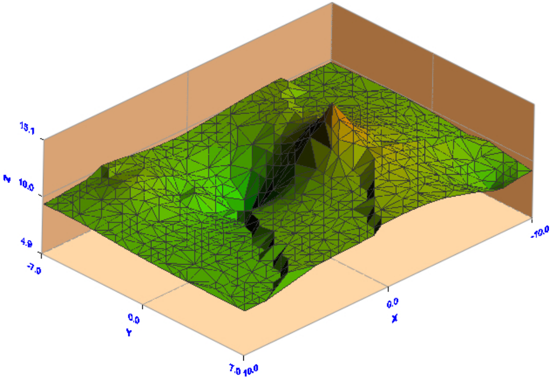


Figure : Taken from graphnow

1D results

In 1D, FEM means just linear interpolation

Let's try our idea, computing the Laplacian of a sine function (periodic b.c.s)

Results are good for the Poisson problem: $h''(x) = f(x)$ given f

Results are not too good for the direct problem: $g(x) = f''(x)$ given f

1D results

In 1D, FEM means just linear interpolation

Let's try our idea, computing the Laplacian of a sine function (periodic b.c.s)

Results are good for the Poisson problem: $h''(x) = f(x)$ given f

Results are not too good for the direct problem: $g(x) = f''(x)$ given f

1D results

In 1D, FEM means just linear interpolation
Let's try our idea, computing the Laplacian of a sine function (periodic b.c.s)
Results are good for the Poisson problem: $h''(x) = f(x)$ given f
Results are not too good for the direct problem: $g(x) = f''(x)$ given f

1D results

In 1D, FEM means just linear interpolation
Let's try our idea, computing the Laplacian of a sine function (periodic b.c.s)
Results are good for the Poisson problem: $h''(x) = f(x)$ given f
Results are not too good for the direct problem: $g(x) = f''(x)$ given f

