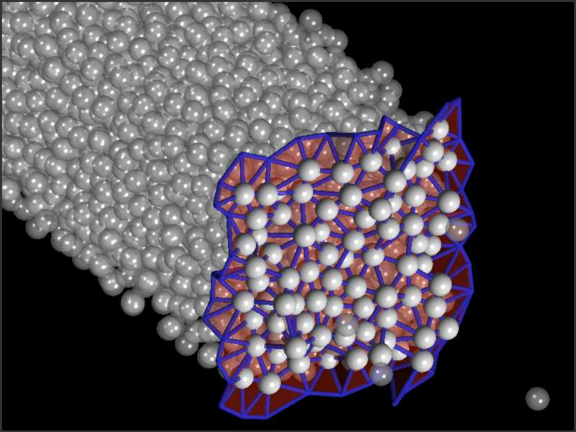
# Statistical mechanics in computational geometry

Daniel Duque
ETSIN, UPM

June 21, 2010

# Motivation

- Notice: in the past, we have applied computational geometry (CG) to statistical mechanics (SM). I.e. the $\alpha$-shapes.
- Today, I'll describe methods *from* SM applied *to* CG.



# The problem

- When discretizing the continuum equations of hydrodynamics using nodes ("particles") one may obtain equations of motion. Direct connection: *Molecular dynamics* No SM, really
- (Actually, interesting SM in the mesoscopic, Brownian regime, but that's not our topic today.)
- How to discretize the equations for some given particles? Some unexpected SM.

# Laws: Navier-Stokes equations

We would be happy to compute these two accurately:
Continuity:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \rho \mathbf{v}$$

Momentum density $\rho v_\alpha$, for each coordinate $\alpha = 1, 2, 3$:

$$\frac{\partial \rho v_\alpha}{\partial t} = -\nabla \cdot \rho v_\alpha \mathbf{v} - \frac{\partial p}{\partial x_\alpha} + \mu \nabla^2 v_\alpha$$

# Euler's view

Use a grid, and finite differences (or finite elements) for equations that are written in this "frame".

# Euler's view

Use a grid, and finite differences (or finite elements) for equations that are written in this "frame".
Notice "particles" are fixed here, are just space nodes. There is usually lots of freedom in choosing them (and refining them).

SERG, U. Sheffield

# Limitations of fixed meshes

Problems arise in many situations.

- When one does not know in advance where more effort (CPU and RAM) will be needed (turbulence, astrophysics. . . )

# Limitations of fixed meshes

Problems arise in many situations.

- When one does not know in advance where more effort (CPU and RAM) will be needed (turbulence, astrophysics...)
- When the boundary is also moving, "free boundary problems". E.g. surface waves (what people call "waves"!). (Spanish: ola $\neq$ onda). I.e. almost *always* in naval engineering.

# Limitations of fixed meshes

Problems arise in many situations.

- When one does not know in advance where more effort (CPU and RAM) will be needed (turbulence, astrophysics. . . )
- When the boundary is also moving, "free boundary problems". E.g. surface waves (what people call "waves"!). (Spanish: ola $\neq$ onda). I.e. almost *always* in naval engineering.

This calls for a *Lagrangian* approach, and computational methods that will be either *meshless* or have a *moving mesh*.

# Lagrange's view

No need to go into details today. In a nutshell, "particles" are defined that follow the flow (pathlines).

# Lagrange's view

No need to go into details today. In a nutshell, "particles" are defined that follow the flow (pathlines).
In this frame, the equations simplify remarkably. Now, we would be happy just with these.
Lagrangian particles:

$$\frac{D\mathbf{r}}{Dt} = \mathbf{v}$$

Continuity:

$$\frac{DM}{Dt} = 0$$

Momentum:

$$\frac{M}{V}\frac{D\mathbf{v}}{Dt} = -\nabla p + \mu\nabla^2\mathbf{v}.$$

# Computing with fluid particles

Given a set of particles with

- positions $\{\mathbf{r}_a\}$,
- velocities $\{\mathbf{v}_a\}$,
- and pressures $\{p_a\}$

# Computing with fluid particles

Given a set of particles with

- positions $\{\mathbf{r}_a\}$,
- velocities $\{\mathbf{v}_a\}$,
- and pressures $\{p_a\}$

$$\frac{D\mathbf{r}_a}{Dt} = \mathbf{v}_a$$

$$\frac{DM_a}{Dt} = 0$$

$$\frac{M_a}{V_a}\frac{D\mathbf{v}_a}{Dt} = -(\nabla p)_a + \mu(\nabla^2 \mathbf{v})_a.$$

# Computing with fluid particles

So, we need to provide expressions for:

- particles' volumes $\{V_a\}$
- pressure gradients $\{(\nabla p)_a\}$
- velocity Laplacians $\{(\nabla^2 \mathbf{v})_a\}$

# Anyway!

Mathematically, our problem boils down to:

- given the values of a function $u(x)$ on a set of nodes $\{x_a\}$, $\{u_a\}$

# Anyway!

Mathematically, our problem boils down to:

- given the values of a function $u(x)$ on a set of nodes $\{x_a\}$, $\{u_a\}$
- find the derivatives at the nodes: $\{\nabla u\}_a$, $\{\nabla^2 u\}_a$, etc.

