Recovering numerical reproducibility in hydrodynamics simulations

R. Nheili [1] P. Langlois [1] C. Denis [2]

[1]University of Perpignan Via Domitia

[2]CMLA - ENS Cachan

June 29th, 2016

Plan

1. Introduction
2. Reproducibility failures in a finite element simulation
3. Recovering numerical reproducibility
4. Efficiency
5. Conclusion and work in progress

Plan

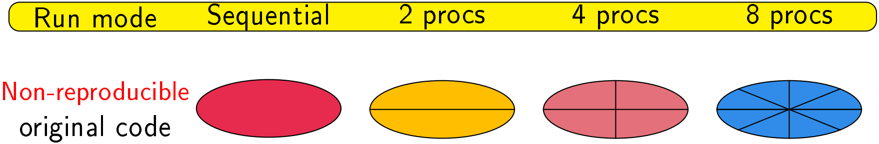
Reproducibility?

- Getting bitwise identical result for every p-parallel run; p>1
- Reproducibility $\neq$ accuracy

Run mode	Sequential	2 procs	4 procs	8 procs
Original code				

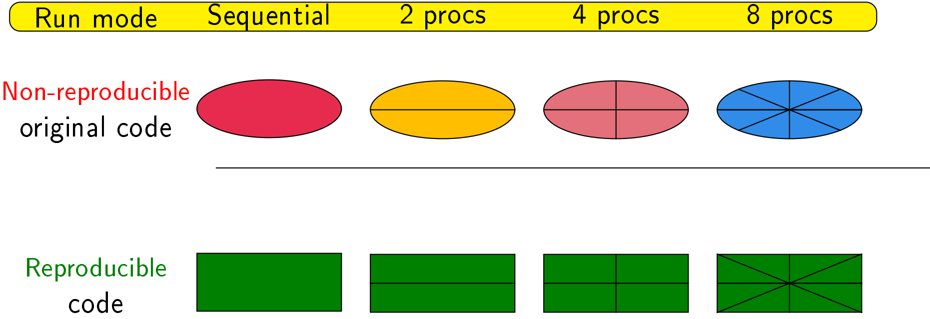
Reproducibility?

- Getting bitwise identical result for every p-parallel run; p>1
- Reproducibility $\neq$ accuracy



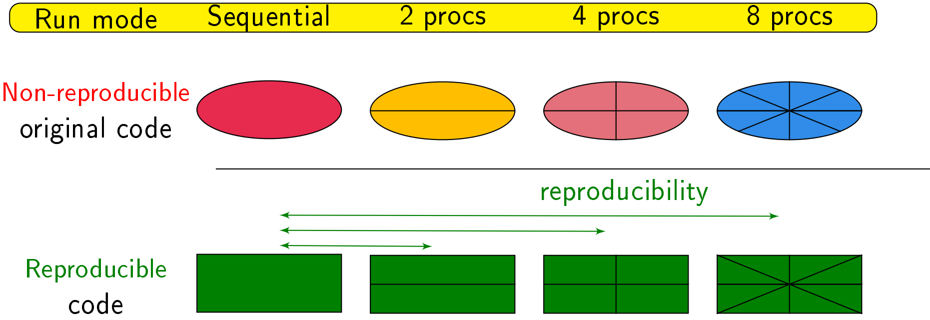
Reproducibility?

- Getting bitwise identical result for every p-parallel run; p>1
- Reproducibility $\neq$ accuracy



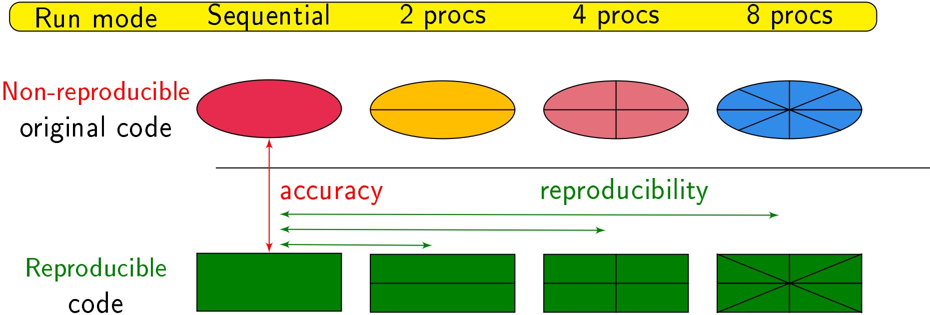
Reproducibility?

- Getting bitwise identical result for every p-parallel run; p>1
- Reproducibility $\neq$ accuracy



Reproducibility?

- Getting bitwise identical result for every p-parallel run; $p>1$
- Reproducibility $\neq$ accuracy



Reproducibility?

- Getting bitwise identical result for every p-parallel run; p>1
- Reproducibility $\neq$ accuracy

Motivation:

- Reproducibility failures in numerical simulation in many domains
- Debug, validate, test and receive legal agreement

Reproducibility failure of one industrial scale simulation code

- Simulation of free-surface flows in 1D-2D-3D hydrodynamics
- 300 000 loc. of open source Fortran 90
- 20 years, 4000 registered users, EDF R&D + international consortium

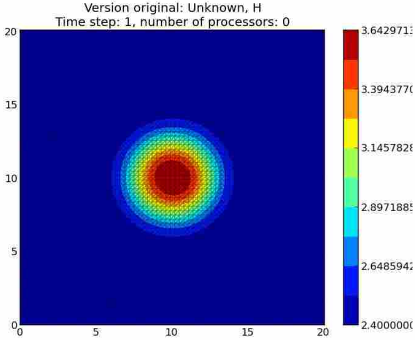
Telemac 2D

- 2D hydrodynamics: Saint Venant equations
- Finite element method, triangular element mesh, sub-domain decomposition for parallel resolution
- Mesh node unknowns: water depth (H) and velocity (U,V)

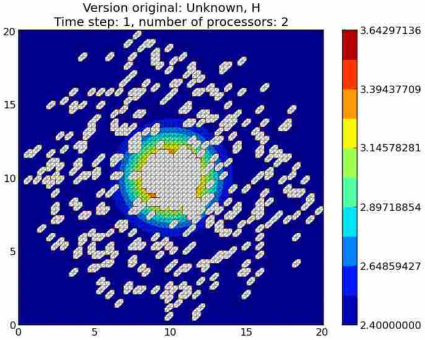
The *gouttedo* simulation

- 2D-simulation of a water drop fall in a square basin
- Unknown: water depth for a 0.2 sec time step
- Triangular mesh of 8978 elements and 4624 nodes

Non-reproducible result!



