# Discrete Gradient Theorem and element-based integration in meshless methods

Guillaume Pierrot [1]    Gabriel Fougeron [1,2]

[1]ESI Group, Rungis, France

[2]CentraleSupelec, Châtenay-Malabry, France

# Rationale

## The curse of meshless methods

- Meshless methods have very appealing properties (versatility, low numerical diffusion..) but they classically exhibit suboptimal convergence rates

## The curse of meshless methods

- Meshless methods have very appealing properties (versatility, low numerical diffusion..) but they classically exhibit suboptimal convergence rates
- Shouldn't numerical integration and numerical differentiation ( discrete gradient) satisfy some *compatibility* conditions as the mesh-based methods do ?

## The curse of meshless methods

- Meshless methods have very appealing properties (versatility, low numerical diffusion..) but they classically exhibit suboptimal convergence rates
- Shouldn't numerical integration and numerical differentiation ( discrete gradient) satisfy some *compatibility* conditions as the mesh-based methods do ?

## Discrete Gradient Theorem as the cornerstone [Bonet & al]

get it right®

## The curse of meshless methods

- Meshless methods have very appealing properties (versatility, low numerical diffusion..) but they classically exhibit suboptimal convergence rates
- Shouldn't numerical integration and numerical differentiation ( discrete gradient) satisfy some *compatibility* conditions as the mesh-based methods do ?

## Discrete Gradient Theorem as the cornerstone [Bonet & al]

$$\int_{\mathcal{C}} \nabla u = \int_{\partial \mathcal{C}} u$$

# Outline

1. Compatibility in the context of nodal integration

2. Approximate compatibility

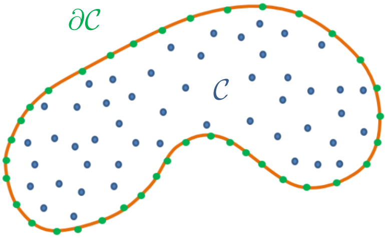
3. Towards element-based integration

# Outline

1. Compatibility in the context of nodal integration
2. Approximate compatibility
3. Towards element-based integration

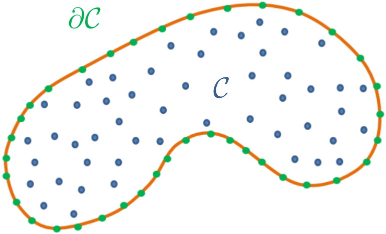
Cloud of points : $\mathcal{C}$

Boundary nodes $\partial\mathcal{C} \subset \mathcal{C}$



Cloud of points : $\mathcal{C}$

Boundary nodes $\partial\mathcal{C} \subset \mathcal{C}$

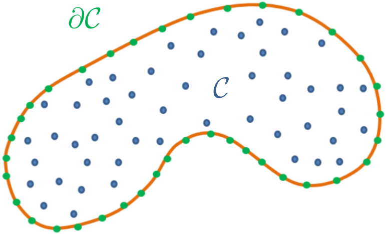



Meshless operators :

- Nodal volume quadrature : $\displaystyle \oint_{\mathcal{C}} f = \sum_{i \in \mathcal{C}} V_i f_i$
- Boundary quadrature : $\displaystyle \oint_{\partial\mathcal{C}} f = \sum_{i \in \partial\mathcal{C}} f_i \mathbf{\Gamma}_i$

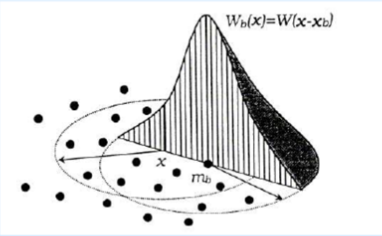
Cloud of points : $\mathcal{C}$

Boundary nodes $\partial\mathcal{C} \subset \mathcal{C}$



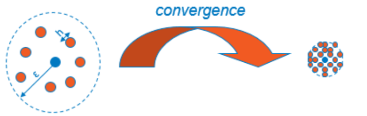

Meshless operators :

- Nodal volume quadrature : $\displaystyle \oiiint_{\mathcal{C}} f = \sum_{i \in \mathcal{C}} V_i f_i$
- Boundary quadrature : $\displaystyle \oiint_{\partial\mathcal{C}} f = \sum_{i \in \partial\mathcal{C}} f_i \mathbf{\Gamma}_i$
- Meshless gradient : $\displaystyle V_i \nabla_i f = \sum_{j \in \mathcal{N}(i)} \mathbf{A}_{i,j} f_j$




## SPH gradient

$$\nabla_i^O f = -\oint_{\mathcal{C}} f_\star \nabla W_\epsilon \left(\mathbf{x}_\star - \mathbf{x}_i\right)$$
$$= -\sum_{j\in\mathcal{C}} V_j\, f_j\, \nabla W_\epsilon(\mathbf{x}_j - \mathbf{x}_i)$$



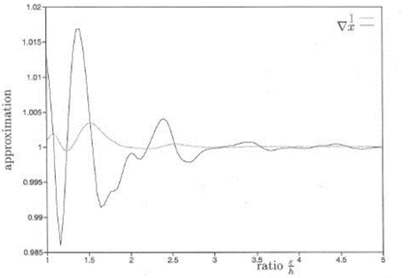




FIG. 2.2: *Approximation de $\nabla x$ et 1 sur une répartition régulière de particules en 2D*