# Anyway!

Mathematically, our problem boils down to:

- given the values of a function $u(x)$ on a set of nodes $\{x_a\}$, $\{u_a\}$
- find the derivatives at the nodes: $\{\nabla u\}_a$, $\{\nabla^2 u\}_a$, etc.
- we are not alone: many people are considering related problems in the field of imaging. E.g. reconstruction: $\{u_a\} \rightarrow u(x)$. Sometimes even $\{u_a\} \rightarrow \nabla^2 u(x)$ (edge detection).

# Anyway!

Mathematically, our problem boils down to:

- given the values of a function $u(x)$ on a set of nodes $\{x_a\}$, $\{u_a\}$
- find the derivatives at the nodes: $\{\nabla u\}_a$, $\{\nabla^2 u\}_a$, etc.
- we are not alone: many people are considering related problems in the field of imaging. E.g. reconstruction: $\{u_a\} \rightarrow u(x)$. Sometimes even $\{u_a\} \rightarrow \nabla^2 u(x)$ (edge detection).
- also: we are unable to choose "nice" nodes. Even if we do, they will move around, to who knows where (This is a huge difference with the fixed grid community, such as FEMs).

# Anyway!

Mathematically, our problem boils down to:

- given the values of a function $u(x)$ on a set of nodes $\{x_a\}$, $\{u_a\}$
- find the derivatives at the nodes: $\{\nabla u\}_a$, $\{\nabla^2 u\}_a$, etc.
- we are not alone: many people are considering related problems in the field of imaging. E.g. reconstruction: $\{u_a\} \to u(x)$. Sometimes even $\{u_a\} \to \nabla^2 u(x)$ (edge detection).
- also: we are unable to choose "nice" nodes. Even if we do, they will move around, to who knows where (This is a huge difference with the fixed grid community, such as FEMs).

The standard approach is to introduce a set of weight functions $\{p_a(x)\}$, from which:

$$u(x) = \sum_a u_a p_a(x) \qquad \nabla u(x) = \sum_a u_a \nabla p_a(x) \qquad \ldots$$

(Actually, some quadrature may be needed on top of this.)

# Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)

# Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)
- $\boxed{\text{0-cons}}$ $\sum_a p_a = 1$

## Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)
- $\boxed{\text{0-cons}}$ $\sum_a p_a = 1$
- $\boxed{\text{1-cons}}$ $\sum_a p_a x_a = x$

# Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)
- $\boxed{\text{0-cons}}$ $\sum_a p_a = 1$
- $\boxed{\text{1-cons}}$ $\sum_a p_a x_a = x$
- $\boxed{\text{2-cons}}$ $\sum_a p_a x_a^2 = x^2$

# Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)
- $\boxed{\text{0-cons}}$ $\sum_a p_a = 1$
- $\boxed{\text{1-cons}}$ $\sum_a p_a x_a = x$
- $\boxed{\text{2-cons}}$ $\sum_a p_a x_a^2 = x^2$
- $\boxed{\text{local}}$

# Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)
- $\boxed{\text{0-cons}}$ $\sum_a p_a = 1$
- $\boxed{\text{1-cons}}$ $\sum_a p_a x_a = x$
- $\boxed{\text{2-cons}}$ $\sum_a p_a x_a^2 = x^2$
- $\boxed{\text{local}}$
- $\boxed{\text{mesh-free}}$

# Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)
- $\boxed{\text{0-cons}}$ $\sum_a p_a = 1$
- $\boxed{\text{1-cons}}$ $\sum_a p_a x_a = x$
- $\boxed{\text{2-cons}}$ $\sum_a p_a x_a^2 = x^2$
- $\boxed{\text{local}}$
- $\boxed{\text{mesh-free}}$
- $\boxed{\text{boundary}}$

# Desired features

- $\boxed{\text{posit}}$ $p_a > 0$ (all of today's talk, but MLS)
- $\boxed{\text{0-cons}}$ $\sum_a p_a = 1$
- $\boxed{\text{1-cons}}$ $\sum_a p_a x_a = x$
- $\boxed{\text{2-cons}}$ $\sum_a p_a x_a^2 = x^2$
- $\boxed{\text{local}}$
- $\boxed{\text{mesh-free}}$
- $\boxed{\text{boundary}}$

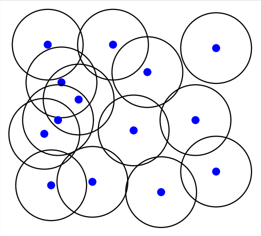
We would like these be satisfied, in some sense, at least. In order of niceness:

- Identically
- For lots of particles ($O(N)$ effort)
- For finer resolution ($O(N^2)$ effort, or worse)

# Popular choices: SPH

$$p_a = \frac{1}{C} \exp[-\beta(x - x_a)^2]$$

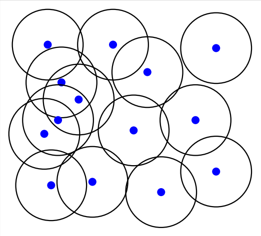
| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |



# Popular choices: SPH

$$p_a = \frac{1}{C} \exp[-\beta(x - x_a)^2]$$

| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |



- Quite bad consistency (needs resolution increase to converge), but *very* easy to implement
- Uses: fluids with interfaces, both in science and in animation industry (films, commercials...). Next Limit's Real Flow: LotR, Avatar, Charlie and the Chocolate, Ice Age 1-3,

# Popular choices: SPH-Shepard

$$s_a = \exp[-\beta(x - x_a)^2]$$

$$p_a = \frac{s_a}{Z}$$

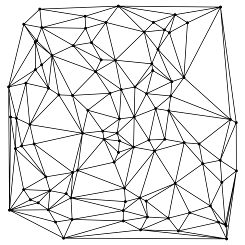
$$Z = \sum_a s_a$$

| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |

- 0-consistency by construction, the rest is as bad as SPH.

