Parallel streamline tracking for Telemac

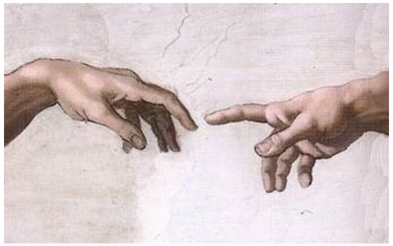
Jacek A. Jankowski

BAW

Department Inland
Waterways Engineering

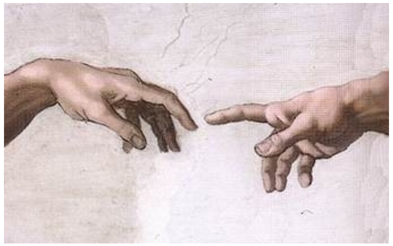
Fifteenth Telemac Users' Club
Chatou, 9-10 October 2008

Semi-Lagrangian advection



- **The method of characteristics:** the most elegant numerical scheme for solving the advection step
- The main advantages:
  - computing with Courant-Nr. > 1
  - can be efficiently implemented
- The main disadvantages:
  - a non-conservative scheme (in efficient implementations)
  - complex numerical features / programming

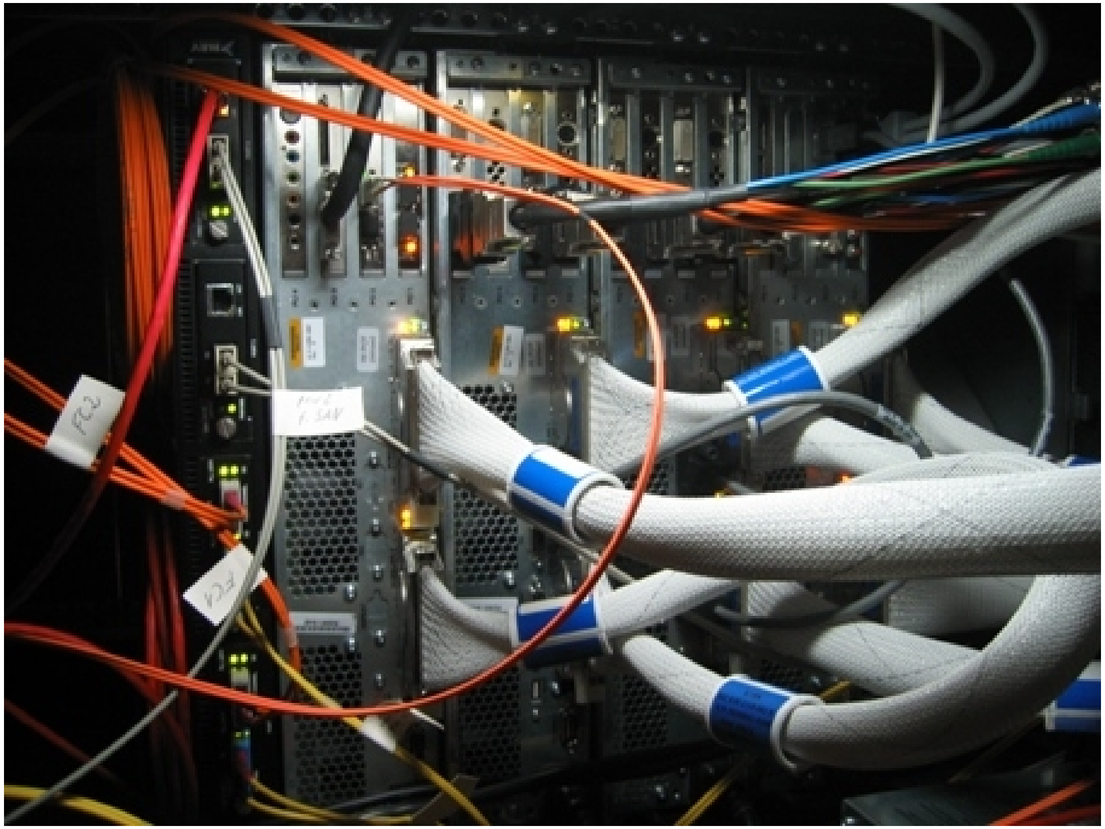
Parallel implementation



- Streamline tracking is awkward to parallelise – attempts for Telemac:
  - R. Hinkelmann ca. 1996-97 (time step reduction)
  - J.A. Jankowski ca. 2001, the present idea, but abandoned!
  - J.-M. Hervouet ca. 2001, for “mild cases”
  - J.A. Jankowski:
    - 2007 for UnTRIM
    - 2008 for Telemac

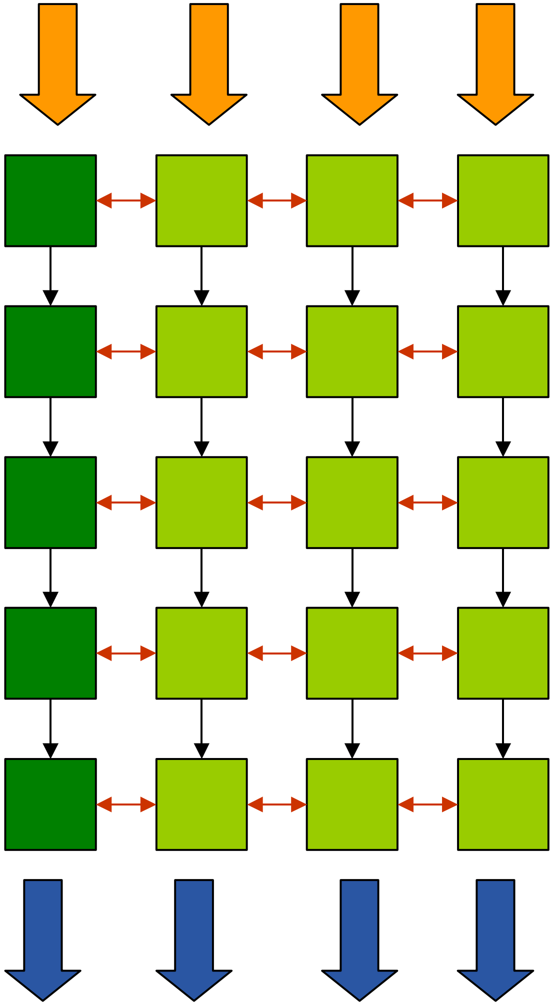
Parallel computing

A few words...



Message-passing parallelism

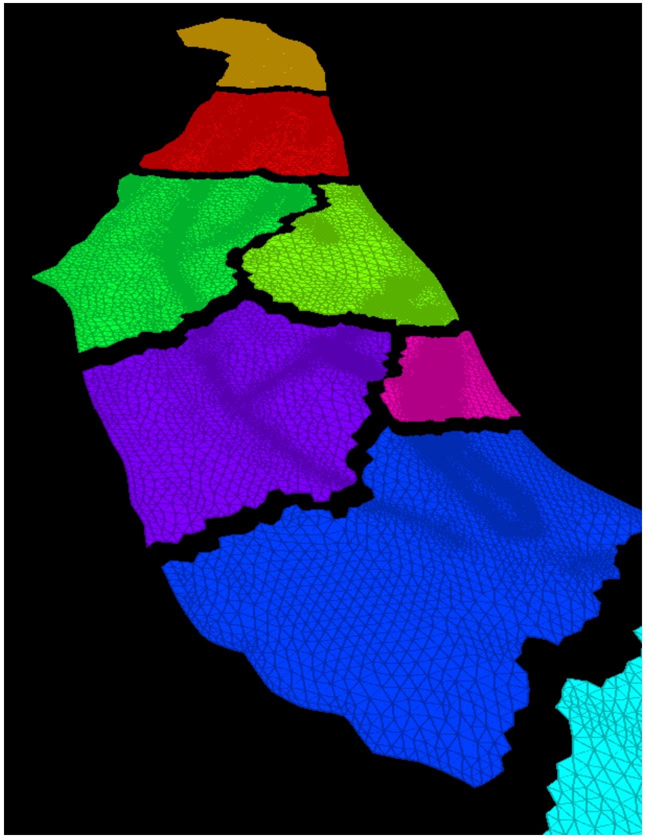
- Each processor executes a program copy with its own data
- Communication limits the scalability of the code
  - preparing data for sending
  - communication *itself*
  - integrating the received data



```
mpirun -np 4 ./prog.exe
```

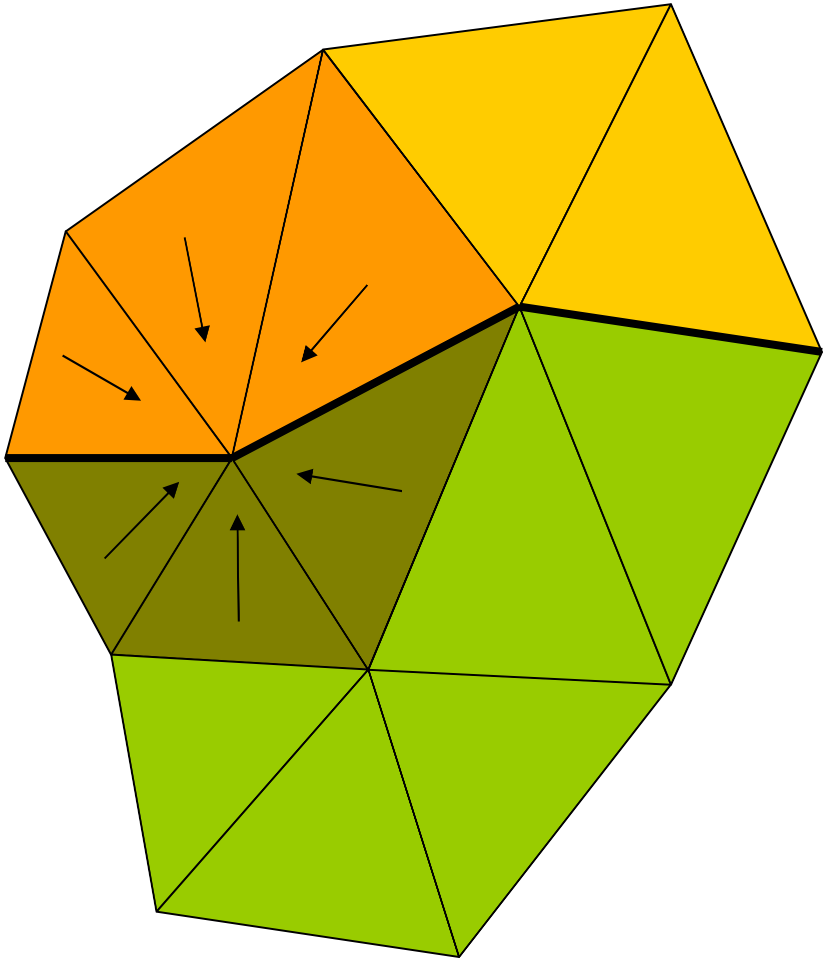
Domain decomposition method

- Parallel implementation with domain decomposition and *non-overlapping* mesh partitions (FEM)
- This leads to *point-to-point* communication between neighbouring partitions for *interface node* values
- Semi-Lagrangian advection methods do not fit well to this scheme: *global* communication