## Recovery of $\mathbb{P}_1$-consistency

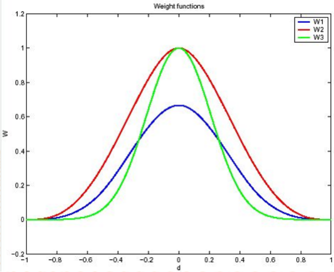
$$\begin{cases} \nabla_i^{\mathrm{R1}} f &= -\oint_{\mathcal{C}} (f_\star - f_i)\, \mathbb{B}_i \, \nabla W_h \, (\mathbf{x}_\star - \mathbf{x}_i) \\ &= -\sum_{j \in \mathcal{C}} V_j \, \mathbb{B}_i \, \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(f_j - f_i) \\ \mathrm{I}_d &= -\oint_{\mathcal{C}} (\mathbf{x}_\star - \mathbf{x}_i) \otimes \mathbb{B}_i \, \nabla W_h \, (\mathbf{x}_\star - \mathbf{x}_i) \end{cases}$$

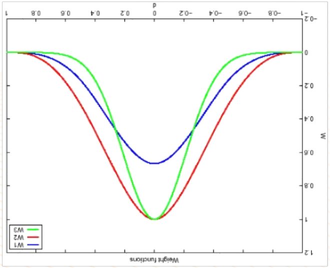
## Recovery of $\mathbb{P}_1$-consistency

$$\begin{cases} \nabla_i^{\mathrm{R1}} f & = & -\oint_{\mathcal{C}} (f_\star - f_i)\, \mathbb{B}_i\, \nabla W_h\, (\mathbf{x}_\star - \mathbf{x}_i) \\ & = & -\sum_{j \in \mathcal{C}} V_j\, \mathbb{B}_i\, \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(f_j - f_i) \\ \mathrm{I}_d & = & -\oint_{\mathcal{C}} (\mathbf{x}_\star - \mathbf{x}_i) \otimes \mathbb{B}_i\, \nabla W_h\, (\mathbf{x}_\star - \mathbf{x}_i) \end{cases}$$

this time gradient convergence is ensured for $\epsilon \propto h$

get it right®




## Standard Formulation

$$\nabla_i^{\mathrm{LS}} u = \operatorname{argmin}_{\mathbf{b}} \sum_{j \in \mathrm{N}_i} W_{ij} \left(u_j - u_i - \mathbf{b} \cdot (x_j - x_i)\right)^2$$

## Alternative formulation [Levin]

$$\{\mathbf{B}_{ij}\} = \underset{\{\mathbf{C}_{ij}\}}{\operatorname{argmin}} \sum_{j \in \mathrm{N}_i} \mathcal{W}_{ij}^{-1} \mathbf{C}_{ij}^2$$

$$s.t. \quad \sum_{j \in \mathrm{N}_i} \mathbf{C}_{ij} \otimes (x_j - x_i) = Id$$

$$\nabla_i^{\mathrm{LS}} u = \sum_{j \in \mathrm{N}_i} \mathbf{B}_{ij} \ (u_j - u_i)$$

# A dual operator

## Integration by parts formula

$$\int_{\Omega} f \nabla \cdot \mathbf{g} + \mathbf{g} \cdot \nabla f \, \mathrm{d}V = \int_{\partial\Omega} f \mathbf{g} \cdot \mathrm{d}\mathbf{S}$$

# A dual operator

## Integration by parts formula

$$\int_{\Omega} f \nabla \cdot \mathbf{g} + \mathbf{g} \cdot \nabla f \; \mathrm{d}V = \int_{\partial \Omega} f \mathbf{g} \cdot \mathrm{d}\mathbf{S}$$

## Discrete counterpart : definition of a dual gradient $\mathbb{V}^*$

$$\int_{\mathcal{C}} f \mathbb{V} \cdot \mathbf{g} + \mathbf{g} \cdot \mathbb{V}^* f = \int_{\partial \mathcal{C}} f \mathbf{g}$$

# A dual operator

## Integration by parts formula

$$\int_{\Omega} f\nabla\cdot\mathbf{g}+\mathbf{g}\cdot\nabla f\;\mathrm{d}V=\int_{\partial\Omega} f\mathbf{g}\cdot\mathrm{d}\mathbf{S}$$

## Discrete counterpart : definition of a dual gradient $\mathbb{V}^*$

$$\oint_{\mathcal{C}} f\mathbb{V}\cdot\mathbf{g}+\mathbf{g}\cdot\mathbb{V}^* f=\oint_{\partial\mathcal{C}} f\mathbf{g}$$

## Explicit formula for the dual gradient

$$V_i\mathbb{V}_i^* f=\sum_{j\in\mathcal{N}(i)}\left(-\mathbf{A}_{j,i}+\delta_{i,j}\boldsymbol{\Gamma}_i\right)f_j$$

## Continuous weak formulation

Find $u \in \mathcal{H}^1(\Omega)$ such that :

$$\begin{cases} \displaystyle\int_\Omega \nabla u \cdot \nabla v = \int_\Omega sv \quad \forall v \in \mathcal{H}_0^1(\Omega) \\ u_{|\partial\Omega} = u_0 \end{cases}$$

# Plug & play discretization of diffusion equation

## Continuous weak formulation

Find $u \in \mathcal{H}^1(\Omega)$ such that :

$$\begin{cases} \displaystyle\int_\Omega \nabla u \cdot \nabla v = \int_\Omega sv \quad \forall v \in \mathcal{H}^1_0(\Omega) \\ u_{|\partial\Omega} = u_0 \end{cases}$$