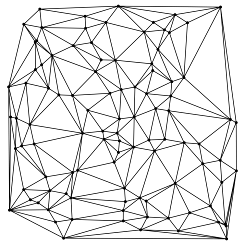
# Popular choices: FEM

$p_a$ = finite elements on Delaunay mesh



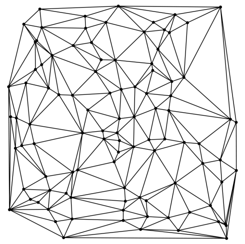
# Popular choices: FEM

$$p_a = \text{finite elements on Delaunay mesh}$$



- The very famous Finite Element Method of engineering
- Notice the elements are triangles in 1D, pyramids in 2D, but in the later there are many possible triangulations on which to build them
- The Delaunay lattice is the "best" in many ways: it is the one with more open angles. Plus, its dual is the Voronoi tesselation.

# FEM



| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |

# Popular choices: MLS

$$p_a = f_a(x)\exp[-\beta(x - x_a)^2]$$

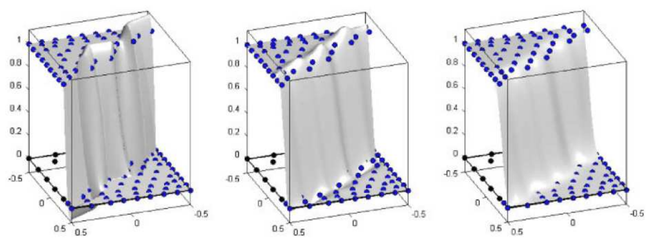
| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |

# Popular choices: MLS

$$p_a = f_a(x)\exp[-\beta(x - x_a)^2]$$

| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |

- Exact consistencies by construction.
- But notice: serious stability issues due to fitting.
- In fact, it violates positivity, $p_a < 0$ sometimes.



# Our little experience with FEM and MLS

FEM seemed really the way to go. But! it *never* converges to the proper $\nabla^2$! (on arbitrary point sets).

# Our little experience with FEM and MLS

FEM seemed really the way to go. But! it *never* converges to the proper $\nabla^2$! (on arbitrary point sets).
This was quite a shock, it being such stablished in engineering.

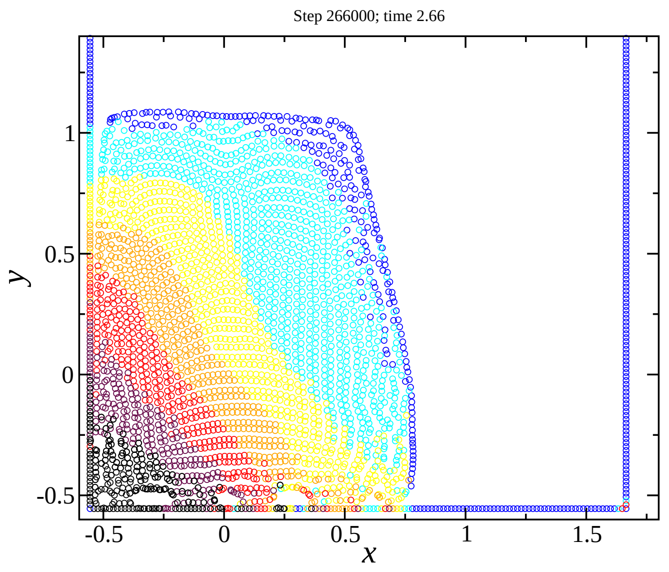
# Our little experience with FEM and MLS

FEM seemed really the way to go. But! it *never* converges to the proper $\nabla^2$! (on arbitrary point sets).
This was quite a shock, it being such stablished in engineering.
Then, we (re)discovered MLS. It did converge, but when used it is very unstable. Beware of overfitting!