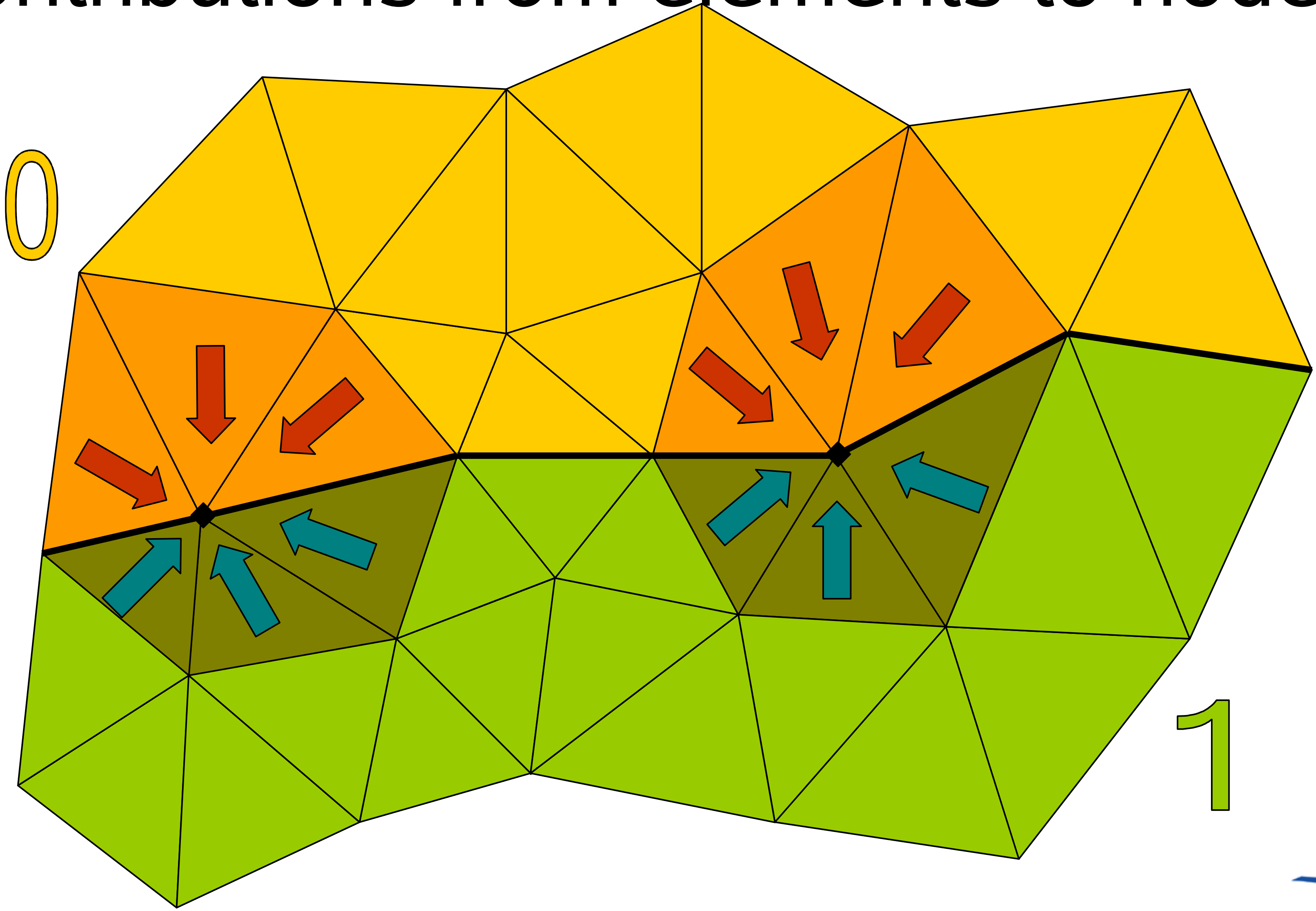


Point-to-point communication

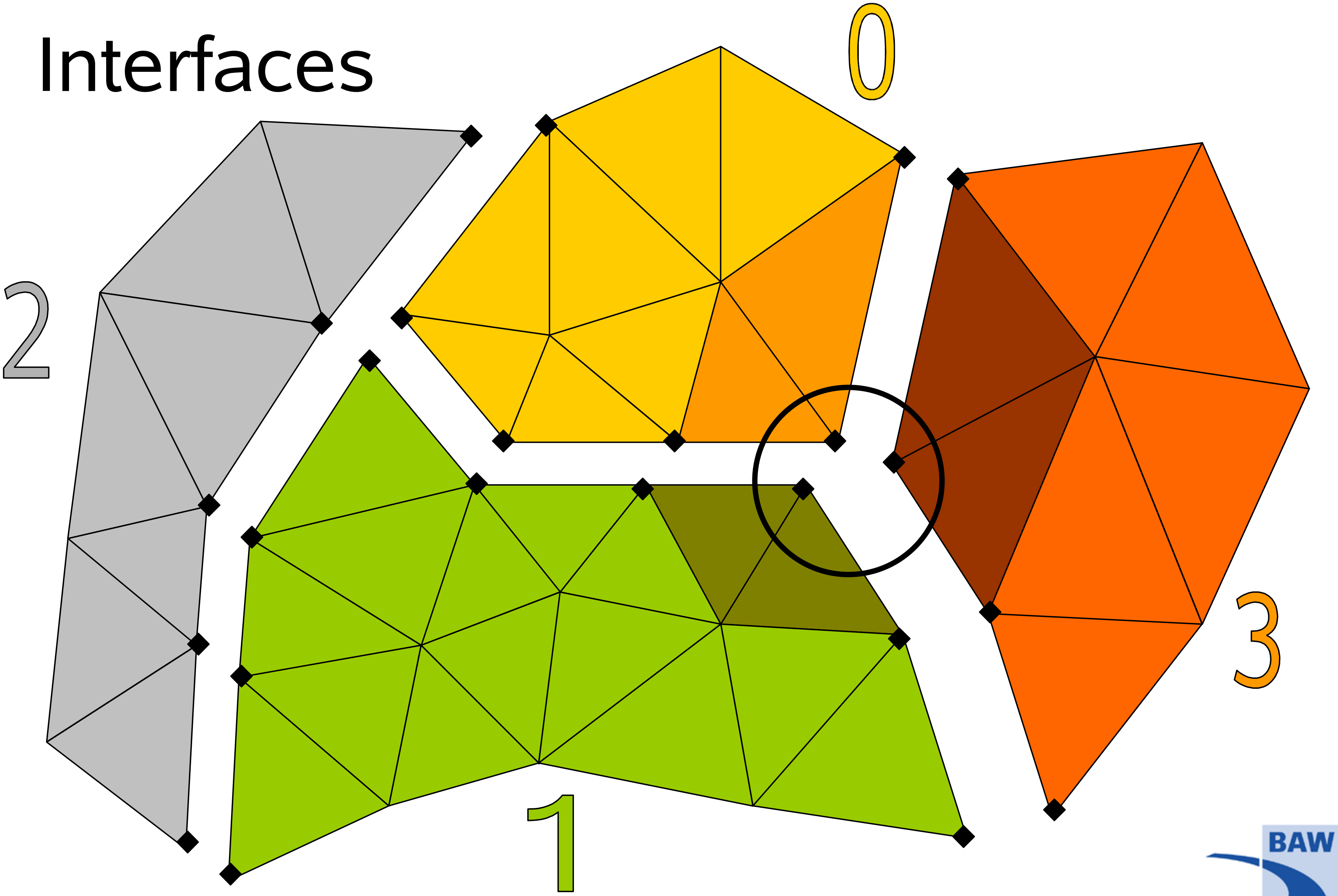
Dealing with the Finite Element Method (FEM)



