Unstructured PHOENICS June, 2009



## Summary

This presentation, consisting of contributions by:

- Valeriy Artemov,
- Alexey Ginevsky and
- Brian Spalding,

describes the current status of ‘USP’, *i.e.*
Un-Structured PHOENICS,
mainly by way of examples.

# Contents list





## The topics considered include:

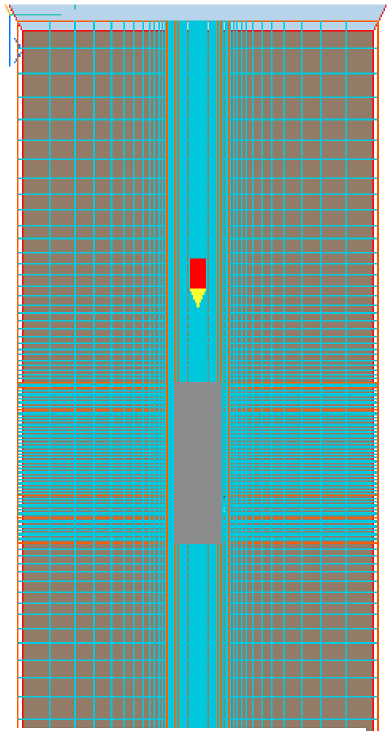
# Why USP is being developed: economy of computer time & storage

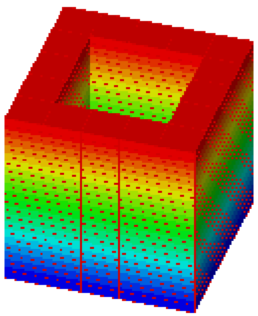




The **motive** for introducing it has **not** been (as it may be for competitors) to handle **curved-surface** bodies; for **PARSOL** handles these satisfactorily.

Instead, the motive is to reduce the **waste of time and storage** entailed by the un-needed fine-grid regions which PHOENICS (in structured- grid mode) generates far from the bodies, as seen on the right.



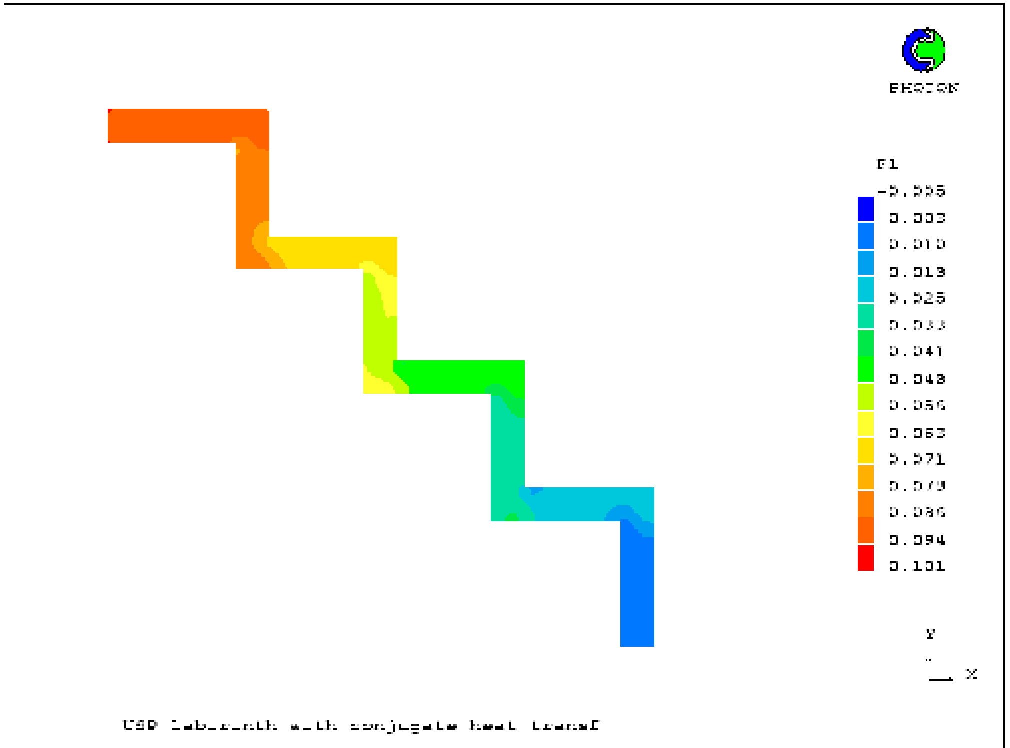


For the hollow-box heat-conduction problem on the left, **SP (structured PHOENICS)** pays attention also to the **empty central** volume; **USP does not.**

# Another example of USP's ignoring unimportant regions



Here is shown a non-straight duct contained within a solid block. To compute the flow within it, SP uses a grid which covers the **whole block**. Moreover it repeatedly visits all cells in the grid and **re-computes** the (zero) velocities in the solid.




USP, by contrast, has few cells in the solid region, or even none at all; and it makes calculations only for cells which lie within the duct. (See library case u009)

# General description; the unstructured grid

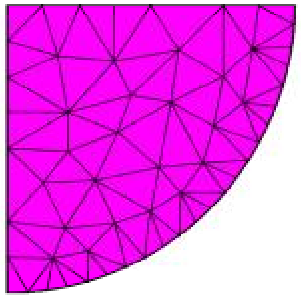


USP is a part of the standard PHOENICS package, which can therefore work in structured or unstructured modes at user's choice.

Setting USP=T in the Q1 file is the first step. Then the user must make decisions about the computational grid which is to be used.

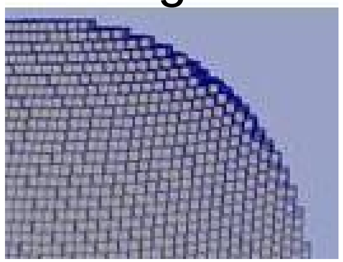
All USP grids consist of Cartesian (*i.e.)* brick-shaped cells.

The general polygonal shapes such as this **à** used in other codes have been judged to be needlessly complex.

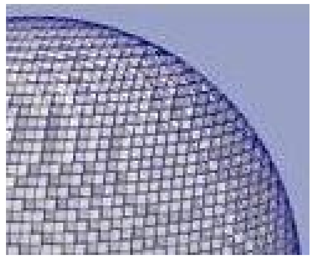


USP cells adjoining objects with curved surfaces can be distorted so as to fit them better, as shown on the right **à**

rectangular



distorted



# General description; how the grid is created



USP employs a standard-PHOENICS cartesian grid as its starting point.

If this is a very fine one it proceeds by **coarsening**, *i.e.* by replacing pairs, quartets or octets of cells by single cells, until the required economical grid is arrived at.

Alternatively, it may start from an already coarse grid and proceed by **refining** it, *i.e.* by halving cells systematically until the grid is sufficiently fine in the regions of special interest.

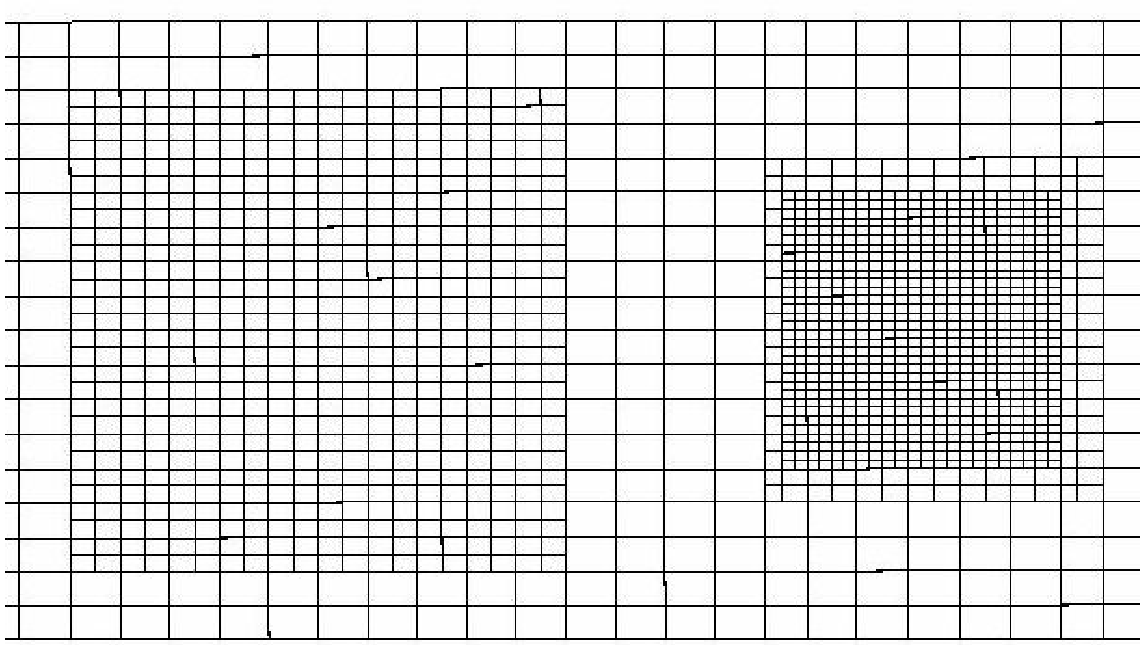
The recently-developed AGG (**Automatic Grid Generator**) module proceeds by way refinement, guided by settings made by the user and by what VR-objects it finds to have been introduced.

AGG is described in more detail elsewhere (AGG.ppt)

# General description; the unstructured grid



Unstructured grids may look, in two dimensions like this **à**



This is the grid which is used for the two-sphere comparison below.

In a USP grid, faces of larger cells may adjoin 2 smaller cells, or 4 in three-dimensional cases, but no more.

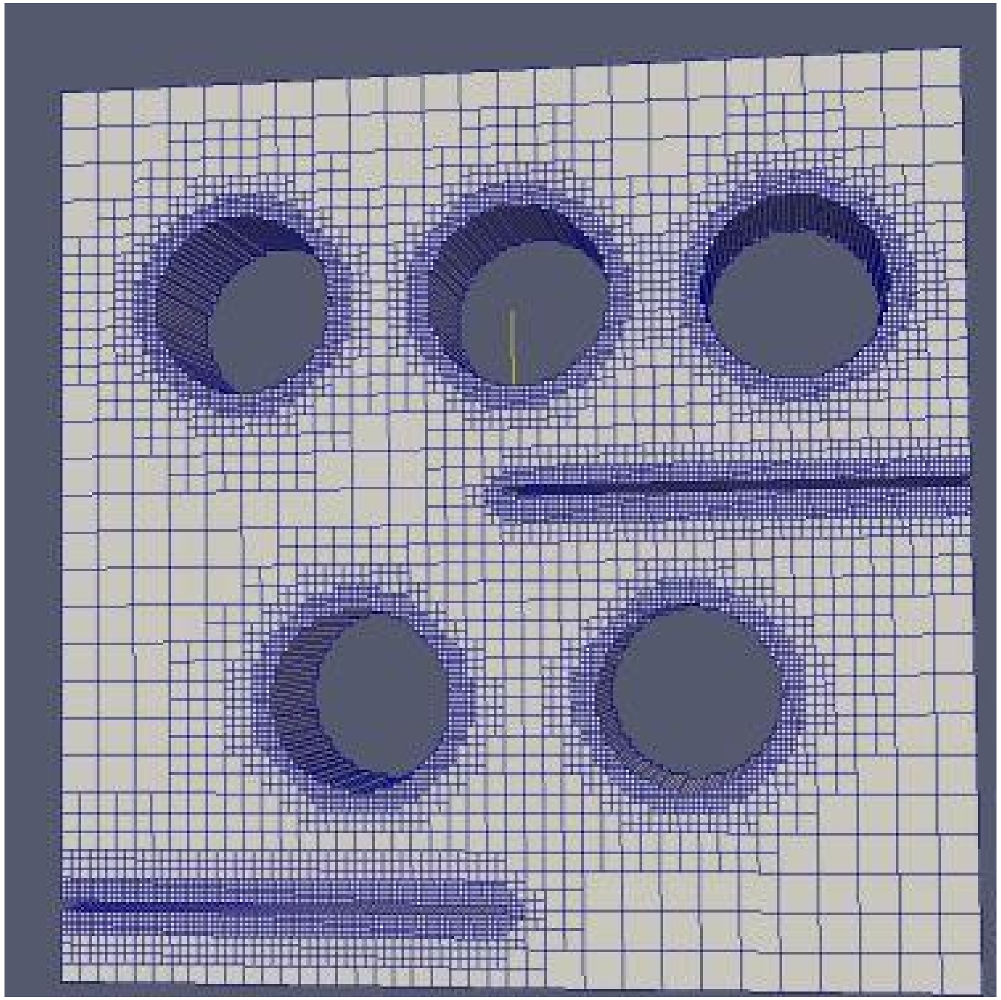
# General description; another 2D grid





This grid was created by means of AGG, the Automatic Grid Generator, a utility program which is supplied with the PHOENICS package.

AGG detects the presence, size and location of facetted 'virtual-reality' objects, and then fits layers of small cells to their surfaces.



# Examples of unstructured-grid flow simulations with AGG-generated grids





The examples concern:

1. heat conduction, two-dimensional
2. heat conduction, three-dimensional
3. flow around a cylinder
4. Mixing hot and cold water in a faucet (tap)

# USP and AGG: Example #1
## 2D Heat conduction in plate with holes

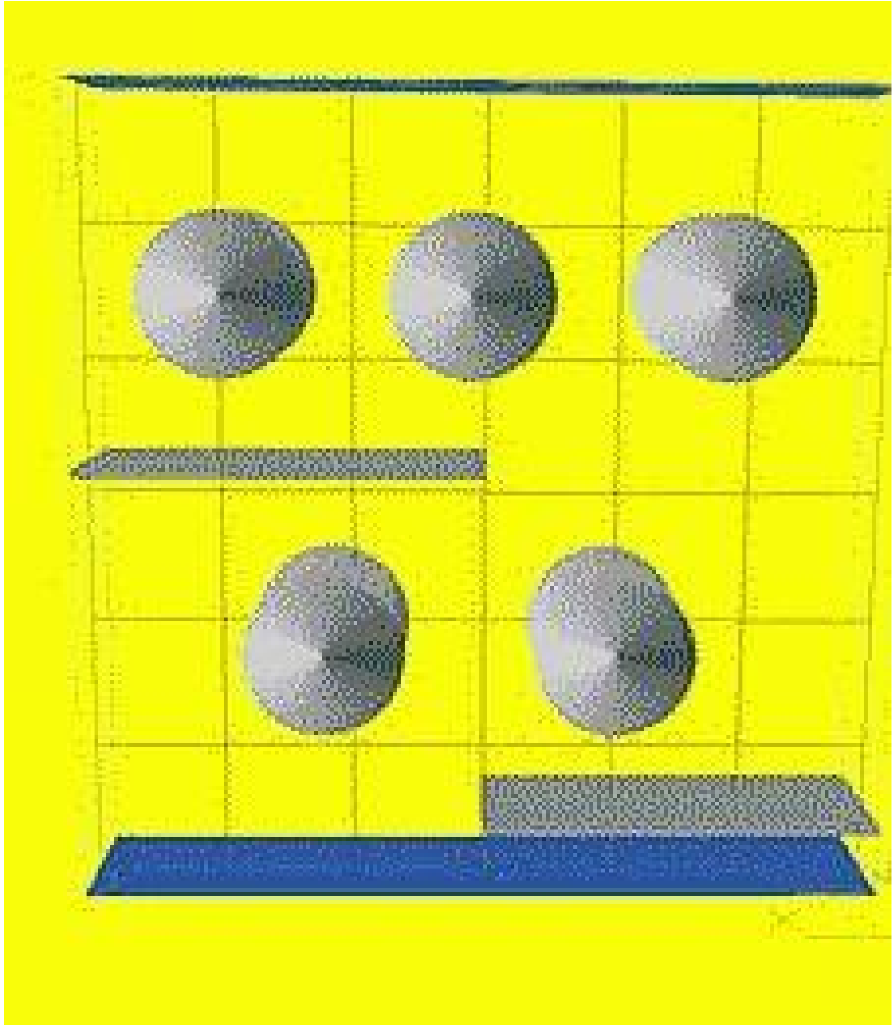


A plate is perforated by holes and slots.

Heat is conducted from the top boundary
at 10 degrees

to the bottom boundary
at 0 degrees.

The **coarse grid** from which AGG starts is shown by the dark lines.



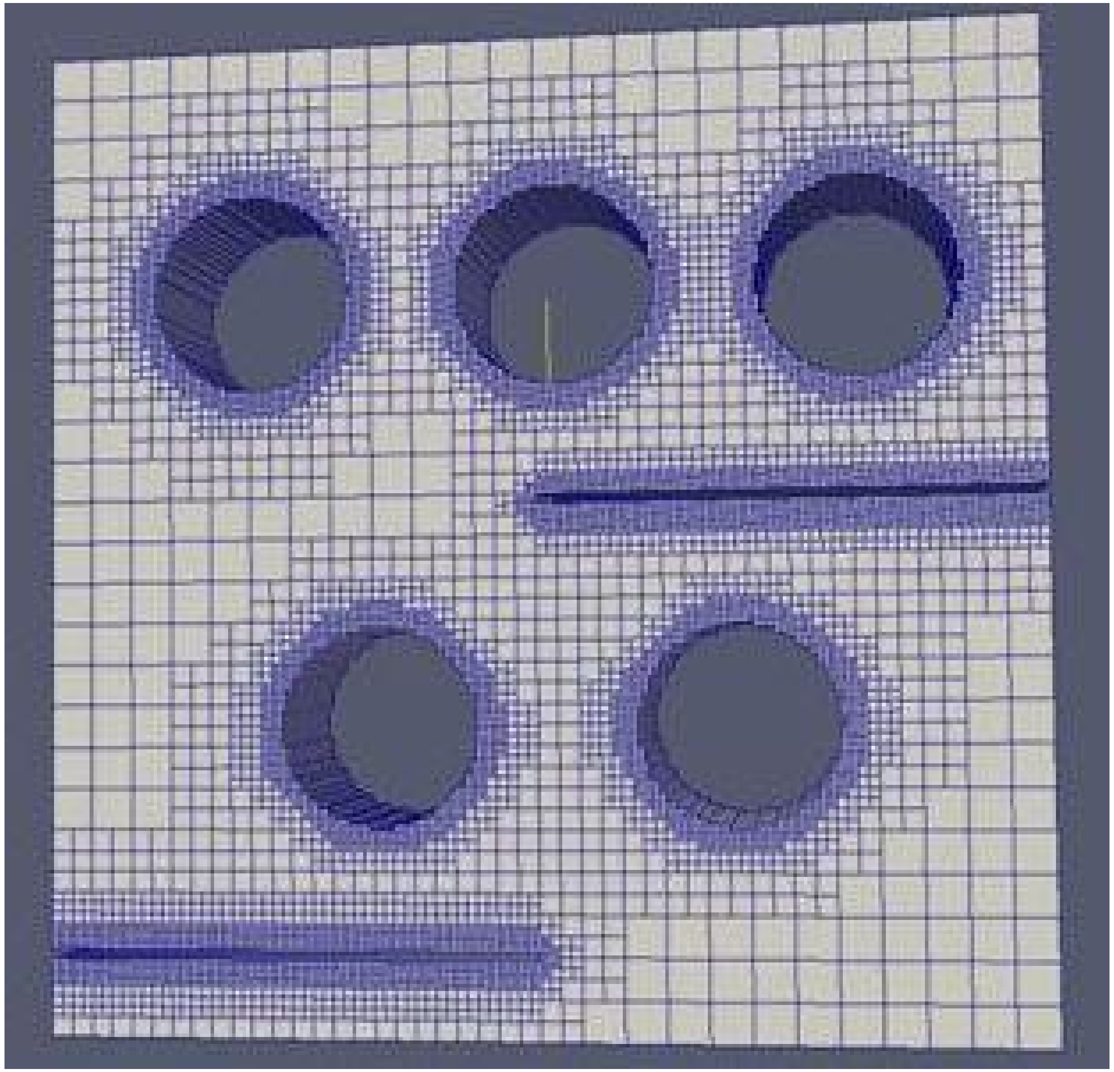
# USP and AGG: Example #1
# 2D Heat conduction





**AGG**, following a few **user-given** instructions in the Q1 file concerning **number of refinement levels** and **how many layers of cells** are to be used at each level, then **creates the grid** shown on the right.

Cells are **smallest** at hole and slot **surfaces**.



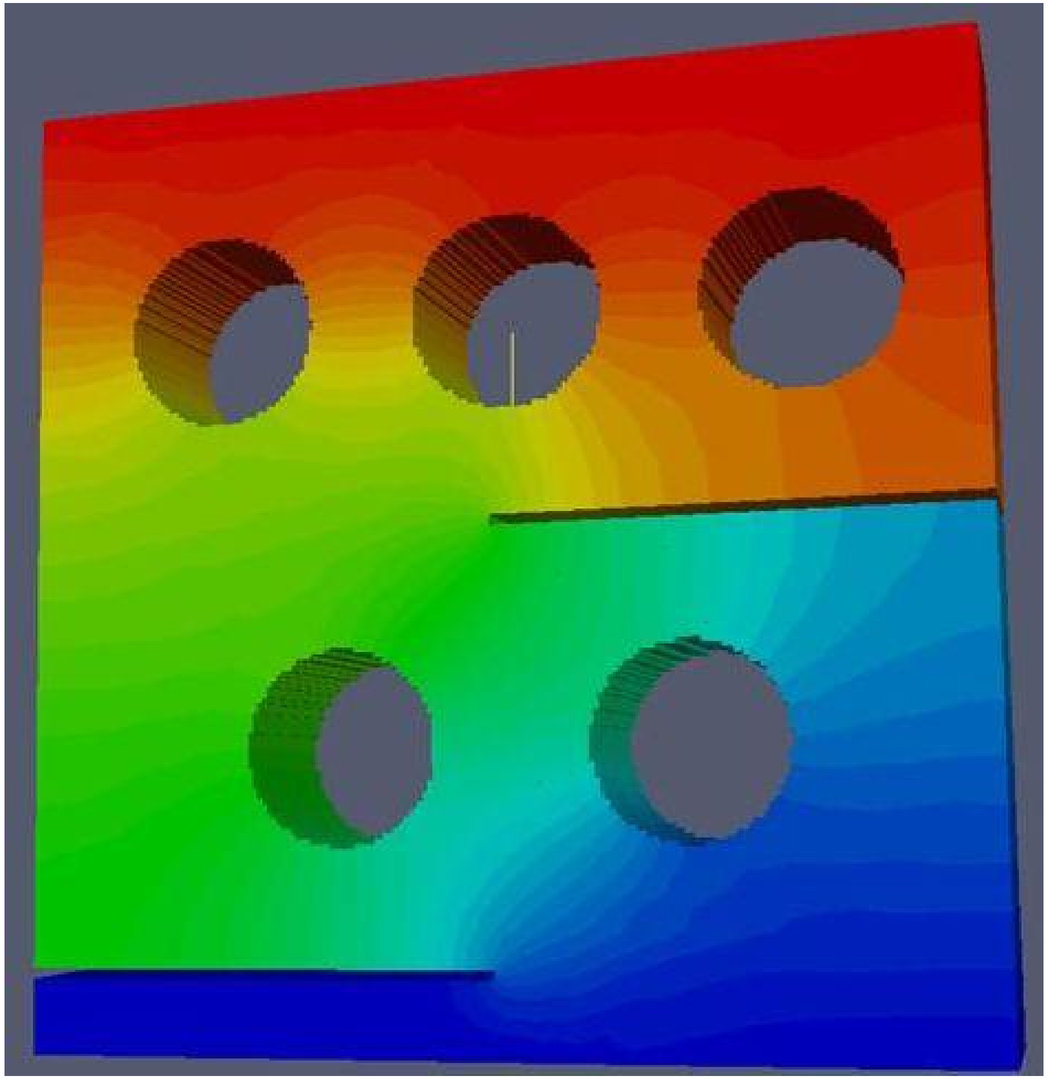
# USP and AGG: Example #1; the results of calculation



The resulting temperature contours reveal the expected effects:

the slots and holes serve as barriers to the flow of heat.