## Discrete weak formulation

Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\begin{cases} \displaystyle\int_{\mathcal{C}} \nabla u \cdot \nabla v = \int_{\mathcal{C}} sv \quad \forall v : \mathcal{C} \backslash \partial\mathcal{C} \to \mathbb{R} \\ u_{|\partial\mathcal{C}} = u_0 \end{cases}$$

# Plug & play discretization of diffusion equation

## Continuous weak formulation

Find $u \in \mathcal{H}^1(\Omega)$ such that :

$$\begin{cases} \displaystyle\int_\Omega \nabla u \cdot \nabla v = \int_\Omega s v \quad \forall v \in \mathcal{H}^1_0(\Omega) \\ u_{|\partial\Omega} = u_0 \end{cases}$$

## Discrete weak formulation

Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\begin{cases} a_{\text{stab}}(u, v) + \displaystyle\int_{\mathcal{C}} \nabla u \cdot \nabla v = \int_{\mathcal{C}} s v \quad \forall v : \mathcal{C} \backslash \partial\mathcal{C} \to \mathbb{R} \\ u_{|\partial\mathcal{C}} = u_0 \end{cases}$$

# Linear patch test

## Equivalent nodewise formulation

Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\left\{ \begin{aligned} -\mathbb{V}_i^* \cdot \mathbb{V} u &= s_i \qquad \forall i \in \mathcal{C} \backslash \partial\mathcal{C} \\ u_{|\partial\mathcal{C}} &= u_0 \end{aligned} \right.$$

## Equivalent nodewise formulation

Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\begin{cases} a_{\text{stab}}\left(u, \delta_i\right) - \mathbb{V}_i^* \cdot \mathbb{V}u = s_i & \forall i \in \mathcal{C} \backslash \partial\mathcal{C} \\ u_{|\partial\mathcal{C}} = u_0 & \end{cases}$$

## Equivalent nodewise formulation

Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\begin{cases} a_{\text{stab}}\left(u, \delta_i\right) - \mathbb{V}_i^* \cdot \mathbb{V}u = s_i & \forall i \in \mathcal{C}\backslash\partial\mathcal{C} \\ u_{|\partial\mathcal{C}} = u_0 & \end{cases}$$

## Necessary conditions for the linear patch test

- $\mathbb{V}x = \mathrm{I}_d$

## Equivalent nodewise formulation

Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\begin{cases} a_{\text{stab}}\left(u, \delta_i\right) - \mathbb{V}_i^* \cdot \mathbb{V}u = s_i & \forall i \in \mathcal{C} \backslash \partial\mathcal{C} \\ u_{|\partial\mathcal{C}} = u_0 & \end{cases}$$

## Necessary conditions for the linear patch test

- $\mathbb{V}x = \mathrm{I}_d$
- $\mathbb{V}^* 1 = 0$

## Equivalent nodewise formulation

Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\begin{cases} a_{\text{stab}}(u, \delta_i) - \mathbb{V}_i^* \cdot \mathbb{V} u = s_i & \forall i \in \mathcal{C} \backslash \partial \mathcal{C} \\ u_{|\partial \mathcal{C}} = u_0 & \end{cases}$$

## Necessary conditions for the linear patch test

- $\mathbb{V} x = \mathrm{I}_d$
- $\mathbb{V}^* 1 = 0$ Compatibility !

## Equivalent nodewise formulation

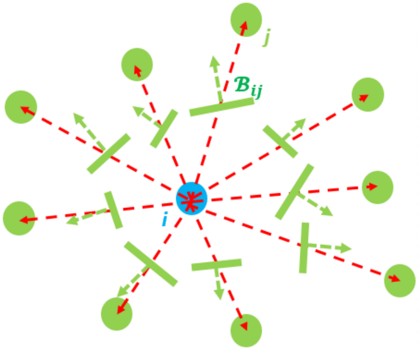
Find $u : \mathcal{C} \to \mathbb{R}$ such that :

$$\begin{cases} a_{\text{stab}}\left(u, \delta_i\right) - \nabla_i^* \cdot \nabla u = s_i & \forall i \in \mathcal{C} \backslash \partial\mathcal{C} \\ u_{|\partial\mathcal{C}} = u_0 & \end{cases}$$

## Necessary conditions for the linear patch test

- $\nabla x = \mathrm{I}_d$
- $\nabla^* 1 = 0$

$\Leftrightarrow$ Discrete Gradient Theorem : $\oint_{\mathcal{C}} \nabla u = \oint_{\partial\mathcal{C}} u$




The anti-symmetric edge coefficient $\mathbf{B}_{ij} = \mathbf{A}_{ij} - \mathbf{A}_{ji}$ does fulfill a *volume closedness* property :

$$\sum_{j \in \mathcal{N}(i)} \mathbf{B}_{j,i} + \delta_{i \in \partial \mathcal{C}} \mathbf{\Gamma}_i = 0$$

$\Rightarrow$ similar to vertex-centered FV discretizations

## Corrected gradient

$$\nabla_i^c u = \nabla_i u + \sum_{j \in \mathcal{N}(i)} \boldsymbol{\mu}_{i,j}(u_j - u_i - \nabla_i u \cdot (\mathbf{x}_j - \mathbf{x}_i))$$