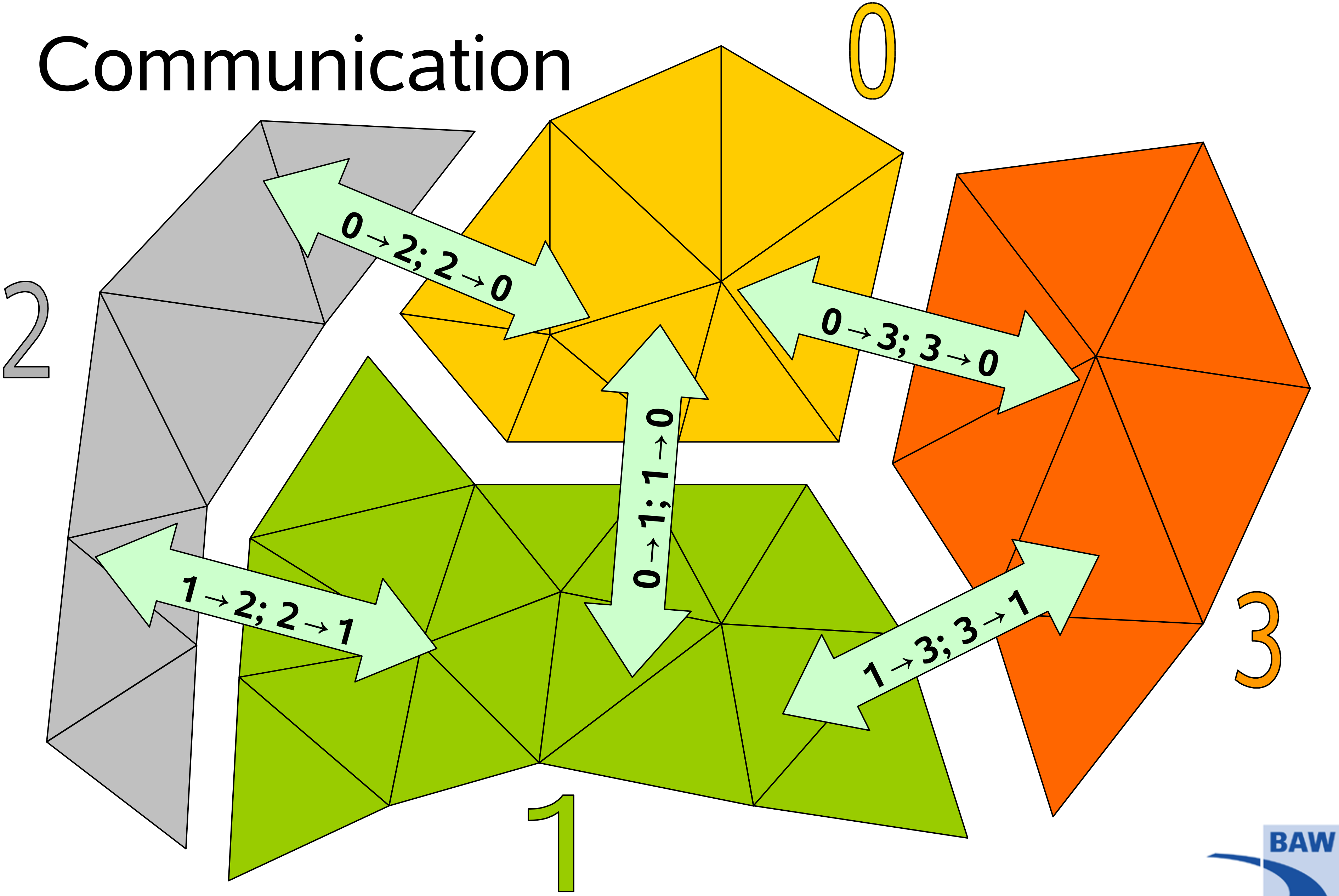
Contributions from elements to nodes



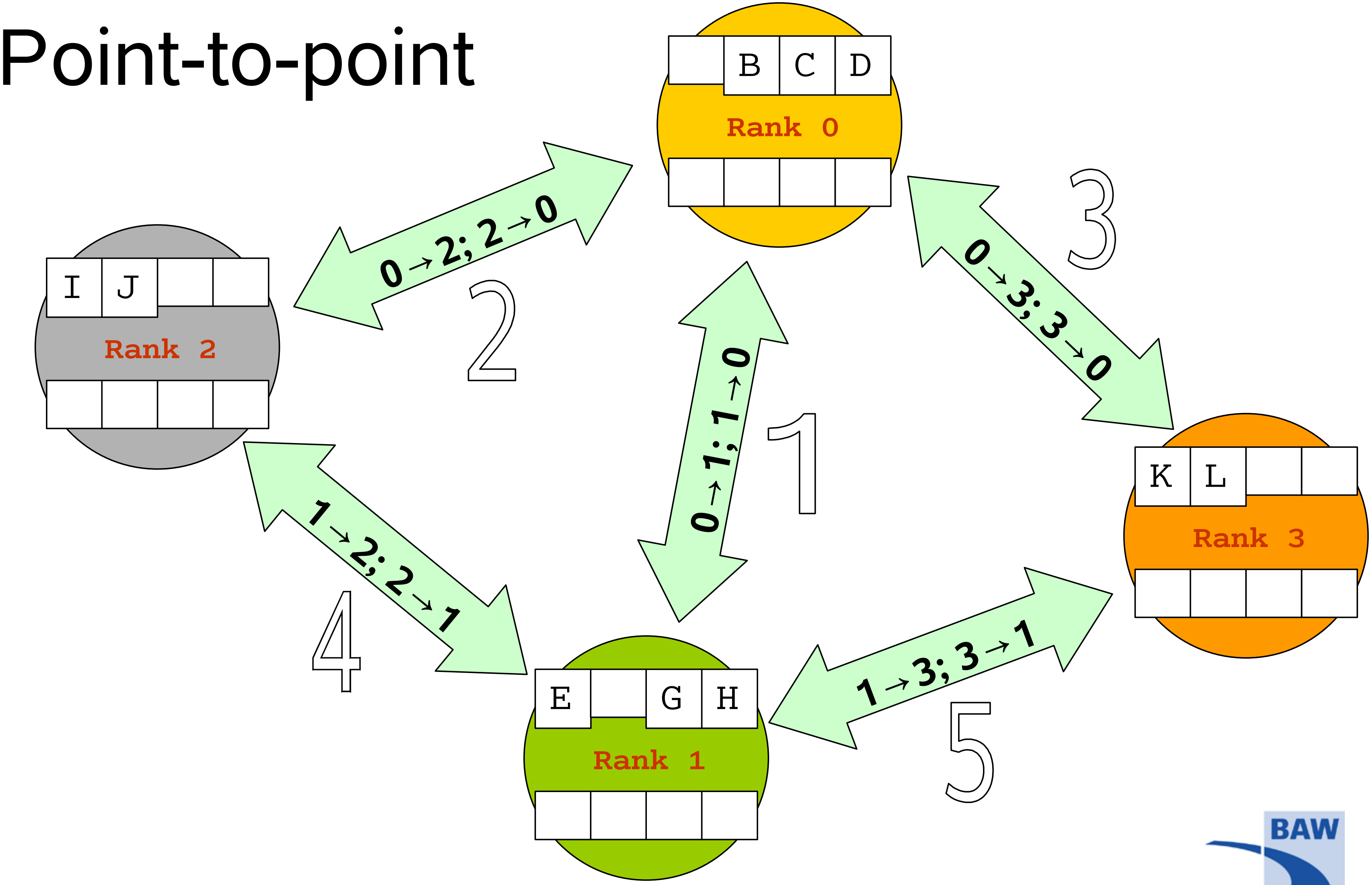
Interfaces



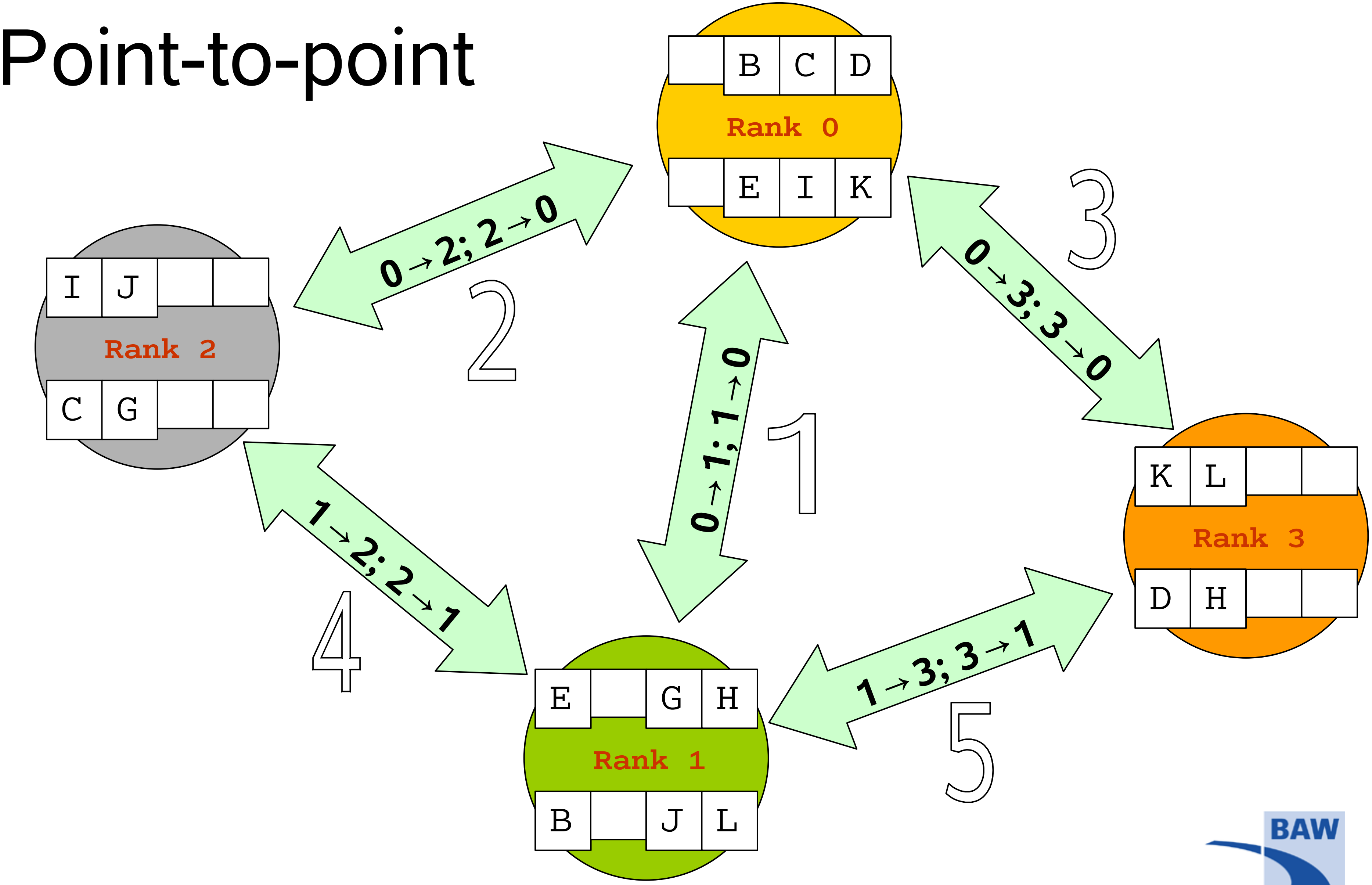
Communication



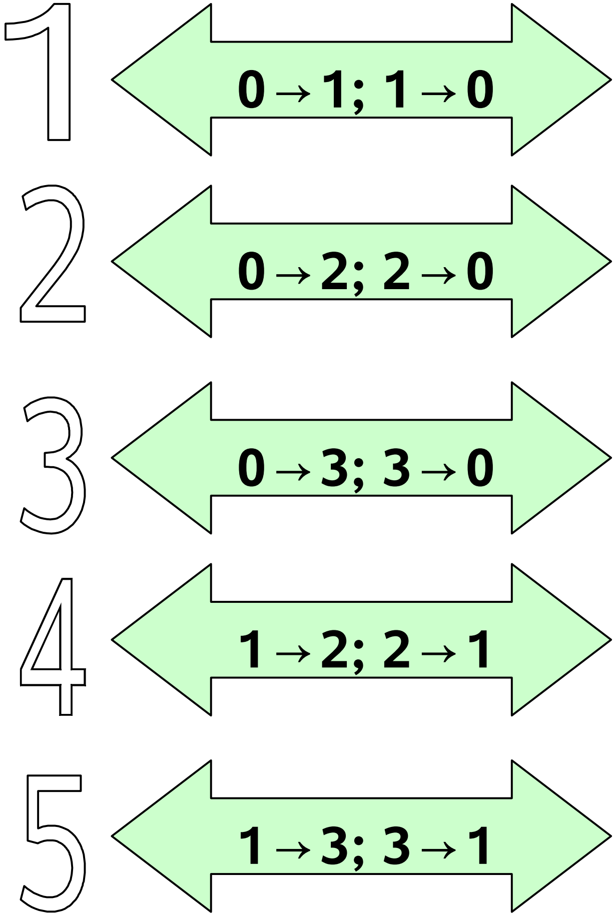
Point-to-point



Point-to-point



MPI_Send, MPI_Recv



1 0 → 1; 1 → 0

2 0 → 2; 2 → 0

3 0 → 3; 3 → 0

4 1 → 2; 2 → 1

5 1 → 3; 3 → 1

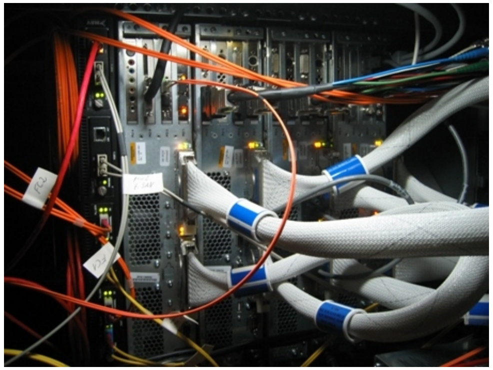
- Point-to-point communication
- Exchanging contributions for interface nodal values
- Assembling (adding them up)

Global communication

Dealing with the Lagrangian advection

(the method of characteristics)

A semi-Lagrangian advection treatment

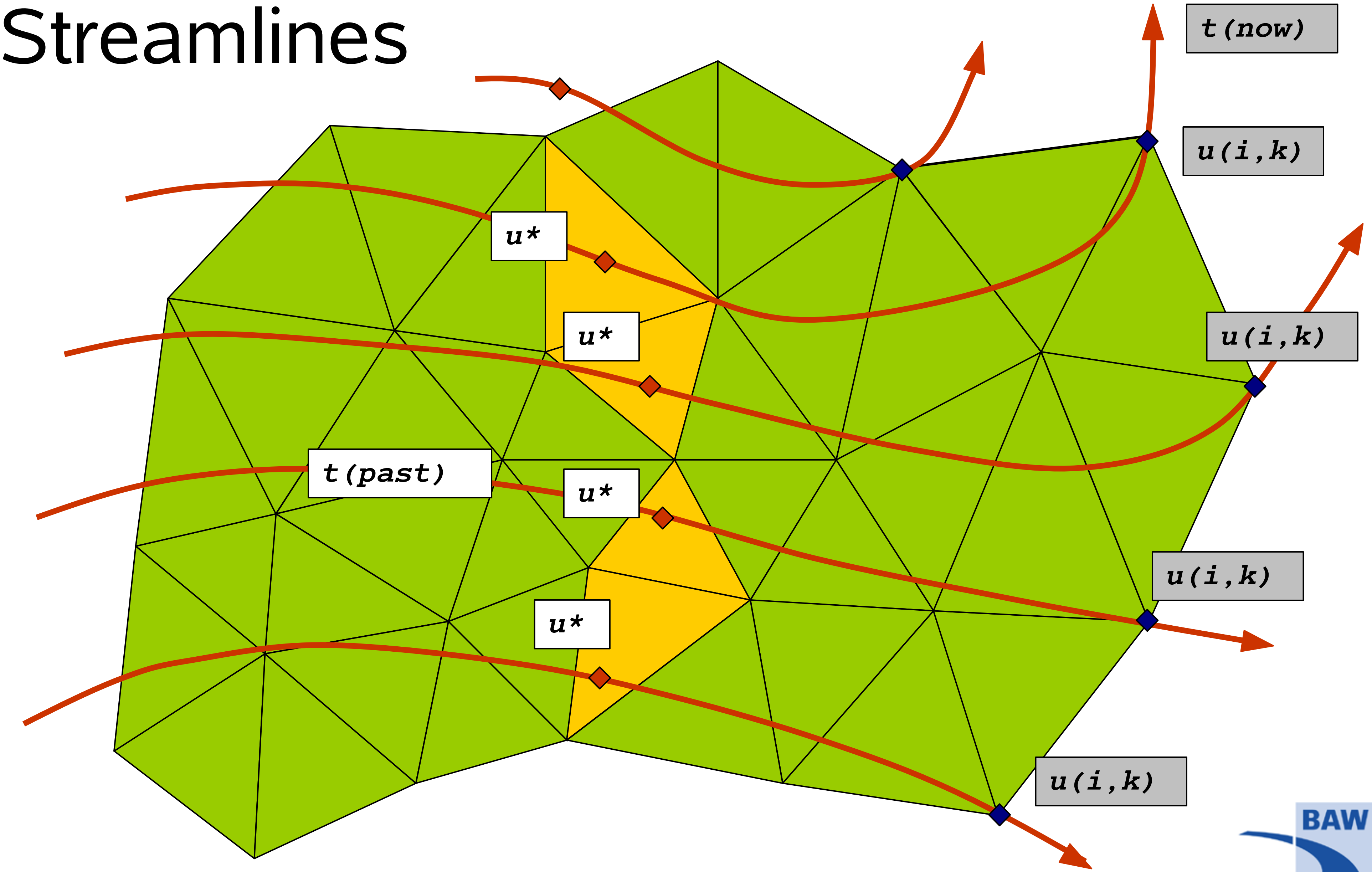


The pure advection – variable values remain constant along a streamline

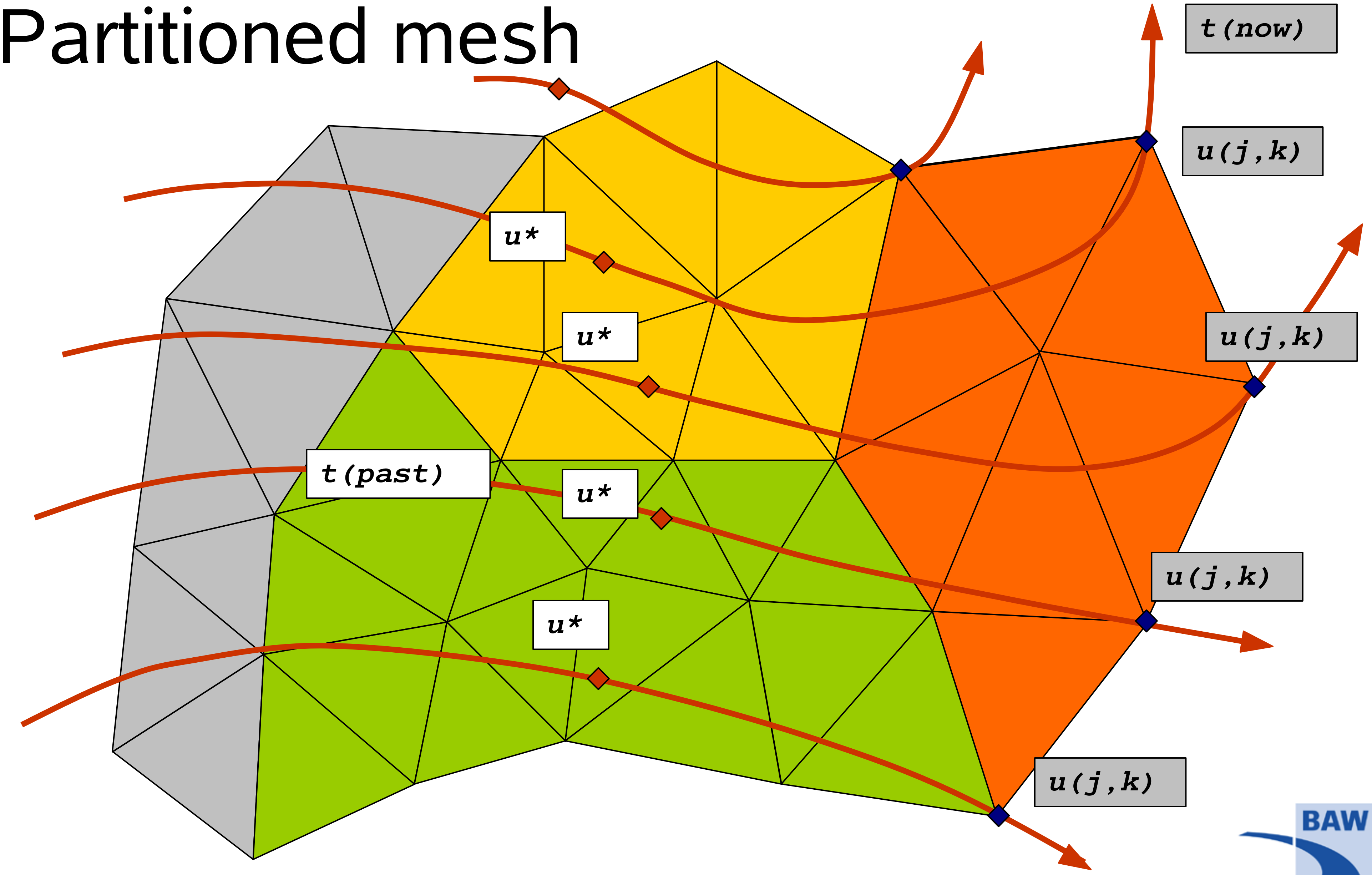
streamline tracking over the mesh – backward in time
interpolating the value at a located point in the mesh
applying the found value further on

semi – actually, the *discretised* values are applied, defined on an Eulerian mesh