Sequential

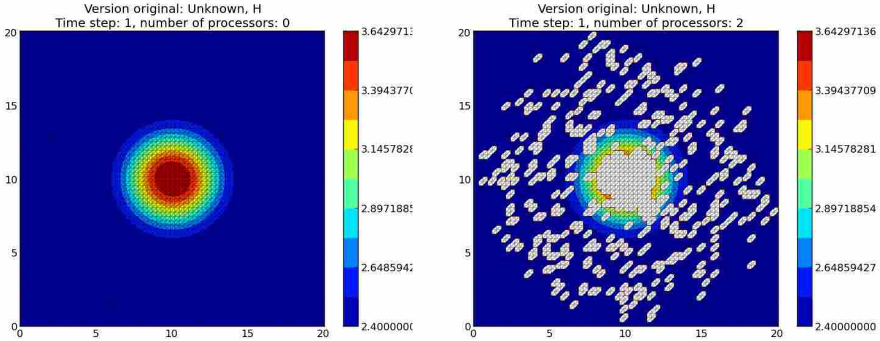


Parallel: 2 procs

A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

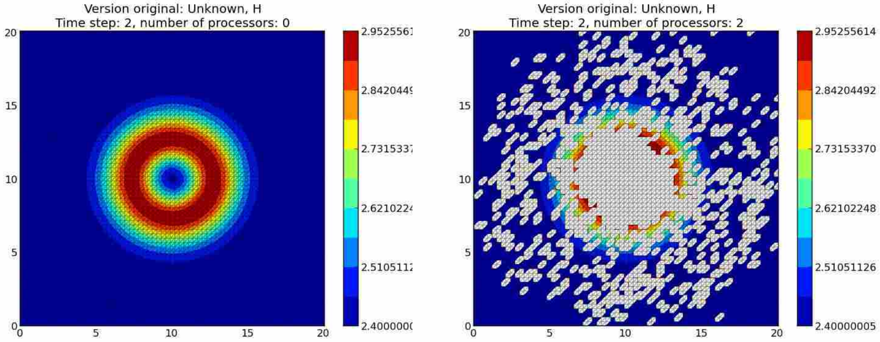
time step = 1



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

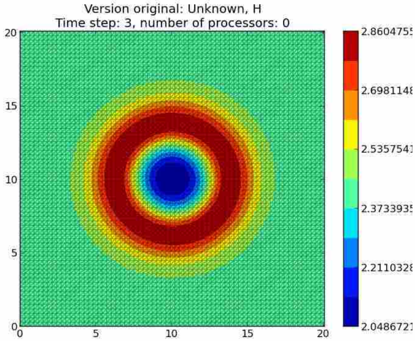
time step = 2

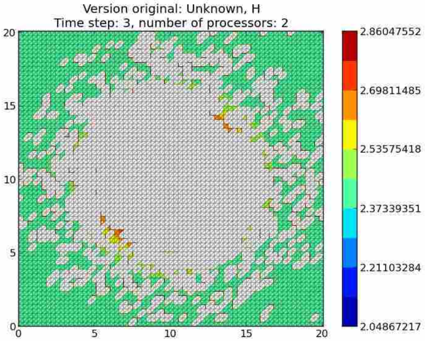


A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

time step = 3

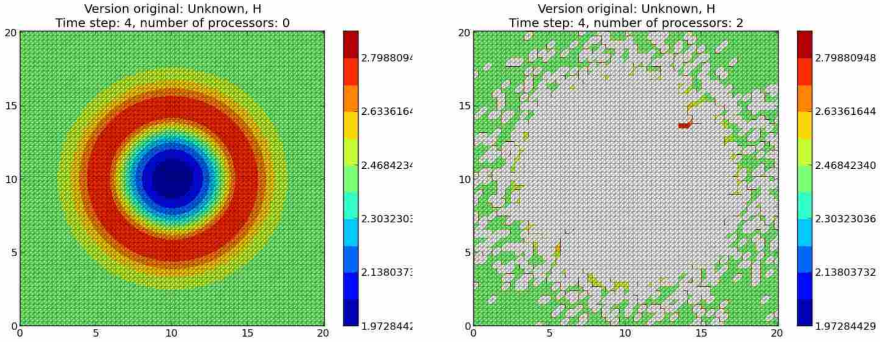




A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

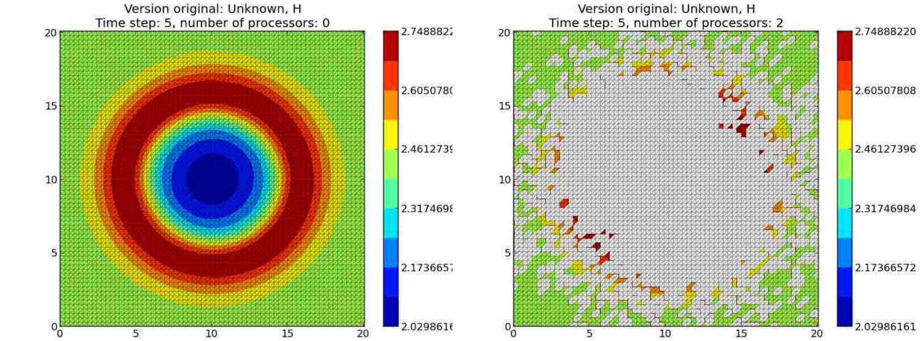
time step = 4



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

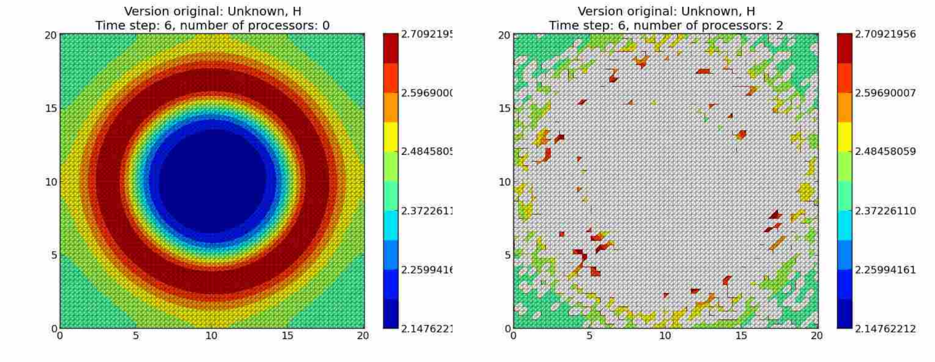
time step = 5



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

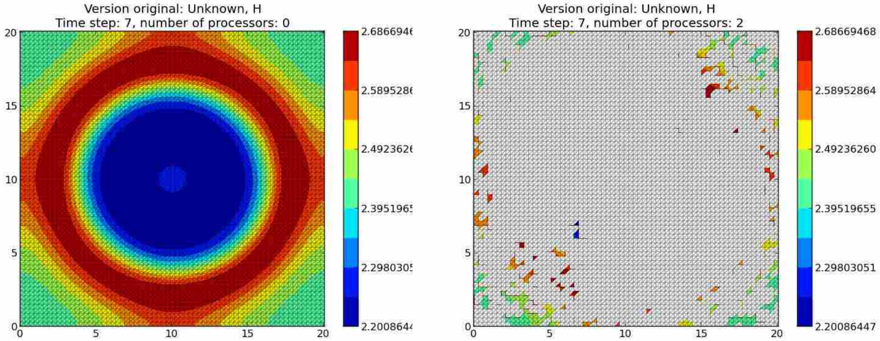
time step = 6



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

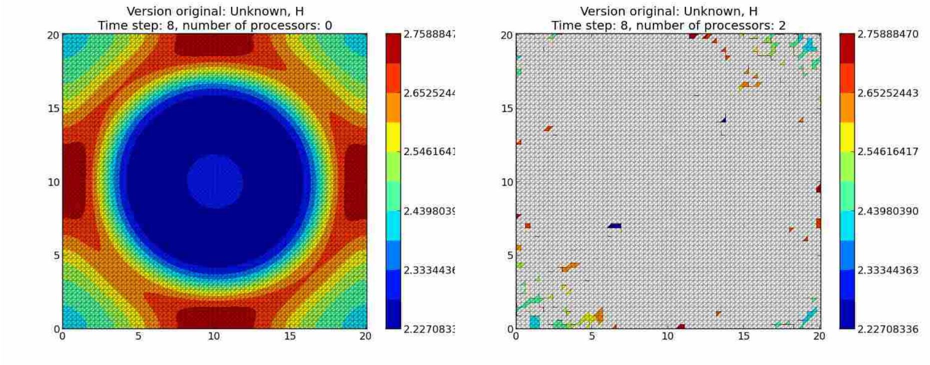
time step = 7



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

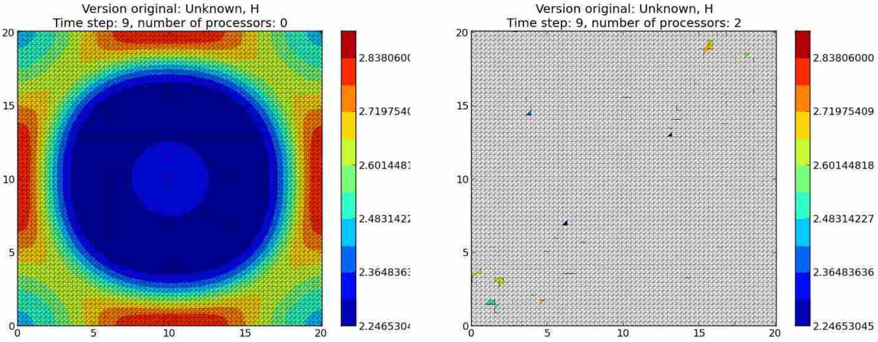
time step = 8



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

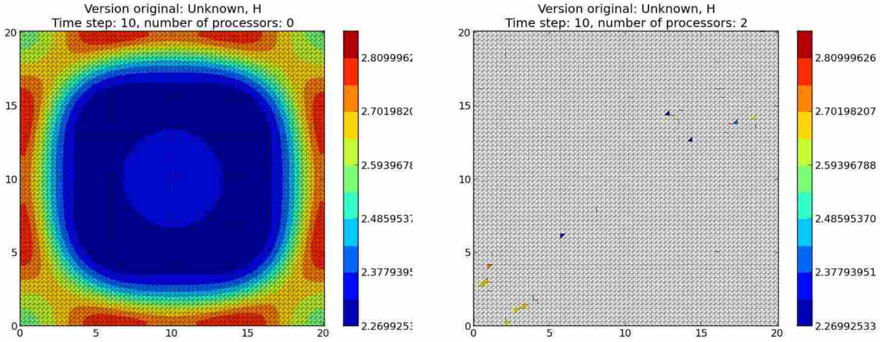
time step = 9



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

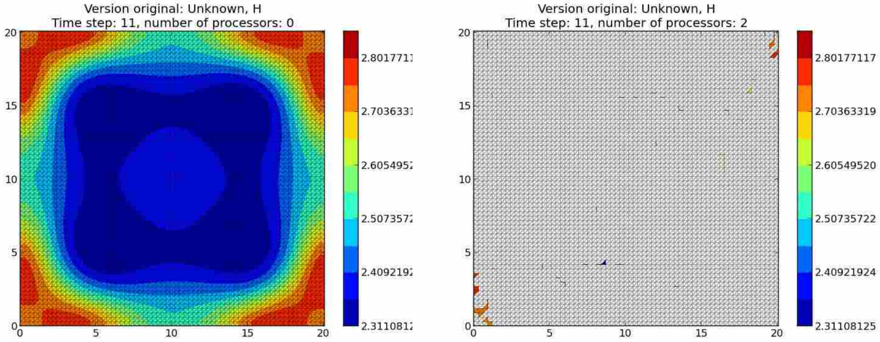
time step = 10



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

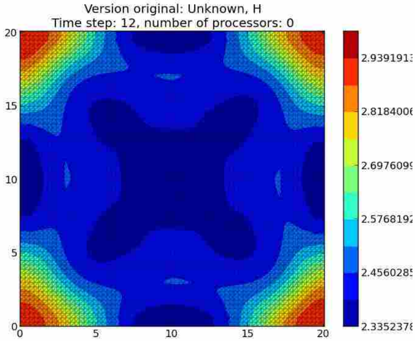
time step = 11

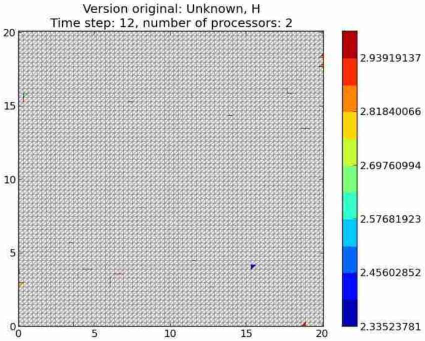


A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

time step = 12

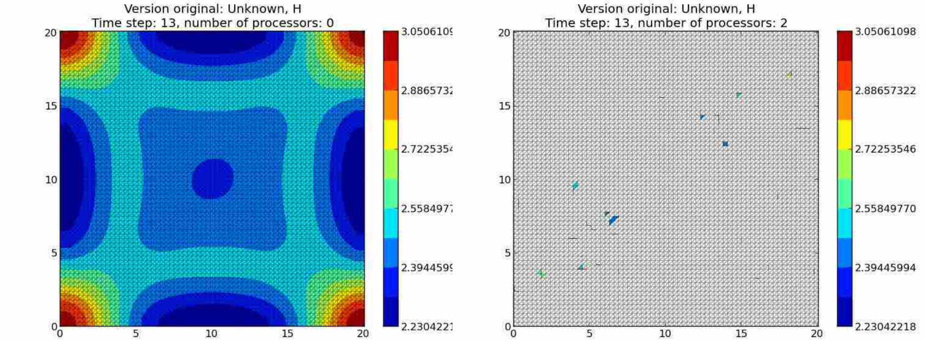




A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

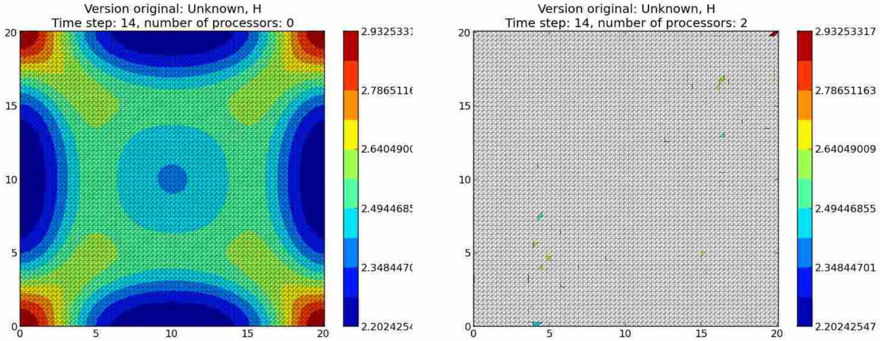
time step = 13



A white plot displays a non-reproducible value

Sequential *vs.* parallel (2 procs)

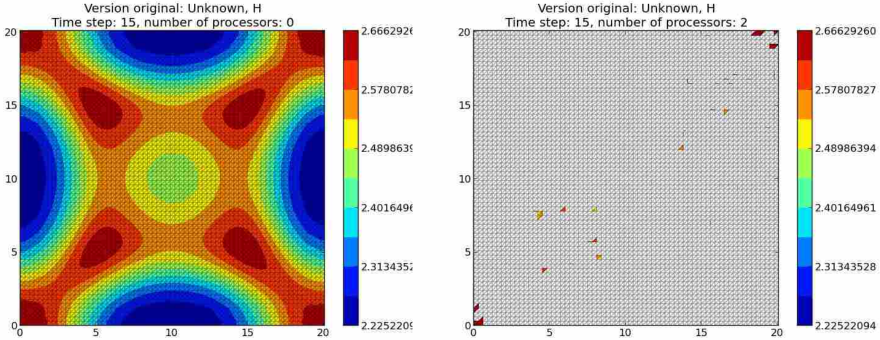
time step = 14



A white plot displays a non-reproducible value

Numerical reproducibility ?

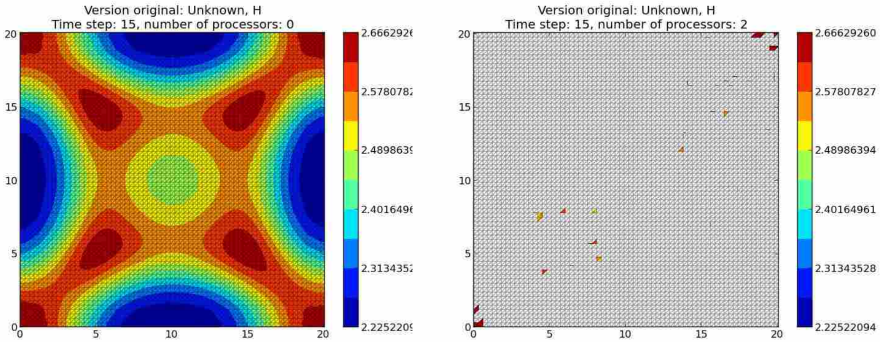
time step = 15



A white plot displays a non-reproducible value

Numerical reproducibility ?

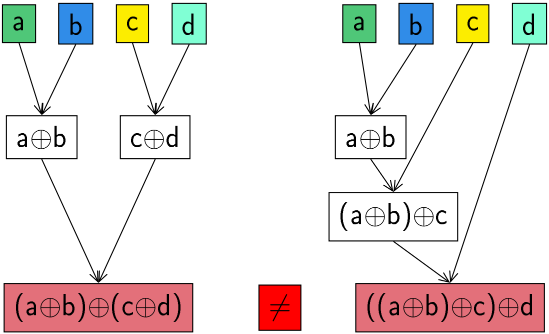
time step = 15



Weakness of floating point arithmetic