## Corrected gradient

$$\mathbb{V}_i^c u = \mathbb{V}_i u + \sum_{j \in \mathcal{N}(i)} \boldsymbol{\mu}_{i,j} (u_j - u_i - \mathbb{V}_i u \cdot (\mathbf{x}_j - \mathbf{x}_i))$$

Correction preserves linear consistency :

$$\forall \boldsymbol{\mu}_{i,j}, \; \mathbb{V}\mathbf{x} = \mathrm{I}_d \Rightarrow \mathbb{V}^c \mathbf{x} = \mathrm{I}_d$$

# Compatibility recovery : correction

## Corrected gradient

$$\mathbb{V}_i^c u = \mathbb{V}_i u + \sum_{j \in \mathcal{N}(i)} \boldsymbol{\mu}_{i,j} (u_j - u_i - \mathbb{V}_i u \cdot (\mathbf{x}_j - \mathbf{x}_i))$$

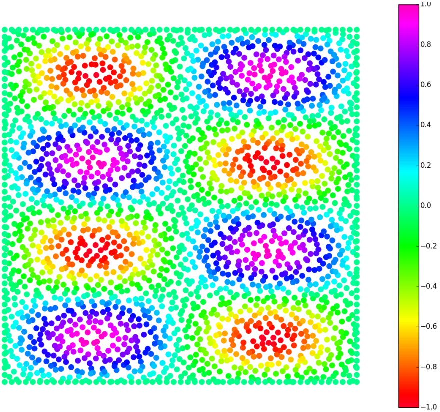
Correction preserves linear consistency :

$$\forall \boldsymbol{\mu}_{i,j}, \; \mathbb{V}\mathbf{x} = \mathrm{I}_d \Rightarrow \mathbb{V}^c \mathbf{x} = \mathrm{I}_d$$

## Correction equations

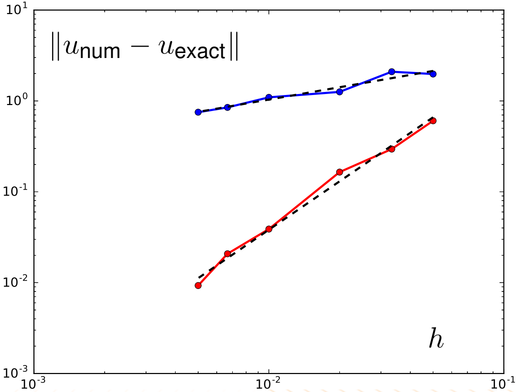
Solve $\mathbb{V}^{c*} 1 = 0$ for $\boldsymbol{\mu}_{i,j}$ (in the least-norm sense) given $\mathbb{V}$

# Manufactured solution




Halton sequences

$$s = 20\pi^2 \sin(2\pi x)\sin(4\pi y)$$
$$u = \sin(2\pi x)\sin(4\pi y)$$




**Non-corrected**

$\|u_{\text{num}} - u_{\text{exact}}\| \propto h^{0.45}$

**Corrected**

$\|u_{\text{num}} - u_{\text{exact}}\| \propto h^{1.77}$

$$\mathbb{V}_i^{\text{R1}} f = -\sum_{j \in \mathcal{C}} V_j \mathbb{B}_i \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(f_j - f_i)$$

$$\mathbb{B}_i^{-1} = -\sum_{j \in \mathcal{C}} V_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T$$

# Outline

# Towards a cheap compatibility correction

## Rationale

- Compatibility correction works well but it comes with a cost

## Rationale

- Compatibility correction works well but it comes with a cost
- Do we really need to correct up to machine (or say, very good) precision ?

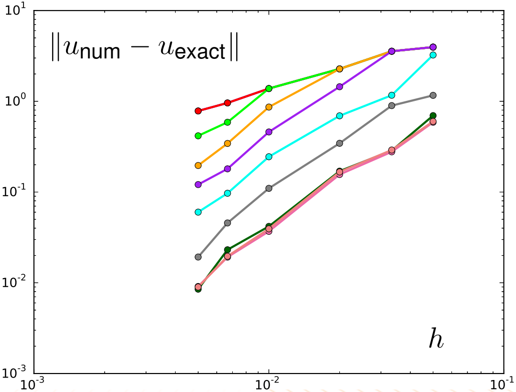
# Towards a cheap compatibility correction

## Rationale

- Compatibility correction works well but it comes with a cost
- Do we really need to correct up to machine (or say, very good) precision ?

## Order of magnitude of the compatibility defect

$$\begin{aligned}
\mathbb{\nabla}_i^* 1 &= \frac{1}{V_i} \left\{ \oint_{\partial \mathcal{C}} \delta_i - \oint_{\mathcal{C}} \mathbb{\nabla}\, \delta_i \right\} \\
&= \frac{1}{V_i} \sum_{j \in \mathcal{N}(i)} \left( -\mathbf{A}_{j,i} + \delta_{i,j} \mathbf{\Gamma}_i \right) \\
&= \mathcal{O}\left( \frac{1}{h} \right)
\end{aligned}$$