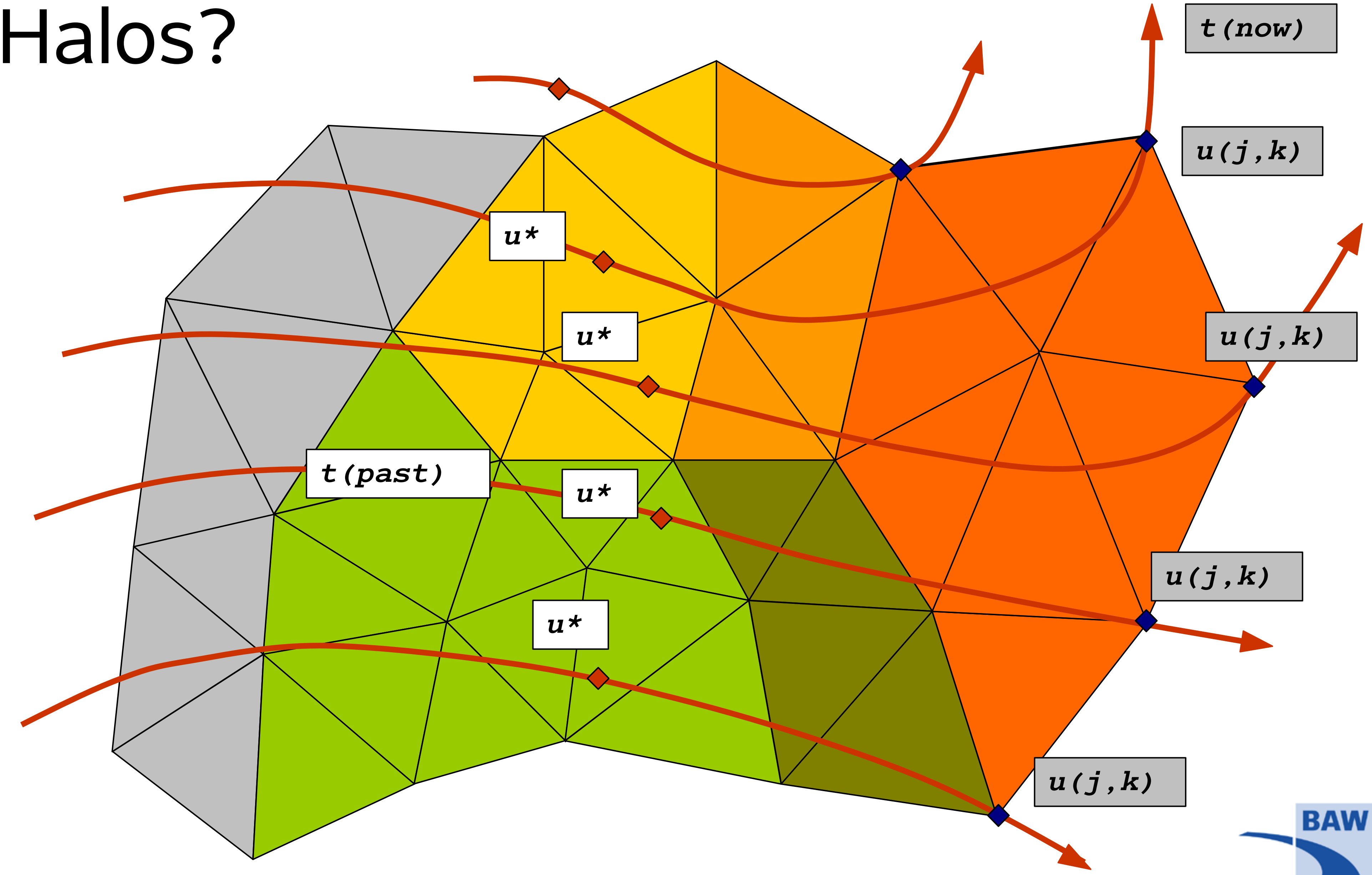
Streamlines



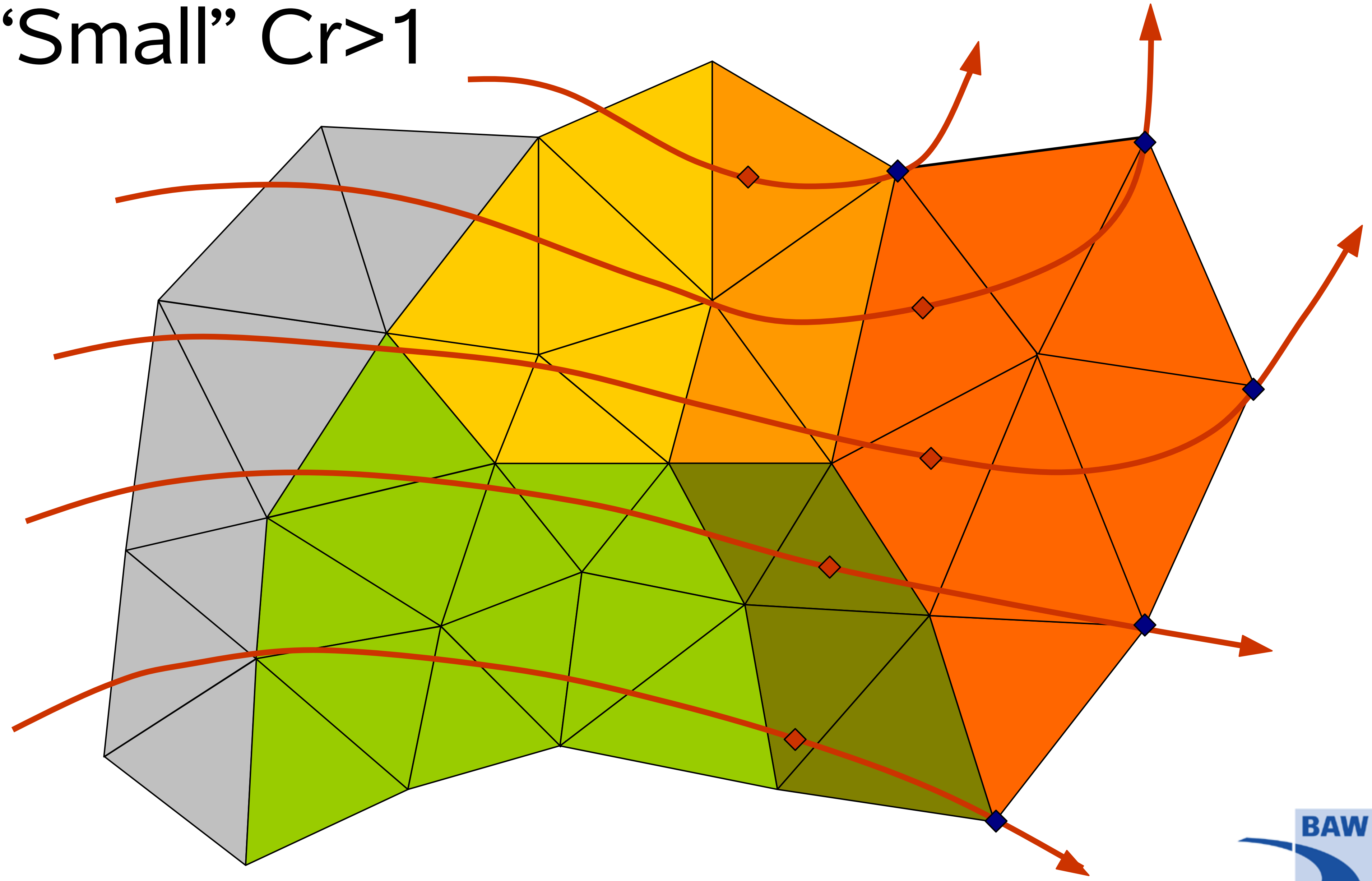
Partitioned mesh



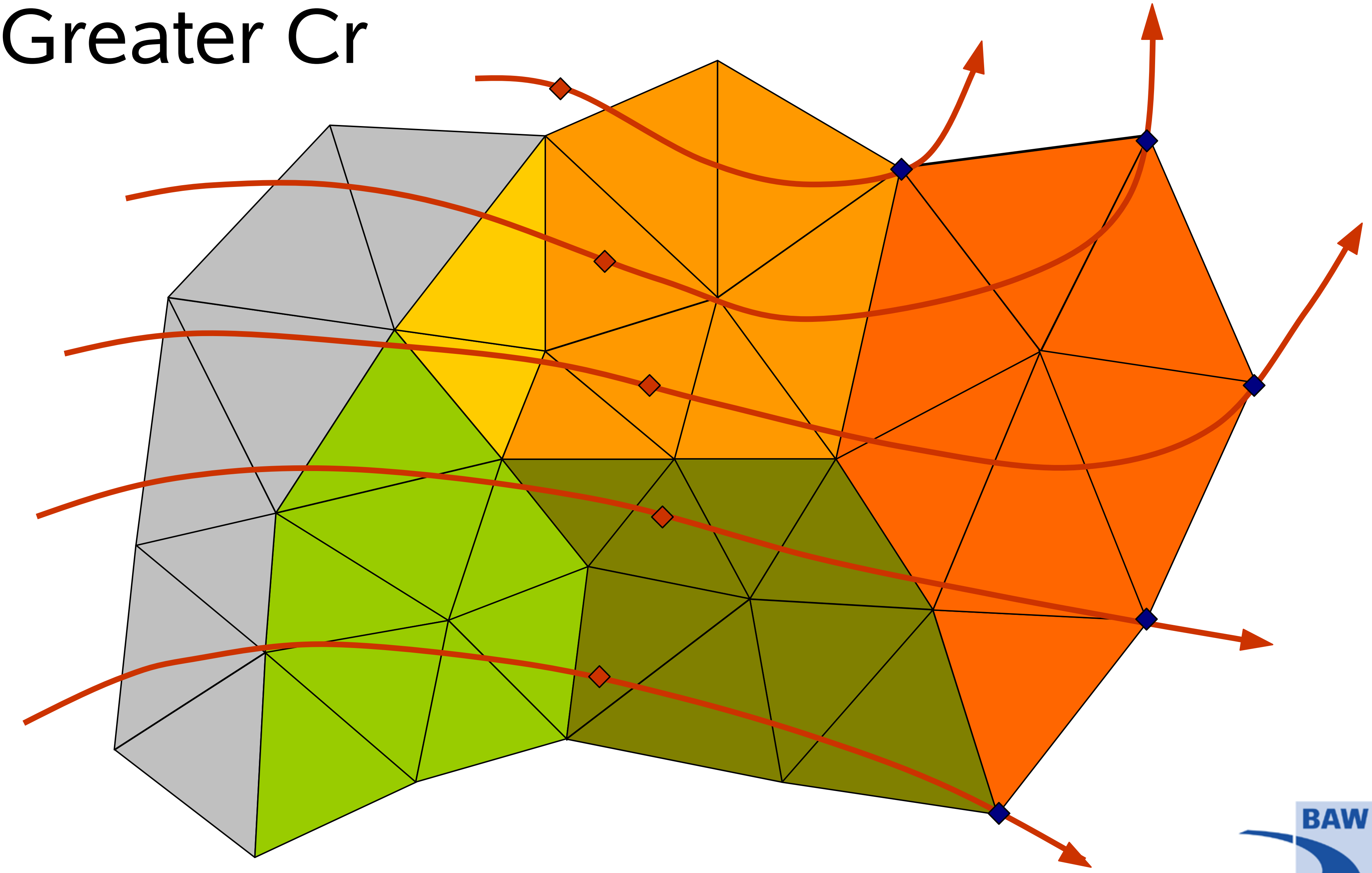
Halos?