- Rounding errors, non associative floating-point addition
- The computed values depend on the operation order

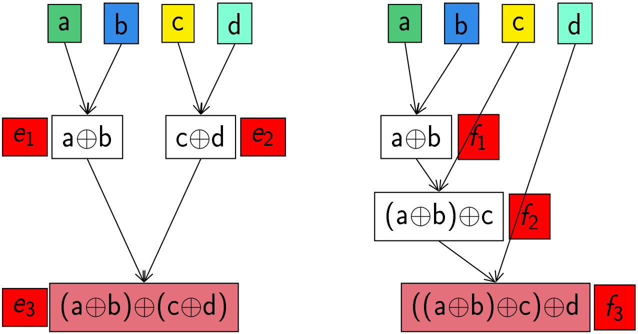
Parallel reduction: undefined reduction order ⇒ non-reproducibility



Weakness of floating point arithmetic

- Rounding errors, non associative floating-point addition
- The computed values depend on the operation order

Compensation principle



$$((a \oplus b) \oplus (c \oplus d)) \oplus e_1 \oplus e_2 \oplus e_3) = (((a \oplus b) \oplus c) \oplus d) \oplus ((f_1 \oplus f_2) \oplus f_3)$$

Rounding errors are computed with error-free transformations

function [x,y] = 2Sum(a,b)

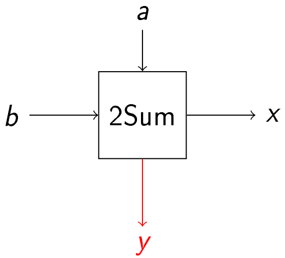
$x = a \oplus b$
$z = x \ominus a$
$y = (a \ominus (x \ominus z)) \oplus (b \ominus z)$

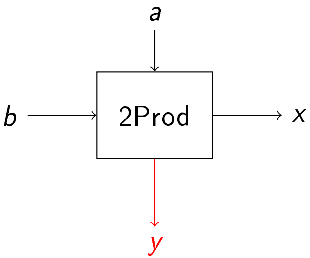
function [x,y] = 2Product(a,b)

$x = a \otimes b$
$[ah, al] = \text{Split}(a)$
$[bh, bl] = \text{Split}(b)$
$y = (al \otimes bl \ominus (((x \ominus ah.bh) \ominus al.bh) \ominus ah.bl))$

function [ah,al] = Split(a)

$c = 2^{27} + 1 \otimes a$
$ah = c \ominus (c \ominus a)$
$al = a \ominus ah$



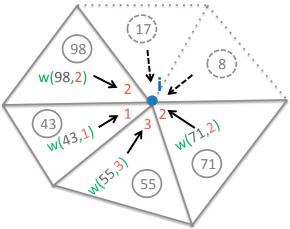


Plan

Finite element assembly: the sequential case

The assembly step: $V(i) = \sum_{el=1}^{nel} W_{el}(i)$

- computes the inner node values $V(i)$
- accumulating local W_{el} for every el that contains i



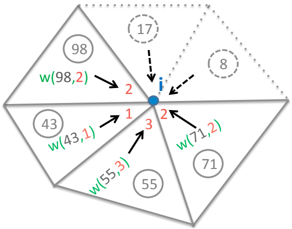
The assembly loop:

```
for  dp = 1, ndp          //dp: triangular local number(ndp=3)
  for el = 1, nel
     i = IKLE(el,  dp)
     V(i) = V(i) +  W(el, dp)   //i: domain global number
```

Finite element assembly: the sequential case

The assembly step: $V(i) = \sum_{el=1}^{nel} W_{el}(i)$

- computes the inner node values $V(i)$
- accumulating local W_{el} for every el that contains i



The assembly loop: managing local vs. global numbers

```
for  dp = 1, ndp          //dp: triangular local number(ndp=3)
  for el = 1, nel
     i = IKLE(el,  dp)        <-- LOOP INDEX INDIRECTION
     V(i) = V(i) +  W(el, dp)   //i: domain global number
```