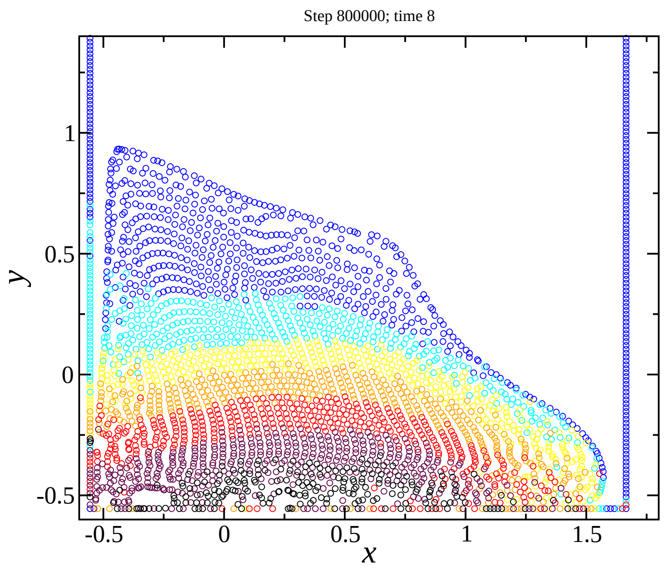
# Our little experience with FEM and MLS

FEM seemed really the way to go. But! it *never* converges to the proper $\nabla^2$! (on arbitrary point sets).
This was quite a shock, it being such stablished in engineering.
Then, we (re)discovered MLS. It did converge, but when used it is very unstable. Beware of overfitting!
All in all, the FEM can already be used for reasonable results.

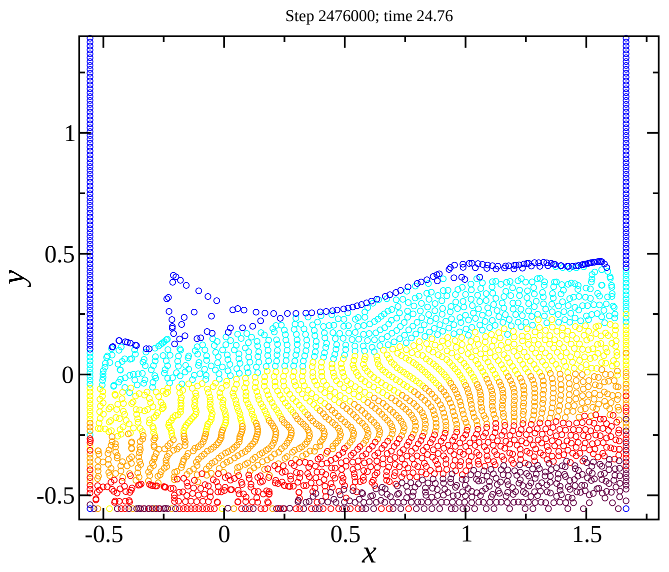
# FEM results: dam break



# FEM results: dam break



# FEM results: dam break



# Introducing: LME

A Copernican revolution:

- instead of $p_a(x) \longrightarrow u(x) \longrightarrow \nabla u$

# Introducing: LME

A Copernican revolution:

- instead of $p_a(x) \longrightarrow u(x) \longrightarrow \nabla u$
- Rather: fix $x$, find $\{p_a\}$ with the desired properties.

# Introducing: LME

A Copernican revolution:

- instead of $p_a(x) \to u(x) \to \nabla u$
- Rather: fix $x$, find $\{p_a\}$ with the desired properties.
- E.g. find $\{p_a\}$ such that $\sum p_a(x) = 1$, $\sum p_a(x - x_a) = 0$, plus of course more requirements. Perhaps some variational conditions (min or max something?)

# Introducing: LME

A Copernican revolution:

- instead of $p_a(x) \to u(x) \to \nabla u$
- Rather: fix $x$, find $\{p_a\}$ with the desired properties.
- E.g. find $\{p_a\}$ such that $\sum p_a(x) = 1$, $\sum p_a(x - x_a) = 0$, plus of course more requirements. Perhaps some variational conditions (min or max something?)
- Notice: no functional form to the $p_a$. This is both a source of freedom and a computational nuisance.

# Previous hint: Rajan

- Impose $\sum p_a(x) = 1$, $\sum p_a(x - x_a) = 0$.

# Previous hint: Rajan

- Impose $\sum p_a(x) = 1$, $\sum p_a(x - x_a) = 0$.
- Consider the squared weighted distance from $x$ to all $\{x_a\}$: $U = \sum p_a(x - x_a)^2$. (Notice: all values $\{x - x_a\}$ are fixed: it's the weights we can play with).

# Previous hint: Rajan

- Impose $\sum p_a(x) = 1$, $\sum p_a(x - x_a) = 0$.
- Consider the squared weighted distance from $x$ to all $\{x_a\}$: $U = \sum p_a(x - x_a)^2$. (Notice: all values $\{x - x_a\}$ are fixed: it's the weights we can play with).
- Minimize $U$ subject to the constraints.

# Previous hint: Rajan

- Impose $\sum p_a(x) = 1$, $\sum p_a(x - x_a) = 0$.
- Consider the squared weighted distance from $x$ to all $\{x_a\}$: $U = \sum p_a(x - x_a)^2$. (Notice: all values $\{x - x_a\}$ are fixed: it's the weights we can play with).
- Minimize $U$ subject to the constraints.
- Guess what we get?

# Previous hint: Rajan

- Impose $\sum p_a(x) = 1$, $\sum p_a(x - x_a) = 0$.
- Consider the squared weighted distance from $x$ to all $\{x_a\}$: $U = \sum p_a(x - x_a)^2$. (Notice: all values $\{x - x_a\}$ are fixed: it's the weights we can play with).
- Minimize $U$ subject to the constraints.
- Guess what we get?