$\|\nabla^* 1\| \leq 50$

$\|\nabla^* 1\| \leq 20$

$\|\nabla^* 1\| \leq 10$

$\|\nabla^* 1\| \leq 5$

$\|\nabla^* 1\| \leq 2$

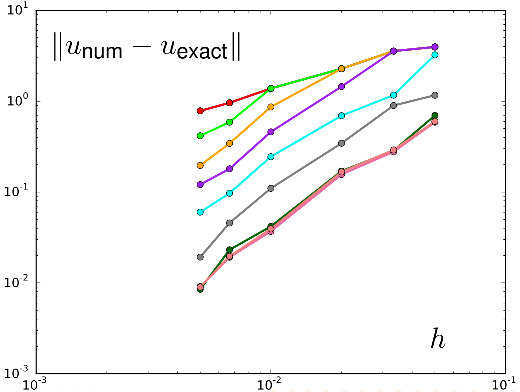
$\|\nabla^* 1\| \leq 1$

$\|\nabla^* 1\| \leq 0.5$

$\|\nabla^* 1\| \leq 0.2$

$\|\nabla^* 1\| \leq 0.1$

# Convergence for different values of $\|\nabla^* 1\|$




$\|\nabla^* 1\| \leq 50$

$\|\nabla^* 1\| \leq 20$

$\|\nabla^* 1\| \leq 10$

$\|\nabla^* 1\| \leq 5$

$\|\nabla^* 1\| \leq 2$

$\|\nabla^* 1\| \leq 1$

$\|\nabla^* 1\| \leq 0.5$

$\|\nabla^* 1\| \leq 0.2$

$\|\nabla^* 1\| \leq 0.1$

Observation : Keep $\|\nabla^* 1\| = \mathcal{O}(1)$ instead of $\|\nabla^* 1\| = \mathcal{O}(h^{-1})$

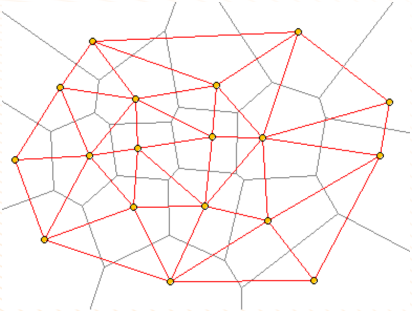
$\Rightarrow$ recover almost second order convergence !

# Outline

# Are we restricted to nodal integration ?

- Element-based integration might prove beneficial in term of algebraical complexity (assembly, connectivities..) as well as numerical stability (nodal integration is deemed to yield spurious modes)

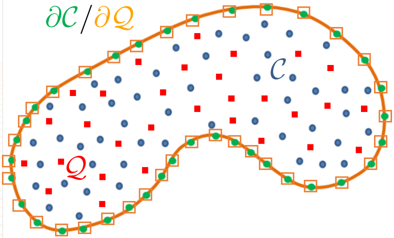
- Should be regarded as a natural generalization (nodal integration fits gracefully in the extended framework)



Cloud of nodes : $\mathcal{C}$

Integration points : $\mathcal{Q}$

Boundary $\partial\mathcal{Q} = \partial\mathcal{C}$

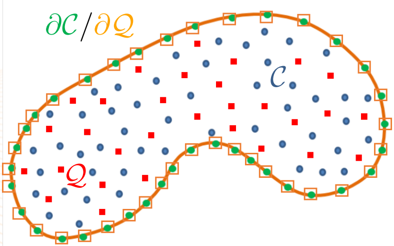


# New discretization set-up

Cloud of nodes : $\mathcal{C}$

Integration points : $\mathcal{Q}$

Boundary $\partial\mathcal{Q} = \partial\mathcal{C}$




Meshless operators :

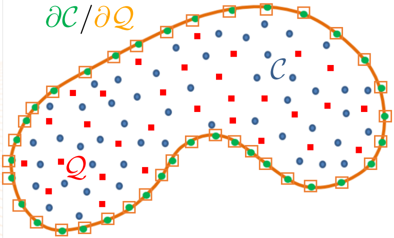
- Meshless reconstruction : $f_e = \sum_{i \in \mathcal{V}_e} \Phi_{i,e} f_i$

# New discretization set-up

Cloud of nodes : $\mathcal{C}$

Integration points : $\mathcal{Q}$

Boundary $\partial\mathcal{Q} = \partial\mathcal{C}$




Meshless operators :

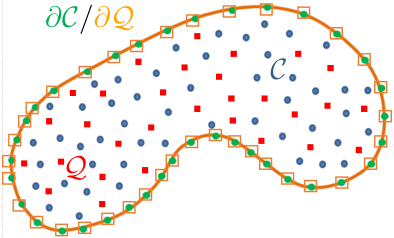
- Meshless reconstruction : $f_e = \sum_{i \in \mathcal{V}_e} \Phi_{i,e} f_i$
- Element-based cubature : $\oint_{\mathcal{Q}} f = \sum_{e \in \mathcal{Q}} V_e f_e$
- Boundary quadrature : $\oint_{\partial\mathcal{Q}} f = \sum_{i \in \partial\mathcal{Q}} f_i \mathbf{\Gamma}_i$

# New discretization set-up

Cloud of nodes : $\mathcal{C}$

Integration points : $\mathcal{Q}$

Boundary $\partial\mathcal{Q} = \partial\mathcal{C}$




Meshless operators :

- Meshless reconstruction : $f_e = \sum_{i \in \mathcal{V}_e} \Phi_{i,e} f_i$
- Element-based cubature : $\oint_{\mathcal{Q}} f = \sum_{e \in \mathcal{Q}} V_e f_e$
- Boundary quadrature : $\oint_{\partial\mathcal{Q}} f = \sum_{i \in \partial\mathcal{Q}} f_i \mathbf{\Gamma}_i$
- Meshless gradient : $V_e \nabla_e f = \sum_{i \in \mathcal{V}(e)} \mathbf{A}_{i,e} f_i$