Of course, structured PHOENICS could have solved this problem easily with a uniformly fine grid, but at greater expense.



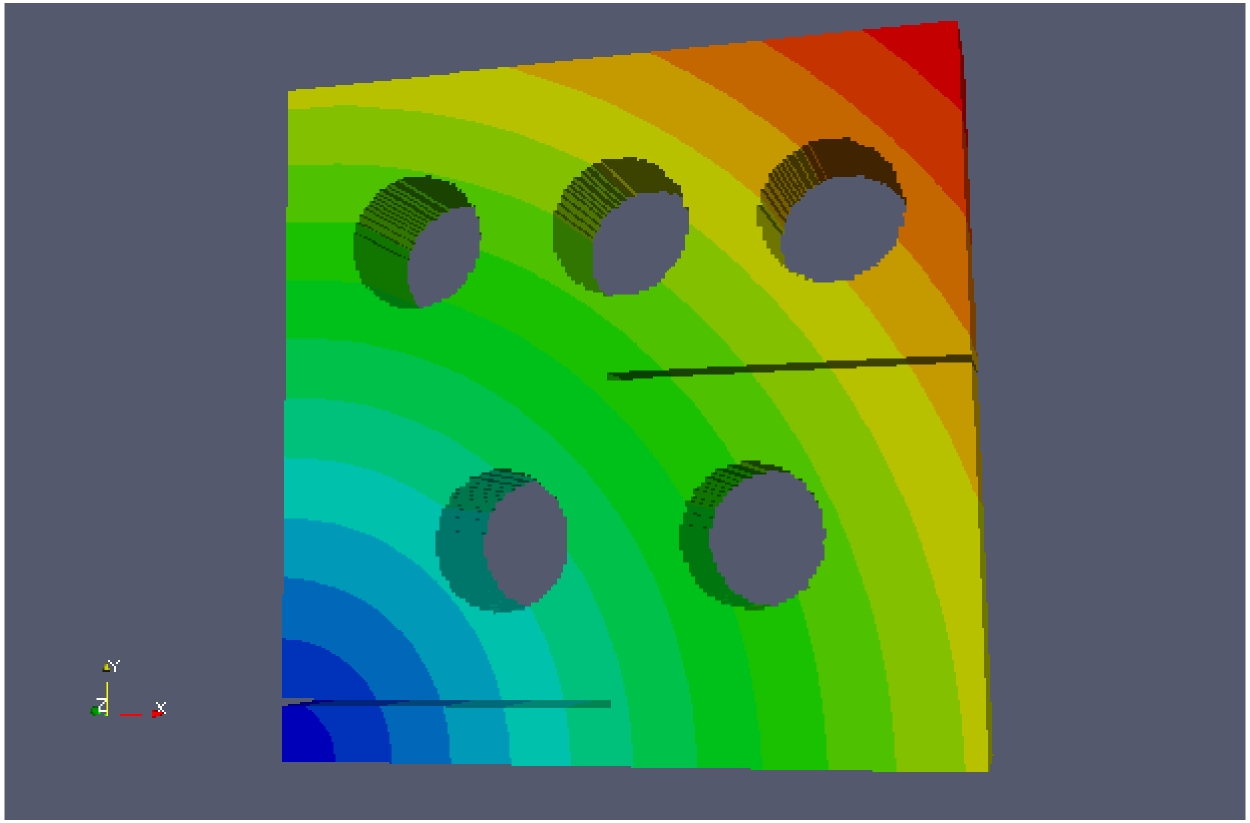
Unstructured PHOENICS
June, 2009

# USP and AGG: Example #1
# In-Form ‘stored’ statement

PHOENICS User Meetings, 2009

Most **In-Form** statements work for USP in the same way as for SP.
Here are shown contours of A001, defined by:

**(STORED of A001 is Rho1*SQRT(XG^2+YG^2) with SWPFIN)**



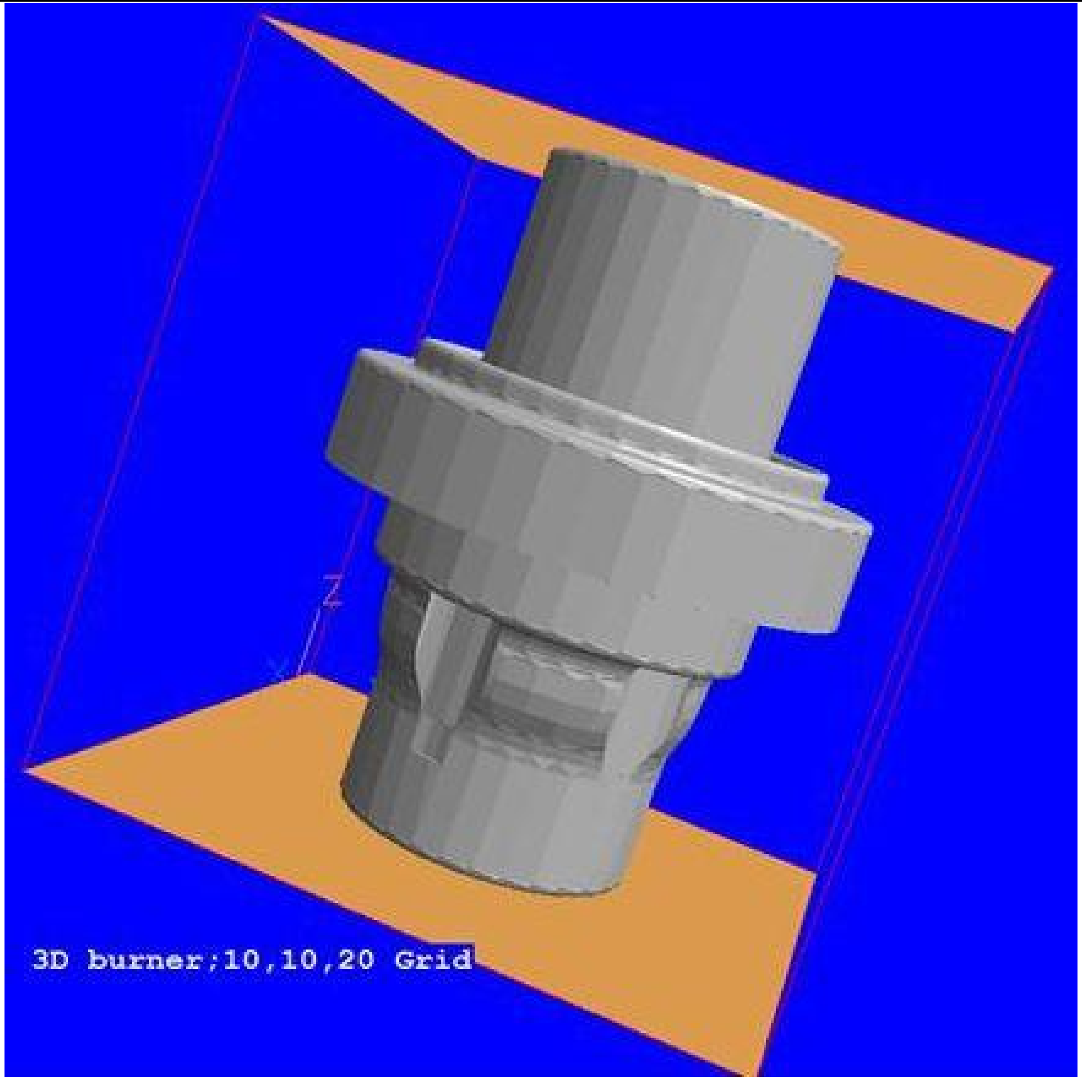
# USP and AGG: Example #2; 3D Heat conduction

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**

Heat flows from the bottom boundary of a hollow 3D object at 10 degrees C to the top boundary at 0 degrees C.

If SP were used:
- a fine grid would have to be used for the whole of the bounding-box space
- most of the computing time would have been wasted.



# USP and AGG: Example #2
# “Heat conduction”



On the right are shown the cells which touch the inner and outer surfaces of the solid body.

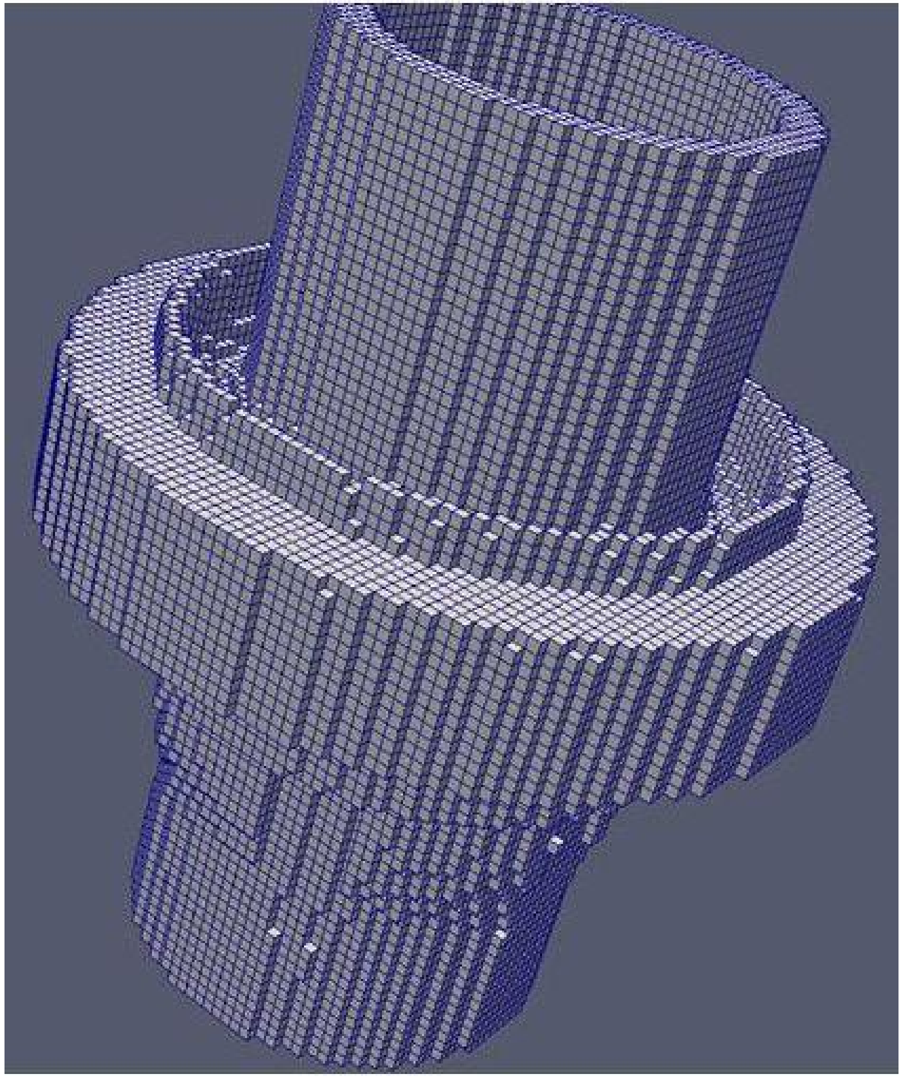
They are of a uniformly small size.

Larger cells fill the remainder of the volume of the object.

No cells exist at all in the non-solid spaces.

AGG has therefore built a grid of maximum economy.

Cell distortion for better fitting is **not** used here.



USP and AGG: Example #2
Computed temperature distribution



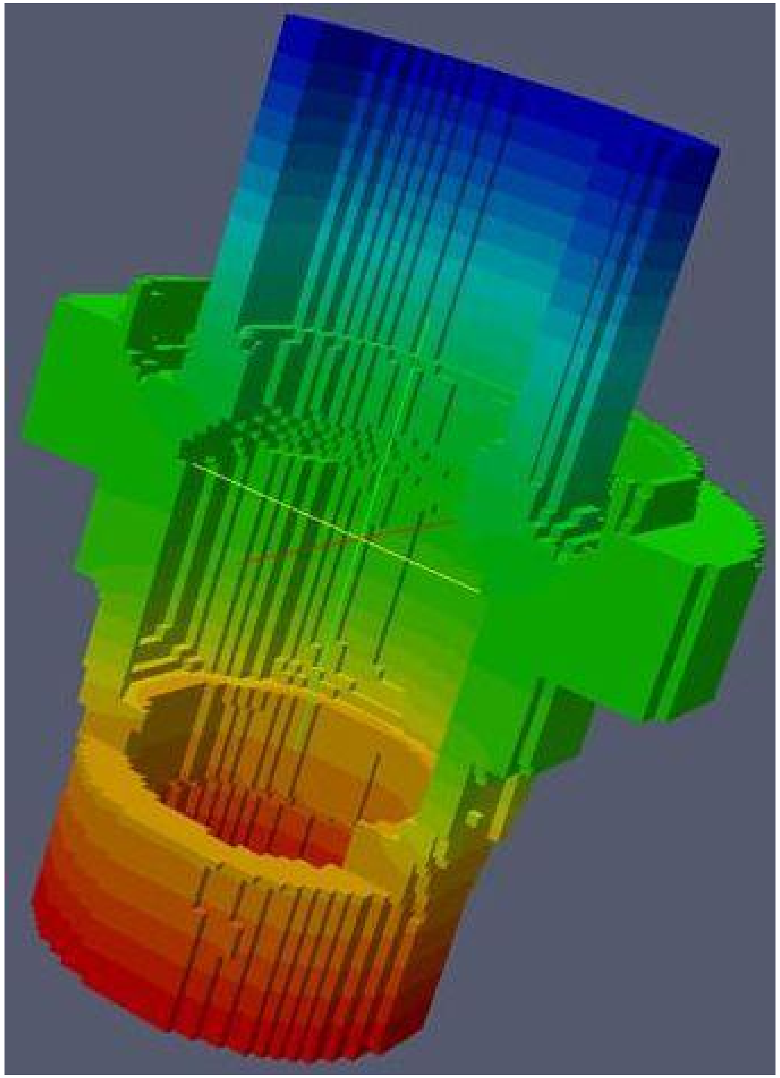


The **temperature contours** are shown on the right.

Part of the body has been cut away in order that the contours on the inner surface can be seen.

If there had been **fluid** inside and outside the body, AGG would have created cells in those regions also.

Then USP would have calculated the temperatures there too; and also velocities and pressures, **there only**.



# USP and AGG: Example #3; Flow around a cylinder



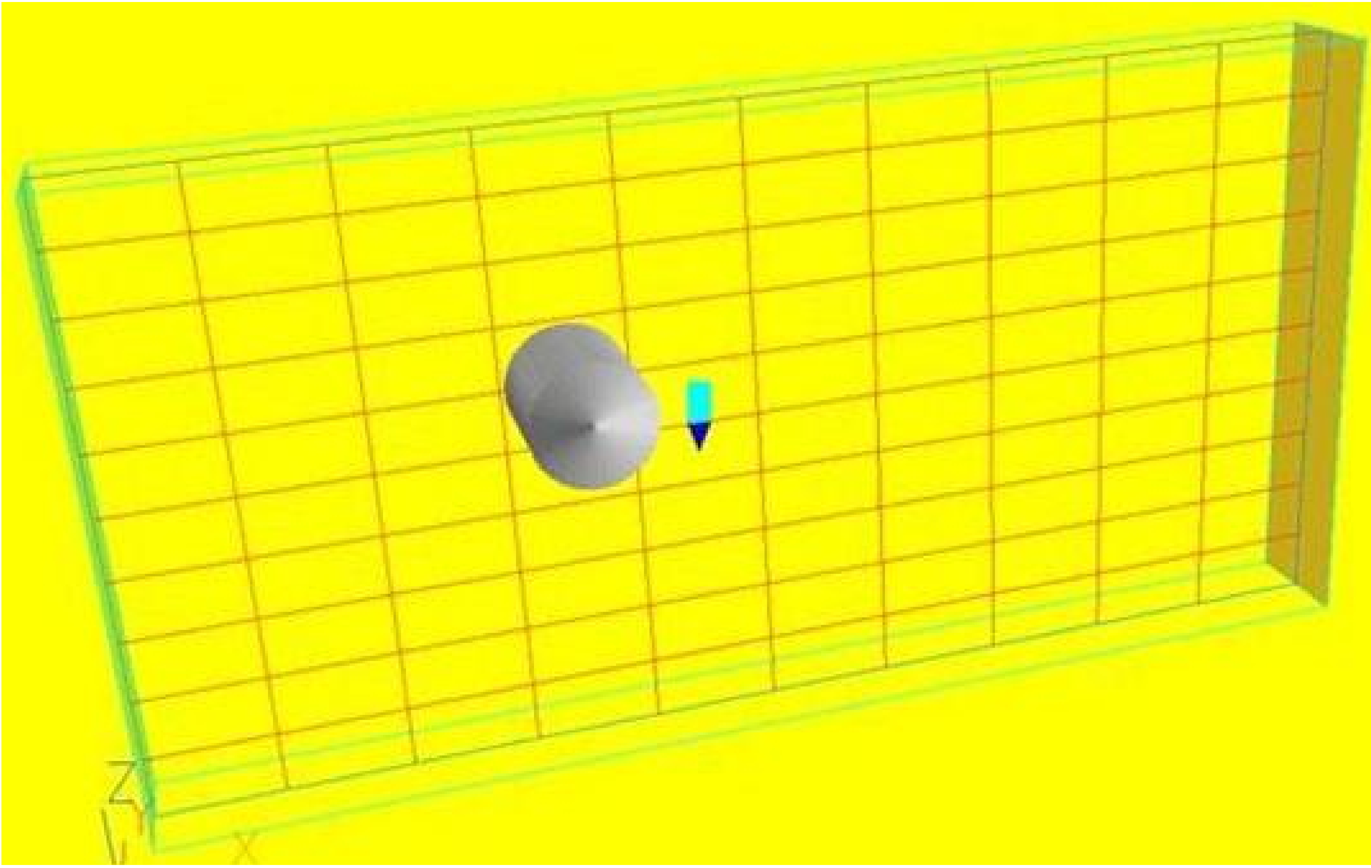


Flow is present in this third example which concerns steady laminar flow around a cylinder within a duct of finite width, from left to right.

The geometry is 2D.

The Reynolds number is 40.

AGG starts with the coarse grid.