Mathematically: extremize

$$\mathcal{L} = \beta \sum p_a(x - x_a)^2 + \alpha \left( \sum p_a - 1 \right) + \lambda \sum p_a(x - x_a)$$

## Rajan – FEM

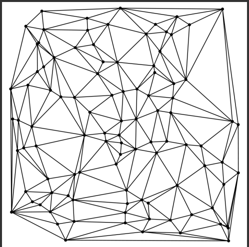
- If the “program” is carried out we recover . . .

# Rajan – FEM

- If the “program” is carried out we recover . . .
- the FEM . . .

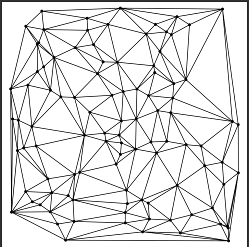
# Rajan – FEM

- If the "program" is carried out we recover …
- the FEM …
- on the Delaunay triangulation!



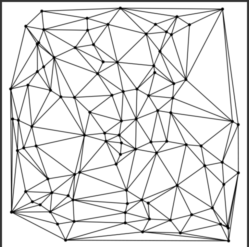
# Rajan – FEM

- If the "program" is carried out we recover ...
- the FEM ...
- on the Delaunay triangulation!
- Notice we don't even compute the triangulation! (which is non-trivial).



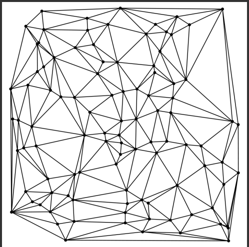
# Rajan – FEM

- If the "program" is carried out we recover ...
- the FEM ...
- on the Delaunay triangulation!
- Notice we don't even compute the triangulation! (which is non-trivial).



# Rajan – FEM

- If the "program" is carried out we recover ...
- the FEM ...
- on the Delaunay triangulation!
- Notice we don't even compute the triangulation! (which is non-trivial).
- Amazing or what? (It is to me.)



# LME, continued

- So, this $U$ looks like an energy, doesn't it: $U = \sum p_a(x - x_a)^2$.
- It would be a sort of mean spherical model (all $p_a$ add up to 1), where each "spin" $p_a$ is coupled only to its field $(x - x_a)^2$.

# LME, continued

- So, this $U$ looks like an energy, doesn't it: $U = \sum p_a (x - x_a)^2$.
- It would be a sort of mean spherical model (all $p_a$ add up to 1), where each "spin" $p_a$ is coupled only to its field $(x - x_a)^2$.

So, what if we tried "entropy" instead of "energy"?

$$\mathcal{L} = \sum p_a \left(\log p_a - 1\right) + \alpha \left( \sum p_a - 1 \right) + \lambda \sum p_a (x - x_a)$$

# LME, continued

- So, this $U$ looks like an energy, doesn't it: $U = \sum p_a (x - x_a)^2$.
- It would be a sort of mean spherical model (all $p_a$ add up to 1), where each "spin" $p_a$ is coupled only to its field $(x - x_a)^2$.

So, what if we tried "entropy" instead of "energy"?

$$\mathcal{L} = \sum p_a \left(\log p_a - 1\right) + \alpha \left(\sum p_a - 1\right) + \lambda \sum p_a (x - x_a)$$

Notice: $S = -\sum p_a \left(\log p_a - 1\right)$, which is maximized.

# LME, continued

Well, in SM the solution to this is well-known:

$$p_a = \frac{1}{Z} \exp\left[-\lambda^*(x - x_a)\right],$$

where $\lambda^*$ is the value:

$$\lambda^* : \min_{\lambda} \log Z$$

# LME, continued

Well, in SM the solution to this is well-known:

$$p_a = \frac{1}{Z} \exp\left[-\lambda^*(x - x_a)\right],$$

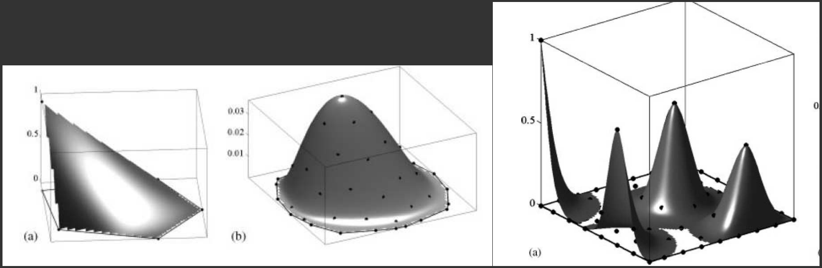
where $\lambda^*$ is the value:

$$\lambda^* : \min_\lambda \log Z$$

Note: this is the physicist's approach to LME. Computing people have an easier time considering this from information theory: $S$ is the information entropy (Shannon's), which reflects how much we know about a system with variables $p_a$. By maximizing it, we choose the least-biased set.

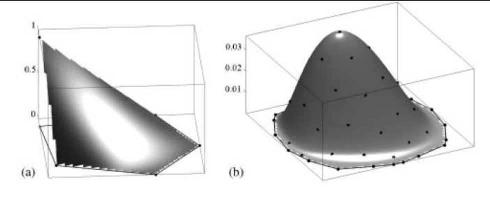
# LME, properties

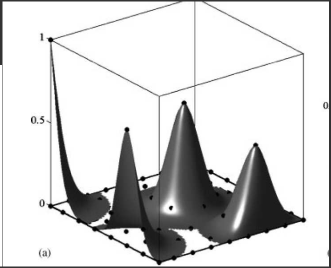
- They look nice and elegant, a bit like splines, not quite as spiky as FEMs. They are quite wide.
- They automatically die out at the boundary, except for nodes already at the boundary (just like FEM).