## Reminder : nodal dual gradient $\mathbb{\nabla}^*$

$$\int_{c} f \mathbb{\nabla} \cdot \mathbf{g} + \int_{c} \mathbf{g} \cdot \mathbb{\nabla}^* f = \int_{\partial c} f \mathbf{g}$$

# A new dual operator

## Reminder : nodal dual gradient $\mathbb{V}^*$

$$\iiint_{\mathcal{C}} f \mathbb{V} \cdot \mathbf{g} + \iiint_{\mathcal{C}} \mathbf{g} \cdot \mathbb{V}^* f = \oint_{\partial \mathcal{C}} f \mathbf{g}$$

## Definition of an elemental dual gradient $\mathbb{V}^\star$

$$\iiint_{\mathcal{Q}} f \mathbb{V} \cdot \mathbf{g} + \iiint_{\mathcal{C}} \mathbf{g} \cdot \mathbb{V}^\star f = \oint_{\partial \mathcal{Q}} f \mathbf{g}$$

# A new dual operator

## Reminder : nodal dual gradient $\mathbb{V}^*$

$$\oint_{\mathcal{C}} f \mathbb{V} \cdot \mathbf{g} + \oint_{\mathcal{C}} \mathbf{g} \cdot \mathbb{V}^* f = \oint_{\partial \mathcal{C}} f \mathbf{g}$$

## Definition of an elemental dual gradient $\mathbb{V}^\star$

$$\oint_{\mathcal{Q}} f \mathbb{V} \cdot \mathbf{g} + \oint_{\mathcal{C}} \mathbf{g} \cdot \mathbb{V}^\star f = \oint_{\partial \mathcal{Q}} f \mathbf{g}$$

## Explicit formula for the dual gradient

$$V_i \mathbb{V}_i^\star f = \sum_{\mathcal{V}(e) \ni i} (-\mathbf{A}_{e,i} + \delta_{e,i} \mathbf{\Gamma}_i) f_e$$

# How to define *meshless* elements ?

## Intuition : place integration points at *holes* location

- Equip each node with a specific smoothing length :$\{\mathbf{x}_i, h_i\}_{i\in\mathcal{C}}$

# How to define *meshless* elements ?

## Intuition : place integration points at *holes* location

- Equip each node with a specific smoothing length :$\{\mathbf{x}_i, h_i\}_{i\in\mathcal{C}}$
- Use *kernel estimation* of density function :

$$\rho(\mathbf{x}) = \sum_{i\in\mathcal{C}} W_{h_i}\left(\mathbf{x} - \mathbf{x}_i\right)$$

# How to define *meshless* elements ?

## Intuition : place integration points at *holes* location

- Equip each node with a specific smoothing length : $\{\mathbf{x}_i, h_i\}_{i\in\mathcal{C}}$
- Use *kernel estimation* of density function :

$$\rho(\mathbf{x}) = \sum_{i\in\mathcal{C}} W_{h_i}\left(\mathbf{x} - \mathbf{x}_i\right)$$

- Locate integration points at local minima of the estimated density function :

$$\left\{\xi_e \setminus \nabla\rho(\xi_e) = 0 \text{ and } D^2\rho(\xi_e) \geq 0\right\}$$

# How to define *meshless* elements ?

## Intuition : place integration points at *holes* location

- Equip each node with a specific smoothing length :$\{\mathbf{x}_i, h_i\}_{i\in\mathcal{C}}$
- Use *kernel estimation* of density function :

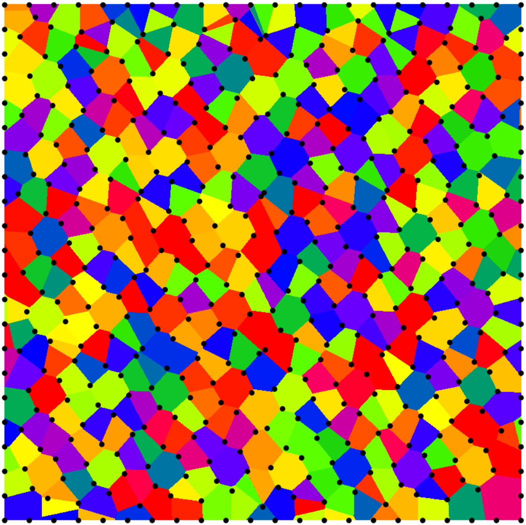
$$\rho(\mathbf{x}) = \sum_{i\in\mathcal{C}} W_{h_i}\left(\mathbf{x} - \mathbf{x}_i\right)$$

- Locate integration points at local minima of the estimated density function :

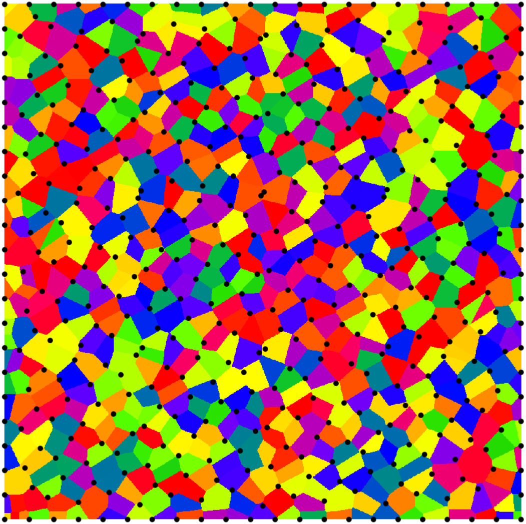
$$\left\{\xi_e \setminus \nabla\rho(\xi_e) = 0 \text{ and } D^2\rho(\xi_e) \geq 0\right\}$$