1D results, FEM — second derivatives

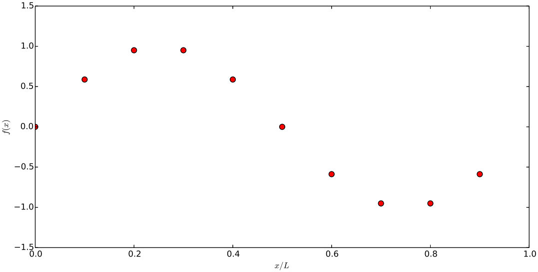


Figure : Original function $f(x)$ on particles

1D results, FEM — second derivatives

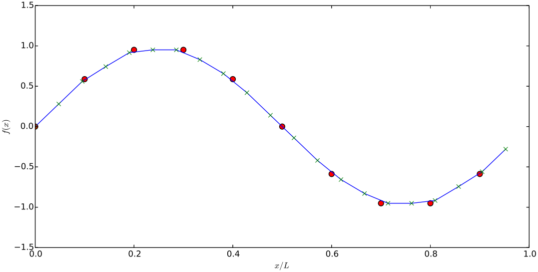


Figure : Function $f(x)$ onto lattice

1D results, FEM — second derivatives

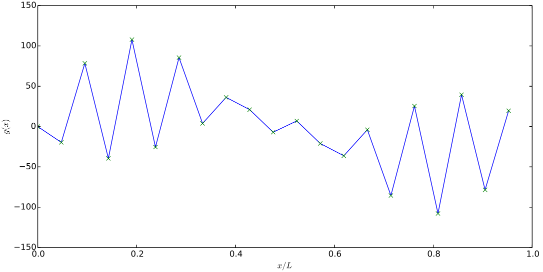


Figure : Second derivative $g(x) = f''(x)$ in lattice

1D results, FEM — second derivatives

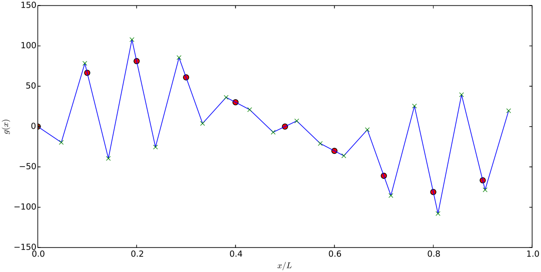


Figure : Second derivative $g(x) = f''(x)$ back on particles

1D results, FEM — second derivatives

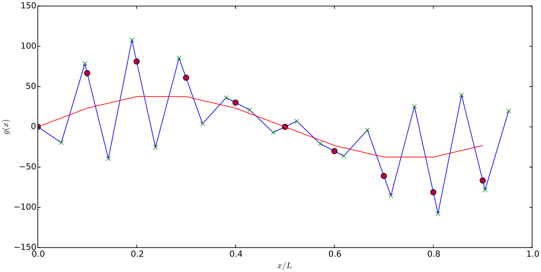


Figure : Second derivative $g(x) = f''(x)$, exact result

Quadratic interpolation

Projection from particles to the lattice seems to be the main culprit

We hereby introduce our (Pep Español, de la Torre, myself) procedure to go from linear to quadratic (sent to journal):

$$\psi_i(\mathbf{r}) = \phi_i(\mathbf{r}) + \sum_{j,k} A_{ijk}\phi_j(\mathbf{r})\phi_k(\mathbf{r})$$

Results are better now

Quadratic interpolation

Projection from particles to the lattice seems to be the main culprit
We hereby introduce our (Pep Español, de la Torre, myself) procedure to go from linear to quadratic (sent to journal):

$$\psi_i(\mathbf{r}) = \phi_i(\mathbf{r}) + \sum_{j,k} A_{ijk}\phi_j(\mathbf{r})\phi_k(\mathbf{r})$$

Results are better now

Quadratic interpolation

Projection from particles to the lattice seems to be the main culprit
We hereby introduce our (Pep Español, de la Torre, myself) procedure to go from linear to quadratic (sent to journal):

$$\psi_i(\mathbf{r}) = \phi_i(\mathbf{r}) + \sum_{j,k} A_{ijk}\phi_j(\mathbf{r})\phi_k(\mathbf{r})$$