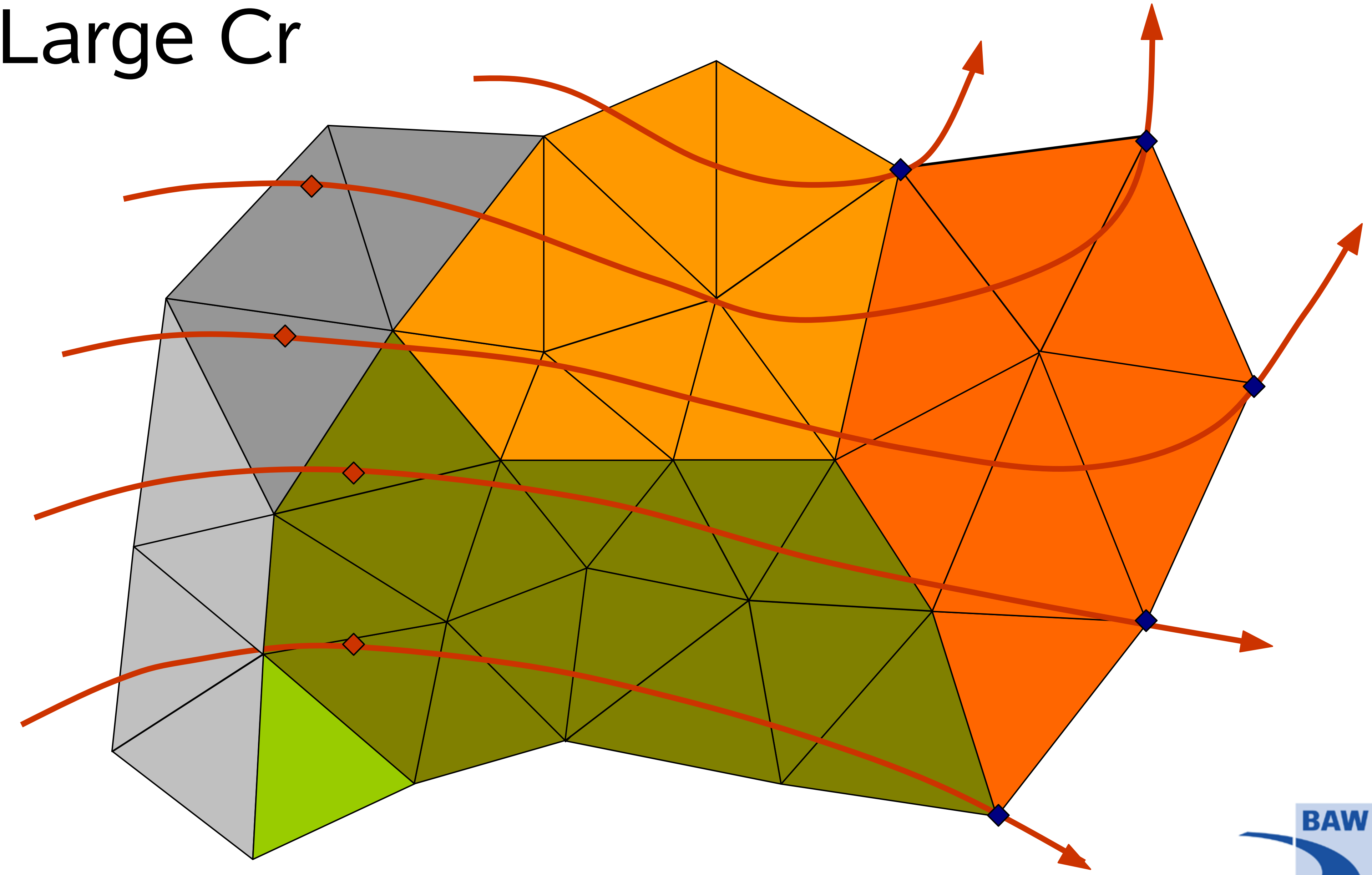
“Small” Cr>1



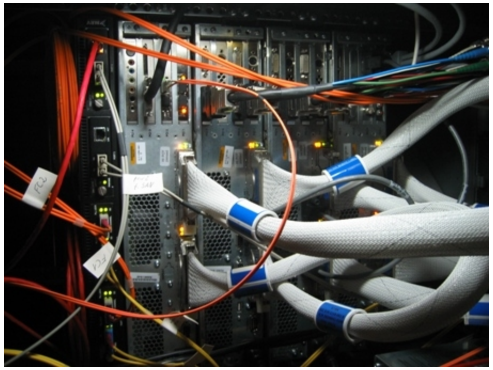
Greater Cr



Large Cr

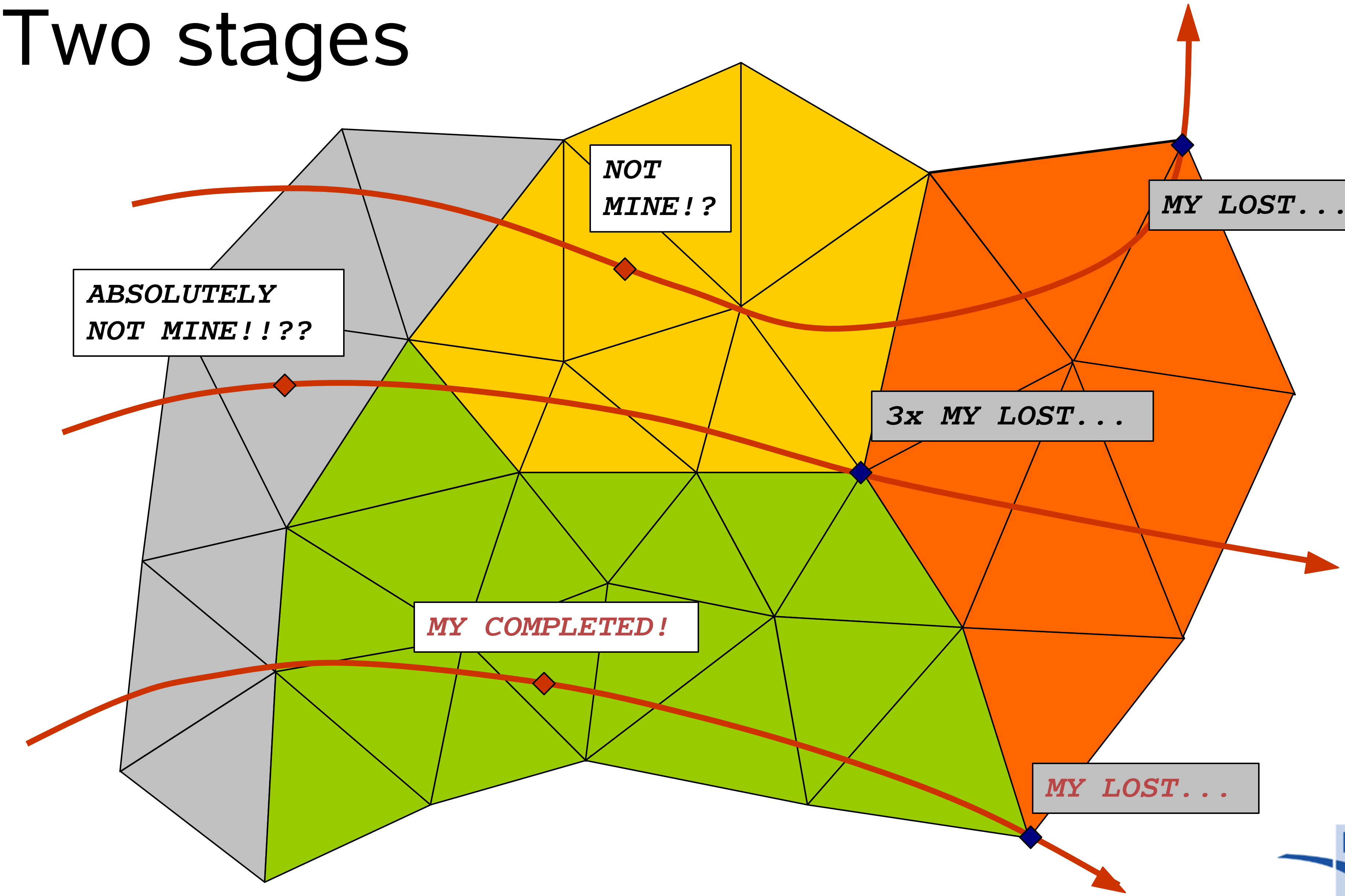


Tracking over partitions

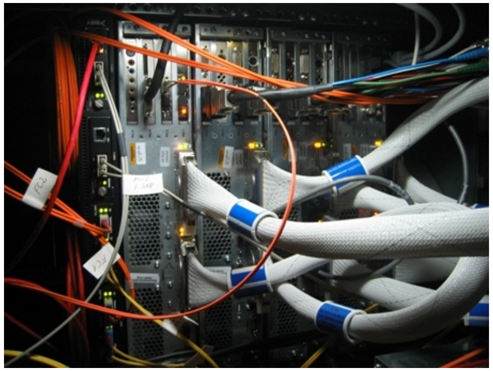


- Streamline tracking is awkward in the point-to-point communication pattern between direct neighbours
- Introducing halos: inefficient for larger Courant numbers (large halos, further neighbours to communicate with…)
- Solution: Tracebacks leaving partitions treated in a ***two-stage algorithm***.

Two stages


NOT MINE!?
MY LOST...
ABSOLUTELY NOT MINE!!??
3x MY LOST...
MY COMPLETED!
MY LOST...

Two stages treatment for lost tracebacks



- If a traceback starting from an *interface node* is completed in one of the neighbouring partitions, the interpolated value is delivered to all these partitions, where this traceback is lost [JMH 2001]

- Only the remaining cases treated in the second stage: tracebacks leaving partitions treated as **separate objects** in an *autonomous* algorithm [JAJ 2007]

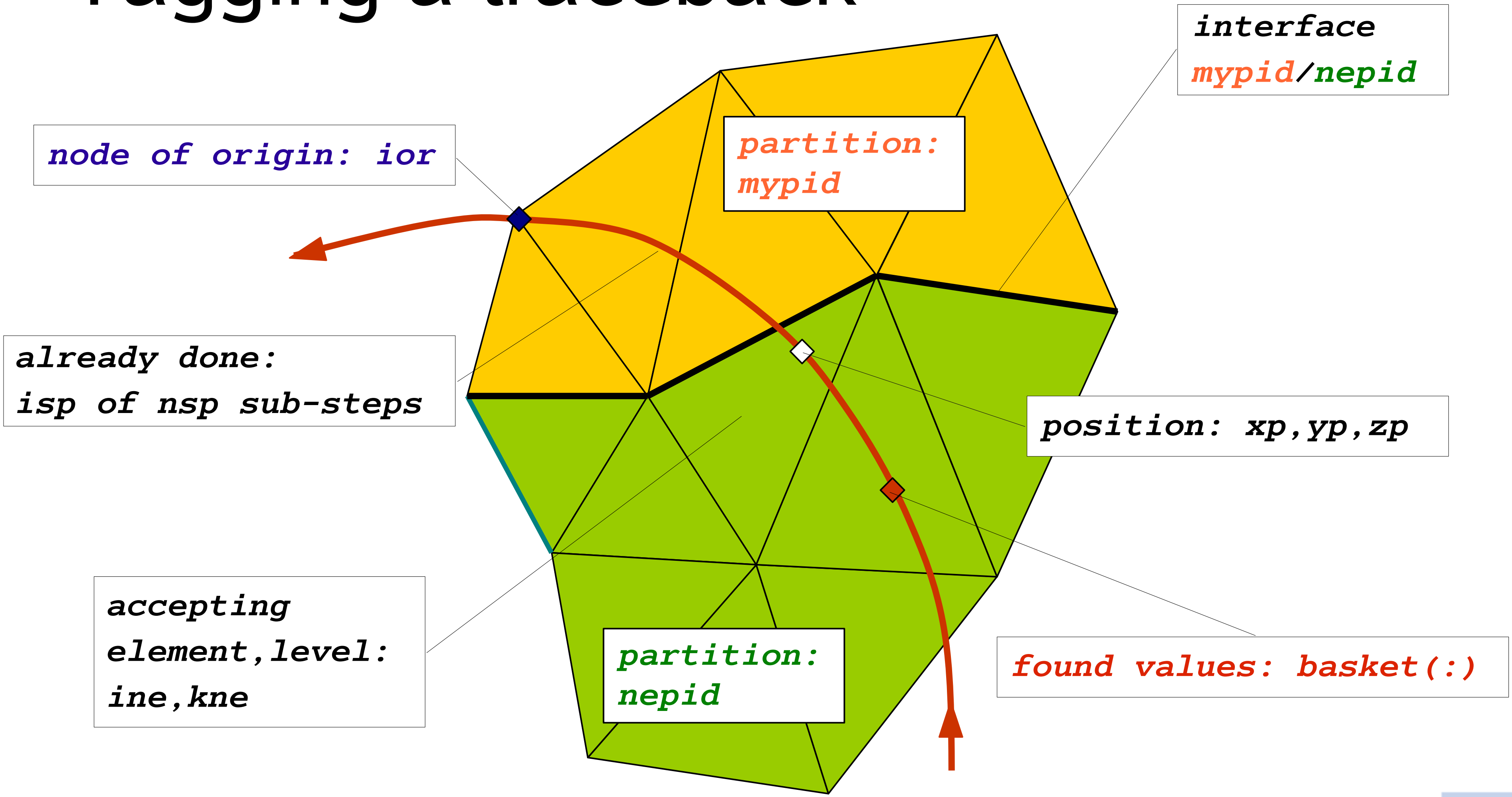
Dog tags for lost tracebacks

Source: dogtagshop.de

An object describing
a 'lost' traceback:

```
TYPE charac_type
   INTEGER   :: mypid,ior
   INTEGER   :: nepid,ine,kne
   INTEGER   :: isp,nsp
   REAL      :: xp,yp,zp
   REAL      :: basket(basket_size)
END TYPE
```