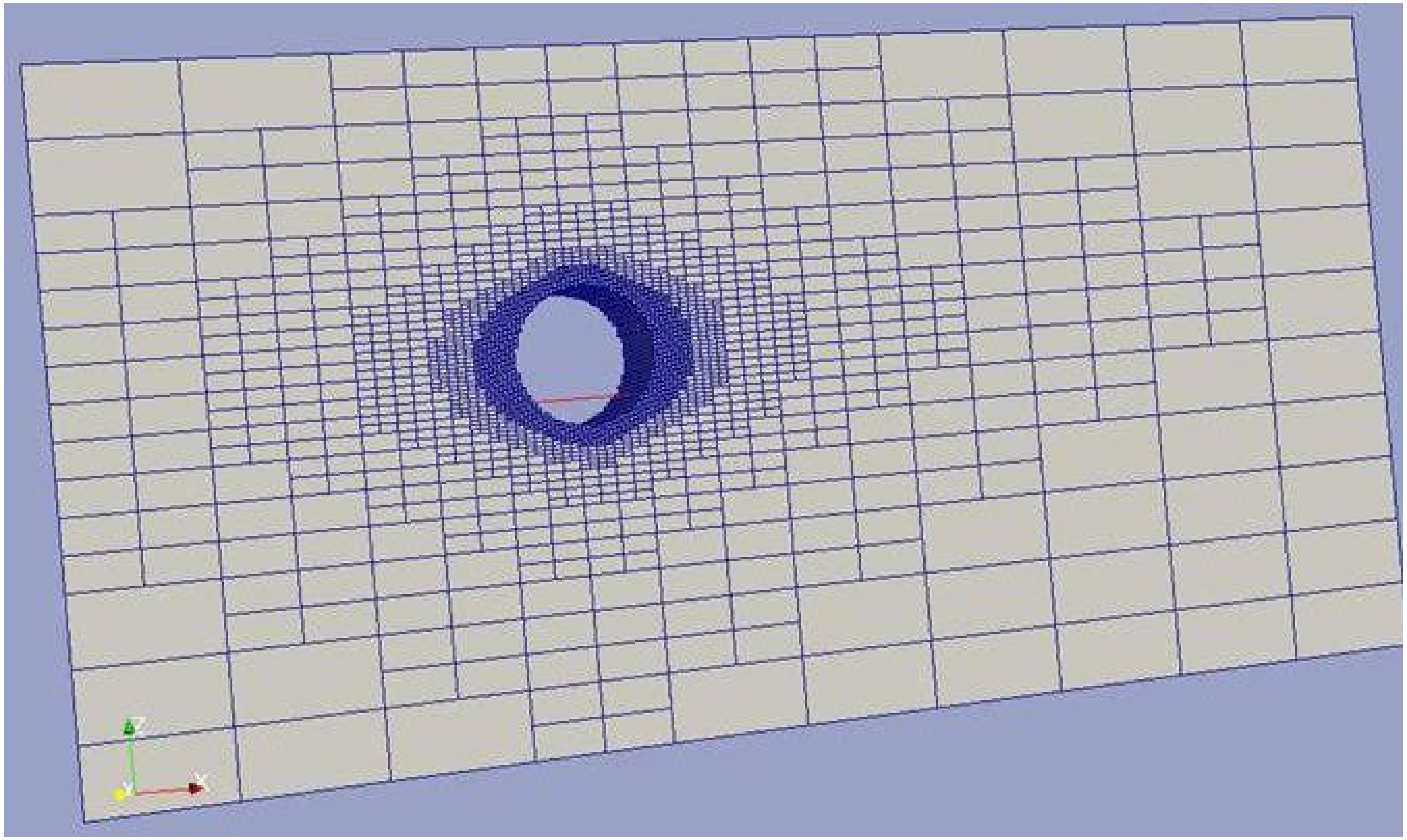


# USP and AGG: Example #3; the unstructured grid





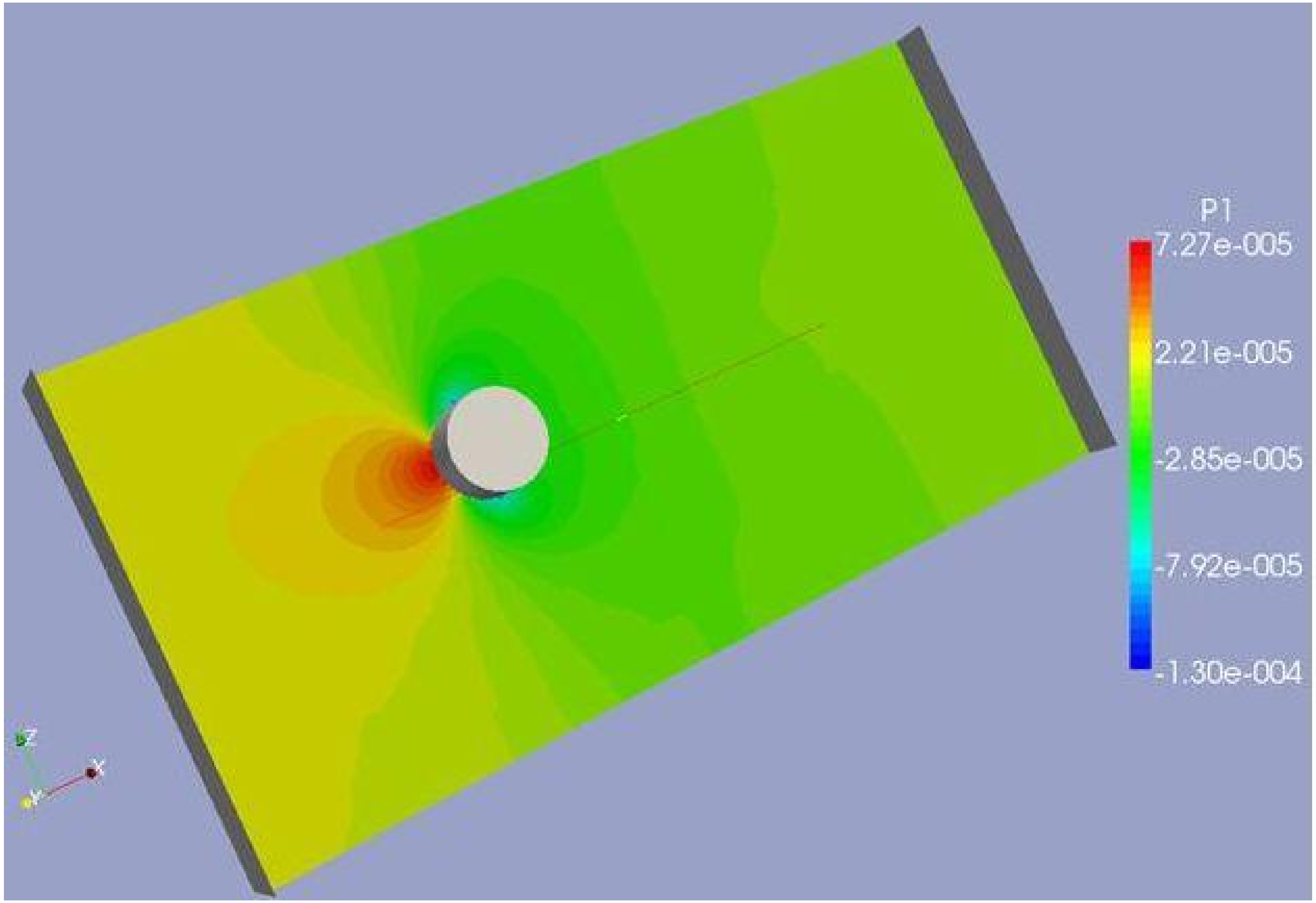
AGG created this grid, with **smallest** cells nearest to the **surface**



# USP and AGG: Example #3
# Computed pressure contours

**Unstructured PHOENICS June, 2009**

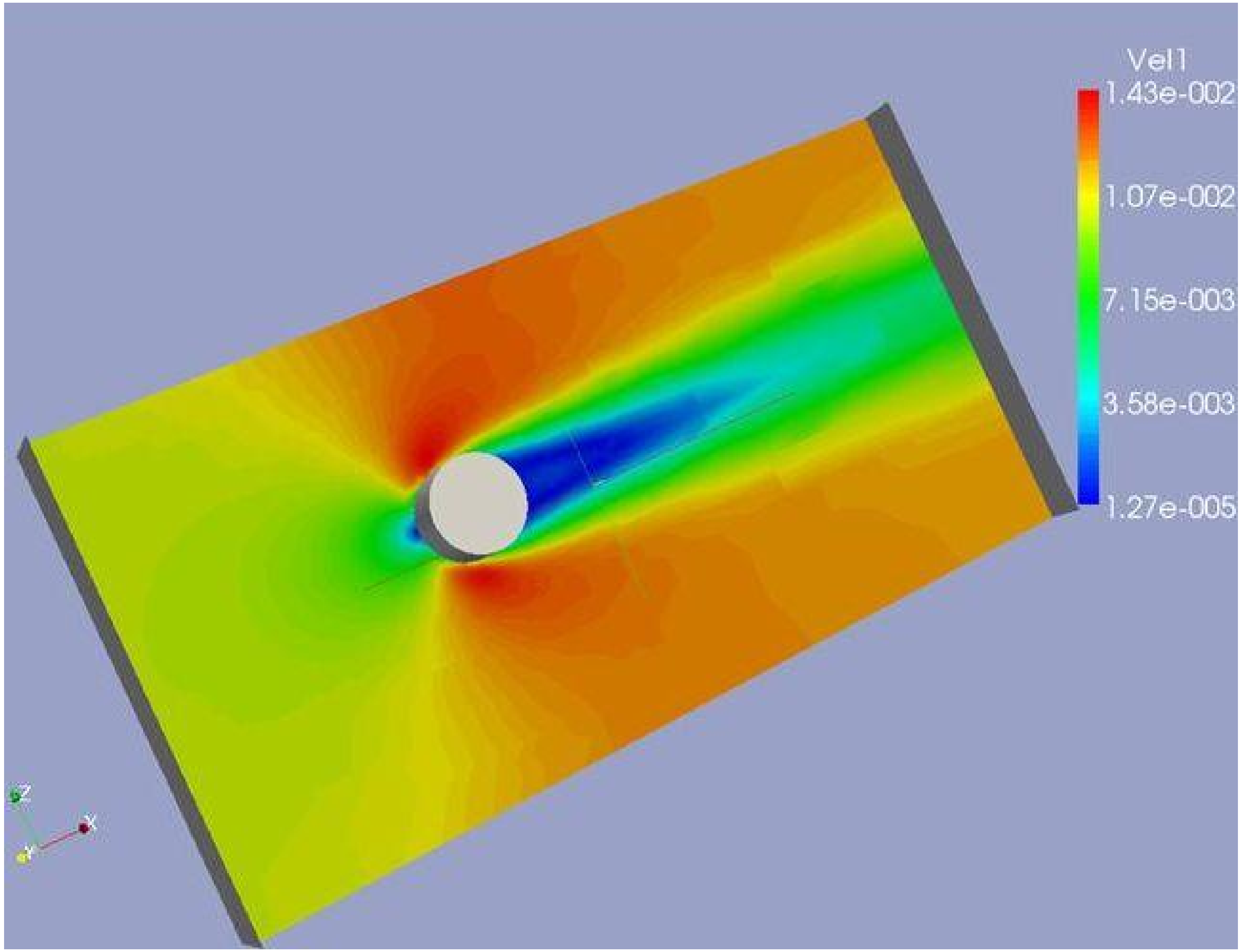
**PHOENICS User Meetings, 2009**



# USP and AGG: Example #3
# Computed velocity contours







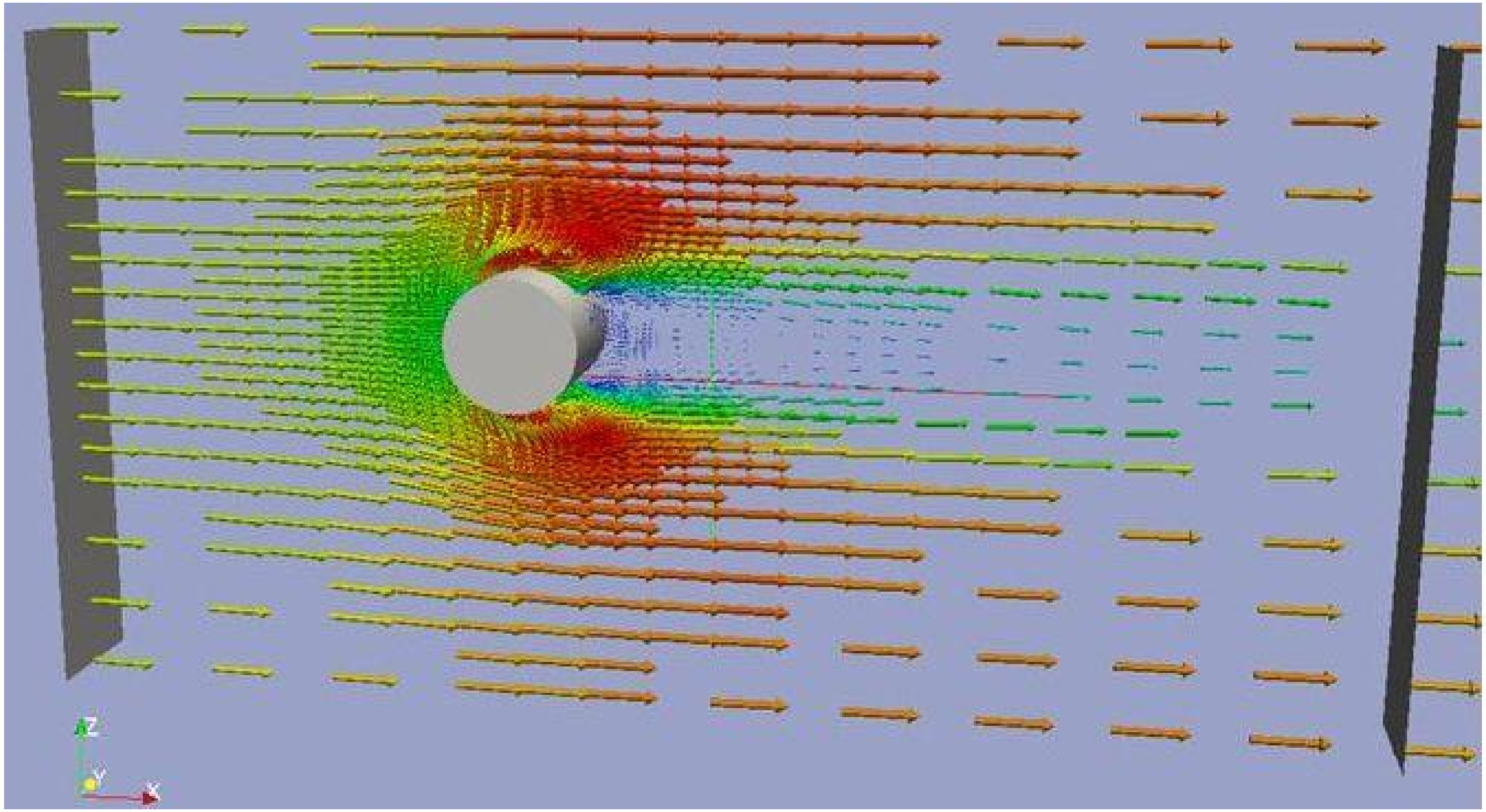
# USP and AGG: Example #3

## Computed velocity vectors

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**

The closeness of the vectors reveals the local grid fineness



# USP and AGG: Example #4; faucet for mixing hot and cold water

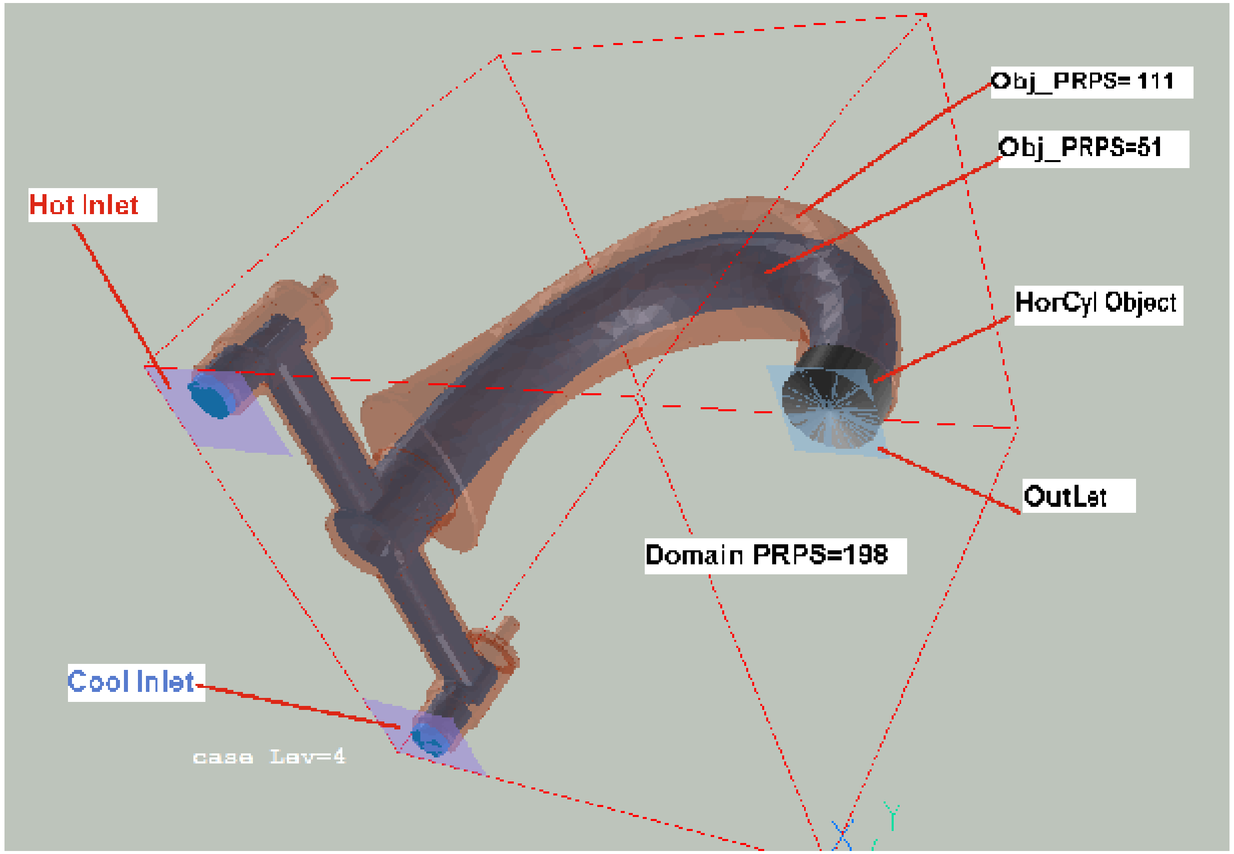
**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**

Structured PHOENICS could have handled example #3 quite well; but it would be **extremely inefficient** if applied to example #4 .

The object represents a domestic hot-&-cold-water tap.

Only internal passages require CFD analysis; but the solid parts conduct heat.



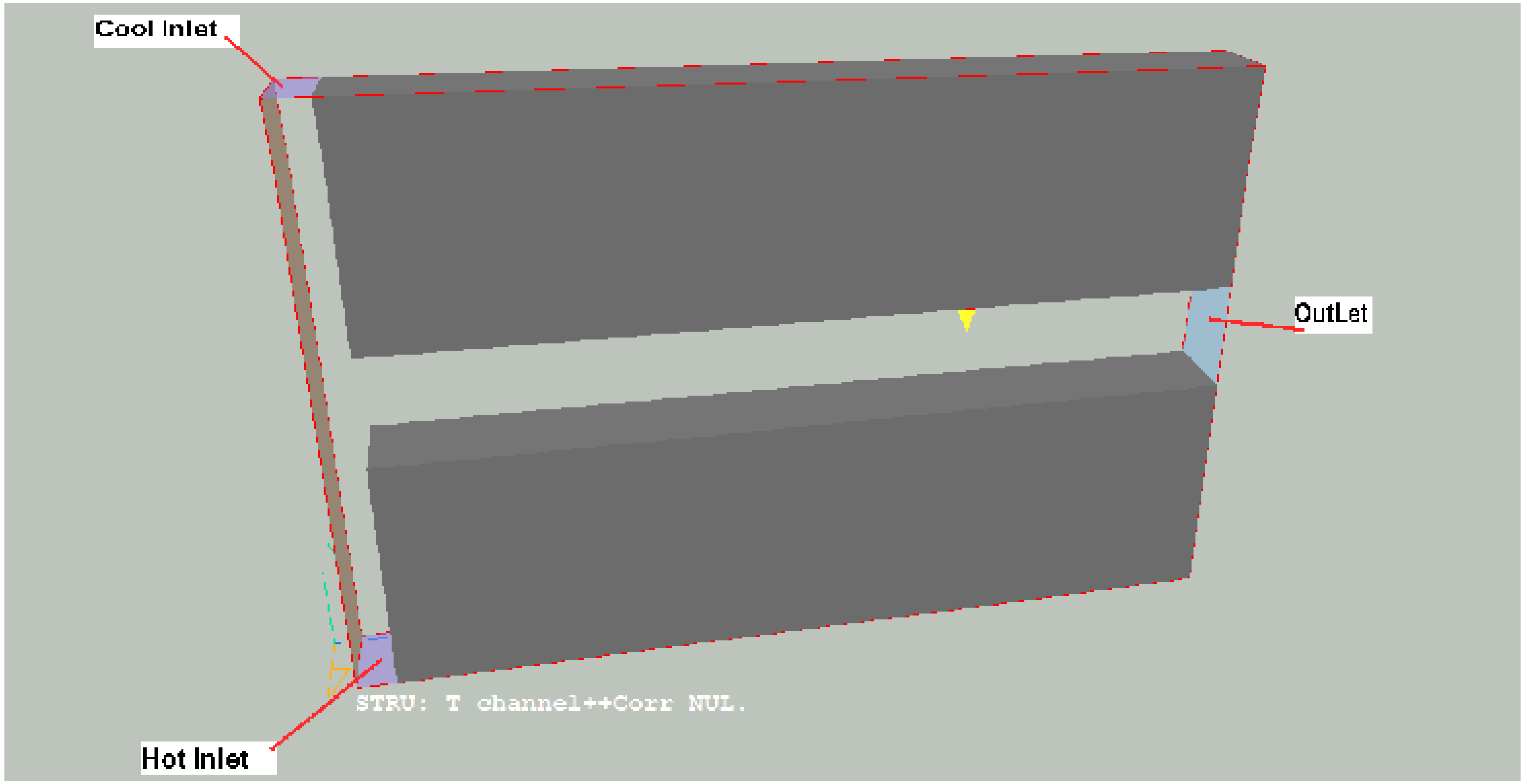
# USP and AGG: Example #4
# Test case: T-channel

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**

A preliminary calculation with simpler geometry was made first with both SP and USP, and with

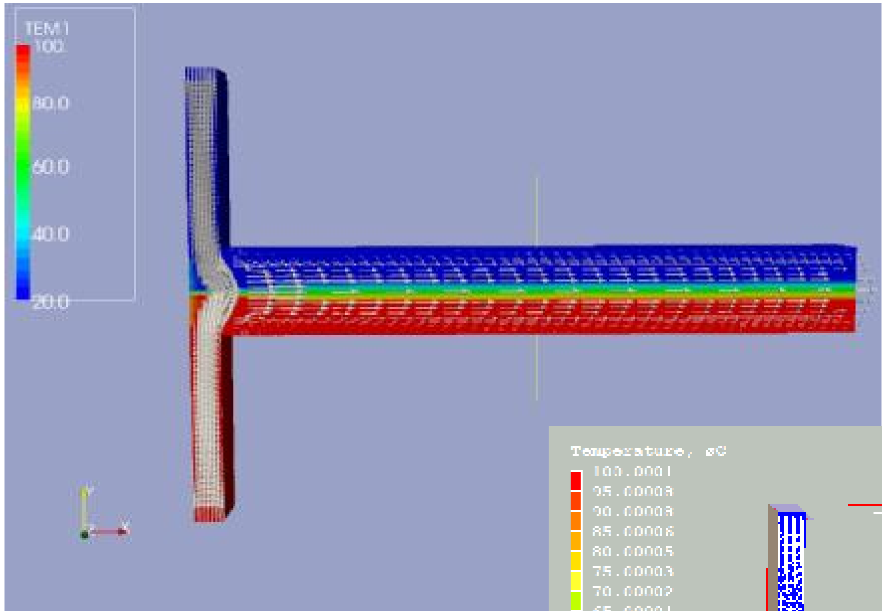
• mass fluxes and temperatures of water, and
• size of channels also the same as in the Faucet.