Results are better now

1D results — two bases

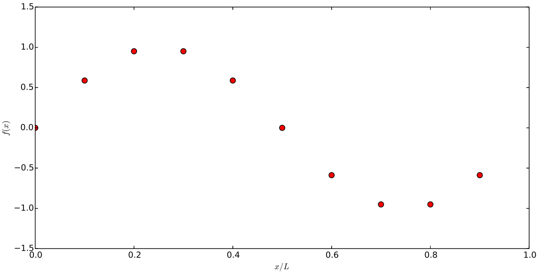


Figure : Original function $f(x)$ on particles

1D results — two bases

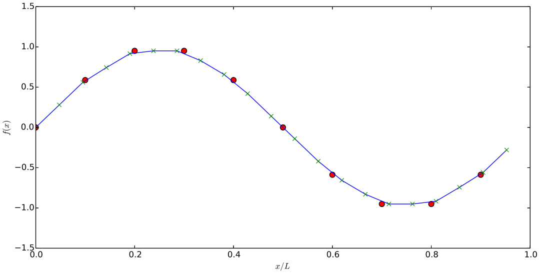


Figure : FEM

1D results — two bases

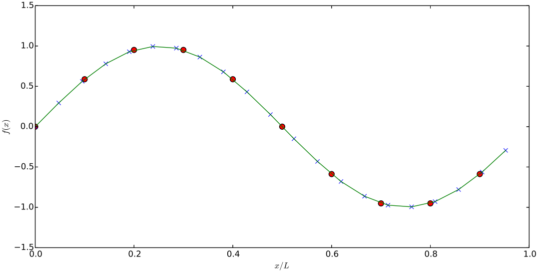


Figure : quad

1D results — two bases

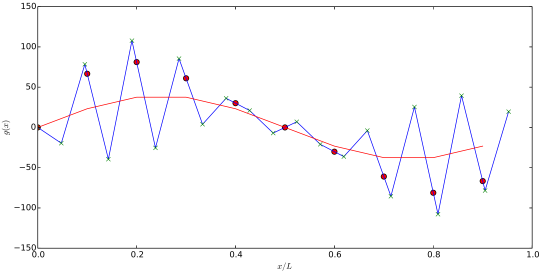


Figure : FEM

1D results — two bases

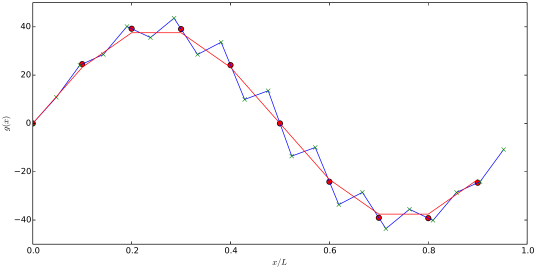
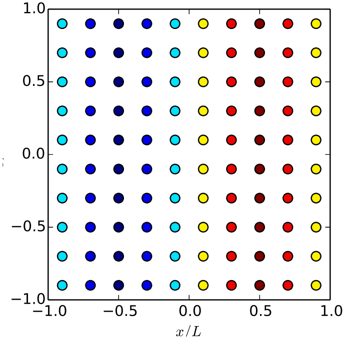


Figure : quad

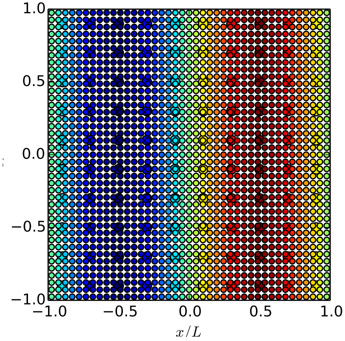
Going to 2D

The whole procedure generalizes to 2D in a straight manner (it would to 3D too). Let's try $f(x, y) = \sin(\pi x)$.



Going to 2D

The whole procedure generalizes to 2D in a straight manner (it would to 3D too). Let's try $f(x, y) = \sin(\pi x)$.