Finite element assembly: the parallel case

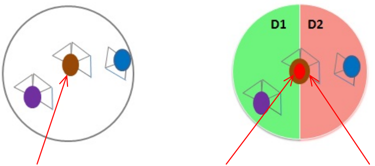
Parallel FE: sub-domain decomposition

IP assembly: communications and reductions

$V(i) = \sum_{D_k} V(i)$ for sub-domains $D_k, k = 1 \ldots p$

Exact arithmetic

sequential parallel



$V(i) = a$ $V_{D_1}(i) = b$ $V_{D_2}(i) = c$

Interface point assembly

$V(i) = b + c = a$

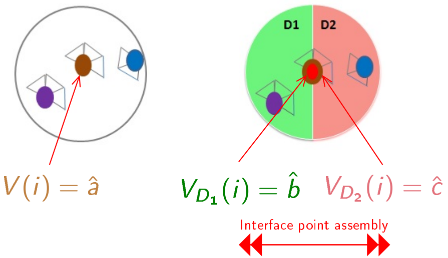
Parallel FE: sub-domain decomposition

IP assembly: communications and reductions

$V(i) = \sum_{D_k} V(i)$ for sub-domains $D_k, k = 1 \ldots p$

Floating point arithmetic

sequential parallel



$$V(i) = \hat{b} \oplus \hat{c} \neq \hat{a}$$

Plan

Sources of non reproducibility in Telemac-2D

Culprits: theory

1. Building step: finite element assembly
2. Solving step: parallel matrix-vector and dot products

Sources of non reproducibility in Telemac-2D

Culprits: theory

1. Building step: finite element assembly
2. Solving step: parallel matrix-vector and dot products

Culprits: practice = optimizations

- Interface point assembly and linear system solving are merged
- Element-by-element (EBE) storage of matrix
    - No BLAS parallel matrix-vector product
    - Everything is vector, no matrix!
- Wave equation, "mass-lumping" and associated algebraic transformations
    - Many diagonal matrices
    - Everything is vector, no matrix!

Sources of non reproducibility in Telemac-2D

Solution U,V

diagonal resolution

Solution H

conjugate gradient

System equation $Ay = b$

$$\begin{pmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{pmatrix} \begin{pmatrix} H \\ U \\ V \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}$$

wave equation

$$\begin{pmatrix} A_{hh} & A_{hu} & A_{hv} \\ A_{uh} & A_{uu} & 0 \\ A_{vh} & 0 & A_{vv} \end{pmatrix} \begin{pmatrix} H \\ U \\ V \end{pmatrix} = \begin{pmatrix} B_h \\ B_u \\ B_v \end{pmatrix}$$

FE assembly +

algebraic computation

Mesh (elements, nodes)

Discretization

Continuous domain

$C_2 = B_u - A_{uh}H,$

$C_3 = B_v - A_{vh}H.$

Interface point assembly:

C_2, C_3

A_2, A_3

Interface point assembly:

A_1 in each iteration

H

A_1, C_1

A_2, A_3 : diagonal matrices,

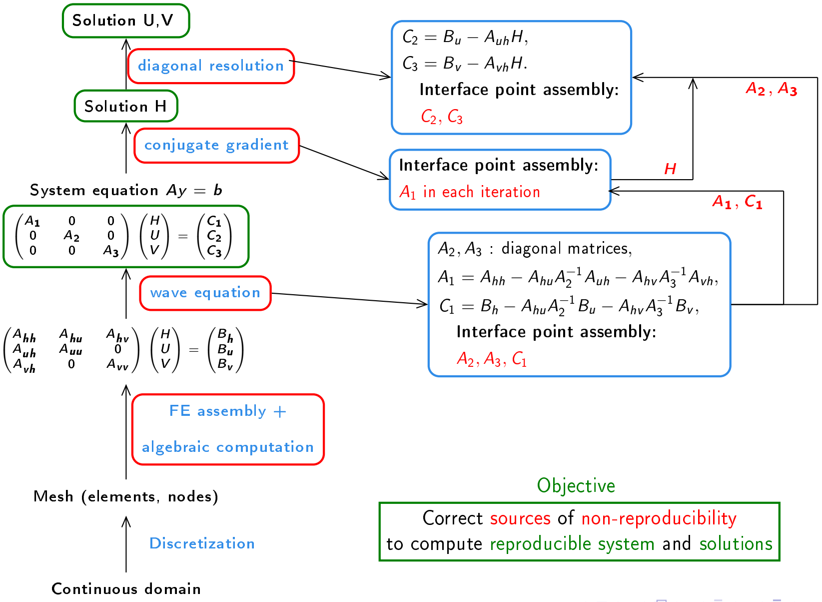
$A_1 = A_{hh} - A_{hu}A_2^{-1}A_{uh} - A_{hv}A_3^{-1}A_{vh},$

$C_1 = B_h - A_{hu}A_2^{-1}B_u - A_{hv}A_3^{-1}B_v,$

Interface point assembly:

A_2, A_3, C_1

Sources of non reproducibility in Telemac-2D



Objective

Correct sources of non-reproducibility
to compute reproducible system and solutions

Plan

Recovering reproducibility in Telemac2D

Sources

- FE assembly: diagonal of matrix and second member
- Resolution: matrix-vector and dot products
- Wave equation: algebraic transformations and diagonal resolutions

Reproducible resolution: principles

- vector $V \rightarrow [V, E_V] \rightarrow V + E_V$
- Computes E_V in the FE assembly of V
- Propagates E_V over each V operation
- Compensates all nodes while assembling the Interface Point
- Compensates the MPI parallel dot products that include MPI reduction

Plan

Reproducible FE assembly

- Contribution $W_{el}(i)$ for every element el belonging to the node i

Original FE assembly: $V(i) = \sum_{elements} W_{el}(i)$

$$V(i) = W_{el_1}(i) + W_{el_2}(i) + \cdots + W_{el_{ni}}(i)$$

Reproducible FE assembly: $V(i) = \text{ReprodAss}_{elements} W_{el}(i)$

$$V(i) = W_{el_1}(i) + W_{el_2}(i) + \cdots + W_{el_{ni}}(i)$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$e_1 \quad e_2 \quad e_{ni}$$

$$E_V(i) = e_1 + e_2 + \cdots + e_{ni-1}$$

Reproducible interface point assembly

- i is an interface point of $D_1, D_2, \cdots, D_{k-1}, D_k$

Original IP assembly: $V(i) = \sum_{D_k} V(i)$

$$V(i) = V_{D_1}(i) + V_{D_2}(i) + \cdots + V_{D_{k-1}}(i) + V_{D_k}(i)$$

Reproducible IP assembly: $V(i) = \text{ReprodAss}_{D_k} V(i)$

$$V(i) = V_{D_1}(i) + \underset{\substack{\downarrow \\ \delta_1}}{V_{D_2}(i)} + \underset{\substack{\downarrow \\ \delta_2}}{\cdots} + \underset{\substack{\downarrow \\ \delta_{k-1}}}{V_{D_{k-1}}(i)} + V_{D_k}(i)$$

$$E_V(i) = (E_{V_{D_1}}(i) + E_{V_{D_2}}(i) + \delta_1) + \cdots + (E_{V_{D_{k-1}}}(i) + E_{V_{D_k}}(i) + \delta_{k-1})$$

The compensation $V + E_V$

Plan

Reproducible vector algebraic operations

- In the library BIEF of Telemac (**BI**bliothèque d'**E**léments **F**inis)
- Entry-wise vector ops: copy, add, sub, Hadamard prod,...
- Applies also for diagonal of matrix
- **Propagate rounding errors to compensate while assembling IP**

Ex. Hadamard product

Original version

$$V(i) = Y(i) \circ Z(i)$$

Reproducible version

$$[V, E_V] = [X, E_X] \circ [Y, E_Y]$$

with:

$$[V(i), e(i)] = 2Prod(X(i), Y(i)),$$

$$E_V = X \circ E_Y + Y \circ E_X + e$$

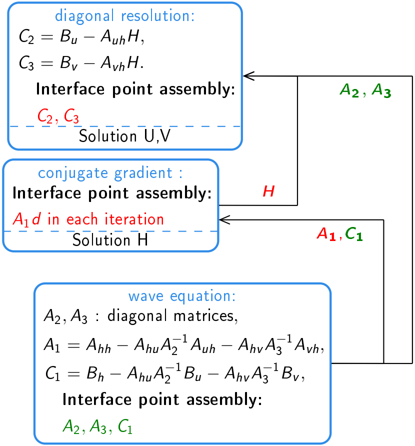
Partially reproducible Telemac-2D

What is reproducible now?

- Most of the linear system:
  - FE assembly
  - algebraic vector operations
  - interface point assembly
- Except:
  - the matrix of the H system
  - its dependencies: the second members of the U and V systems

What else?