# USP and AGG: Example #4
# Test case: comparison of SP and USP



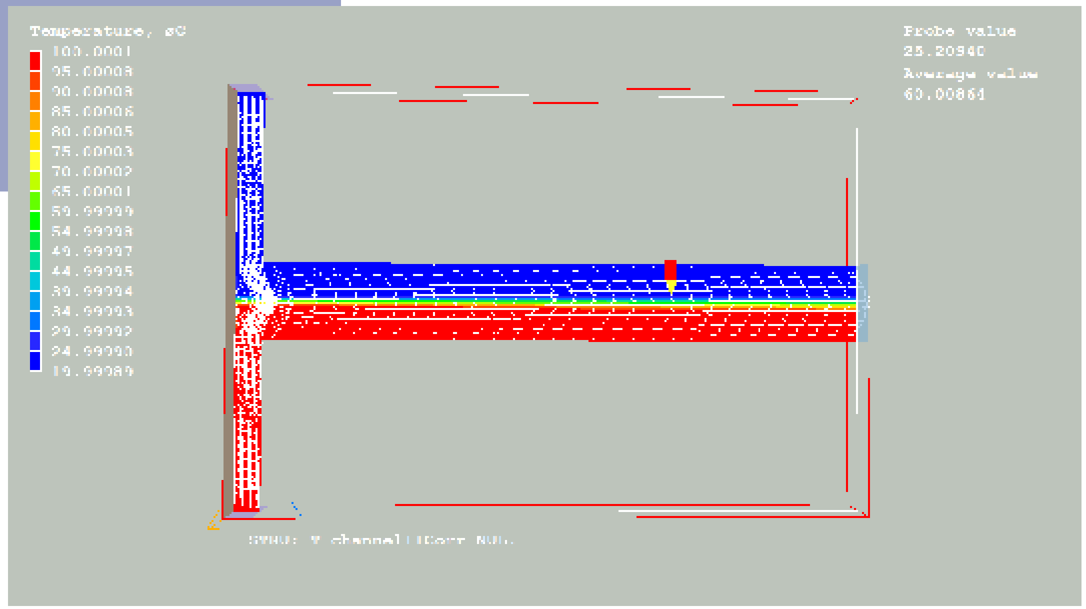





**Velocity** vectors and **temperature** contours

ß USP

SP à

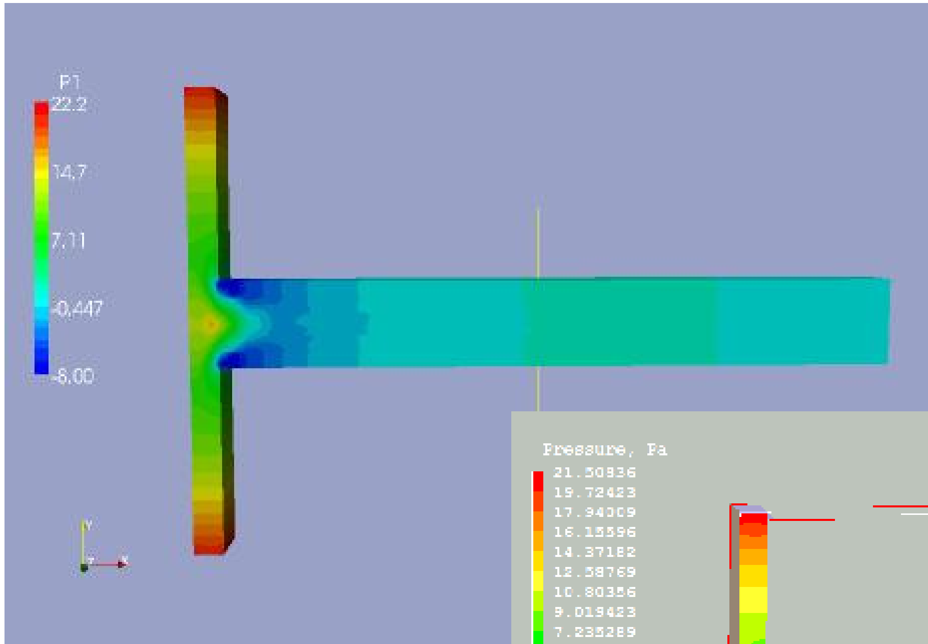



**Note that SP and USP use different display software**

# USP and AGG: Example #4
# Test case: comparison of SP and USP

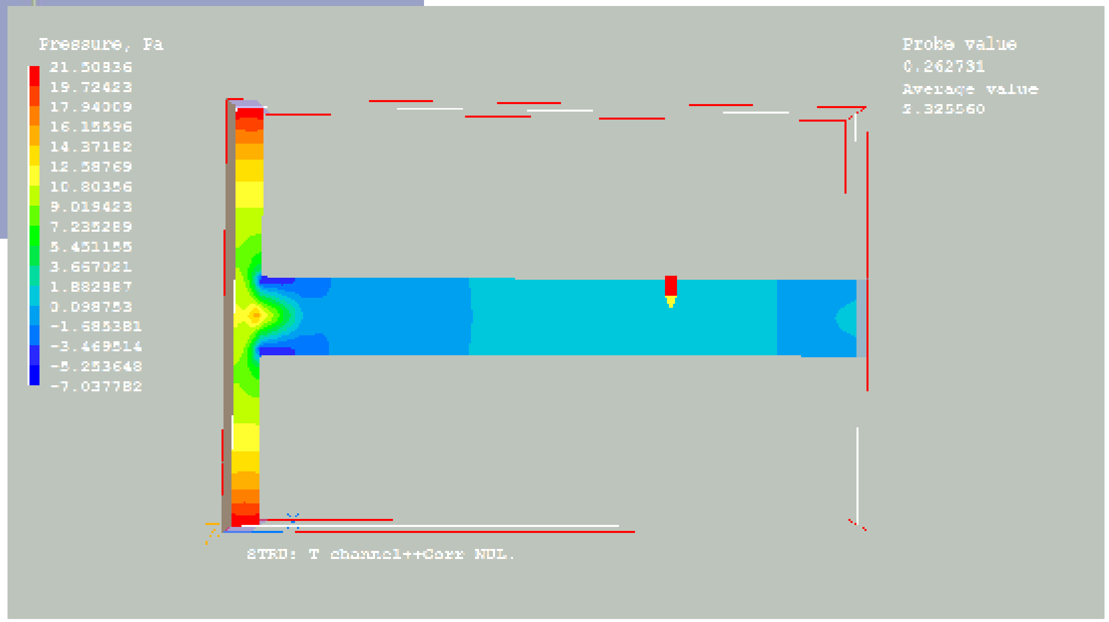


**Pressure** contours




ß USP

P max = 22.2
P min = - 8.0




SP **à**

P max = 21.5
P min = - 7.0

So there are **small differences.**

USP and AGG: Example #4
Grid and PRPS (material index) contours

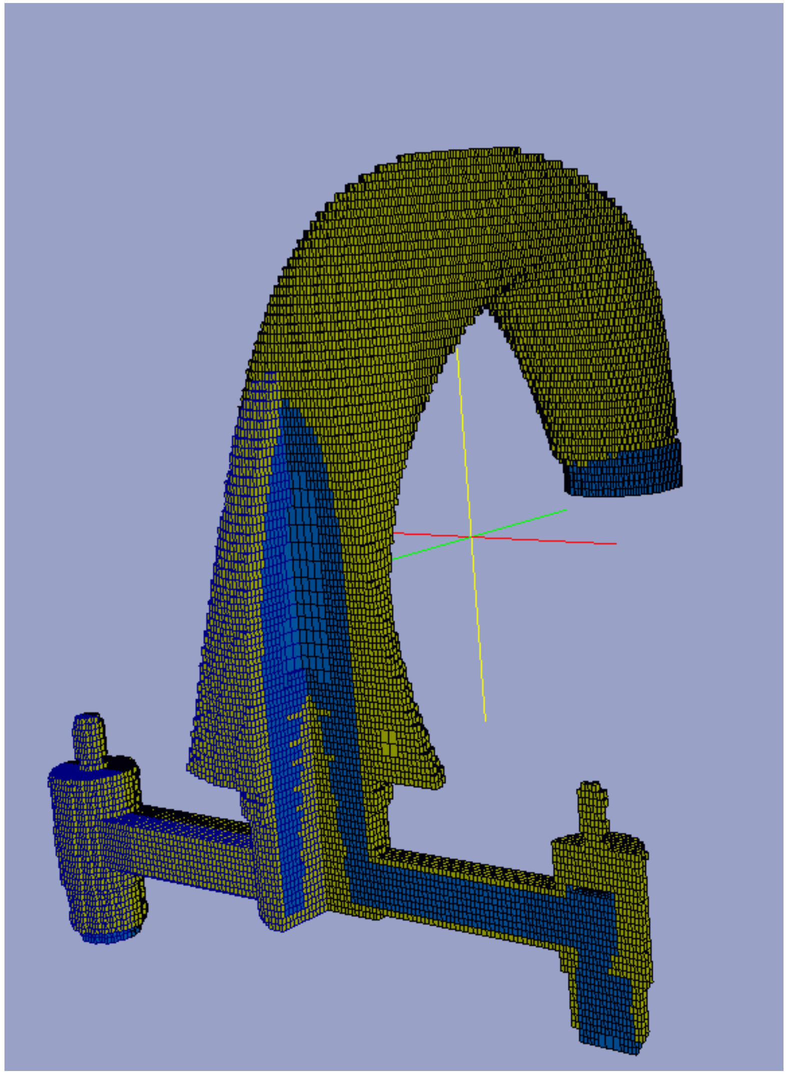




MaxLevel = 4;
*i.e.* there are 4 levels of grid refinement.

The total number of cells is: 174 000

The fluid space is coloured blue; the solid space is coloured olive.



# USP and AGG: Example #4
# Temperature contours

**Unstructured PHOENICS June, 2009**

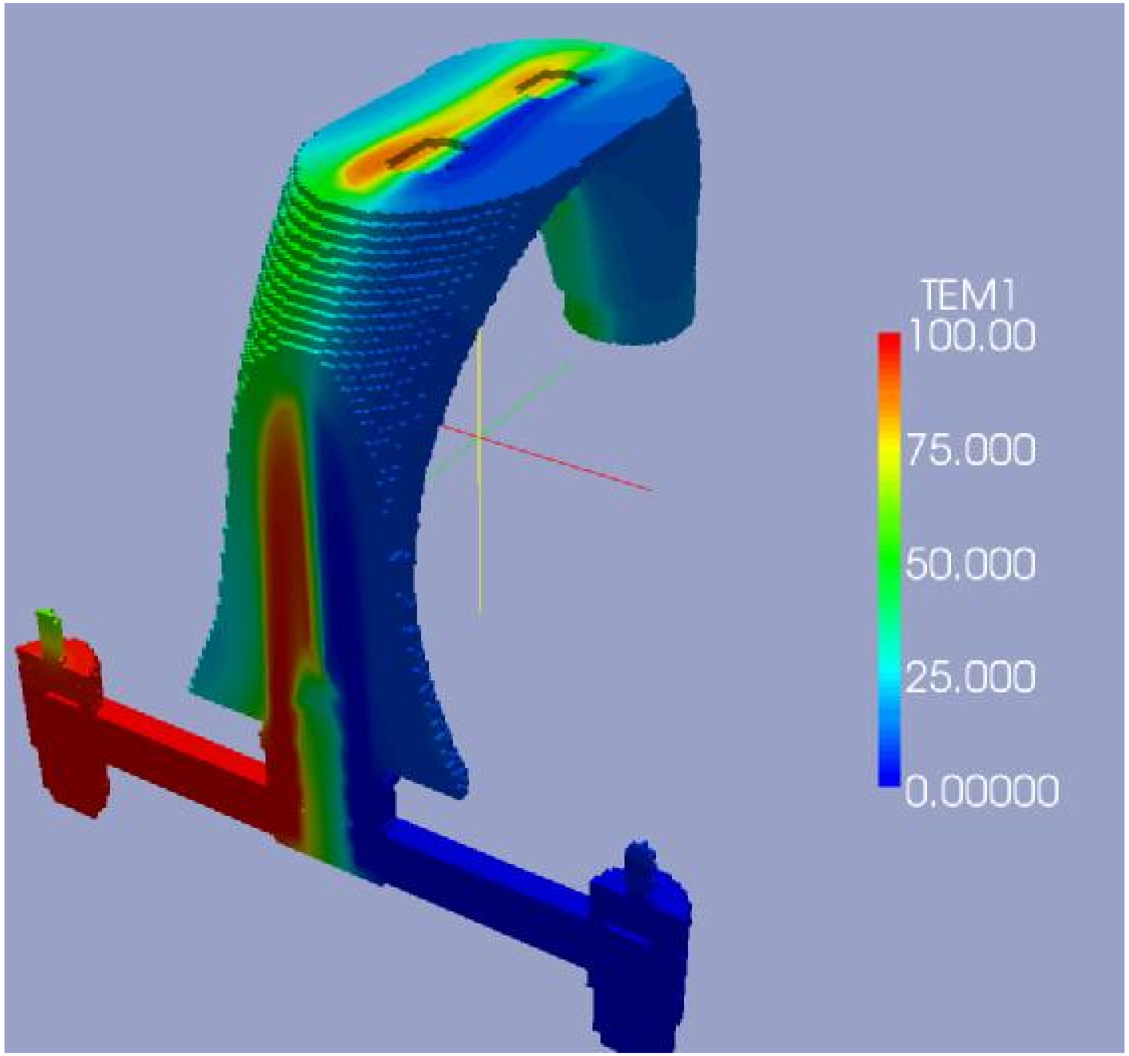
**PHOENICS User Meetings, 2009**

The public-domain package **PARAVIEW** is here used for displaying temperature contours on:

• two cutting planes, and

• part of the outside of the faucet.

The **temperature range** is from **0** to **100** degrees.

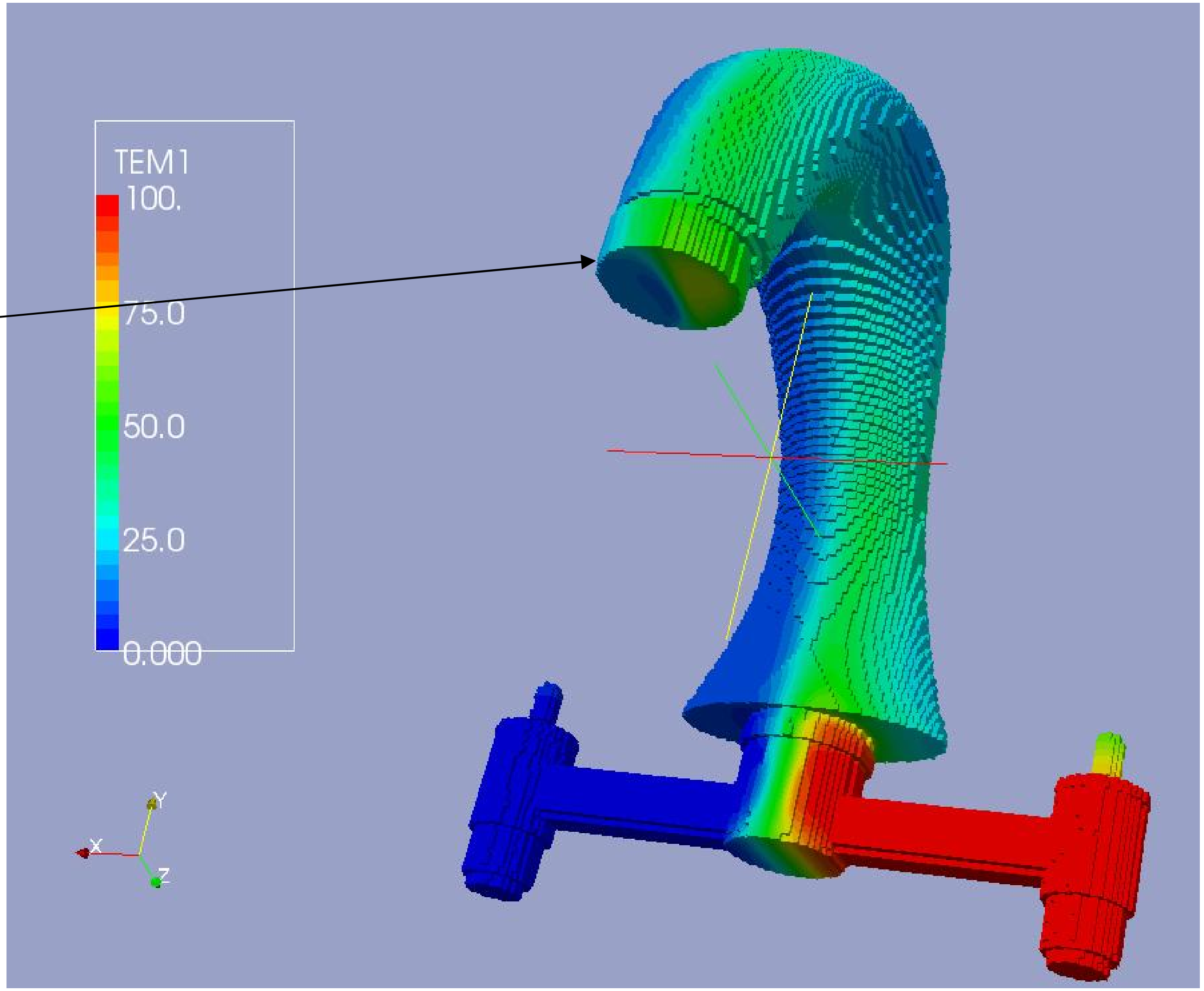


# USP and AGG: Example #4; surface-temperature contours

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**

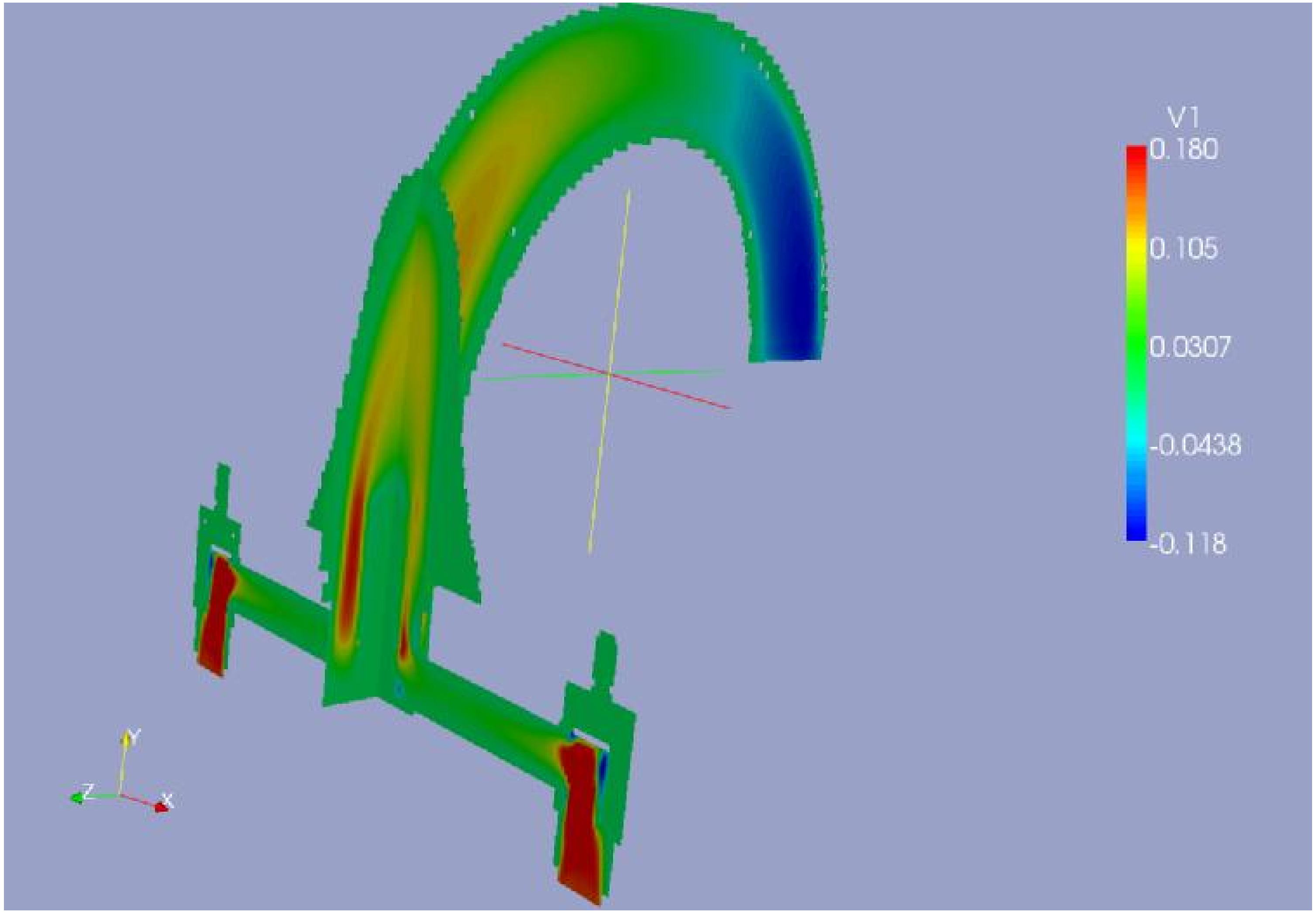
A fictitious cylindrical object has been attached to the outlet so as to enable the outlet pressure to be specified



# USP and AGG: Example #4; Vertical velocity contours

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**



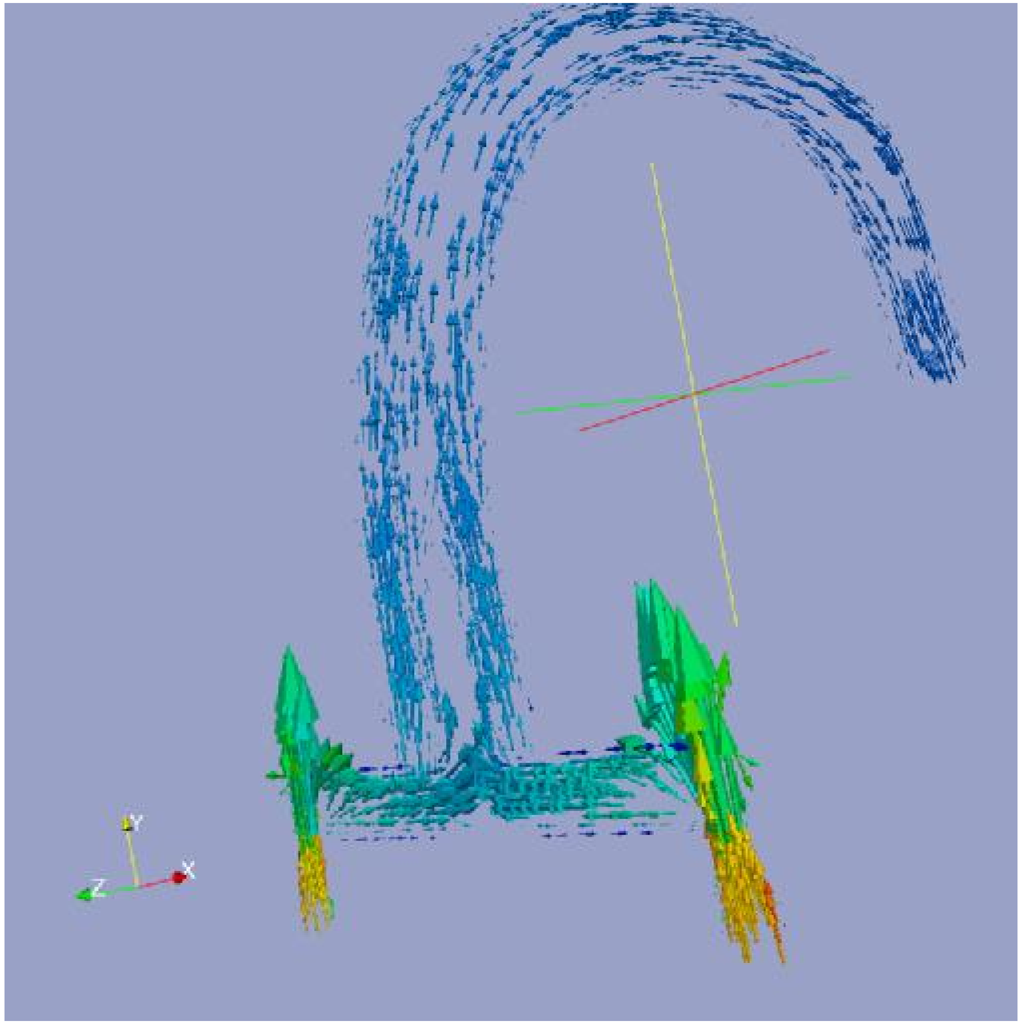
# USP and AGG: Example #4; Velocity vectors (coloured by pressure)





The arrows show the hot and cold entering streams, which flow towards each other.