Tagging a traceback



interface
mypid/nepid

node of origin: ior

partition:
mypid

already done:
isp of nsp sub-steps

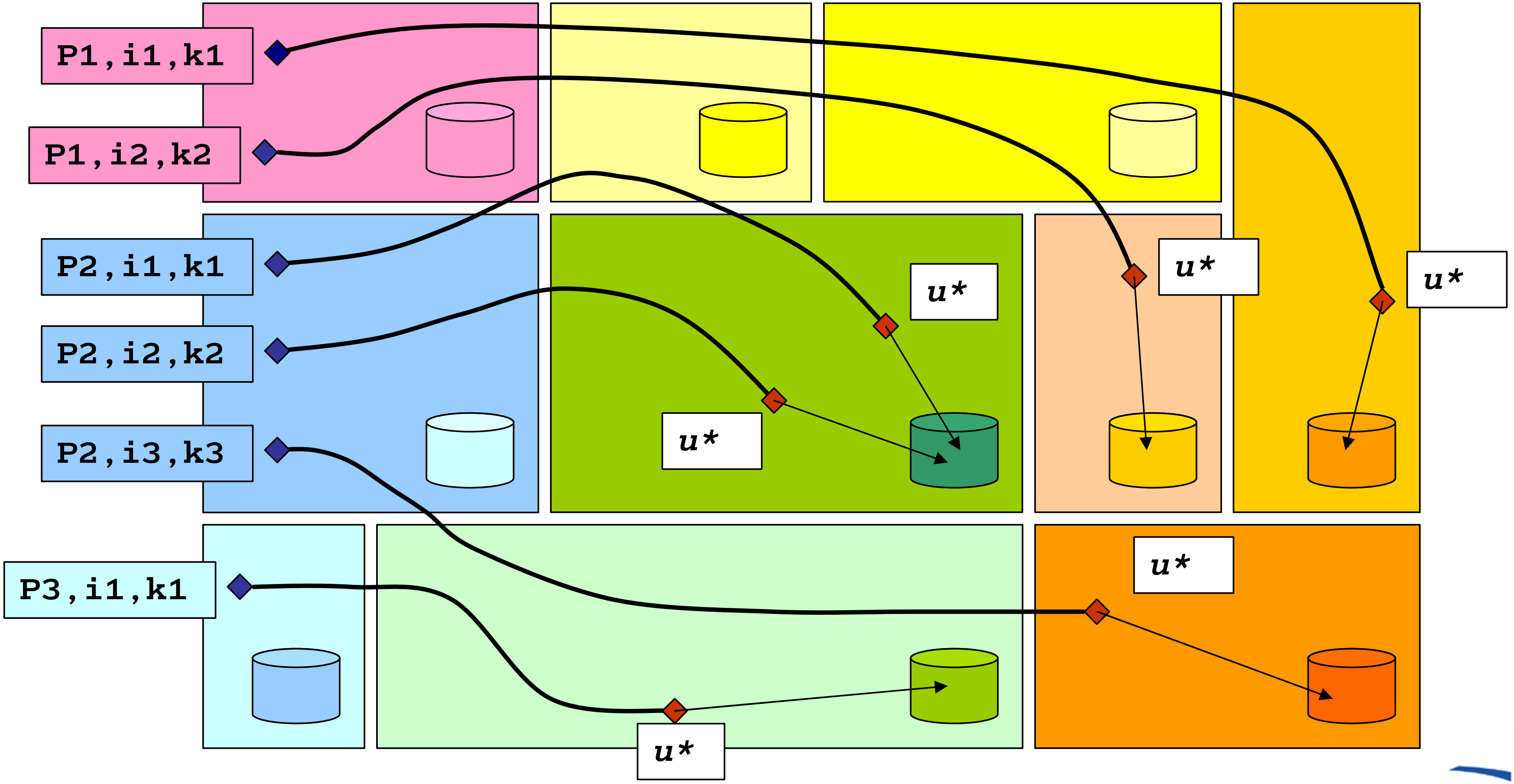
position: xp,yp,zp

accepting
element,level:
ine,kne

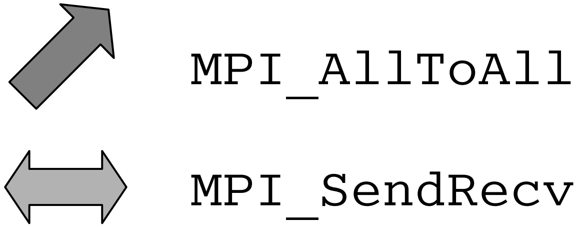
partition:
nepid

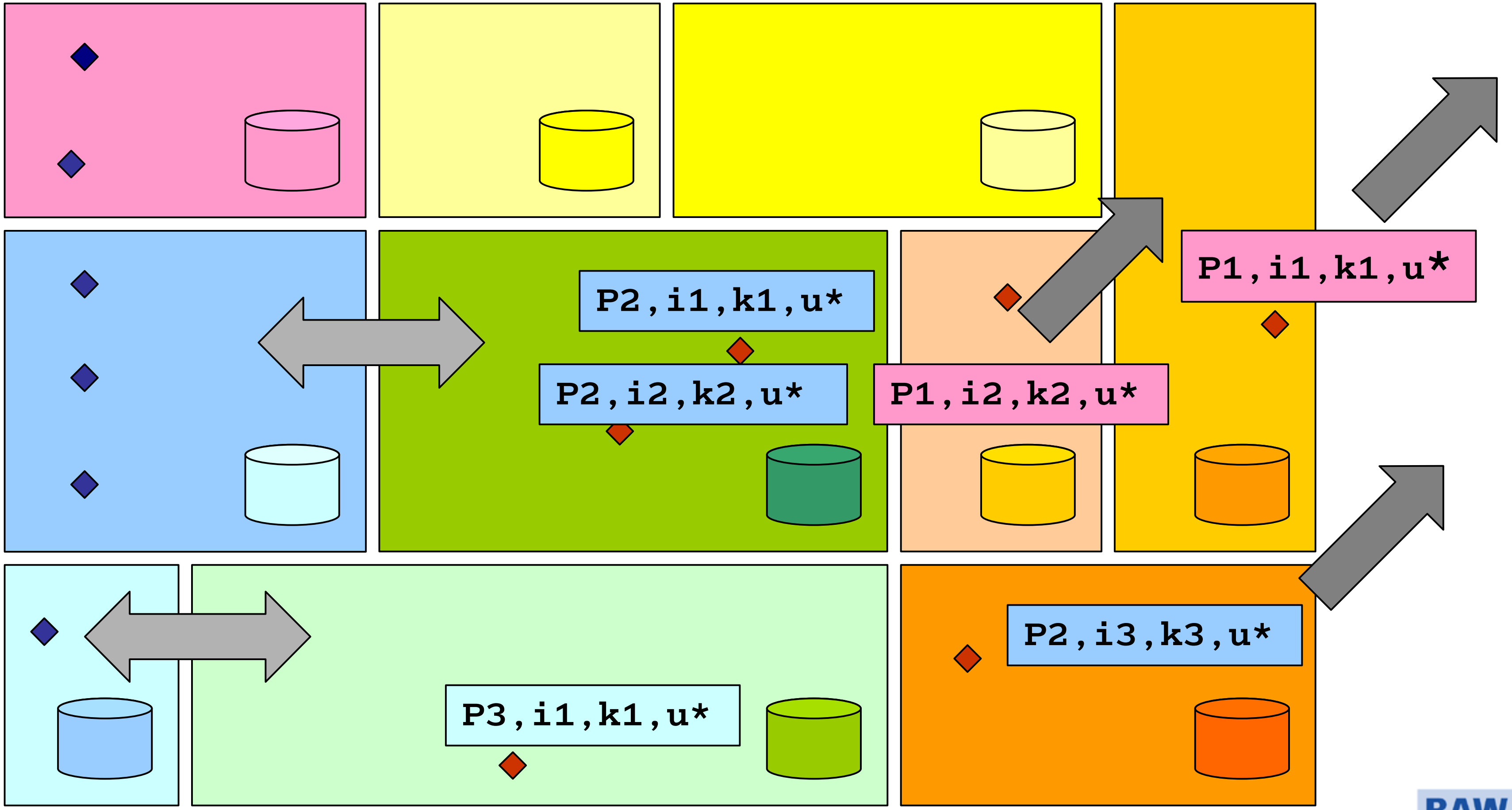
found values: basket(:)

Autonomous tracking

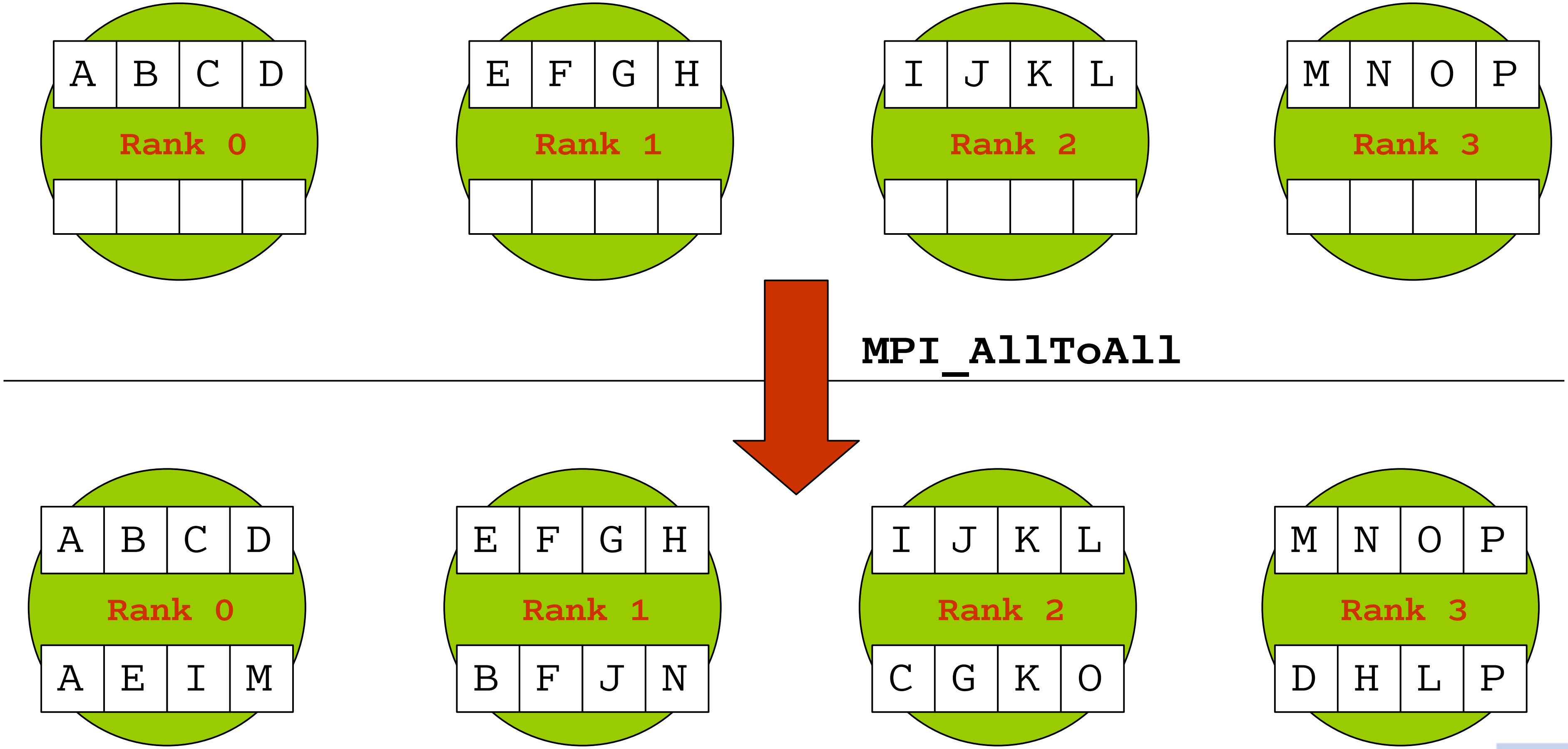


Sending back

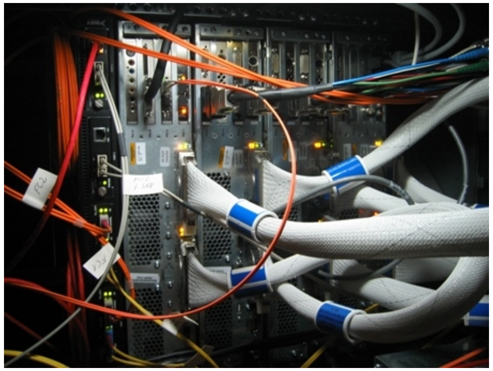




MPI_AllToAll

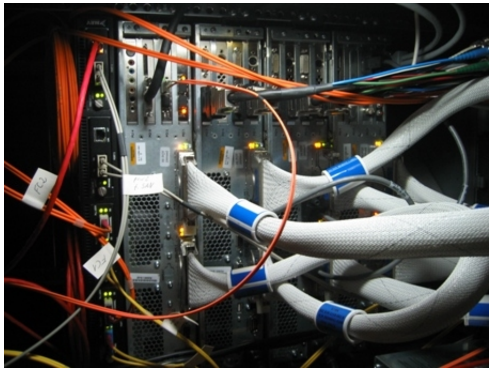


Summary: Communication



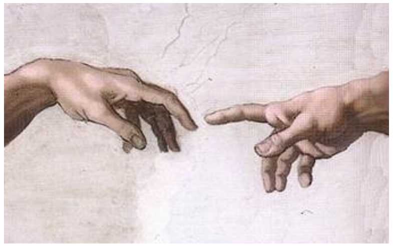
- FEM (Eulerian):
  - exchanging contributions to interface node values: point-to-point communication
- Advection (Lagrangian):
  - streamline tracking treating tracebacks as autonomous objects: global communication

Reached



- *Ten years after:* The parallelisation of the one of the most significant Telemac algorithm options is finished!
- The scalability of Telemac is not adversely affected:
  - minimal amount of data exchanged between processors – only the necessary values
- ...but we are not quite happy...

Parallelising...



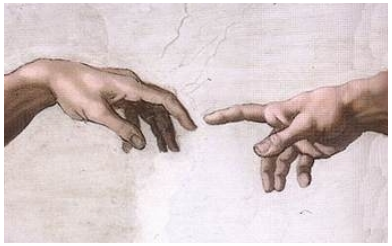
the reproducibility of the results gains in importance

parallel versions of well-verified algorithms bring into light some of their properties

which were treated as unimportant, irrelevant or the matter of some compromises

...and this is annoying!