# LME, properties

- They look nice and elegant, a bit like splines, not quite as spiky as FEMs. They are quite wide.
- They automatically die out at the boundary, except for nodes already at the boundary (just like FEM).




| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |

# Extended LME

The functions are perhaps too wide, so recalling the result by Rajan, we may consider:

$$\begin{aligned}\mathcal{L} &= \beta \sum p_a(x-x_a)^2 + \sum p_a\left(\log p_a - 1\right) + \\ &+ \alpha\left(\sum p_a - 1\right) + \lambda \sum p_a(x-x_a)\end{aligned}$$

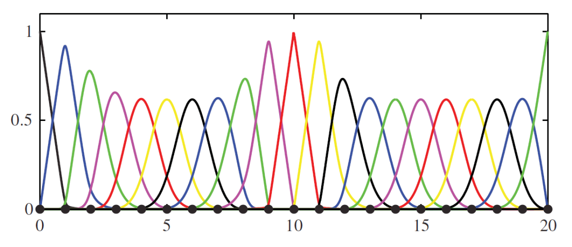
# Extended LME

The functions are perhaps too wide, so recalling the result by Rajan, we may consider:

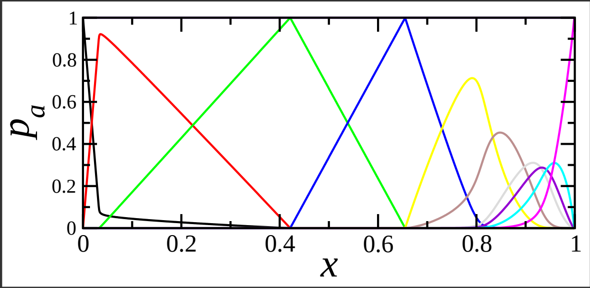
$$\begin{aligned}\mathcal{L} &= \beta \sum p_a (x - x_a)^2 + \sum p_a (\log p_a - 1) + \\ &+ \alpha \left( \sum p_a - 1 \right) + \lambda \sum p_a (x - x_a)\end{aligned}$$

- Pure LEM: $\beta \to 0$. Very hot limit (just entropy)
- Pure Rajan: $\beta \to \infty$. Very cold limit (just energy)
- We can tune $\beta$, even have $\beta = \beta(x)$.

# Extended LME: pictures




Fixed spacing, varying $\beta$




Fixed $\beta$, varying spacing

# LME - SPH connection

Another interesting fact: if we drop 1-cons:

$$\begin{aligned}\mathcal{L} &= \beta \sum p_a (x - x_a)^2 + \sum p_a \left(\log p_a - 1\right) + \\ &+ \alpha \left( \sum p_a - 1 \right),\end{aligned}$$

The solution is just:

$$p_a = \frac{\exp[-\beta (x - x_a)^2]}{Z},$$

SPH-Shepard!

# LME - SPH connection

Another interesting fact: if we drop 1-cons:

$$\begin{aligned}\mathcal{L} &= \beta \sum p_a (x - x_a)^2 + \sum p_a \left(\log p_a - 1\right) + \\ &+ \alpha \left( \sum p_a - 1 \right),\end{aligned}$$

The solution is just:

$$p_a = \frac{\exp[-\beta (x - x_a)^2]}{Z},$$

SPH-Shepard!

I know, this is trivial, but few works bridge these fields: fluids, computational geometry, and statistical physics.

# Limitations of LME

Imagine we would also want 2-cons: $\sum_a p_a(x - x_a)^2 = 0$.

# Limitations of LME

Imagine we would also want 2-cons: $\sum_a p_a(x - x_a)^2 = 0$.
But, the associated Lagrange multiplier already *is* our energy term!

$$\mathcal{L} = \beta \sum p_a(x - x_a)^2 + \dots$$

# Limitations of LME

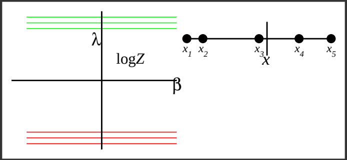
Imagine we would also want 2-cons: $\sum_a p_a (x - x_a)^2 = 0$.
But, the associated Lagrange multiplier already *is* our energy term!

$$\mathcal{L} = \beta \sum p_a (x - x_a)^2 + \dots$$

Clearly, a minimum of $\mathcal{L}$ w.r.t. $\beta$ will likely not be found.
In fact, the minimum will always be $\beta \to \infty$, as we will see next, and FEM will be recovered (which does not have 2-cons!)

# LME's lack of 2-cons

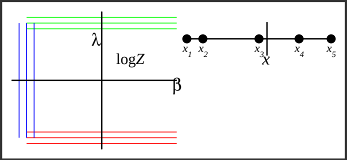
(Argument due to P. Español). The situation is clearer in 1D.