They then join and flow out together along the curved tube to the outlet.



# **Comparisons between SP and USP**; fine-grid embedding



Since the SP technique of **fine-grid embedding already** allows grids to have **varied coarseness** from place to place, **comparison** is possible and interesting.

However, USP uses a **collocated** (*i.e.* not **staggered)** scheme for the pressure~velocity interactions; therefore some differences are to be expected.

The flow around **two spheres** has been calculated in both structured and unstructured modes (Input-file-library **case u208**).

**For equal numbers of cells**, the ratio of computer times was **333 : 72** . So **USP was** more than **four times faster than SP**.

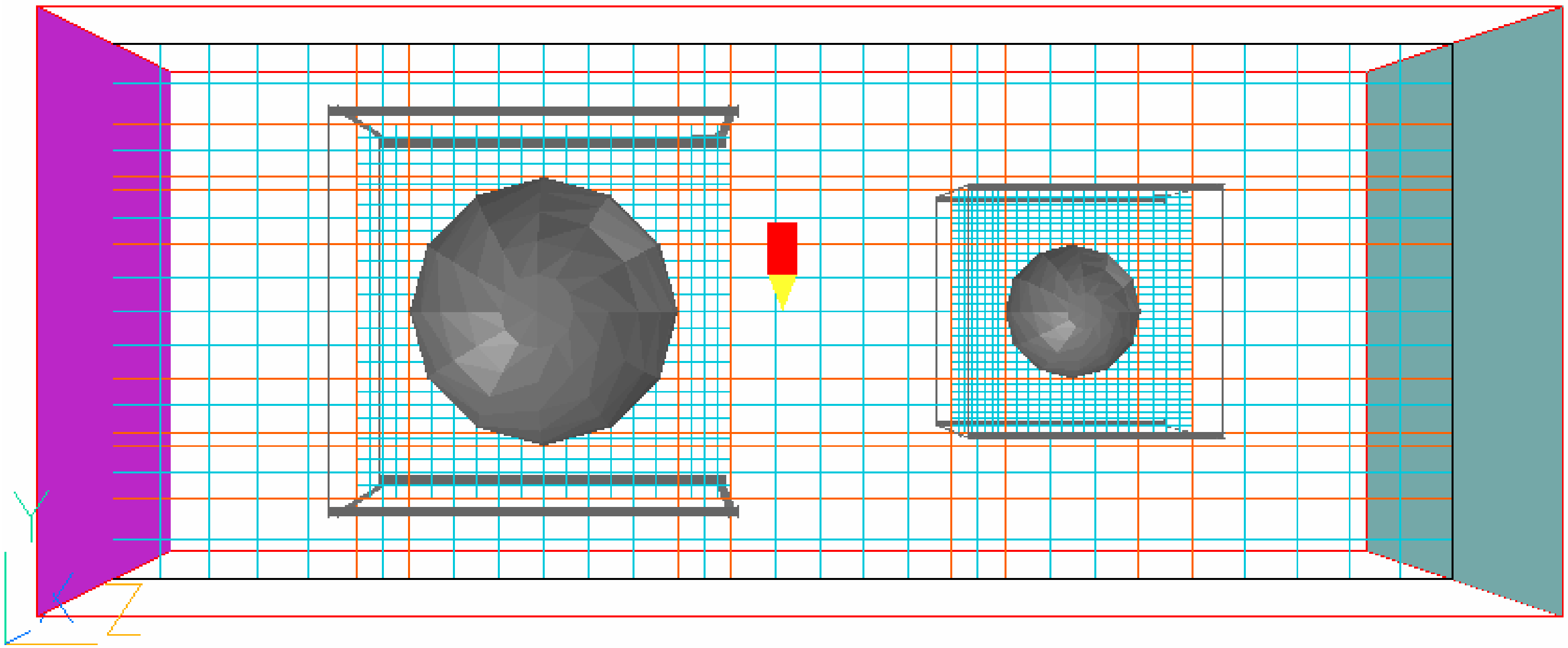
The **results** will now be displayed graphically.

# Comparison USP via SP + FGE for flow around two spheres; grids

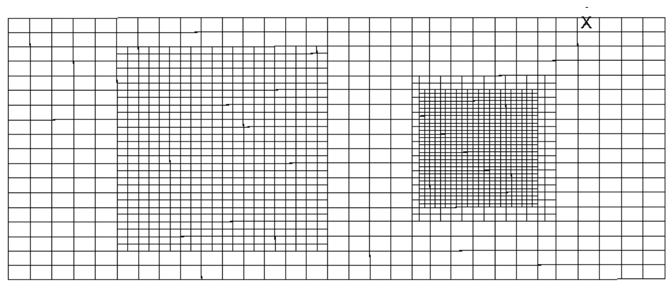
**Unstructured PHOENICS June, 2009**

PHOENICS User Meetings, 2009

**The SP** grid, with fine-grid embedding is shown below




The corresponding **unstructured** grid was as shown here **à**
(with a smaller scale)

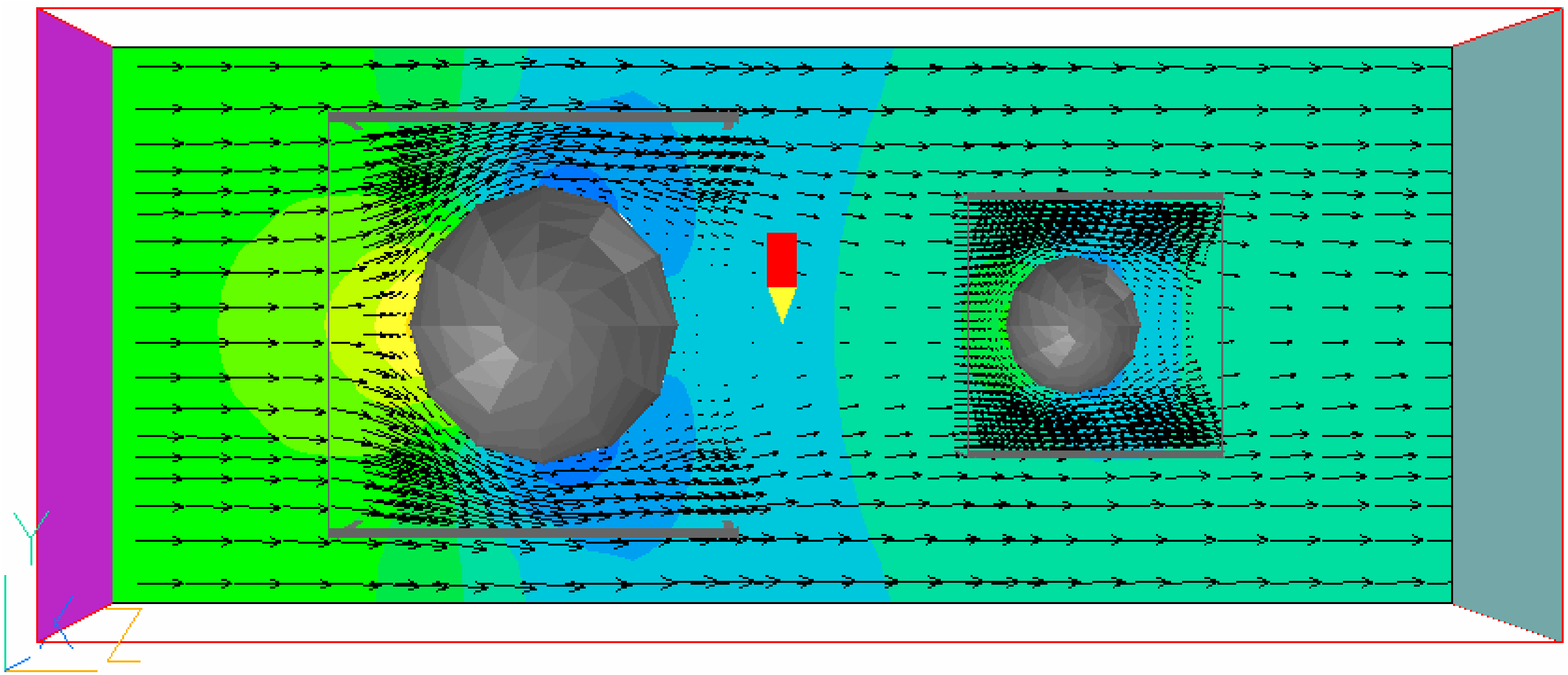


# Comparison USP via SP + FGE for flow around two spheres: SP

Unstructured PHOENICS June, 2009

PHOENICS User Meetings, 2009

## SP+FGE: results



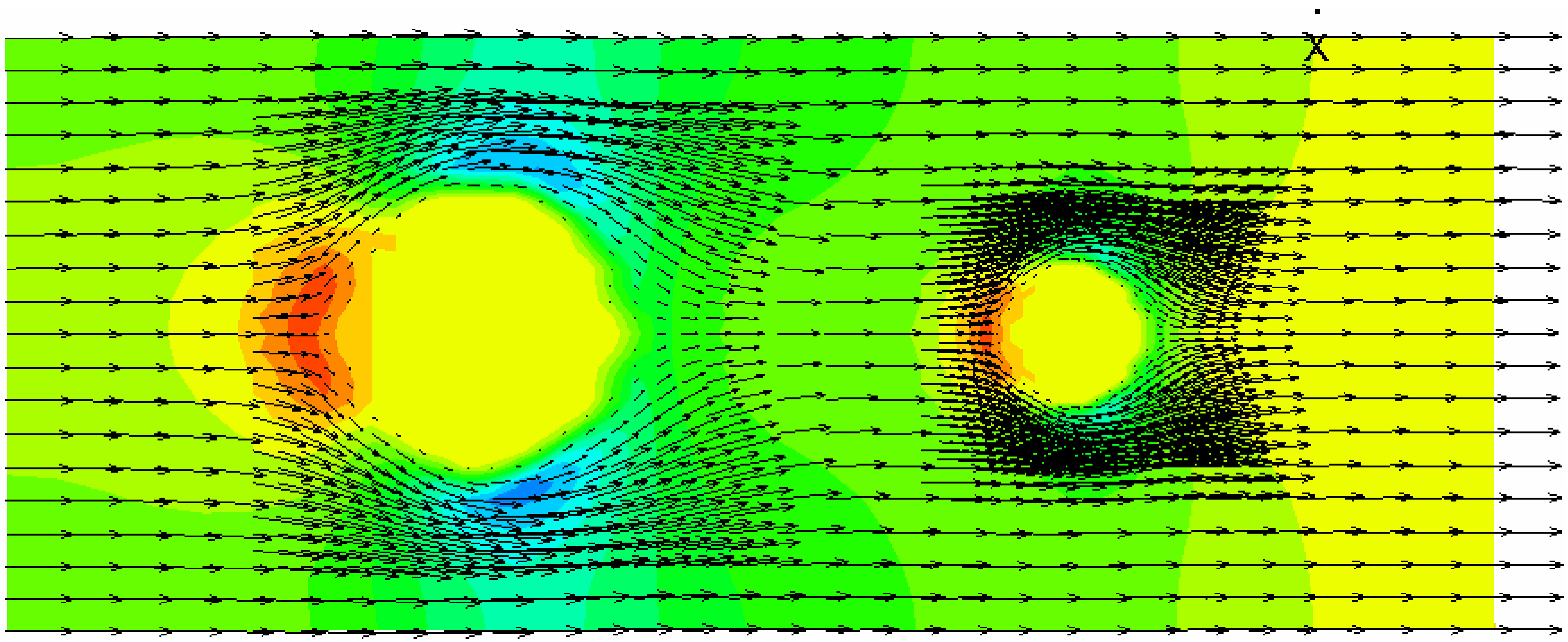
**Elapsed time is 333 seconds on PC pentium-IV, 2.4 GHz**

Comparison USP via SP + FGE
for flow around two spheres: SP





## USP: results




**Elapsed time is 72 seconds on PC pentium-IV, 2.4 GHz**

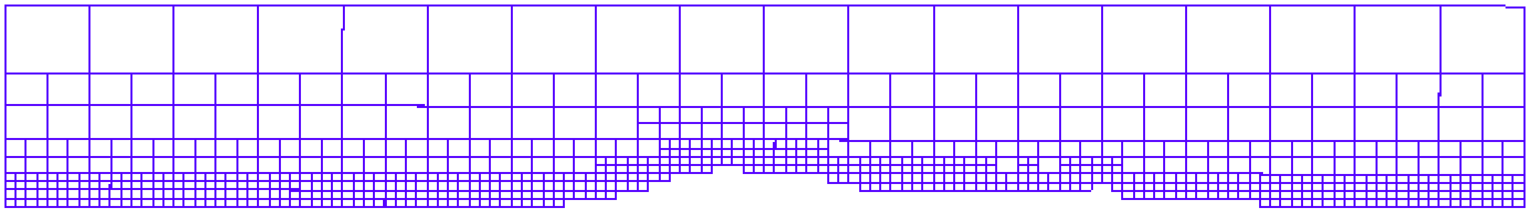
The results of SP and USP were essentially similar; **but** the latter were obtained much more rapidly.

# Further comparisons of SP and USP; flow over terrain

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**

USP is particularly useful for **flow-over-terrain** problems,where **fine** grids are required **near the ground,** whereas **coarser** ones suffice for **higher altitudes**.



For given fineness near the ground, USP **uses fewer** cells than SP.

For the same number of cells, USP's grid is **finer** near the ground.

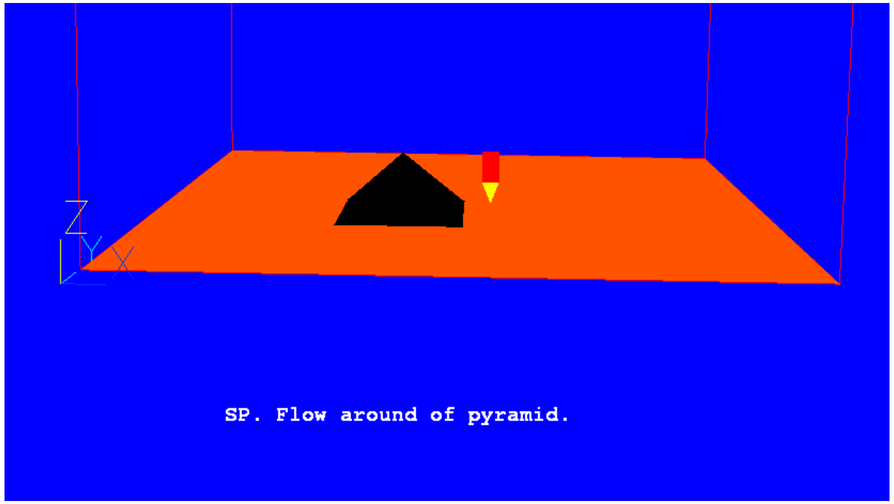
The results of two test cases are shown below:

1. Flow over a **pyramid-shaped mountain**
2. Flow over **natural terrain.**

## Case 1. Flow around pyramid

Size of domain: 10x10x4m
Inlet Velocity: 1 m/s
Effective viscosity: m**2/s

Sizes of smallest cells are same for SP and USP

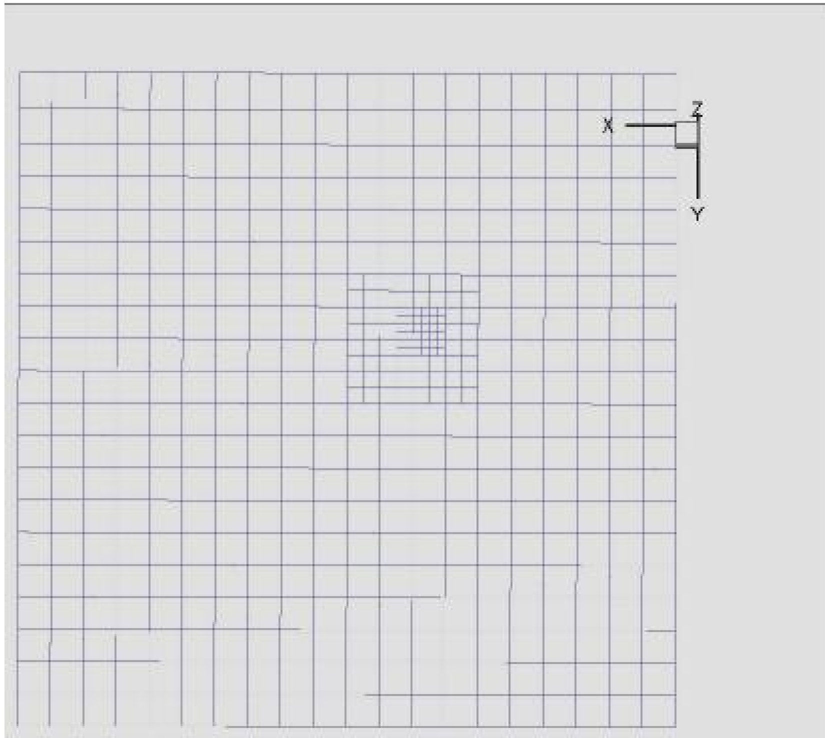



**Structured** grid is uniform with 80x80x32=
**204,800** cells.

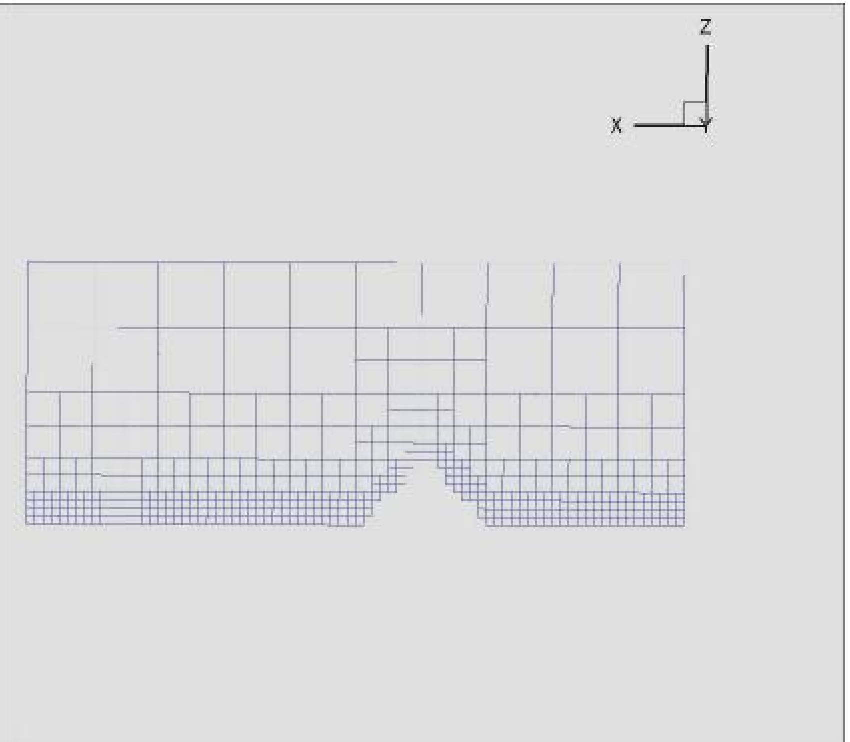
**Unstructured** grid has
**29,778** cells
96,934 faces
Refinement level = 4.