- conjugate gradient



Plan

Towards a reproducible conjugate gradient

A=[A_1, E_{A_1}] → ; B=C_1 → ; → X=H

Initialization:

$$r^0 = AX^0 - B; \text{ a given } d^0$$

$$\rho^0 = \frac{(r^0, d^0)}{(Ad^0, d^0)}; \quad X^1 = X^0 - \rho^0 d^0$$

Iterations until stopping criterion:

$$r^m = r^{m-1} - \rho^{m-1} Ad^{m-1}$$

$$d^m = r^m + \frac{(r^m, r^m)}{(r^{m-1}, r^{m-1})} d^{m-1}$$

$$\rho^m = \frac{(r^m, d^m)}{(d^m, Ad^m)}$$

$$X^{m+1} = X^m - \rho^m d^m$$

Non-reproducibility: sources

1. EBE matrix-vector product
2. Dot product
    - MPI reduction
    - the weighting of interface point shared by p sub-domains

The EBE storage

$$M = D + \sum_{el=1}^{nel} X_{el}$$

- nodes $i \in [1, np]$, elements $el \in [1, nel]$, element vertices $i_1, i_2, i_3 \in el$
- M is decomposed as:
  1. diagonal $D[np]$

$$D = [D(1), \cdots\cdots, D(np)]$$

  2. elementary extra-diagonal $X_{el}[6]$

$$X_{el} = \begin{pmatrix} & X_{i_1 i2}(el) & X_{i_1 i3}(el) \\ X_{i_2 i1}(el) & & X_{i_2 i3}(el) \\ X_{i_3 i1}(el) & X_{i_3 i2}(el) & \end{pmatrix}$$

The EBE storage

$$M = D + \sum_{el=1}^{nel} X_{el}$$

- nodes $i \in [1, np]$, elements $el \in [1, nel]$, element vertices $i_1, i_2, i_3 \in el$
- M is decomposed as:
  1. diagonal $D[np]$

     $$D = [D(1), \cdots\cdots, D(np)]$$

  2. elementary extra-diagonal $X_{el}[6]$

     $$X_{el} = [X(el, 1), \cdots X(el, 6)]$$

The EBE storage

$$M = D + \sum_{el=1}^{nel} X_{el}$$

- nodes $i \in [1, np]$, elements $el \in [1, nel]$, element vertices $i_1, i_2, i_3 \in el$
- M is decomposed as:
  1. diagonal $D[np]$

$$D = [D(1), \cdots\cdots, D(np)]$$

  2. elementary extra-diagonal $X_{el}[6] \rightarrow 6 \times nel$

$$X = [X_{el_1}, \cdots X_{el_{nel}}]$$

The EBE Matrix-Vector product

$$RES = M \cdot V = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

Steps of the EBE Matrix-Vector product

1. i $\in [1, np]$ $\qquad R_1(i) = D(i) \cdot V(i)$

The EBE Matrix-Vector product

$$RES = M \cdot V = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

Steps of the EBE Matrix-Vector product

1. i $\in [1, np]$ $\qquad R_1(i) = D(i) \cdot V(i)$
2. $el \in [1, nel]$ $\qquad X_{el}.V_{el} = [X(el,1).V(i_2), X(el,2).V(i_3), \cdots X(el,6).V(i_2)]$

The EBE Matrix-Vector product

$$RES = M \cdot V = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

Steps of the EBE Matrix-Vector product

1. i $\in [1, np]$ $\quad R_1(i) = D(i) \cdot V(i)$
2. $el \in [1, nel]$ $\quad X_{el}.V_{el} = [X(el,1).V(i_2), X(el,2).V(i_3), \cdots X(el,6).V(i_2)]$
3. FE assembly to a vector $R_2[np]$

$$R_2 = \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

The EBE Matrix-Vector product

$$RES = M \cdot V = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

Steps of the EBE Matrix-Vector product

1. i $\in [1, np]$ $\quad R_1(i) = D(i) \cdot V(i)$
2. $el \in [1, nel]$ $\quad X_{el}.V_{el} = [X(el,1).V(i_2), X(el,2).V(i_3), \cdots X(el,6).V(i_2)]$
3. FE assembly to a vector $R_2[np]$

$$R_2 = \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

4. $RES = R_1 + R_2$

The EBE Matrix-Vector product

$$RES = M \cdot V = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

Steps of the EBE Matrix-Vector product

1. i $\in [1, np]$ $\quad R_1(i) = D(i) \cdot V(i)$
2. $el \in [1, nel]$ $\quad X_{el}.V_{el} = [X(el,1).V(i_2), X(el,2).V(i_3), \cdots X(el,6).V(i_2)]$
3. FE assembly to a vector $R_2[np]$

$$R_2 = \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

4. $RES = R_1 + R_2$
5. when i is an IP $\quad RES(i) = \sum_{D_k} RES(i)$

The EBE Matrix-Vector product

$$RES = M \cdot V = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

Steps of the EBE Matrix-Vector product

1. i $\in [1, np]$ $\qquad R_1(i) = D(i) \cdot V(i)$
2. $el \in [1, nel]$ $\qquad X_{el}.V_{el} = [X(el,1).V(i_2), X(el,2).V(i_3), \cdots X(el,6).V(i_2)]$
3. FE assembly to a vector $R_2[np]$

$$R_2 = \sum_{el=1}^{nel} X_{el} \cdot V_{el}$$

4. $RES = R_1 + R_2$
5. when i is an IP

IP assembly

$$RES(i) = RES_{D_1}(i) + RES_{D_2}(i) + \cdots + RES_{D_{k-1}}(i) + RES_{D_k}(i)$$

Reproducible matrix-vector product

Original matrix-vector product

- $\mathrm{RES} = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V_{el}$
- $RES(i) = \sum_{D_k} RES(i)$

Reproducible matrix-vector product

- $[\mathrm{RES}, E_{\mathrm{RES}}] = [D, E_D] \circ V + \mathrm{ReprodAss}_{el=1}^{nel} X_{el} \cdot V_{el}$
- $RES(i) = \mathrm{ReprodAss}_{D_k}[RES(i), E_{\mathrm{RES}}(i)]$
- The compensation $\mathrm{RES} + E_{\mathrm{RES}}$

Towards a reproducible conjugate gradient

Non-reproducibility: sources and solutions

1. EBE matrix-vector product
   - reproducible FE and IP assembly
2. Dot product
   - MPI reduction : a parallel version of the compensated dot2
   - The IP weighting : $(1/k, 1/k, \ldots, 1/k) \rightarrow (1, 0, \ldots, 0)$

A=[A_1, E_{A_1}] →

B=C_1 →

Initialization:

$r^0 = AX^0 - B$; a given d^0

$\rho^0 = \frac{(r^0, d^0)}{(Ad^0, d^0)}$; $X^1 = X^0 - \rho^0 d^0$

Iterations until stopping criterion:

$r^m = r^{m-1} - \rho^{m-1} Ad^{m-1}$

$d^m = r^m + \frac{(r^m, r^m)}{(r^{m-1}, r^{m-1})} d^{m-1}$

$\rho^m = \frac{(r^m, d^m)}{(d^m, Ad^m)}$

$X^{m+1} = X^m - \rho^m d^m$

→ X=H

Reproducible operations → Reproducible results

Towards a reproducible conjugate gradient

Non-reproducibility: sources and solutions

1. EBE matrix-vector product
   - reproducible FE and IP assembly
2. Dot product
   - MPI reduction : a parallel version of the compensated `dot2`
   - The IP weighting : $(1/k, 1/k, \dots, 1/k) \to (1, 0, \dots, 0)$

A=$[A_1, E_{A_1}]$ →

B=C_1 →

```
Initialization:
```

$$r^0 = AX^0 - B; \text{ a given } d^0$$

$$\rho^0 = \frac{(r^0, d^0)}{(Ad^0, d^0)}; \quad X^1 = X^0 - \rho^0 d^0$$

```
Iterations until stopping criterion:
```

$$r^m = r^{m-1} - \rho^{m-1} A d^{m-1}$$

$$d^m = r^m + \frac{(r^m, r^m)}{(r^{m-1}, r^{m-1})} d^{m-1}$$

$$\rho^m = \frac{(r^m, d^m)}{(d^m, Ad^m)}$$

$$X^{m+1} = X^m - \rho^m d^m$$

→ X=H

Same errors in the compensated values
for both sequential and parallel executions

Reproducible *gouttedo*

Telemac-2D finite element method

diagonal resolution:

$C_2 = B_u - A_{uh}H,$

$C_3 = B_v - A_{vh}H.$

Interface point assembly:

C_2, C_3

Solution U,V

A_2, A_3

conjugate gradient :

Interface point assembly:

$A_1 d$; in each iteration

Solution H

H

A_1, C_1

wave equation:

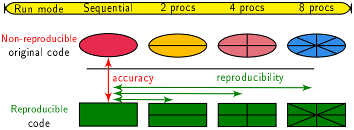
A_2, A_3:diagonal matrices,

$A_1 = A_{hh} - A_{hu}A_2^{-1}A_{uh} - A_{hv}A_3^{-1}A_{vh},$

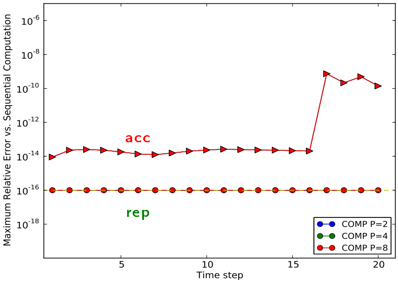
$C_1 = B_h - A_{hu}A_2^{-1}B_u - A_{hv}A_3^{-1}B_v,$

Interface point assembly:

A_2, A_3, C_1



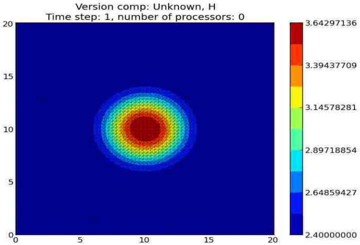
Maximum relative error, *gouttedo* test case



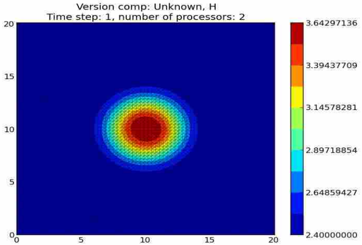
No white spots ⇒ reproducibility everywhere