- Process can be iterated by adding newly computed integration points to the initial cloud

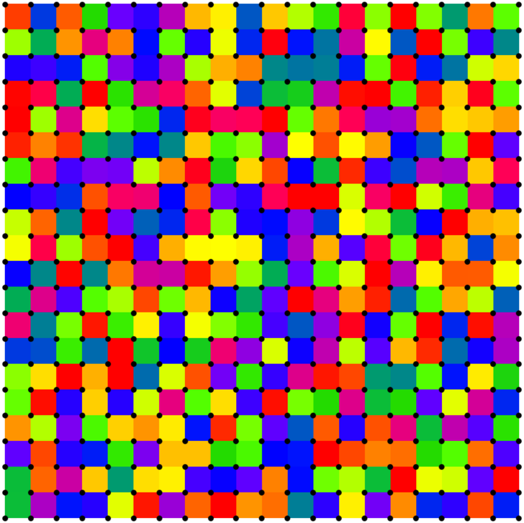
# Meshless Elements : Hammersley sequence

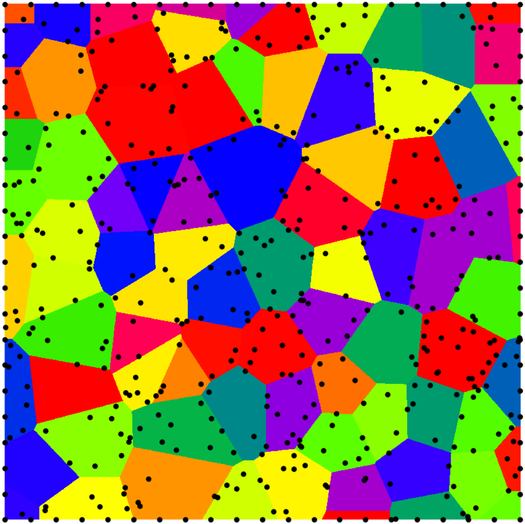


# Iterated elements construction

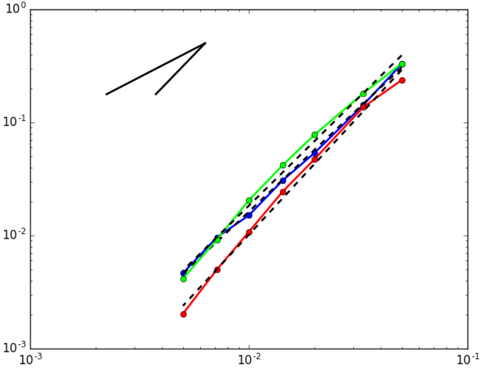


# Meshless Elements : cartesian arrangement





# $h$-analysis for Halton sequences

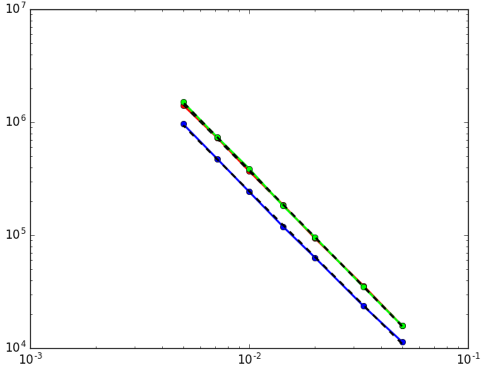



node-based integration

element-based integration

iterated element-based

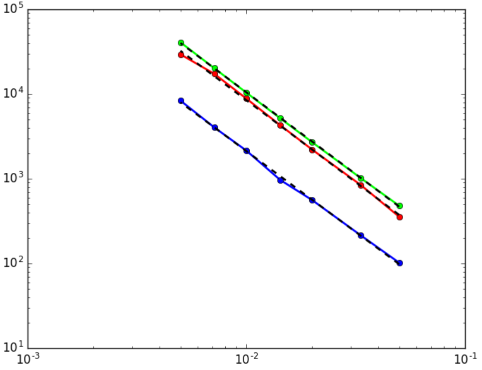
# NNZ comparison




node-based integration

element-based integration

iterated element-based

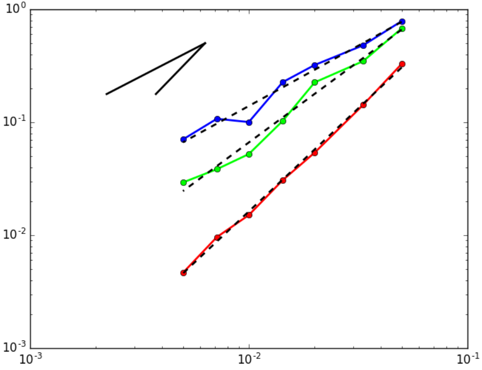



node-based integration

element-based integration

iterated element-based

# Influence of integration weights

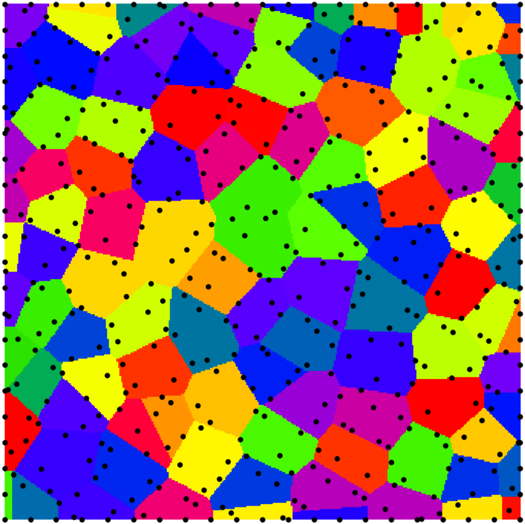



uniform

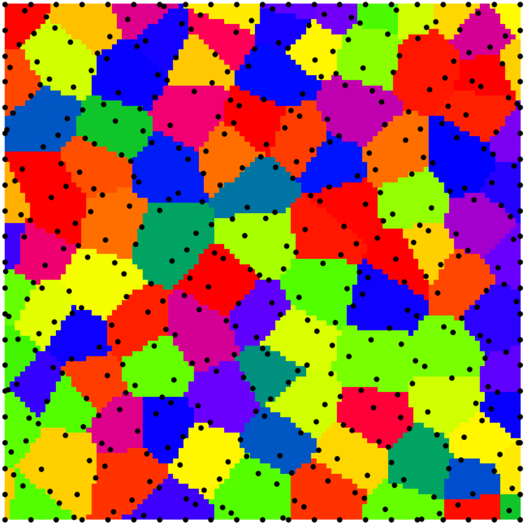
specific volume ($\rho^{-1}$)

Dirichlet regions

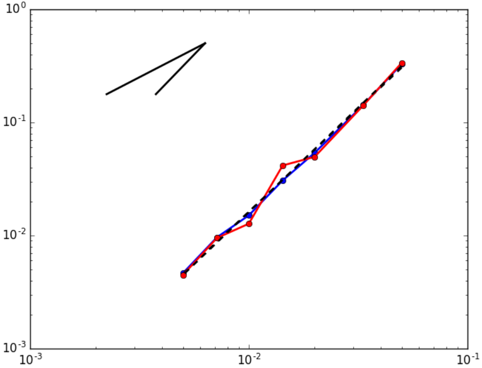
# Influence of aliasing level : $\frac{h_v}{h} = 0.03$



# Influence of aliasing level : $\frac{h_v}{h} = 0.1$



# Influence of aliasing level : convergence analysis




$\frac{h_v}{h} = 0.1$

$\frac{h_v}{h} = 0.03$

## Summary

- Concept of *dual gradient* has been introduced

## Summary

- Concept of *dual gradient* has been introduced
- Discrete Gradient Theorem has been proposed as the corner stone to enforce *compatibility* in between numerical integration and differentiation

# Conclusion

## Summary

- Concept of *dual gradient* has been introduced
- Discrete Gradient Theorem has been proposed as the corner stone to enforce *compatibility* in between numerical integration and differentiation
- An incomplete correction has been proposed with order of magnitude cheaper CPU cost while almost no loss of accuracy

# Conclusion

## Summary

- Concept of *dual gradient* has been introduced
- Discrete Gradient Theorem has been proposed as the corner stone to enforce *compatibility* in between numerical integration and differentiation
- An incomplete correction has been proposed with order of magnitude cheaper CPU cost while almost no loss of accuracy
- A meshless element construction based on kernel estimation has been described

## Summary

- Concept of *dual gradient* has been introduced