# Case 1 Unstructured grid






Z = 1 m

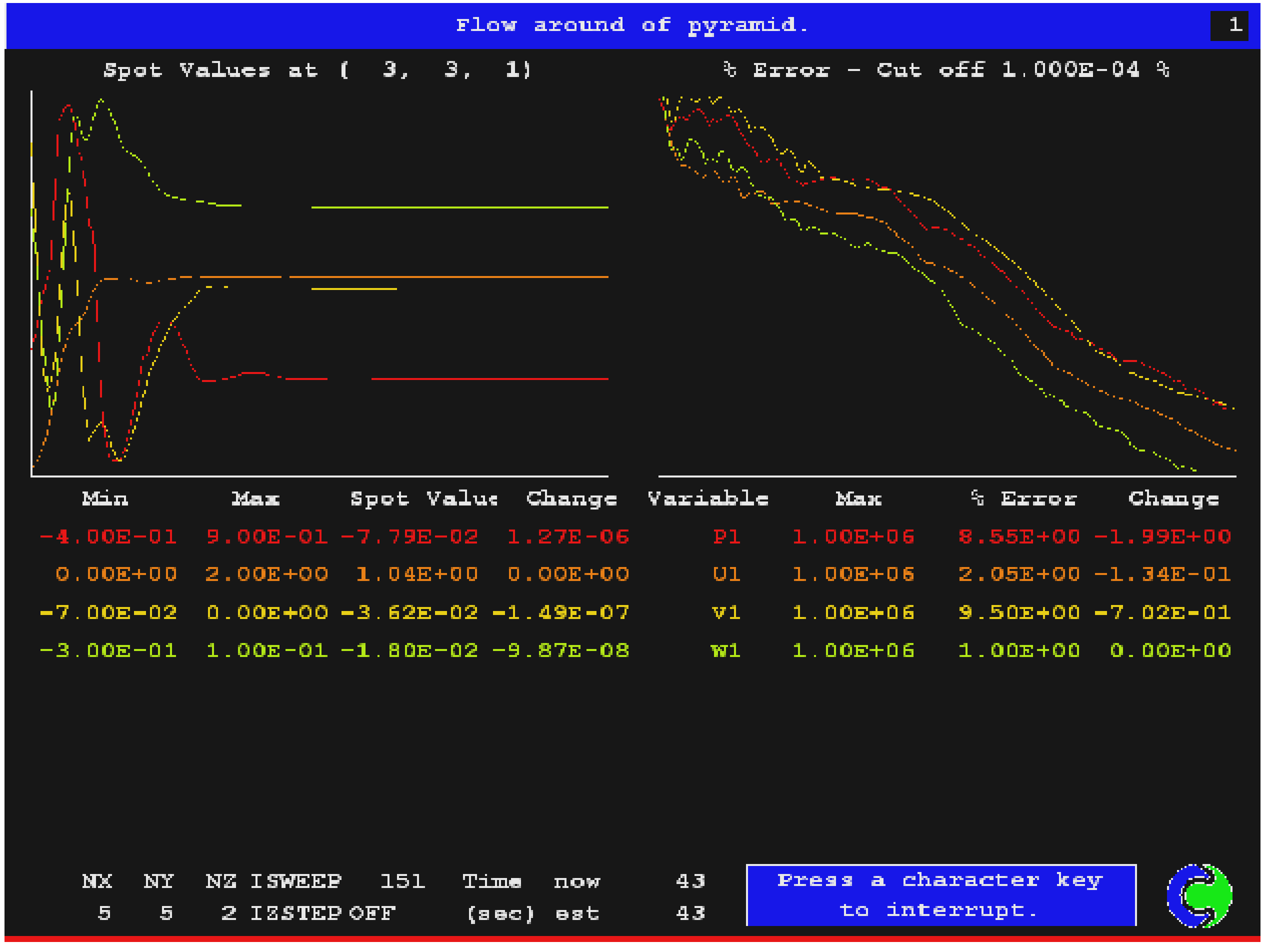



Y = 4 m

# Case 1 Convergence of USP

**Unstructured PHOENICS**
**June, 2009**

**PHOENICS User Meetings, 2009**




LSWEEP = 150, Elapsed time = **51** seconds

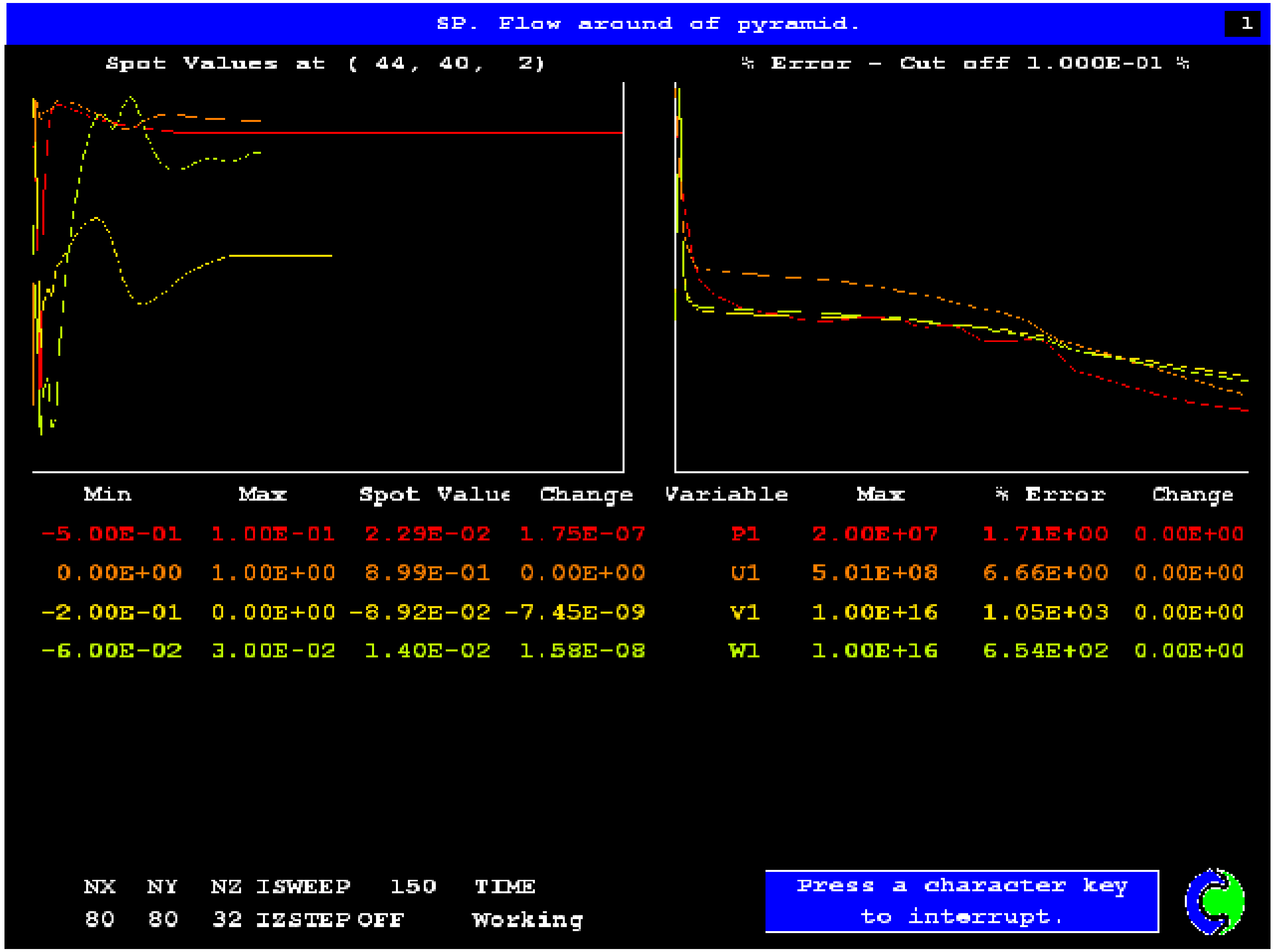
# Case 1 Convergence of SP





LSWEEP = 150, Elapsed time = **458** seconds;

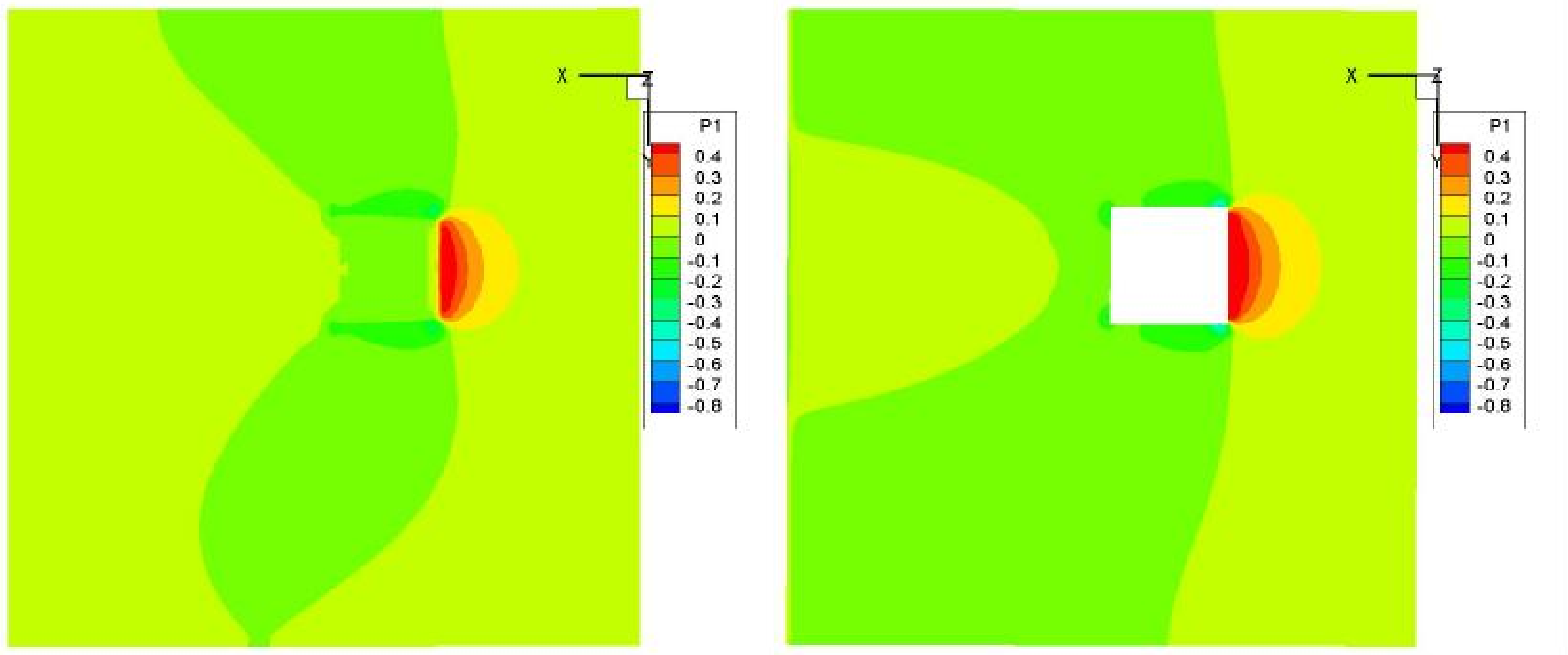
**So USP runs *8,98 times* faster than SP**



# Case 1 Comparison of outcomes.
# Pressure at Z = 0.

**Unstructured PHOENICS**
**June, 2009**

**PHOENICS User Meetings, 2009**



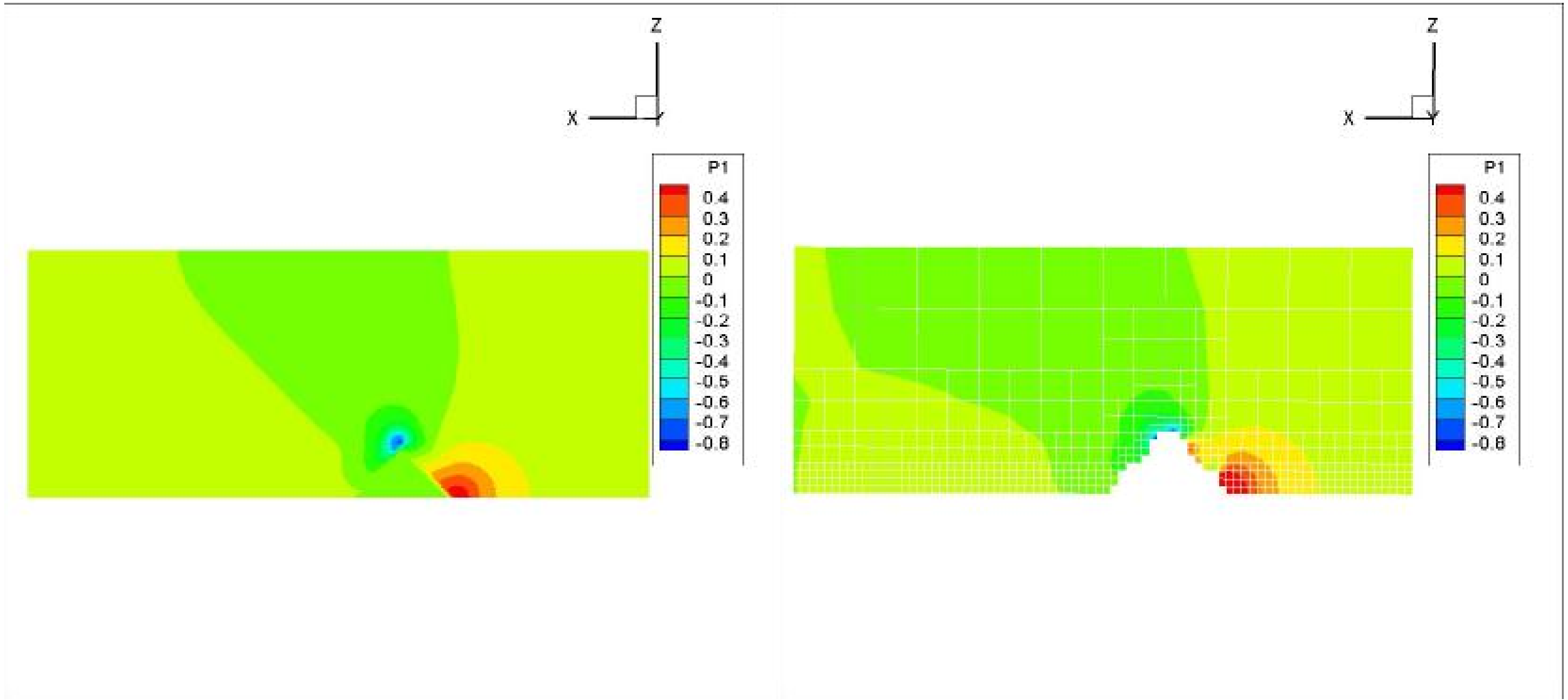

SP USP

The maximum pressures are the same

# Case 1 Comparison of outcomes.
# Pressure at Y = 4 m.

**Unstructured PHOENICS**
**June, 2009**





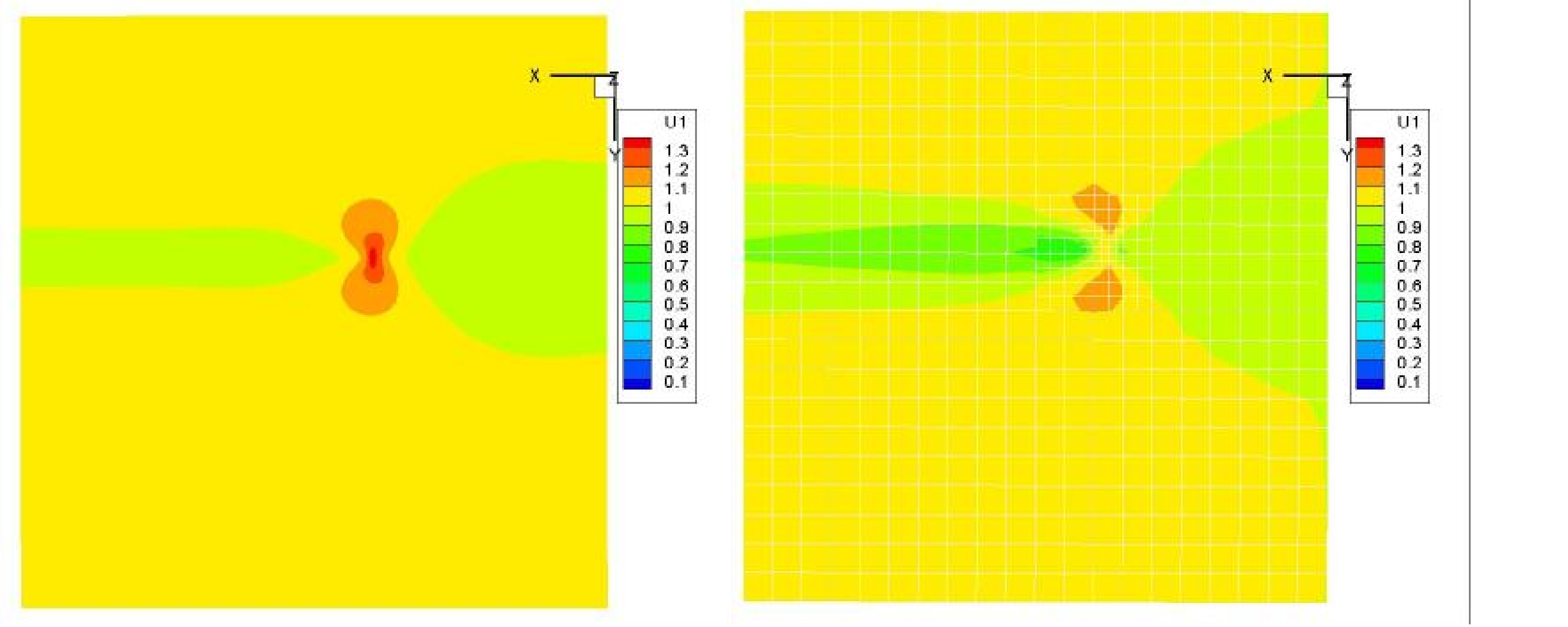

SP

USP

# Case 1 Comparison of outcomes.
# Velocity U1 at Z = 1 m

Unstructured PHOENICS
June, 2009

PHOENICS User Meetings, 2009




SP

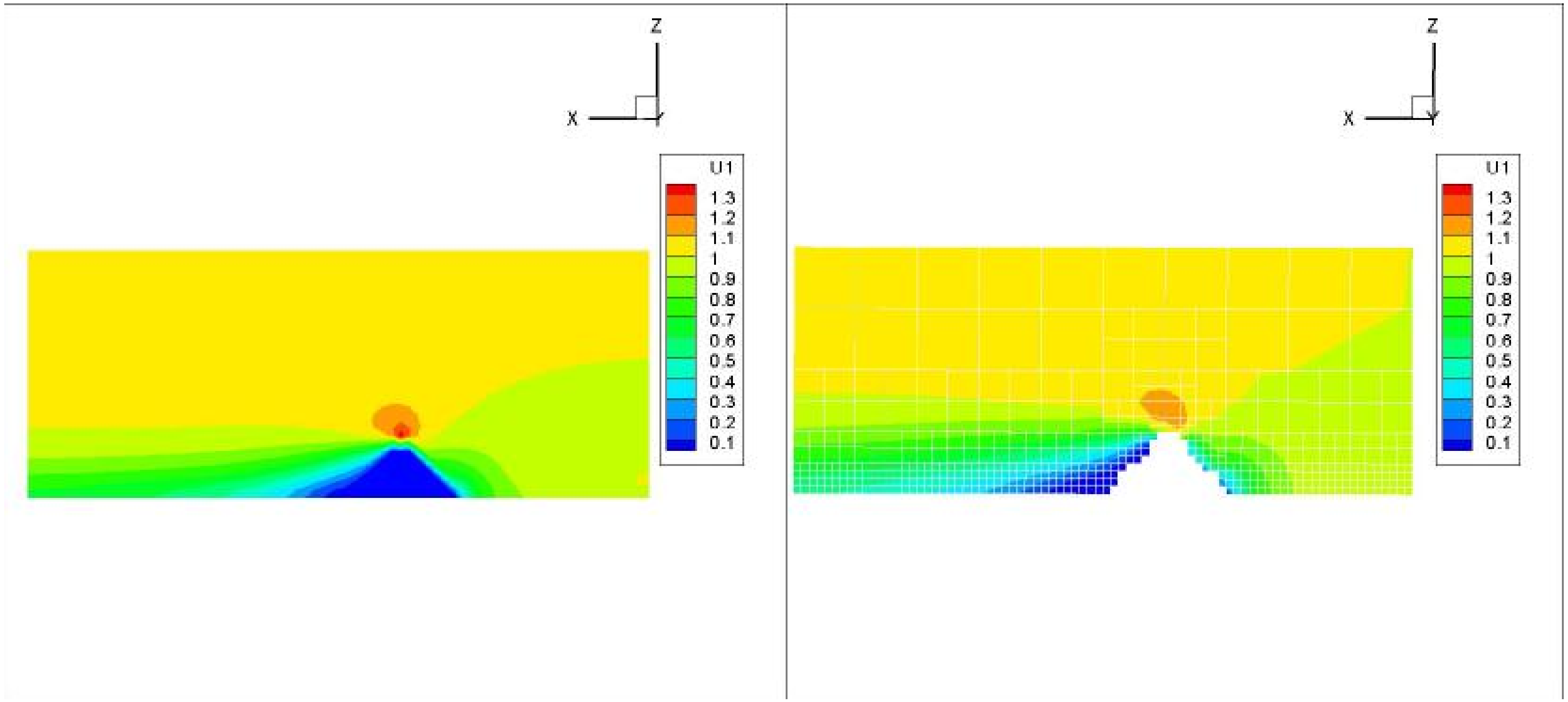
USP

Maxima and minima are the same

# Case 1 Comparison of outcomes.
# Velocity U1 at Y = 4 m








SP

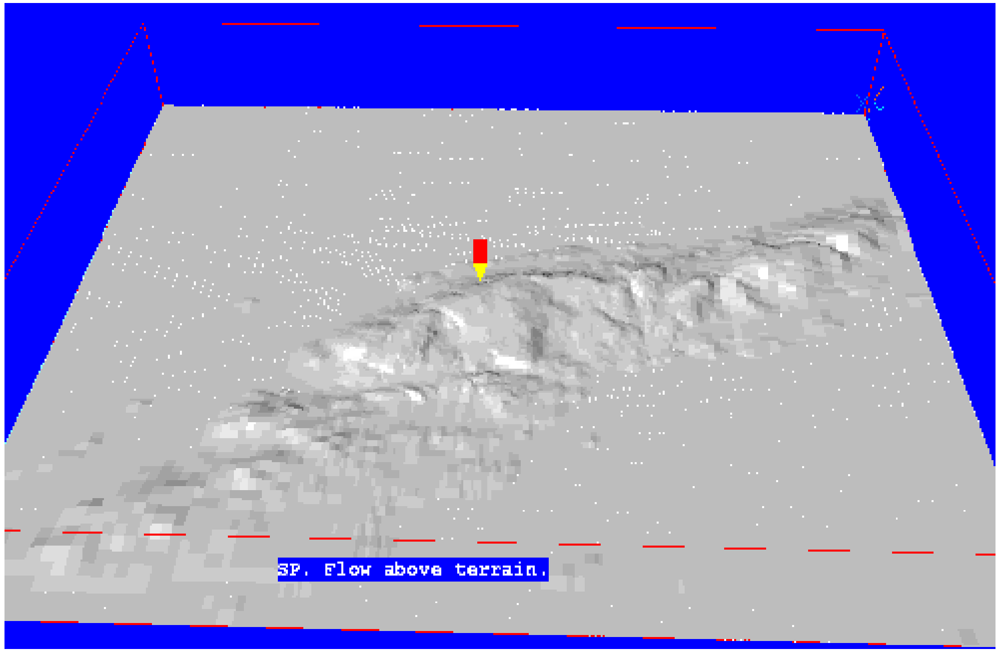
USP

## Case 2. Flow above natural terrain

Size of domain 9.0x7.5x1.2 km
Inlet Velocity 1 m/s
Effective viscosity 10 sq.m/sec

Structured grid is uniform with 144x120x24, *i.e.* 414,720 cells.
Unstructured grid has 77,382 cells *i.e.*19% of structured. 252,289 faces.
Refinement level = 3

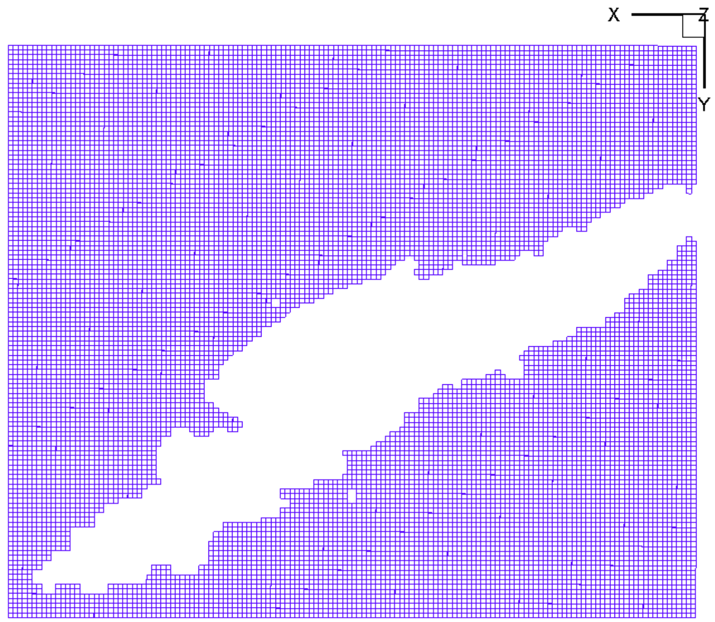
Sizes of smallest cells are the same for SP and USP.