- Low $\lambda < 0$: $Z \approx e^{-\lambda|x-x_1|}$, hence $\log Z \approx -\lambda|x - x_1|$.
- High $\lambda > 0$: $Z \approx e^{\lambda|x-x_5|}$, hence $\log Z \approx \lambda|x - x_5|$.

So, for a fixed $\beta$ we can expect to find a minimum of $\log Z$. This is nothing but the extended LEM.

# LME's lack of 2-cons




- Low $\beta < 0$: $Z \approx e^{-\beta|x-x_3|^2}$, hence $\log Z \approx -\beta|x-x_3|^2$. OK, thus far. But:
- High $\beta > 0$: $Z \approx e^{-\beta|x-x_3|^2}$, hence $\log Z \approx -\beta|x-x_3|^3$. Disaster.

Hence, the minimum squeezes away towards $\beta \to \infty$!

# Introducing SME

OK, now imagine we had:

$$\log Z \approx \beta(g - |x - x_3|^2),$$

with $g > |x - x_3|^2$?

Now, the minimum would not be at $\beta \to \infty$ any more.

# Introducing SME

OK, now imagine we had:

$$\log Z \approx \beta(g - |x - x_3|^2),$$

with $g > |x - x_3|^2$?
Now, the minimum would not be at $\beta \to \infty$ any more.
But then, of course, our original problem has changed to:

$$\begin{aligned}\mathcal{L} &= \sum p_a (\log p_a - 1) + \alpha \left( \sum p_a - 1 \right) + \\ &+ \lambda \sum p_a (x - x_a) + \beta \left( \sum p_a (x - x_a)^2 - g(x) \right).\end{aligned}$$

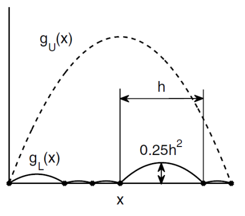
Whose solution is again known from SM:

$$p_a = \frac{1}{Z} \exp\left[ -\lambda^* (x - x_a) - \beta^* \left( (x - x_a)^2 - g(x) \right) \right],$$

$$\lambda^*, \beta^* : \min_{\lambda, \beta} \log Z$$

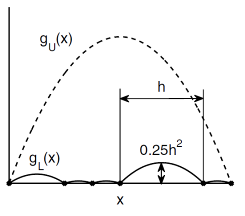
# SME: the gap

Some inspection of the particular requirement $g > |x - x_3|^2$ shows that in general $g(x)$, called "the gap" should be confined to this region:



# SME: the gap

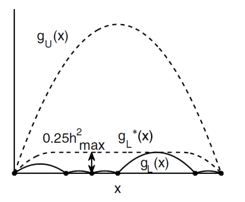
Some inspection of the particular requirement $g > |x - x_3|^2$ shows that in general $g(x)$, called “the gap” should be confined to this region:




How bad is this? After all, 2-cons is lost . . .

# SME: the gap

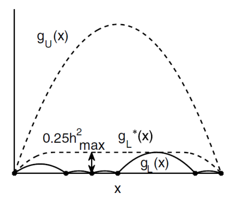
Well, not very bad. Consider the following choice




Here, $g(x)$ is just a constant (except close to the boundaries).

# SME: the gap

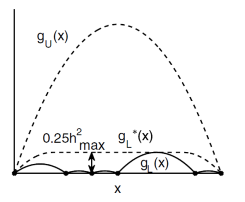
Well, not very bad. Consider the following choice




Here, $g(x)$ is just a constant (except close to the boundaries). Since we have imposed $\sum p_a(x - x_a)^2 - g(x) = 0$, this means a quadratic function will be reconstructed, only shifted by $g(x)$.

# SME: the gap

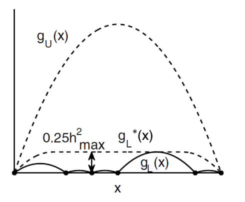
Well, not very bad. Consider the following choice




Here, $g(x)$ is just a constant (except close to the boundaries). Since we have imposed $\sum p_a(x - x_a)^2 - g(x) = 0$, this means a quadratic function will be reconstructed, only shifted by $g(x)$. But, if $g(x)$ is constant, all the derivatives will be exact!! This is exactly what we wanted!!!

# SME: the gap

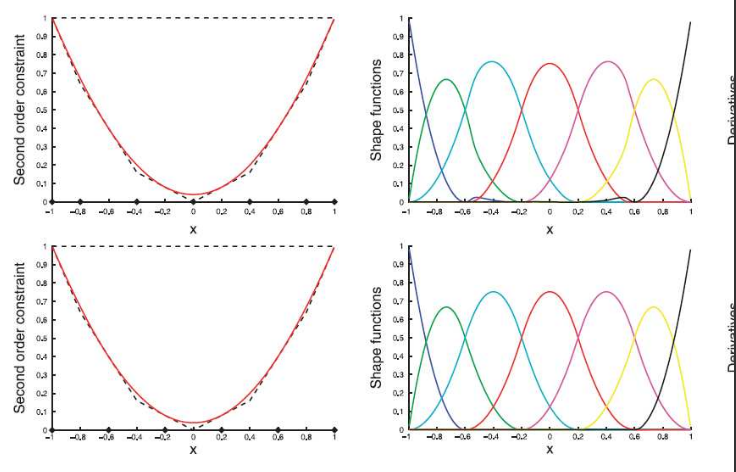
Well, not very bad. Consider the following choice