- Discrete Gradient Theorem has been proposed as the corner stone to enforce *compatibility* in between numerical integration and differentiation
- An incomplete correction has been proposed with order of magnitude cheaper CPU cost while almost no loss of accuracy
- A meshless element construction based on kernel estimation has been described
- The *compatibility* framework has been successfully extended to element-based integration

# Conclusion

## Summary

- Concept of *dual gradient* has been introduced
- Discrete Gradient Theorem has been proposed as the corner stone to enforce *compatibility* in between numerical integration and differentiation
- An incomplete correction has been proposed with order of magnitude cheaper CPU cost while almost no loss of accuracy
- A meshless element construction based on kernel estimation has been described
- The *compatibility* framework has been successfully extended to element-based integration

## Outlook

## Summary

- Concept of *dual gradient* has been introduced
- Discrete Gradient Theorem has been proposed as the corner stone to enforce *compatibility* in between numerical integration and differentiation
- An incomplete correction has been proposed with order of magnitude cheaper CPU cost while almost no loss of accuracy
- A meshless element construction based on kernel estimation has been described
- The *compatibility* framework has been successfully extended to element-based integration

## Outlook

- Further demonstration of the concept and integration within an industrial meshless code

# Conclusion

## Summary

- Concept of *dual gradient* has been introduced
- Discrete Gradient Theorem has been proposed as the corner stone to enforce *compatibility* in between numerical integration and differentiation
- An incomplete correction has been proposed with order of magnitude cheaper CPU cost while almost no loss of accuracy
- A meshless element construction based on kernel estimation has been described
- The *compatibility* framework has been successfully extended to element-based integration

## Outlook

- Further demonstration of the concept and integration within an industrial meshless code
- Are there alternative vehicles than gradient coefficients to enforce compatibility ? (idea : play on nodes positions - see talk of Gabriel Fougeron)

# The End

Thank you !

guillaume.pierrot@esi-group.com gabriel.fougeron@esi-group.com