# Case 2 Unstructured grid

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**




at Z = 0 m

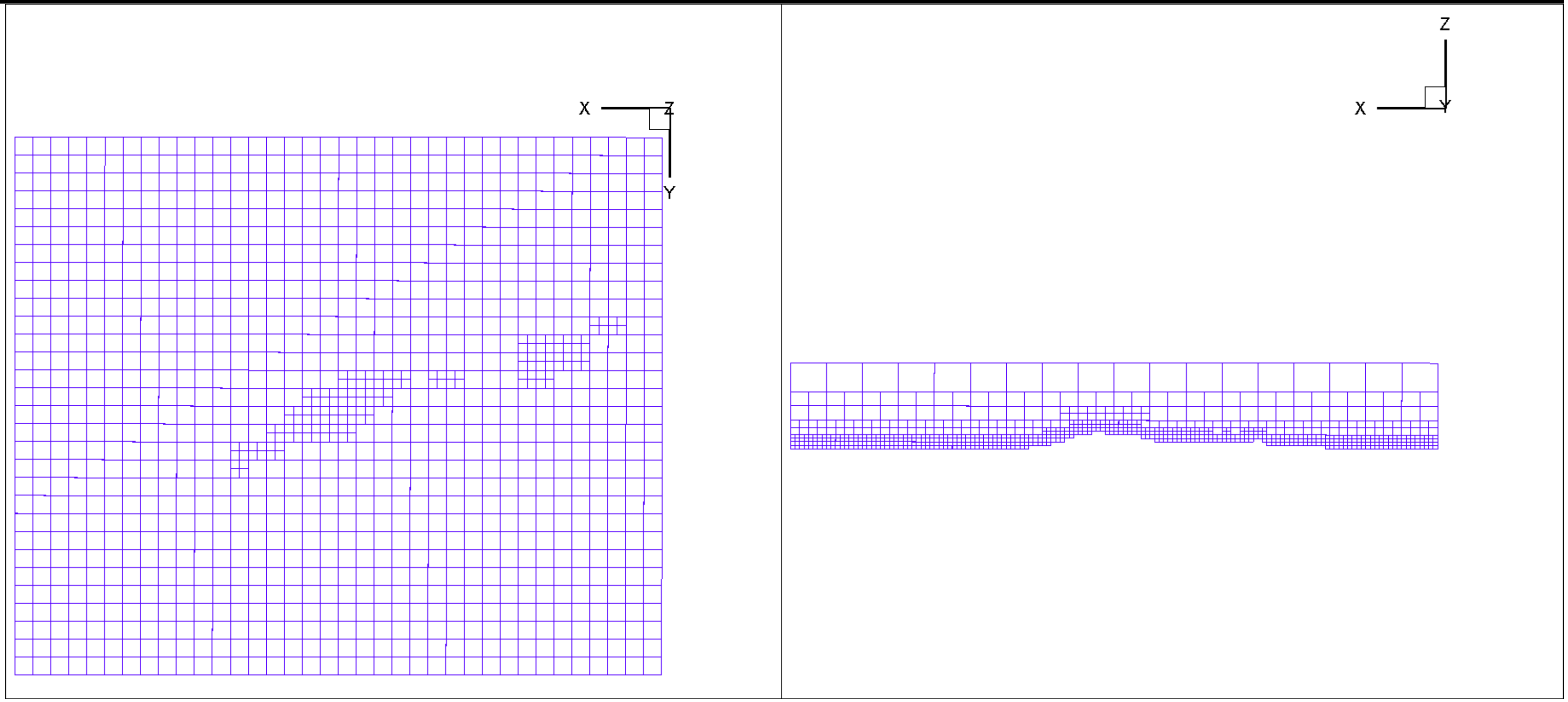

at Z = 200 m

Note that there are **no cells** beneath the ground surface

# Case 2 Unstructured grid

Unstructured
PHOENICS
June, 2009

PHOENICS User Meetings, 2009



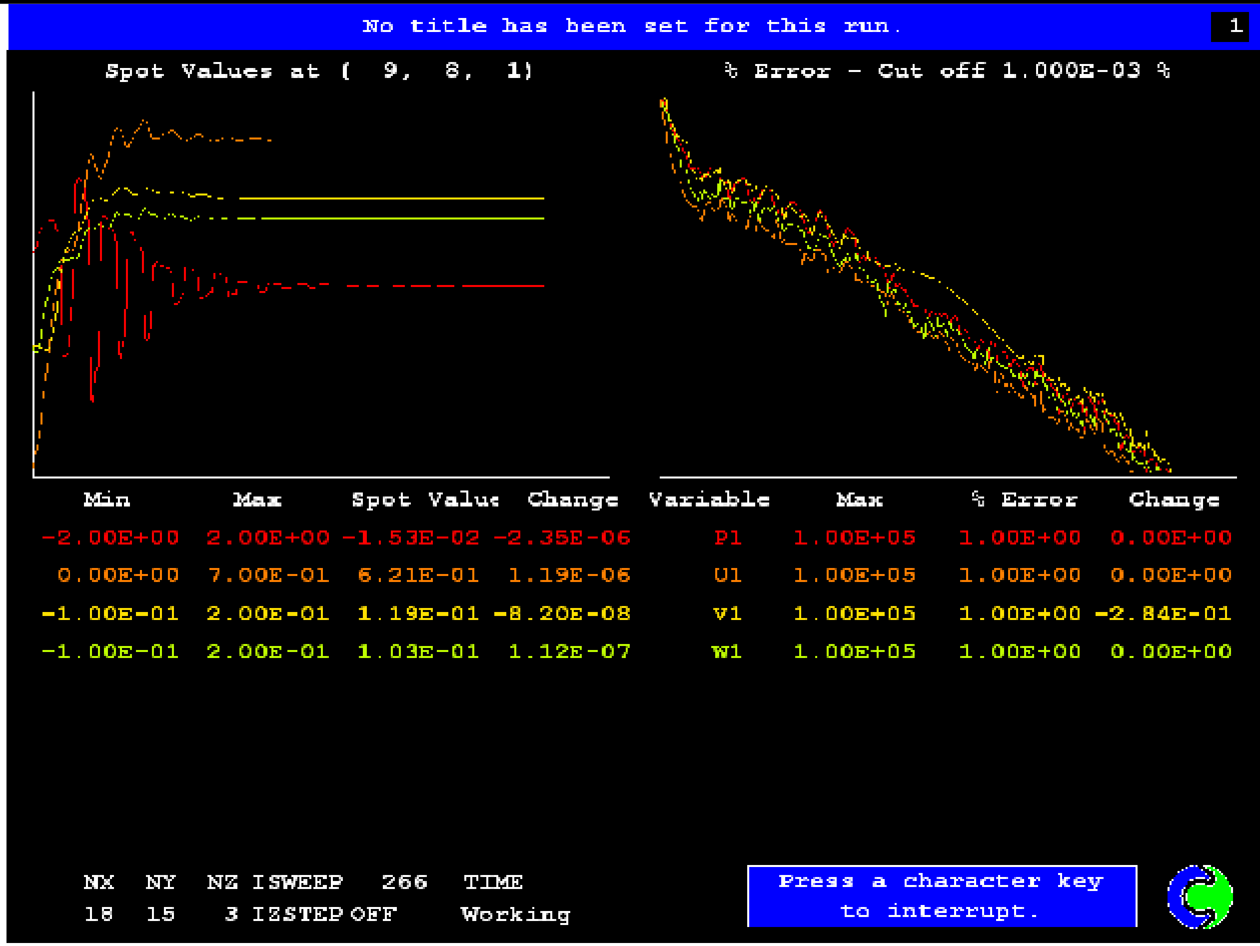

at Z = 400 m

at Y = 3800 m

The cells become larger with increased distance from the ground.

# Case 2 Convergence of USP



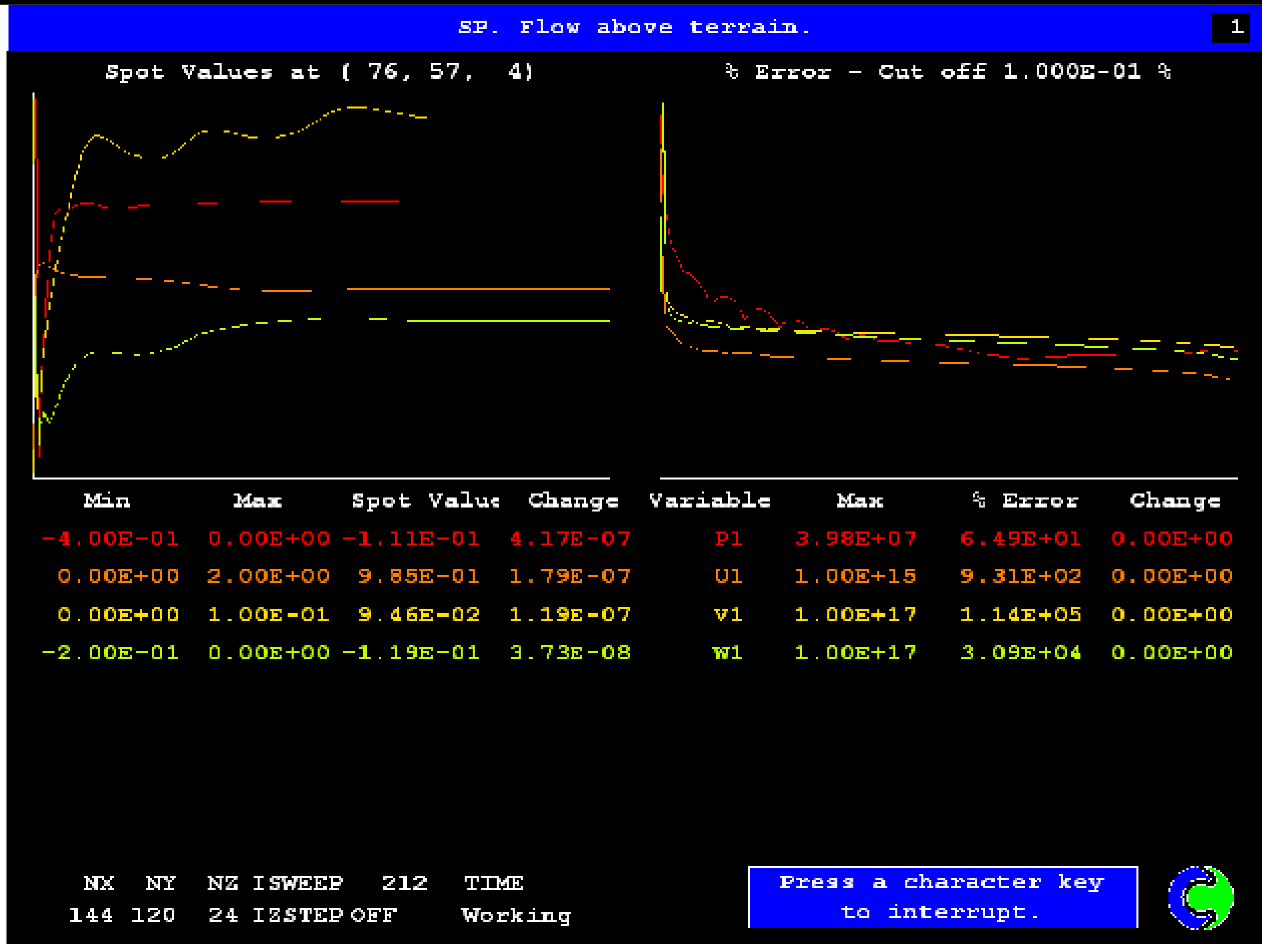



LSWEEP = 266, Elapsed time = **295** seconds

# Case 2 Convergence of SP

**Unstructured PHOENICS**
**June, 2009**

**PHOENICS User Meetings, 2009**



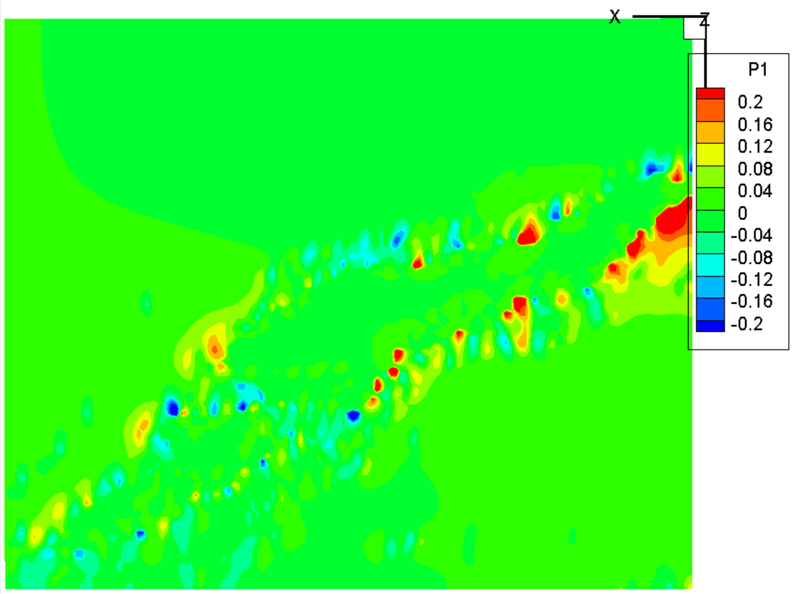

LSWEEP = 210, Elapsed time = **1792** seconds

USP faster by ***6.07 times*** even with more SWEEPs.

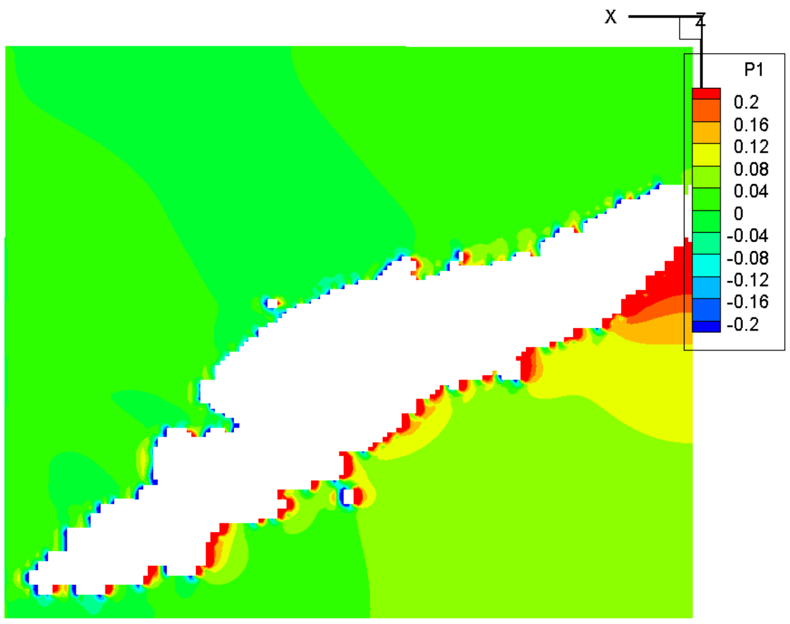
# Case 2 Comparison of outcomes. Pressure at Z = 0.








SP

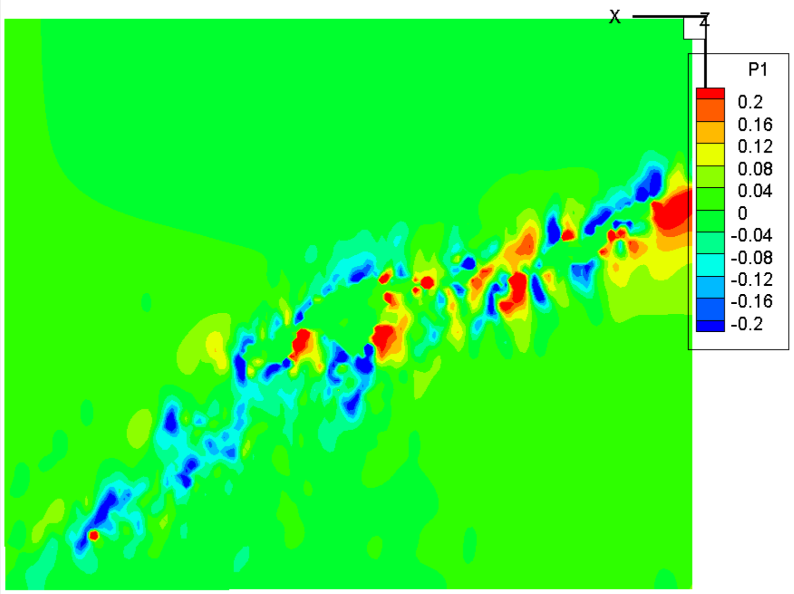



USP

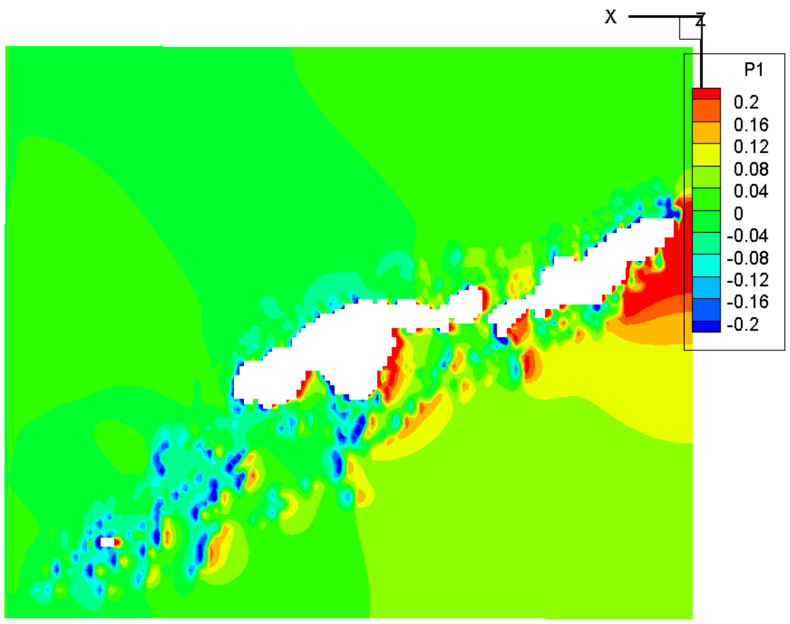
Case 2 Comparison of outcomes.
Pressure at Z = 100 m.








SP

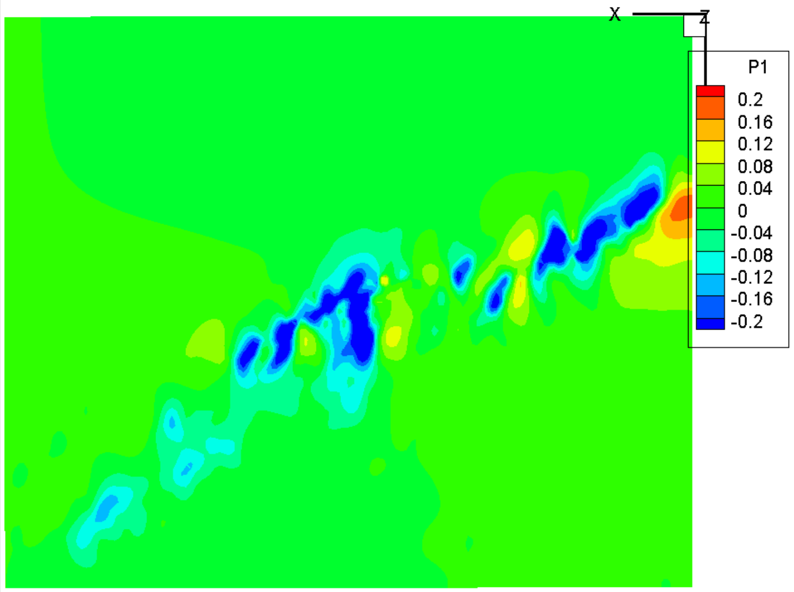



USP

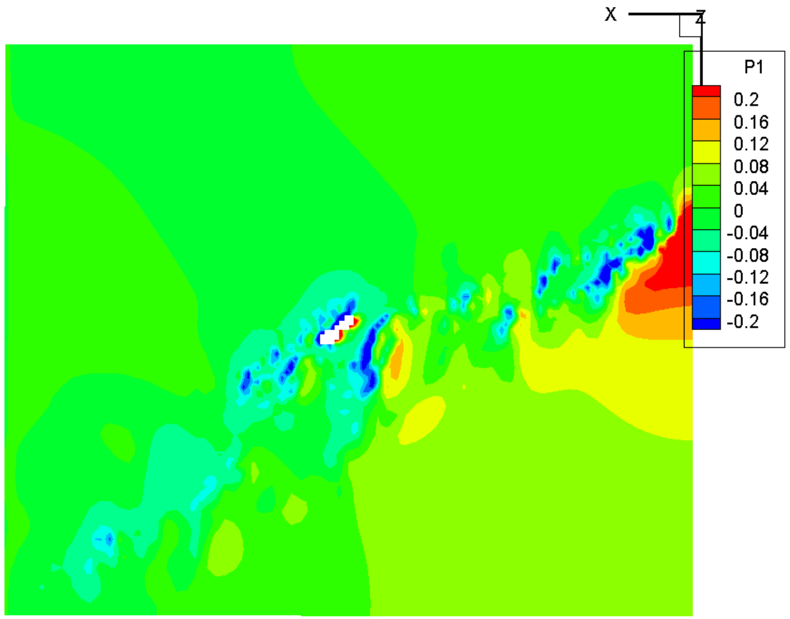
# Case 2 Comparison of outcomes.
# Pressure at Z = 200 m.