Time step 1

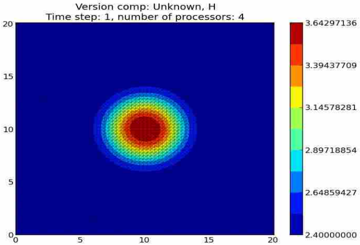
p=1



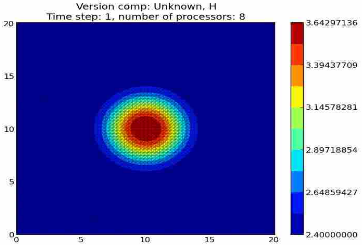
p=2



p=4



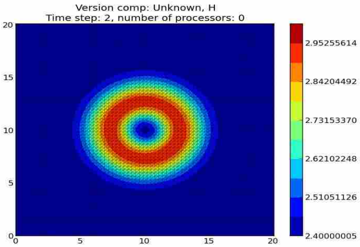
p=8



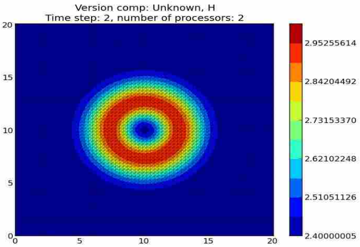
No white spots ⇒ reproducibility everywhere

Time step 2

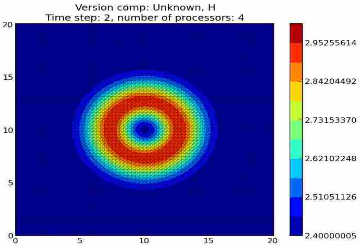
p=1



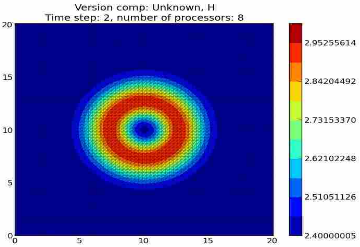
p=2



p=4



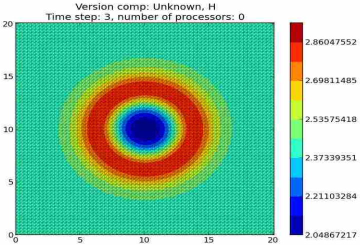
p=8



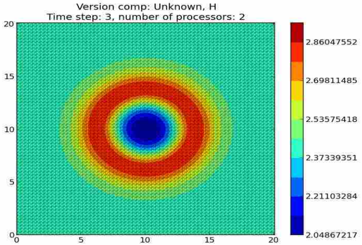
No white spots ⇒ reproducibility everywhere

Time step 3

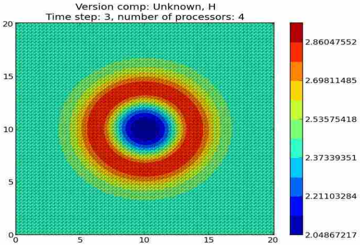
p=1



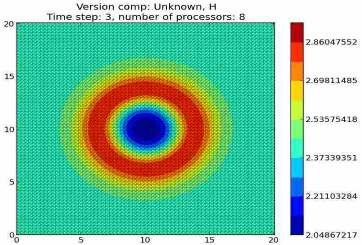
p=2



p=4



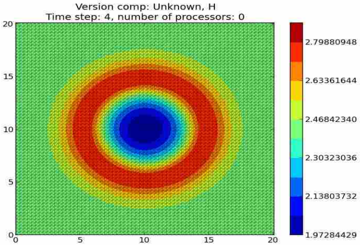
p=8



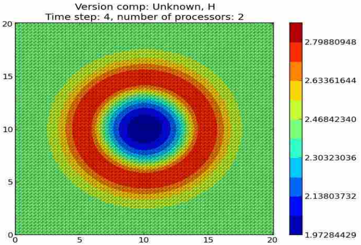
No white spots ⇒ reproducibility everywhere

Time step 4

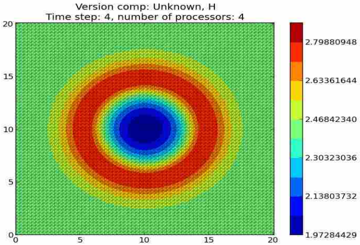
p=1



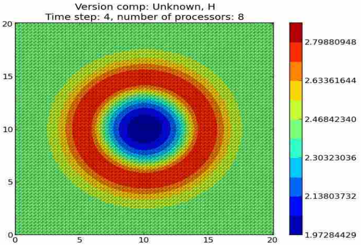
p=2



p=4



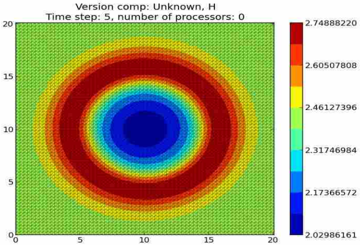
p=8



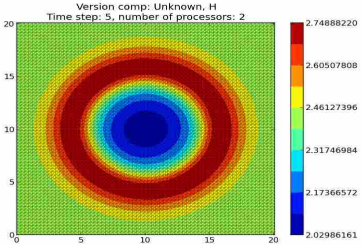
No white spots ⇒ reproducibility everywhere

Time step 5

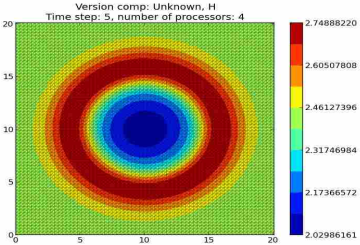
p=1



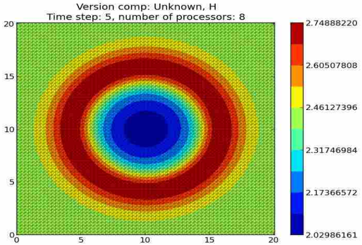
p=2



p=4



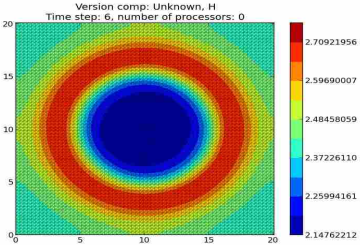
p=8



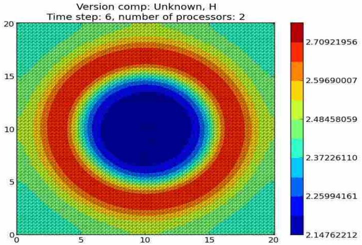
No white spots ⇒ reproducibility everywhere

Time step 6

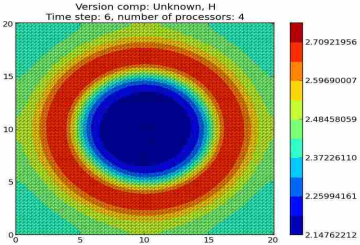
p=1



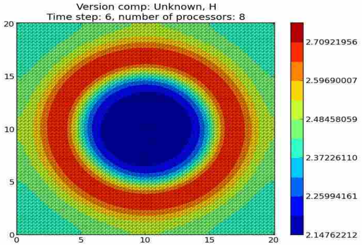
p=2



p=4



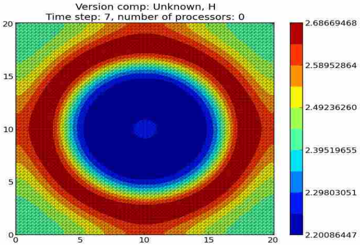
p=8



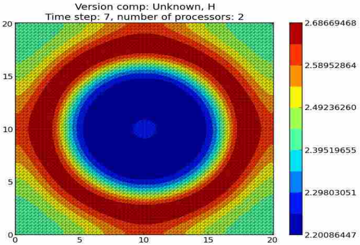
No white spots ⇒ reproducibility everywhere

Time step 7

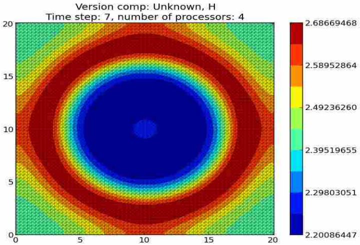
p=1



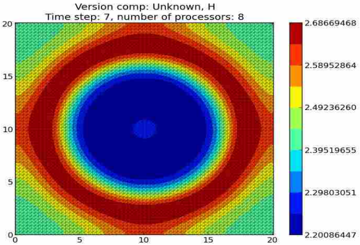
p=2



p=4



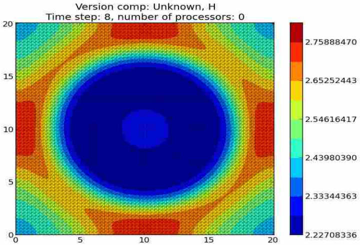
p=8



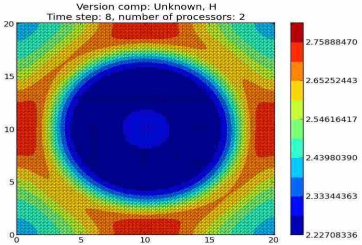
No white spots ⇒ reproducibility everywhere

Time step 8

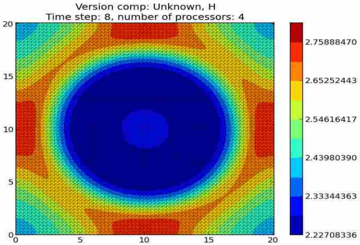
p=1



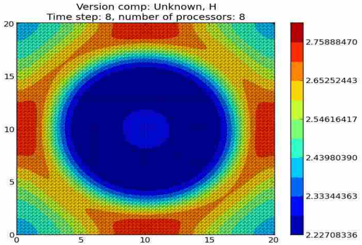
p=2



p=4



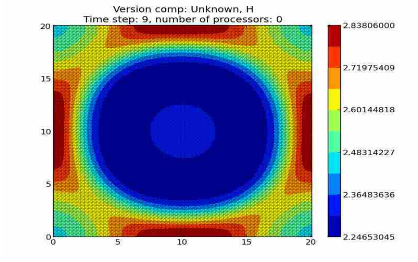
p=8



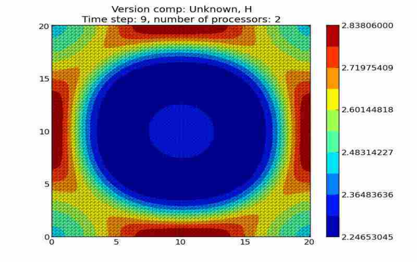
No white spots ⇒ reproducibility everywhere