Verification



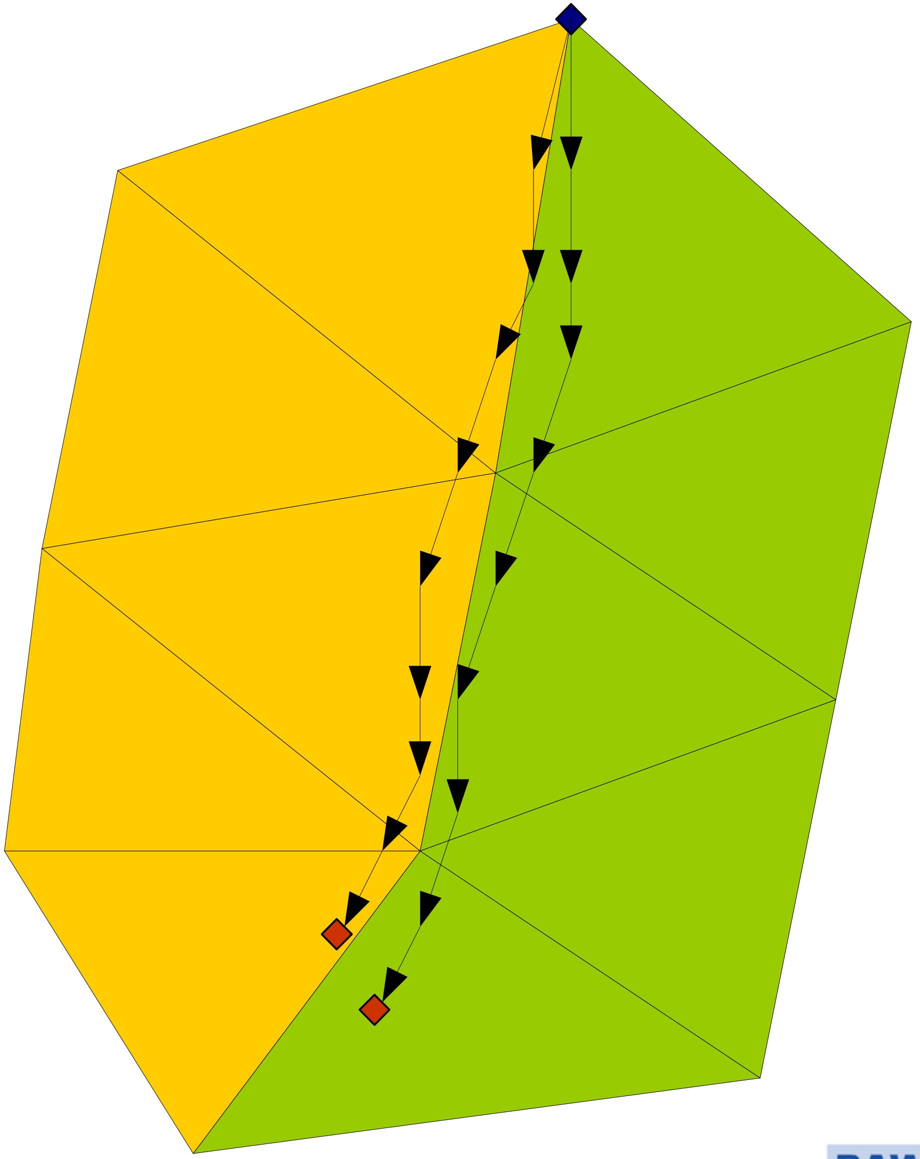
Verification and validation cases deliver in some cases:

- small differences in results *between runs with different partition numbers* in isolated places, also away from interfaces
- reason: reproducibility of the *serial* results is slightly affected by mesh sorting
- *partitioning re-sorts the mesh*, so similar effects

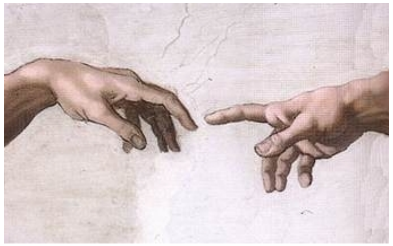
Numerical “dispersion”

- Differently sorted meshes
- The sequence of numeric operations changes
- Starting elements differ
- Tracking paths in sub-steps differ
- Different elements found as the traceback location

- Different values obtained by interpolations there

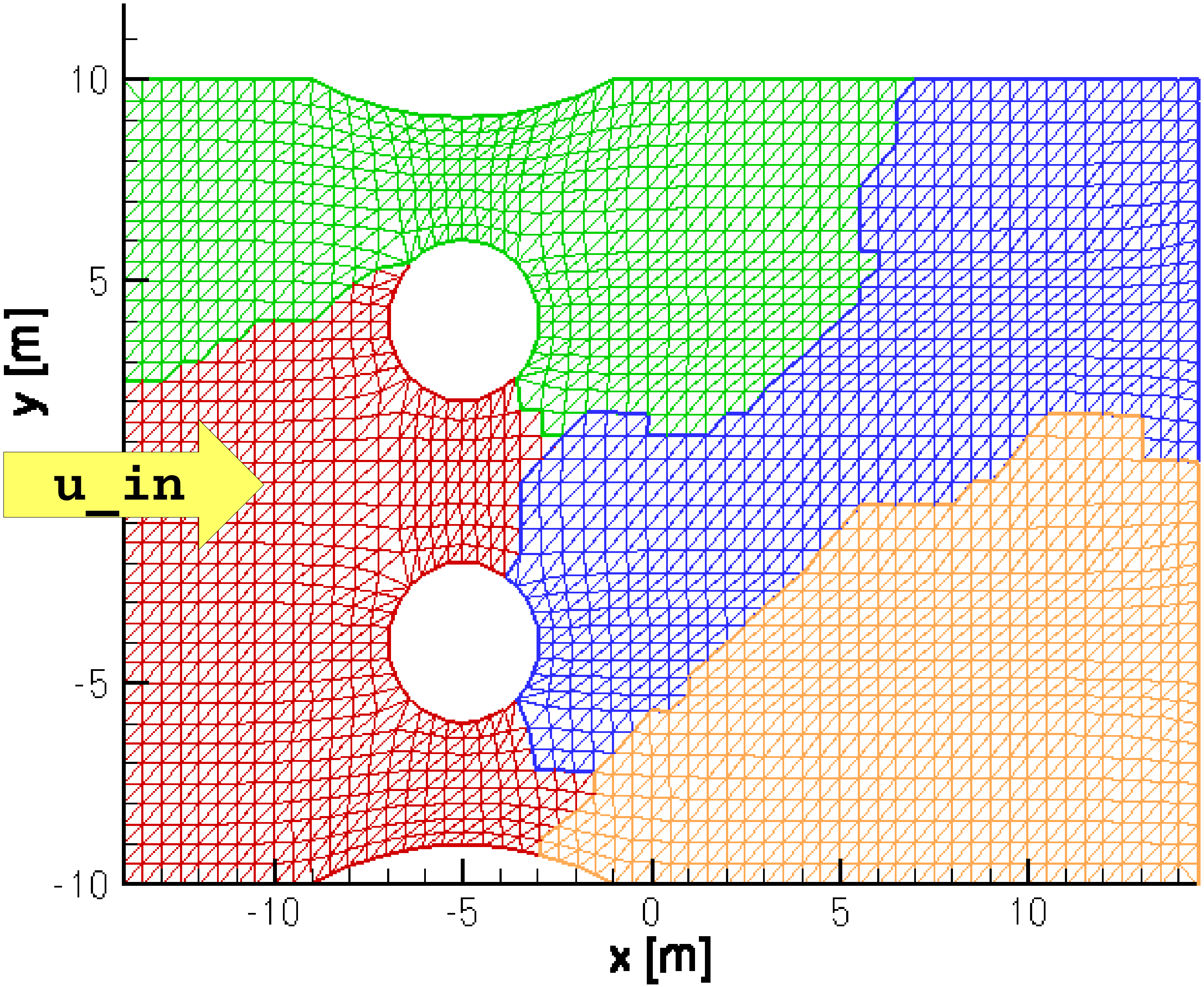


Eddies behind bridge pillars



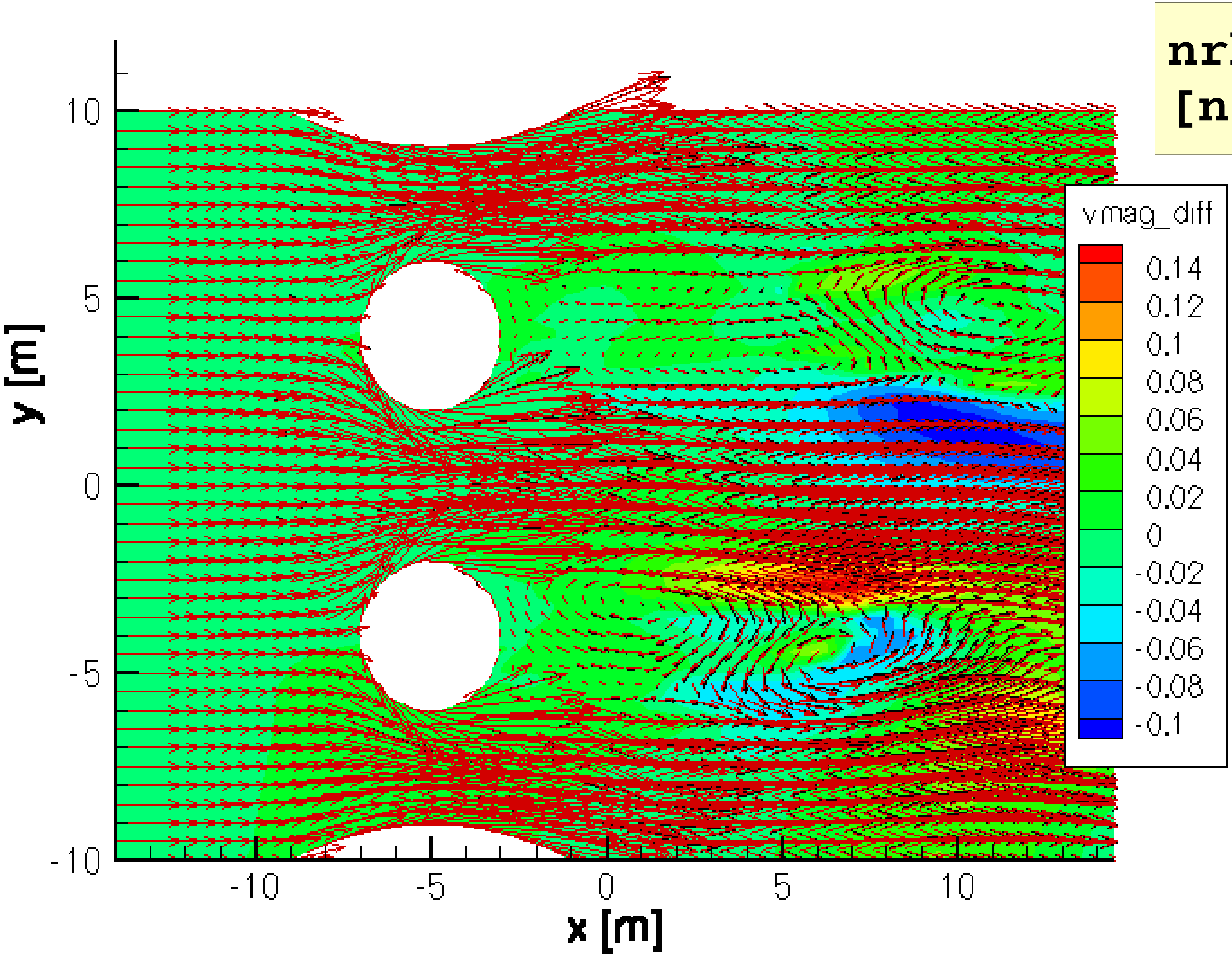
- a standard Telemac2D test case (demanding!)
- comparing parallel (4 processors) and serial results
- influencing the tracking algorithm

- number of sub-steps pro element: nrk=3 (default)
- number of sub-steps pro element: nrk=10
- nrk=10 and taking for each traceback the same number of sub-steps (maximum found) nspmax