Going to 2D

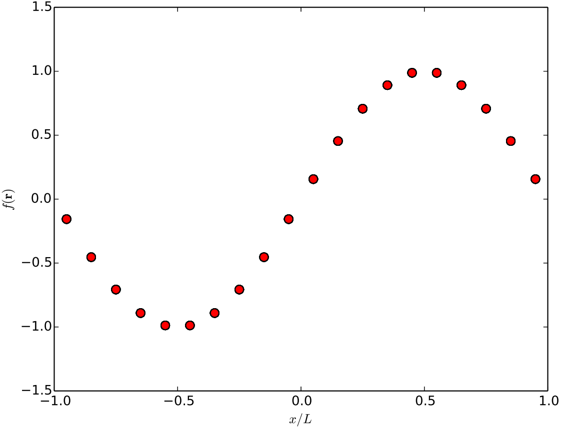


Figure : Original function $f(x)$ on particles

Going to 2D

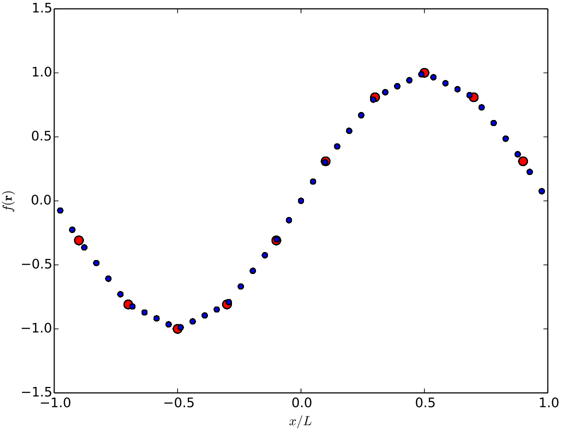


Figure : FEM

Going to 2D

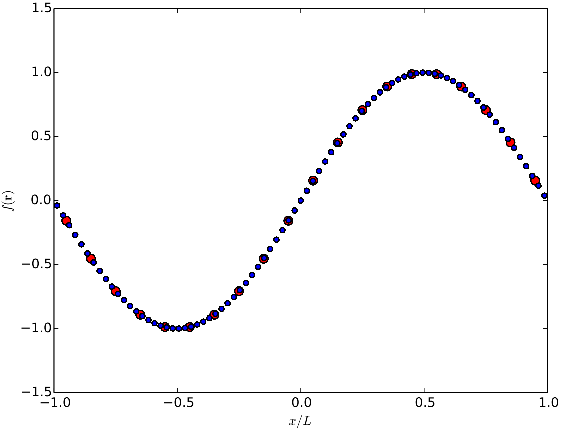


Figure : quad

Taylor-Green vortex sheet

Navier-Stokes for an incompressible fluid:

$$\frac{d\mathbf{u}}{dt} = -\nabla(p/\rho) + \nu\nabla^2\mathbf{u} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

Taylor-Green solution:

$$u_x = A(t)\sin(\pi x)\cos(\pi y) \tag{3}$$

$$u_y = -A(t)\cos(\pi x)\sin(\pi y) \tag{4}$$

$$A(t) = u_0 \exp(-2\pi^2 \nu t) \tag{5}$$

$$p = \frac{1}{4}A(t)^2\left(\cos(2\pi x) + \cos(2\pi y)\right) \tag{6}$$

Taylor-Green vortex sheet

Navier-Stokes for an incompressible fluid:

$$\frac{d\mathbf{u}}{dt} = -\nabla(p/\rho) + \nu\nabla^2\mathbf{u} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

Taylor-Green solution:

$$u_x = A(t)\sin(\pi x)\cos(\pi y) \quad (3)$$

$$u_y = -A(t)\cos(\pi x)\sin(\pi y) \quad (4)$$

$$A(t) = u_0 \exp(-2\pi^2 \nu t) \quad (5)$$

$$p = \frac{1}{4}A(t)^2\left(\cos(2\pi x) + \cos(2\pi y)\right) \quad (6)$$

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla\cdot\mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla \cdot \mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla \cdot \mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla \cdot \mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla \cdot \mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla \cdot \mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla \cdot \mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

Numerical procedure

1. Set up initial conditions
2. Move particles according to $\mathbf{u}_t$
3. Project onto lattice
4. Compute $\mathbf{u}^* = \mathbf{u}_t + (dt)\nu\nabla^2\mathbf{u}_t$
5. Solve PPE $\nabla^2(p/\rho) = 1/(dt)\nabla\cdot\mathbf{u}^*$
6. Compute $\mathbf{u}_{t+1} = \mathbf{u}^* - (dt)\nabla(p/\rho)$
7. Project back onto particles
8. Go to 2

TG vortices – movie

- lattice vs particles ▸ Link
- quad vs FEM ▸ Link
- quad vs pFEM ▸ Link
- Everything looks "nice", but we need to quantify convergence!

TG vortices – movie

- lattice vs particles ▸ Link
- quad vs FEM ▸ Link
- quad vs pFEM ▸ Link
- Everything looks “nice”, but we need to quantify convergence!

TG vortices – movie

- lattice vs particles ▸ Link
- quad vs FEM ▸ Link
- quad vs pFEM ▸ Link
- Everything looks "nice", but we need to quantify convergence!

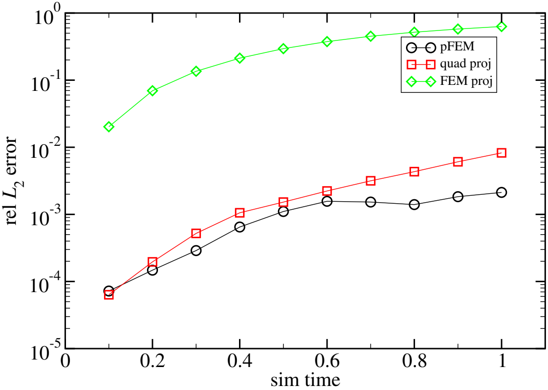
TG vortices – movie

- lattice vs particles ▸ Link
- quad vs FEM ▸ Link
- quad vs pFEM ▸ Link
- Everything looks "nice", but we need to quantify convergence!