Time step 9

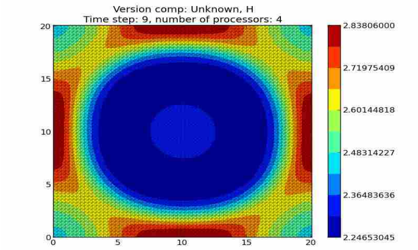
p=1



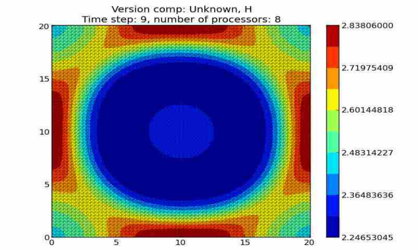
p=2



p=4



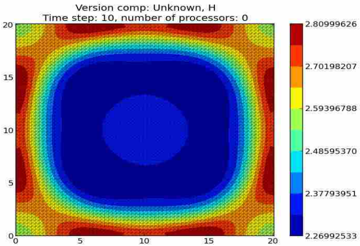
p=8



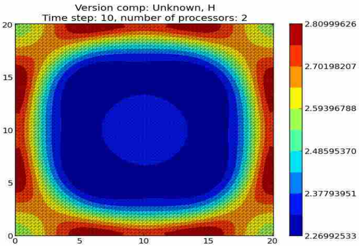
No white spots ⇒ reproducibility everywhere

Time step 10

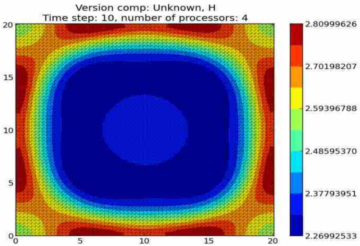
p=1



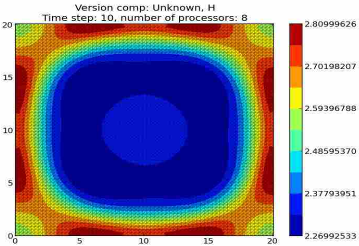
p=2



p=4



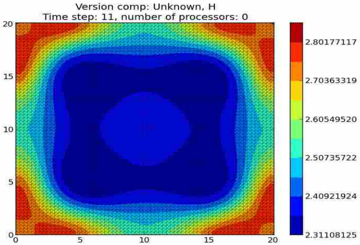
p=8



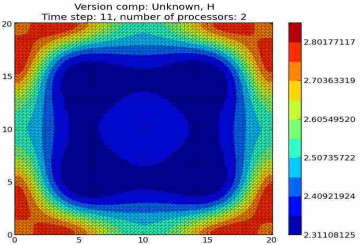
No white spots ⇒ reproducibility everywhere

Time step 11

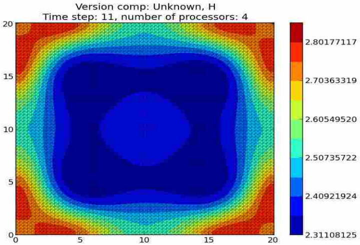
p=1



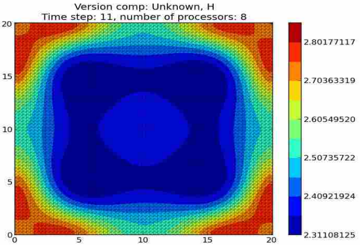
p=2



p=4



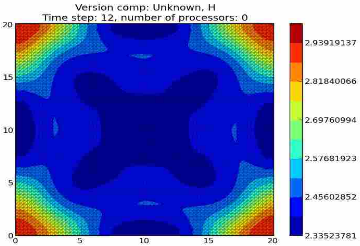
p=8



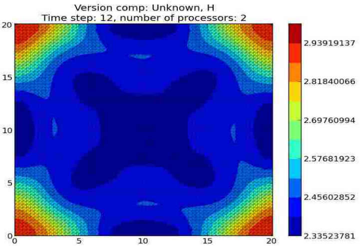
No white spots ⇒ reproducibility everywhere

Time step 12

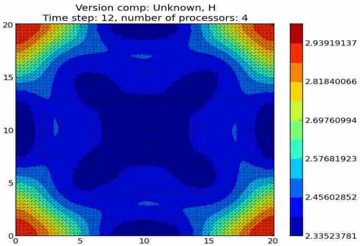
p=1



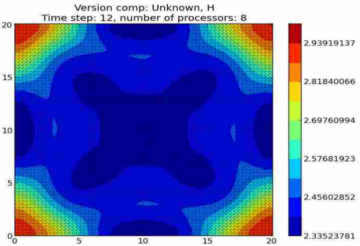
p=2



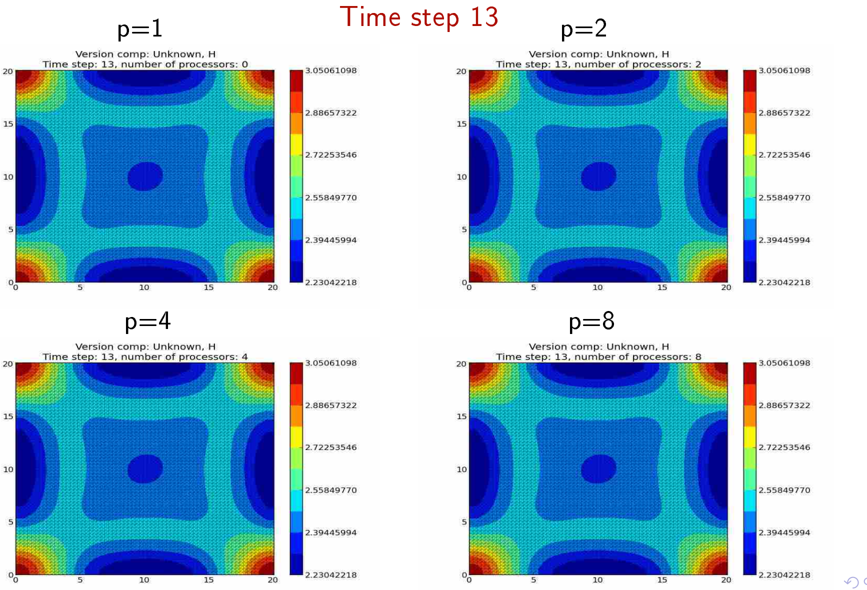
p=4



p=8



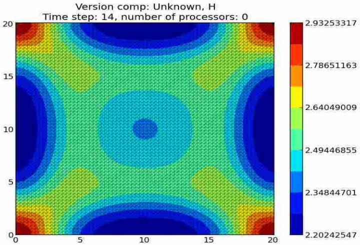
No white spots ⇒ reproducibility everywhere



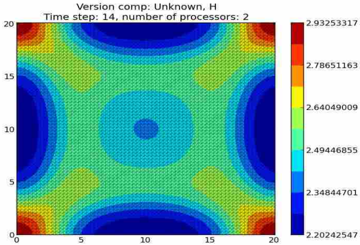
No white spots ⇒ reproducibility everywhere

Time step 14

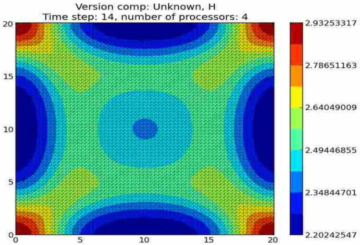
p=1



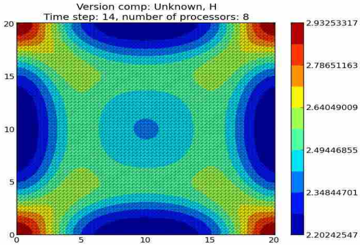
p=2



p=4



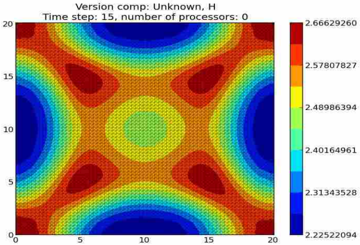
p=8



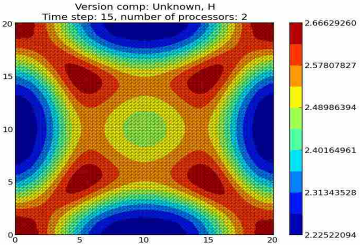
No white spots ⇒ reproducibility everywhere

Time step 15

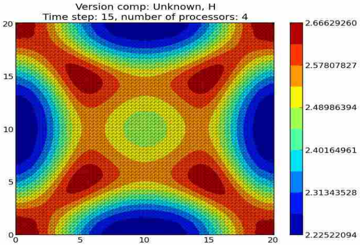
p=1



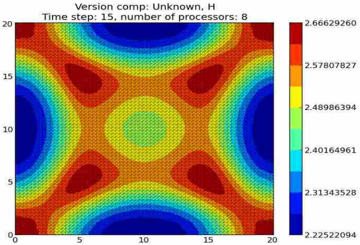
p=2



p=4



p=8



Plan

Runtime extra-cost for reproducible simulations

Measures, test case and mesh size

- hardware cycle counter: *rdtsc*
- Telemac v7.2, *gouttedo*
- mesh sizes: 4624, 18225, 72361