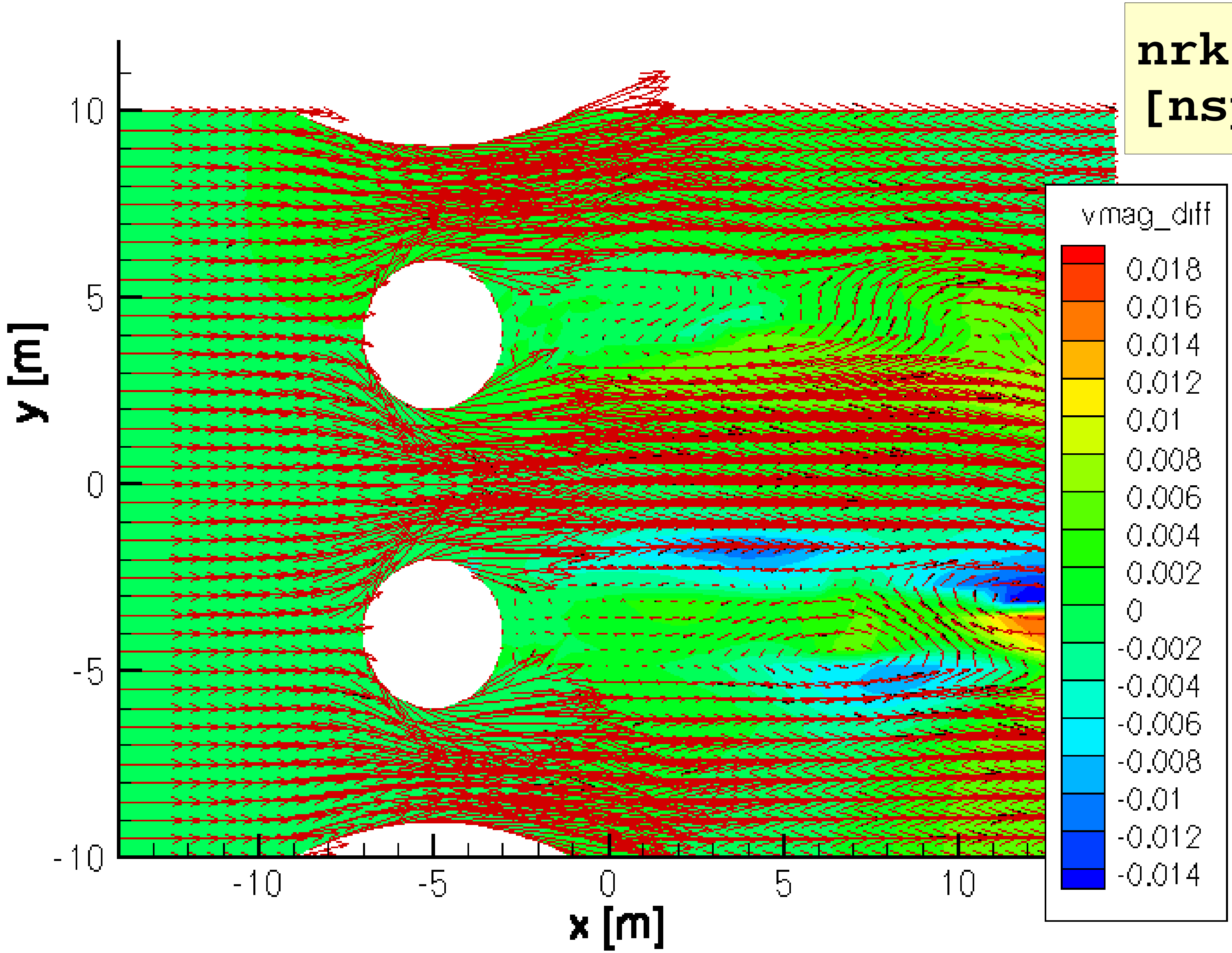


ncsize=4
u_in=1m/s

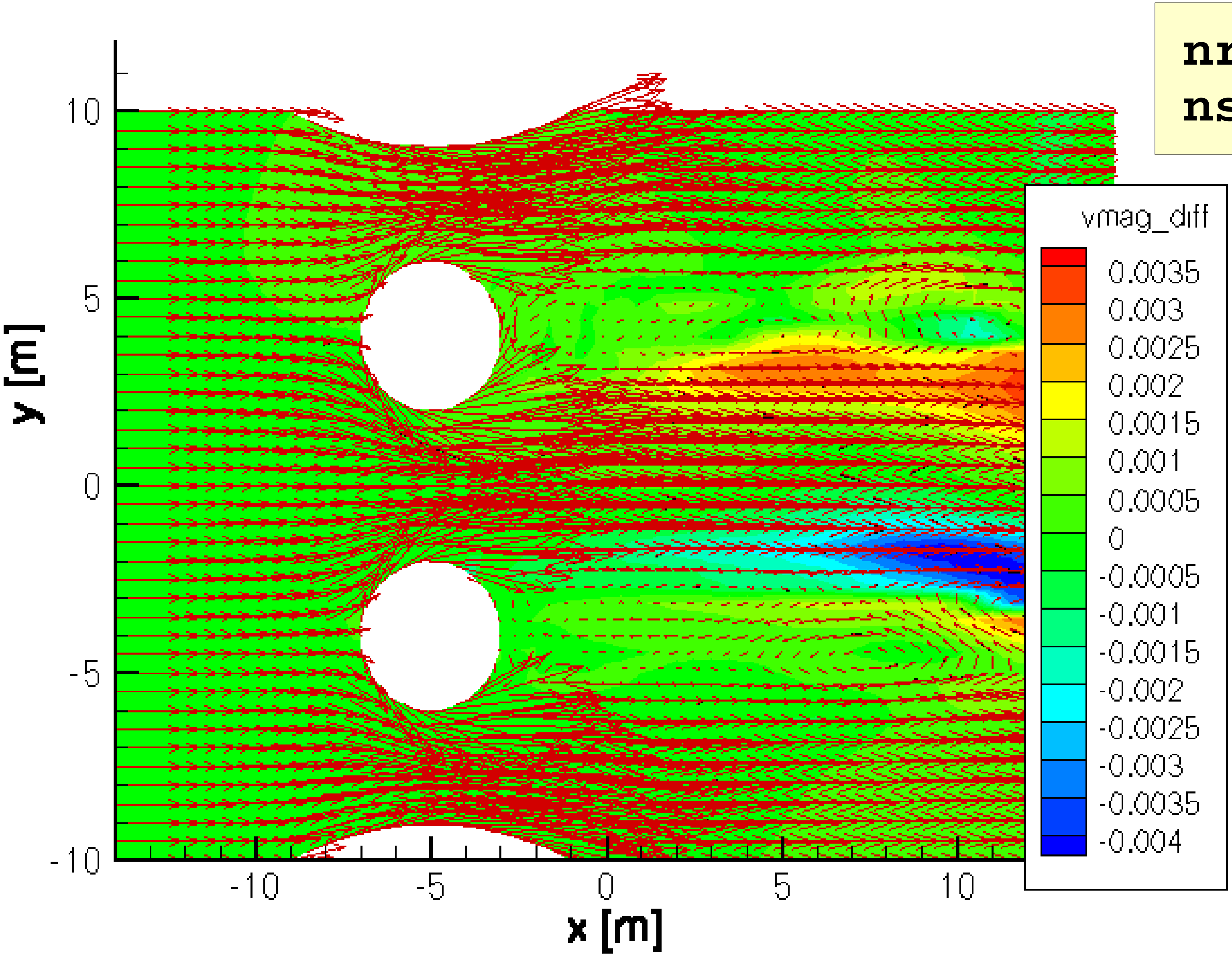
nrk=3
[nspmax=6]



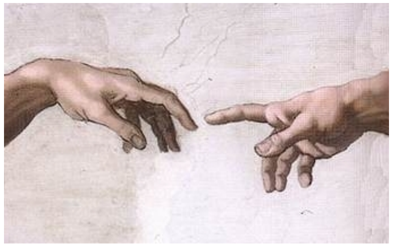
nrk=10
[nspmax=23]



nrk=10
nspmax=23

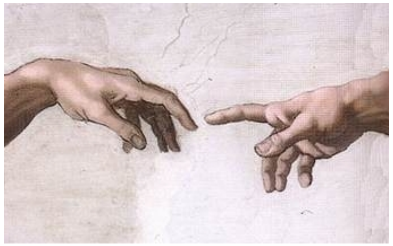


Facts



- fact 1: the results reproducibility is affected by the mesh sorting (partitioning!)
- fact 2: numerical “dispersion” diminishes when tracking algorithm quality is increased
- fact 3: the actual results change as well... (huh!)

Outlook



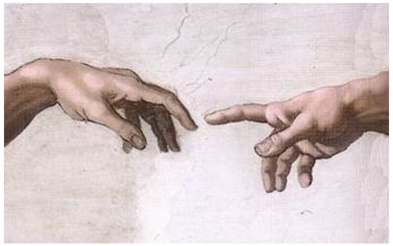
- improving the (serial) algorithm quality?
- a new tracking algorithm? which one?
- finding a new compromise between the quality and the computational efficiency?

- doing nothing? (we lived with it for over 20 years...)

I listen to all questions!

mailto: jacek.jankowski@baw.de

Advection

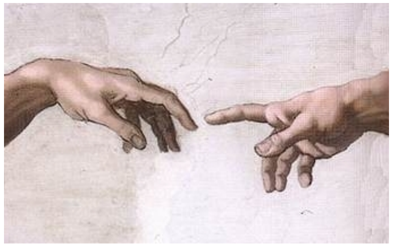


The advection equation says:

the variable f does not change along a characteristic curve (streamline):

$$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = \frac{df}{dt} = 0$$

Streamline tracking

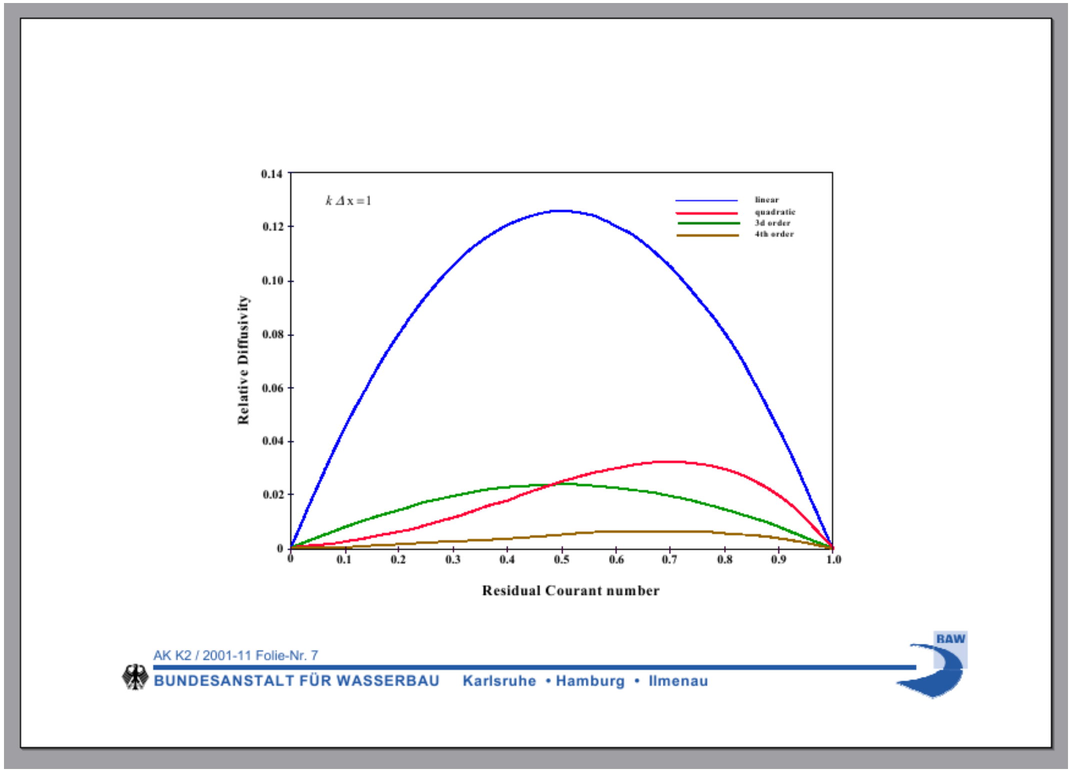


The idea is:

in order to find out the nodal value, follow the characteristic curve backward in time:

$$\frac{dx_j}{dt} = u_j \qquad \Rightarrow \qquad x_j^b = x_i^{n+1} - u_j \Delta t$$

Second order interpolation

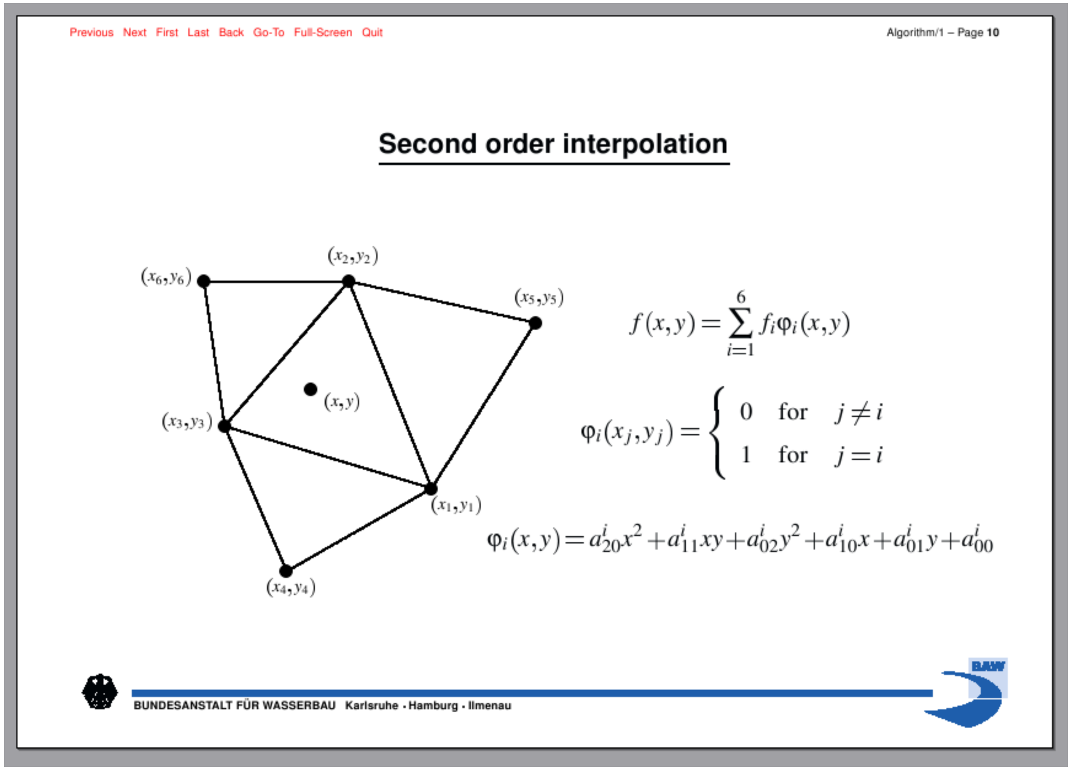


AK K2 / 2001-11 Folie-Nr. 7

BUNDESANSTALT FÜR WASSERBAU Karlsruhe • Hamburg • Ilmenau

- Presentation at TUC2000 by Andreas Malcherek
- Requirement of computationally efficient and accurate advection schemes
- Reducing numerical diffusion by a higher interpolation order

Second order interpolation



Previous Next First Last Back Go-To Full-Screen Quit

Algorithm/1 – Page **10**

Second order interpolation

$$f(x,y) = \sum_{i=1}^{6} f_i \varphi_i(x,y)$$

$$\varphi_i(x_j,y_j) = \begin{cases} 0 & \text{for} \quad j \neq i \\ 1 & \text{for} \quad j = i \end{cases}$$

$$\varphi_i(x,y) = a^i_{20}x^2 + a^i_{11}xy + a^i_{02}y^2 + a^i_{10}x + a^i_{01}y + a^i_{00}$$

BUNDESANSTALT FÜR WASSERBAU Karlsruhe · Hamburg · Ilmenau

- Presentation at TUC2001 by Andreas Malcherek and J.A. Jankowski

- Second order interpolation for Telemac2D
- Good results
- Never implemented in the production code
- Abandoned