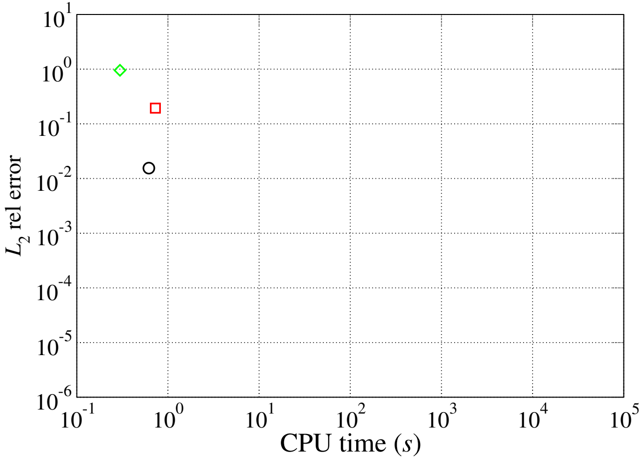
Convergence analysis

Since we know the exact $\mathbf{u} = \bar{\mathbf{u}}$:

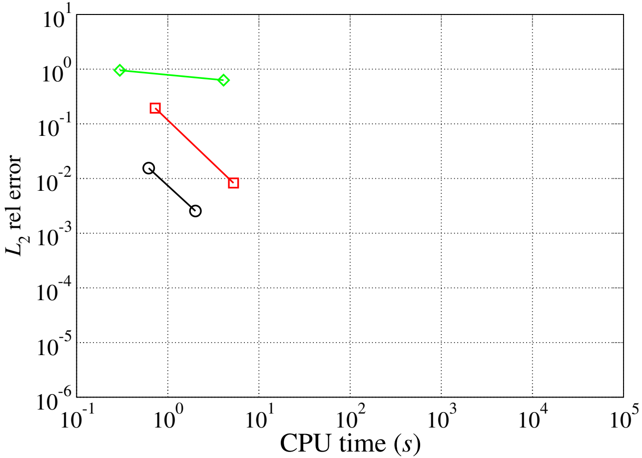
$$L_2 := \frac{\sum_i |\bar{\mathbf{u}}_i - \mathbf{u}_i|^2}{\sum_i |\bar{\mathbf{u}}_i|^2}$$