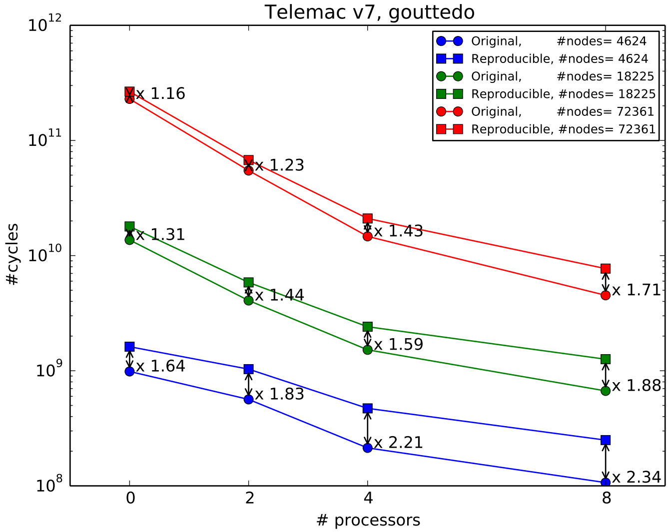
#procs	#IP	#nodes 4624	#nodes 18225	#nodes 72361
	2	72	143	280
	4	304	674	1368
	8	501	1152	2020

Hardware and software env.

- socket: Intel Xeon E5-2660 2.20GHz (L3 cache = 20 M)
- 2 sockets of 8 cores each
- GNU Fortran 4.6.3, -O3
- OpenMPI 1.5.4
- Linux 3.5.0-54-generic

The core runtime extra-cost for reproducible *gouttedo*

The core: no input/output, just building and solving steps



Plan

Conclusion

Feasibility

- Sources of non-reproducibility in a FE simulation?
- How to recover reproducibility? How easily?
- Compensation yields reproducibility here!
- Other existing techniques are also less or more efficient.

Conclusion

Feasibility

- Sources of non-reproducibility in a FE simulation?
- How to recover reproducibility? How easily?
- Compensation yields reproducibility here!
- Other existing techniques are also less or more efficient.

Efficiency

- How much to pay for reproducibility?
- $\times 1.2 \leftrightarrow \times 2.3$ extra-cost which decreases as the problem size increases

Conclusion

Reproducibility at the large scale: the open-Telemac case

- The test cases are significant enough to validate the methodology
- Integration in the next open-Telemac version is in progress

Conclusion

Reproducibility at the large scale: the open-Telemac case

- The test cases are significant enough to validate the methodology
- Integration in the next open-Telemac version is in progress

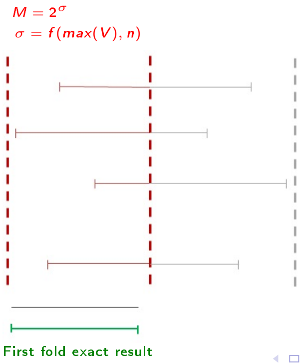
- Difficult to automatize
- Pass the methodology to software developers

Merci pour votre attention

Demmel-Nguyen's reproducible sum (2013)

A parallel K-fold reproducible summation V[np]

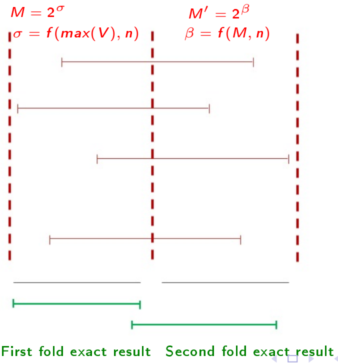
- Exact sum of shrunks defined thanks to:
  - $max|v_i|$ and np
- 2 reductions : max and sum
- K-fold process $\Rightarrow$ more accuracy



Demmel-Nguyen's reproducible sum (2013)

A parallel K-fold reproducible summation V[np]

- Exact sum of shrunks defined thanks to:
  - $max|v_i|$ and np
- 2 reductions : max and sum
- K-fold process $\Rightarrow$ more accuracy



Demmel-Nguyen's reproducible sum (2013)

The assembly loop:

```
for dp = 1, ndp //dp: triangular local number(ndp=3) for el
= 1, nel i = IKLE(el, dp)                              V(i) =
V(i) + W(el,dp) //i: domain global number
```

Loop index indirection forces

- 2 iterations $nel \times ndp$
  1. recover max $W_{el}(i)$
  2. reproducible accumulation $V(i)$
- 2 communications by IP
  1. share max $W_{el}(i)$
  2. assemble $V(i)$

Demmel-Nguyen's reproducible sum (2013)

The assembly loop:

```
for dp = 1, ndp //dp: triangular local number(ndp=3) for el
= 1, nel i = IKLE(el, dp)                            V(i) =
V(i) + W(el,dp) //i: domain global number
```

Loop index indirection forces

- 2 iterations $nel \times ndp$
  1. recover max $W_{el}(i)$
  2. reproducible accumulation $V(i)$
- 2 communications by IP
  1. share max $W_{el}(i)$
  2. assemble $V(i)$

Finite element assembly: the parallel case

Parallel FE: sub-domain decomposition

IP assembly: communications and reductions

$V(i) = \sum_{D_k} V(i)$ for sub-domains $D_k, k = 1 \ldots p$

Original interface point assembly (practically)

$$V_{D_1}(i) = V_{D_1}(i) + V_{D_2}(i) + \cdots + V_{D_{k-1}}(i) + V_{D_k}(i)$$
$$V_{D_2}(i) = V_{D_2}(i) + V_{D_1}(i) + \cdots + V_{D_{k-1}}(i) + V_{D_k}(i)$$
$$V_{D_{k-1}}(i) = V_{D_{k-1}}(i) + V_{D_1}(i) + V_{D_2}(i) + \cdots + V_{D_k}(i)$$
$$V_{D_k}(i) = V_{D_k}(i) + V_{D_1}(i) + V_{D_2}(i) + \cdots + V_{D_{k-1}}(i)$$

Reproducibility failure by varying k

Reproducibility of the conjugate gradient

1. Reproducible matrix-vector product
2. Reproducible dot product
   - the weighting
   - the MPI reduction

Reproducibility of the conjugate gradient

1. Reproducible matrix-vector product

Original Matrix-Vector product

- $\text{RES} = D \cdot V + \sum_{el=1}^{nel} X_{el} \cdot V$
- $RES(i) = \sum_{D_k} RES(i)$

Reproducible Matrix-Vector product

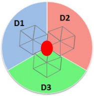
- $[\text{RES}, E_{\text{RES}}] = [D, E_D] \circ V + \text{ReprodAss}_{el=1}^{nel} X_{el} \cdot V$
- $RES(i) = \text{ReprodAss}_{D_k}[RES(i), E_{\text{RES}}(i)]$
- $\text{RES} + E_{\text{RES}}$

2. Reproducible dot product
    - the weighting
    - the MPI reduction

Reproducibility of the conjugate gradient

1. Reproducible matrix-vector product
2. Reproducible dot product
   - the weighting

$$\begin{pmatrix}[x_j]\\ x_i\end{pmatrix}\cdot\begin{pmatrix}[y_j]\\ y_i\end{pmatrix} \mid \begin{pmatrix}[x_l]\\ x_i\end{pmatrix}\cdot\begin{pmatrix}[y_l]\\ y_i\end{pmatrix} \mid \begin{pmatrix}[x_t]\\ x_i\end{pmatrix}\cdot\begin{pmatrix}[y_t]\\ y_i\end{pmatrix}$$



Original version

$\sum_j x_j.y_j + \frac{1}{3}x_i.y_i$

$\sum_l x_l.y_l + \frac{1}{3}x_i.y_i$

$\sum_t x_t.y_t + \frac{1}{3}x_i.y_i$

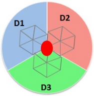
$\frac{1}{3}x_i.y_i \rightarrow$ rounding error!

   - the MPI reduction

Reproducibility of the conjugate gradient

1. Reproducible matrix-vector product
2. Reproducible dot product
   - the weighting

$$\begin{pmatrix}[x_j]\\ x_i\end{pmatrix} \cdot \begin{pmatrix}[y_j]\\ y_i\end{pmatrix} \mid \begin{pmatrix}[x_l]\\ x_i\end{pmatrix} \cdot \begin{pmatrix}[y_l]\\ y_i\end{pmatrix} \mid \begin{pmatrix}[x_t]\\ x_i\end{pmatrix} \cdot \begin{pmatrix}[y_t]\\ y_i\end{pmatrix}$$



Original version

$\sum_j x_j.y_j + \frac{1}{3}x_i.y_i$

$\sum_l x_l.y_l + \frac{1}{3}x_i.y_i$

$\sum_t x_t.y_t + \frac{1}{3}x_i.y_i$

Reproducible version

$\sum_j x_j.y_j + 1x_i.y_i$

$\sum_l x_l.y_l + 0x_i.y_i$

$\sum_t x_t.y_t + 0x_i.y_i$

   - the MPI reduction

Reproducibility of the conjugate gradient

1. Reproducible matrix-vector product
2. Reproducible dot product
   - the weighting
   - the MPI reduction

Original dot product

- $r_p = dot(X, Y)$
- $r = all\_reduce(r_p)$

Reproducible dot product

- $[r_p, e_p] = pdot2(X, Y)$
- all_gather(r_p, e_p)
- $r = Sum2(r_p, e_p)$