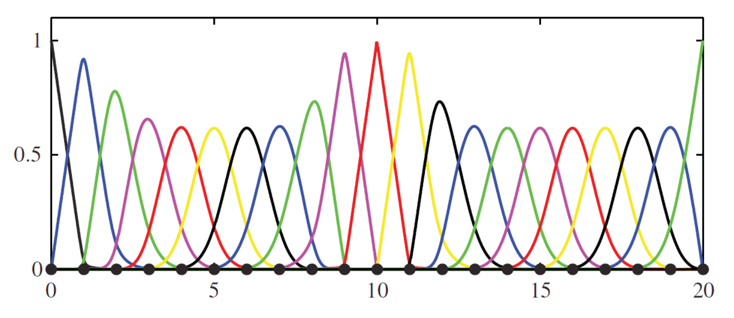

Here, $g(x)$ is just a constant (except close to the boundaries). Since we have imposed $\sum p_a(x - x_a)^2 - g(x) = 0$, this means a quadratic function will be reconstructed, only shifted by $g(x)$. But, if $g(x)$ is constant, all the derivatives will be exact!! This is exactly what we wanted!!!

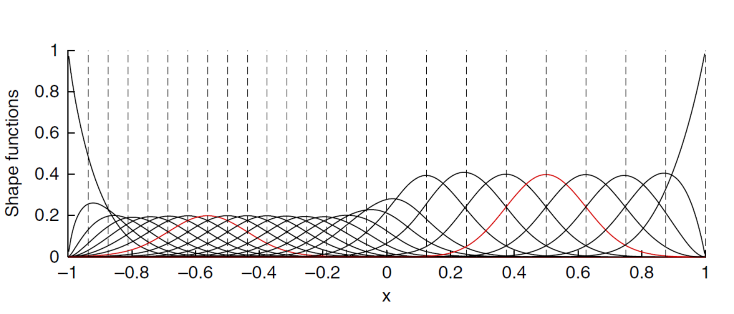
| 0-c | 1-c | 2-c | loc | m-free | bound |
|---|---|---|---|---|---|
| | | | | | |

# SME: some pictures

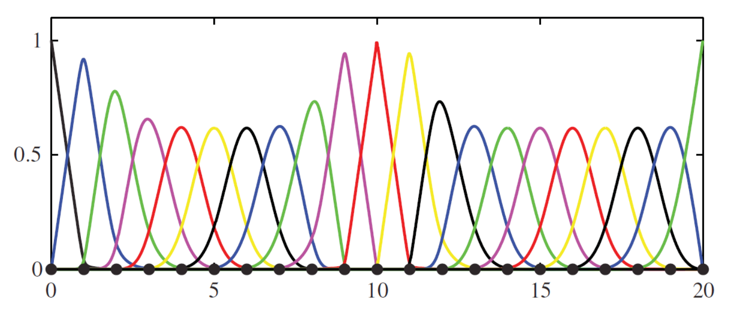


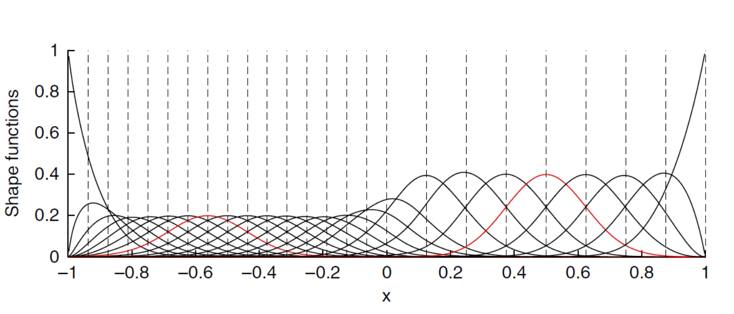
# SME: some pictures



# SME: some pictures



# To do

Basically: go ahead, and implement it on a simulation!

## To do

Basically: go ahead, and implement it on a simulation!
Is it hard? Not really: a minimization should be performed only at each of the particles (not all $x$), which involves only $\partial_\lambda \log Z$ and $\partial_\sigma \log Z$ (these are quite easy).

# To do

Basically: go ahead, and implement it on a simulation!
Is it hard? Not really: a minimization should be performed only at each of the particles (not all $x$), which involves only $\partial_\lambda \log Z$ and $\partial_\sigma \log Z$ (these are quite easy).
For the derivatives, $\partial^2_{\lambda\lambda} \log Z$ etc are needed, but these are only needed when the minimum has been found. Moreover, some methods (quasi-Newton) are supposed to provide these second derivatives automatically.

# References


- *LME* M. Arroyo and M. Ortiz, *Local maximum-entropy approximation schemes: a seamless bridge between nite elements and meshfree methods*. Int. J. Numer. Meth. Engng 2006; 65:21672202. DOI: 10.1002/nme.1534
- *SME* C.J. Cyron, M. Arroyo, and M. Ortiz, *Smooth, second order, non-negative meshfree approximants selected by maximum entropy*. Int. J. Numer. Meth. Engng 2009; 79:16051632. DOI: 10.1002/nme.2597