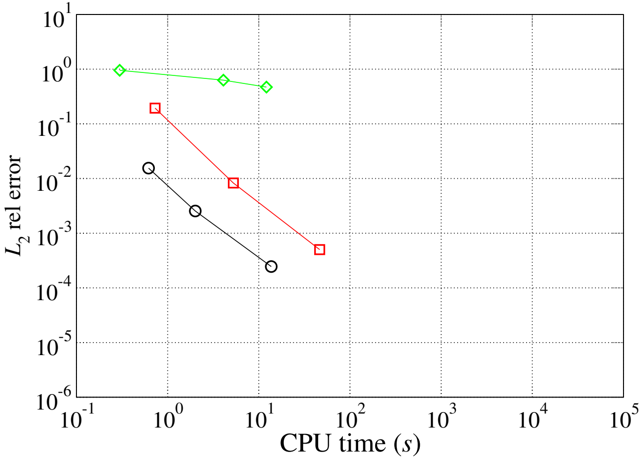
Performance analysis



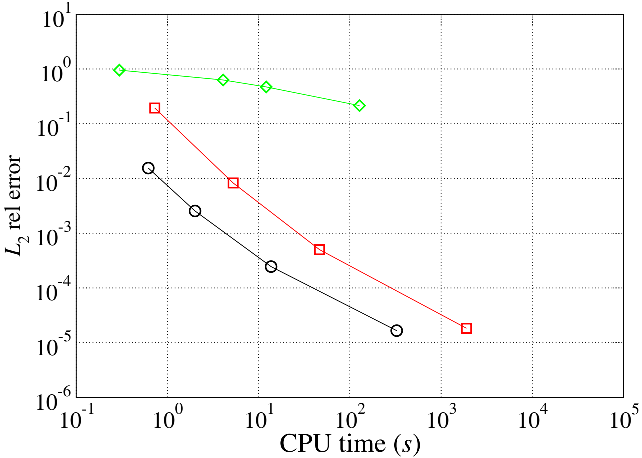
Performance analysis



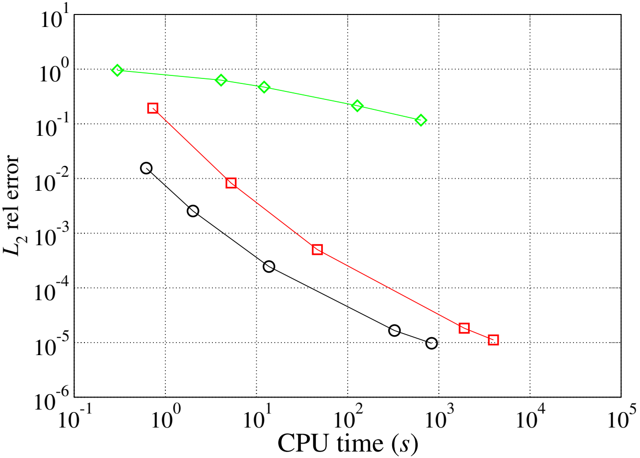
Performance analysis



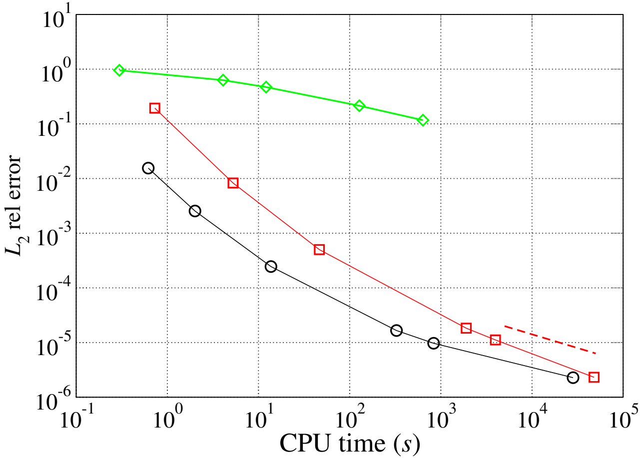
Performance analysis



Performance analysis



Performance analysis



Conclusions

- The idea of projecting the particles' data seems to be viable for large enough systems
- As long as the interpolation from particles to lattice is good !
- The application to explicit integration is an open question
- As is the treatment of the free surface !

Conclusions

- The idea of projecting the particles' data seems to be viable for large enough systems
- As long as the interpolation from particles to lattice is good !
- The application to explicit integration is an open question
- As is the treatment of the free surface !

Conclusions

- The idea of projecting the particles' data seems to be viable for large enough systems
- As long as the interpolation from particles to lattice is good !
- The application to explicit integration is an open question
- As is the treatment of the free surface !

Conclusions

- The idea of projecting the particles' data seems to be viable for large enough systems
- As long as the interpolation from particles to lattice is good !
- The application to explicit integration is an open question
- As is the treatment of the free surface !

Conclusions

Thanks

For the audience and the organizers

Research has received funding from the Spanish Ministry for Science and Innovation under grant TRA2013-41096-P "Optimización del transporte de gas licuado en buques LNG mediante estudios sobre interacción fluido-estructura."

ASME INTERNATIONAL PETROLEUM TECHNOLOGY INSTITUTE

37th International Conference on Ocean, Of shore and Arctic Engineering (OMAE 2018)

Madrid, Spain

June 3-8, 2018

UNIVERSITAT ROVIRA I VIRGILI

IH cantabria INSTITUTO DE HIDRÁULICA AMBIENTAL UNIVERSIDAD DE CANTABRIA

POLITÉCNICA

VIDEOS

https://youtu.be/bVhnFxMANa0

https://youtu.be/r0LWsaKLjqQ

https://youtu.be/P3SsOS7oVtI