SP

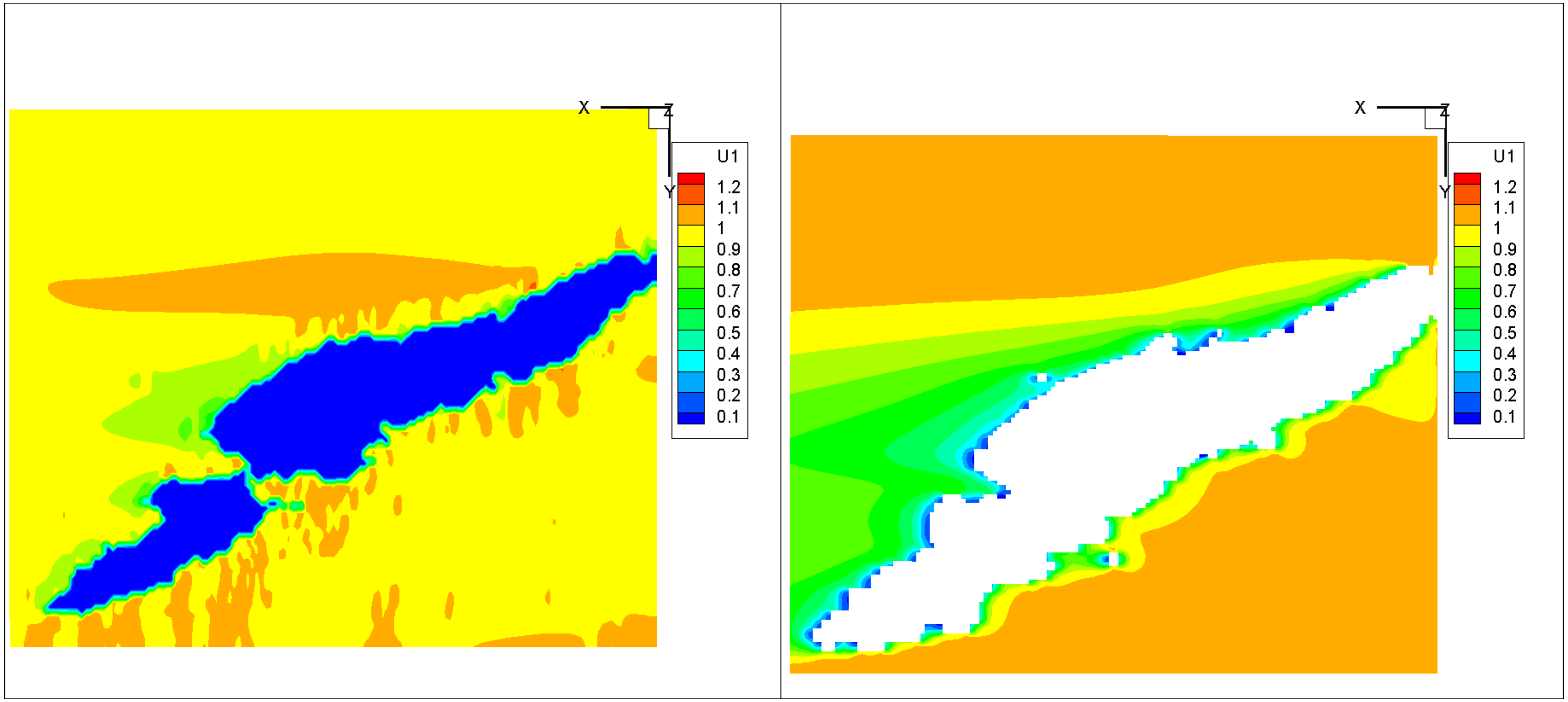



USP

# Case 2 Comparison of outcomes. Velocity U1 at Z = 0 m.

**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**



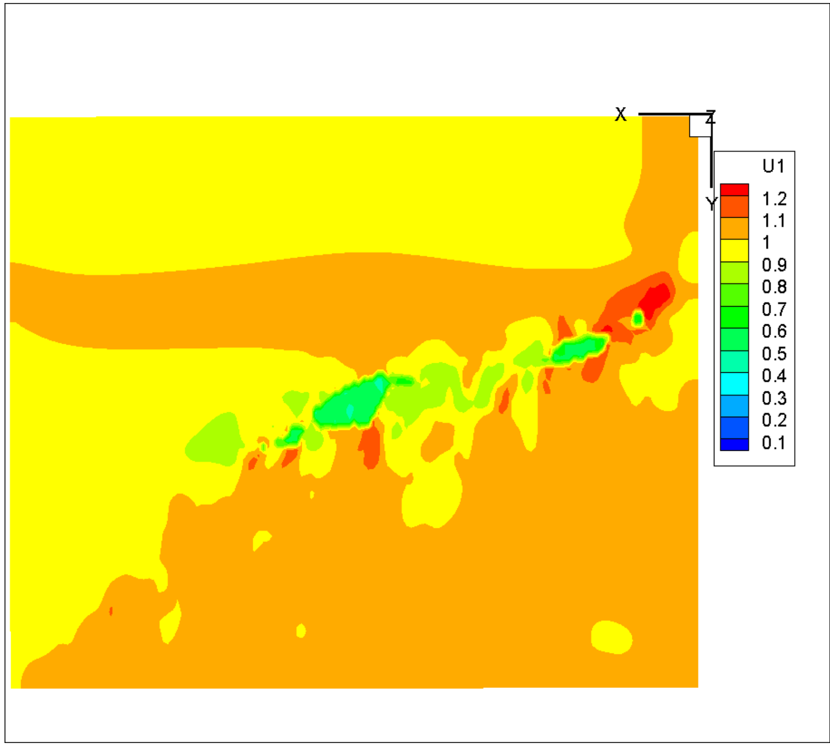

SP

USP

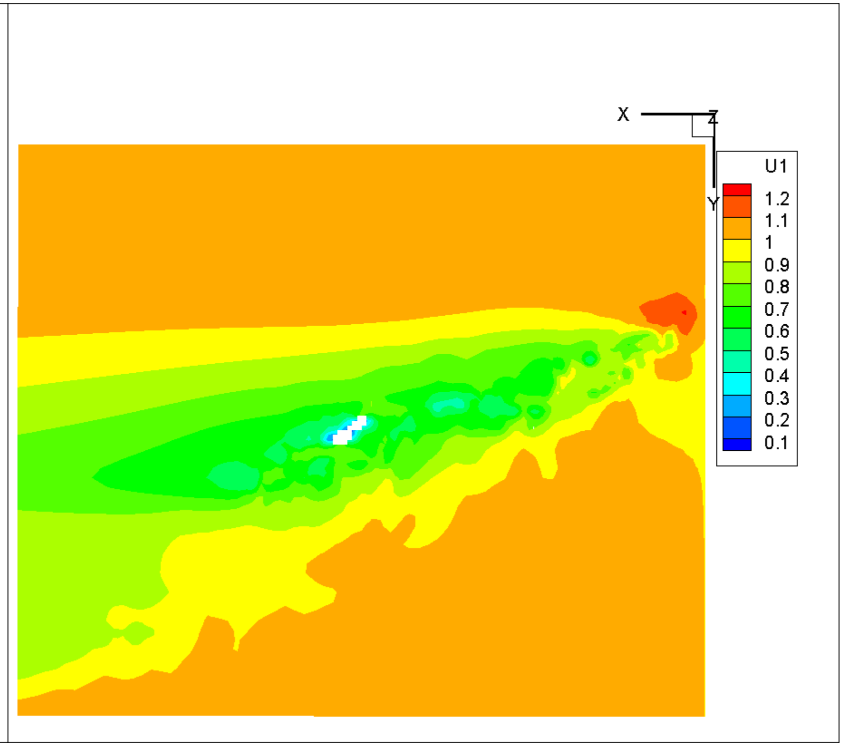
# Case 2 Comparison of outcomes.
# Velocity U1 at Z = 200 m.








SP

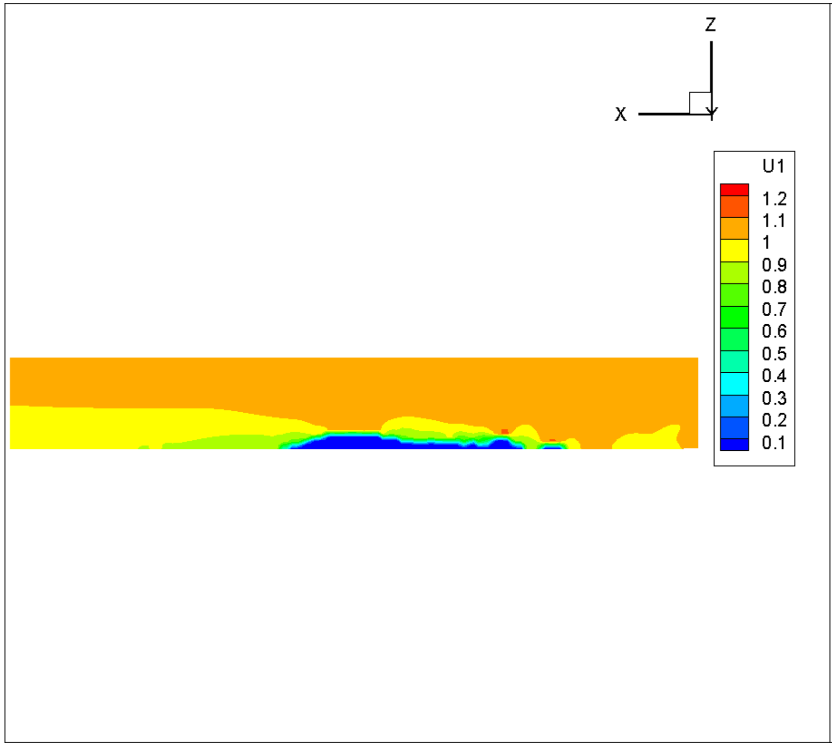


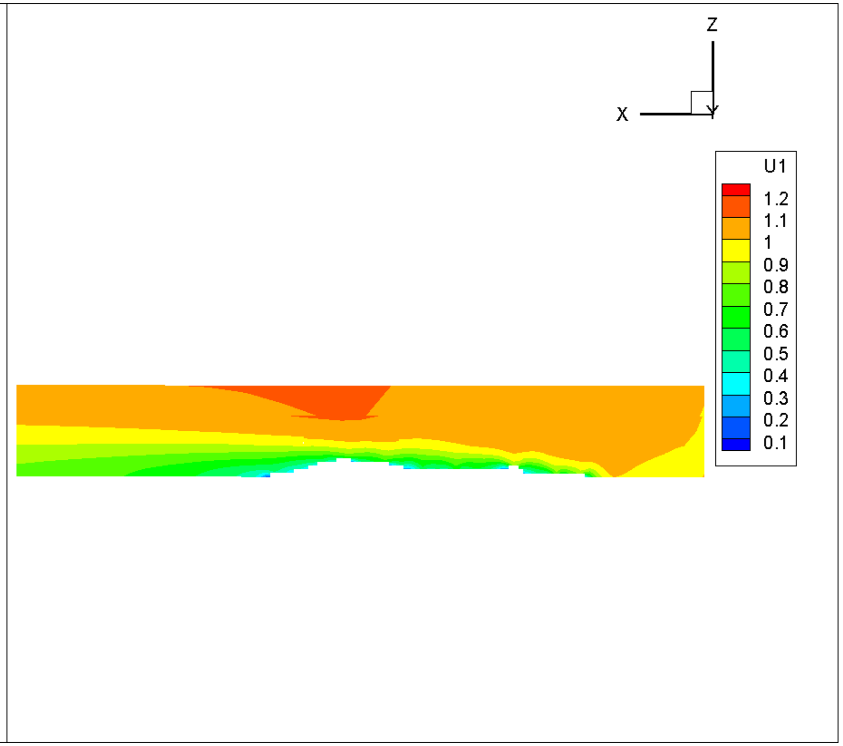

USP

Unstructured PHOENICS June, 2009

# Case 2 Comparison of outcomes. Velocity U1 at Y = 3800 m.






SP

USP

## Summary of conclusions

1. The expected reduction in computer times has been demonstrated.
2. The computed results of SP and USP agree in all important respects.
3. Much more testing is needed before the full benefits can be assessed.

Some unstructured-grid solutions:
stress & strain in long cylinder.



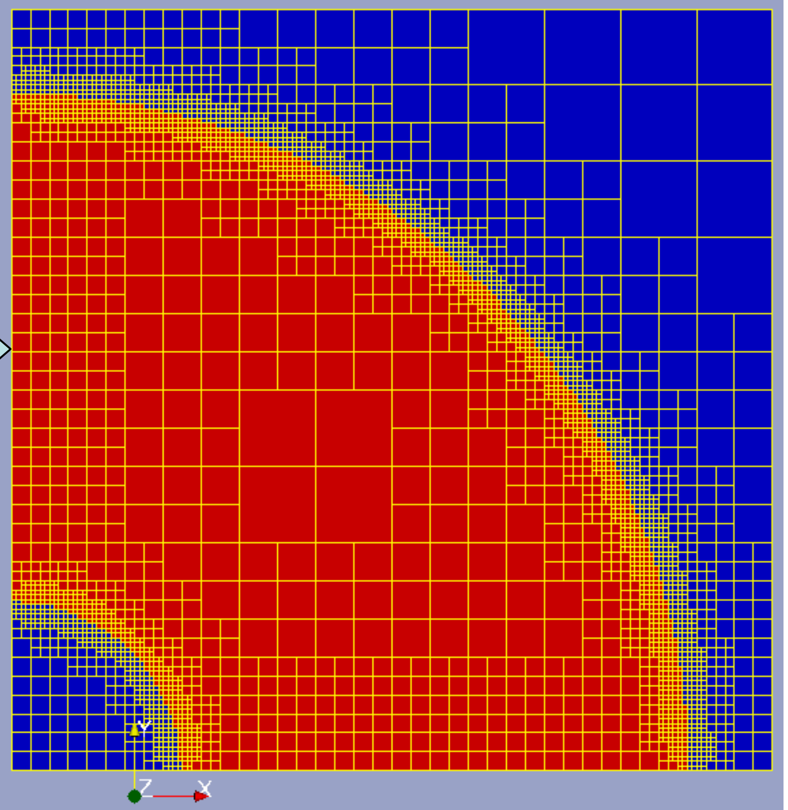


**The problem**:

A long, **hollow**, **thick-walled** cylinder, immersed in an outer fluid, contains a second fluid having a **different pressure**.

The picture on the right shows the so-called '**unstructured**' **grid** used for its solution.

The **smallest** cells are placed near the **boundaries** of the cylinder, so as to represent their **curved** shapes.



# The unstructured-grid solution for the pressurised long cylinder.





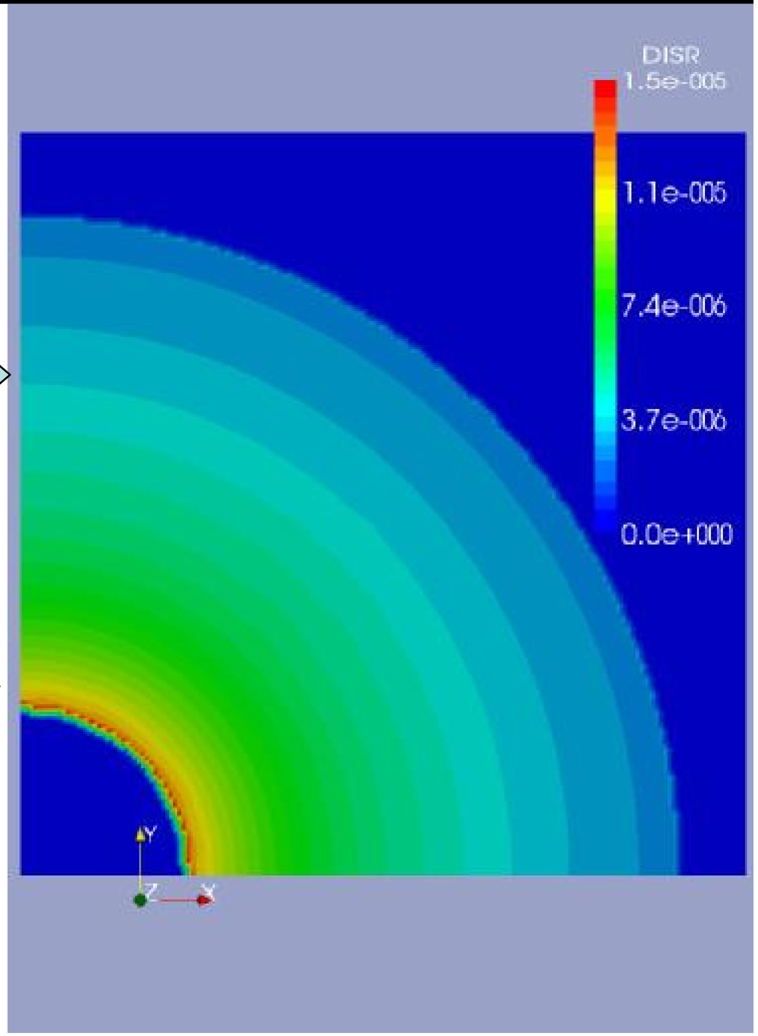
On the right are shown contours of the **displacement** of the material.

The **highest** are **red**, the **smallest blue**; so, understandably, the displacements are largest at the centre, where the **pressure-gradient** is highest.

The contours are **perfectly circular** in shape, despite the fact that the grid is basically a **cartesian** one.

But are the values to which they correspond **correct**?
Because there is an exact analytical solution for this problem, the question can be answered by **comparison**.

The **next slide** shows the evidence.



# Comparison of the numerical with the analytical solution.

Unstructured PHOENICS
June, 2009

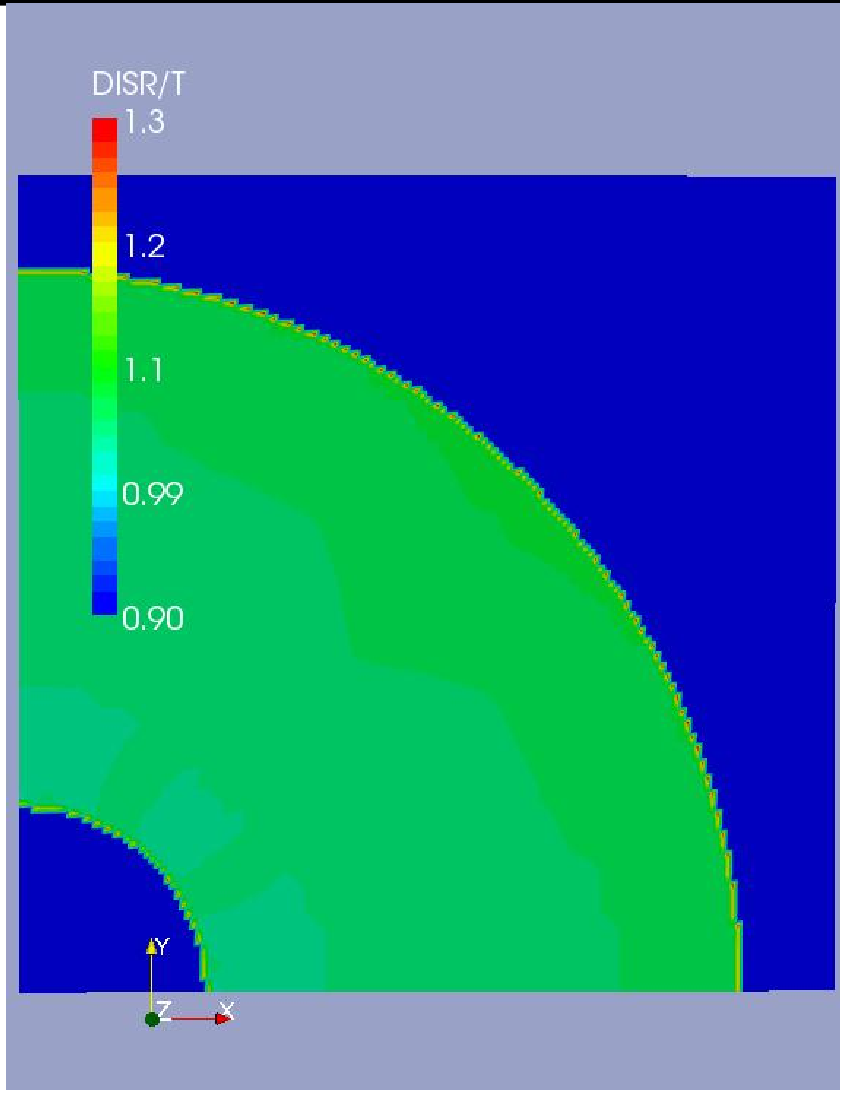
PHOENICS User Meetings, 2009

The contours shown here are of the **ratio** of **numerically-computed** displacement to the **analytically-derived** displacement.

This should equal precisely 1.0 everywhere.

The scale of contours is from 0.9 (blue) to 1.3 (red).

The nearly-uniform bluish-green of the contours in the cylinder shows that the numerically obtained values **agree with** the analytical ones **very well**.



# The SBC (Smoothing Boundary Cell) algorithm, not yet incorporated into AGG



## The basic Ideas of SBC

- All cells having at least **one** edge intersecting a VR-object surface are marked as CutCells.
- Vertices of CutCells are moved to their nearest intersection points.
- No vertex may be moved **more than once**.

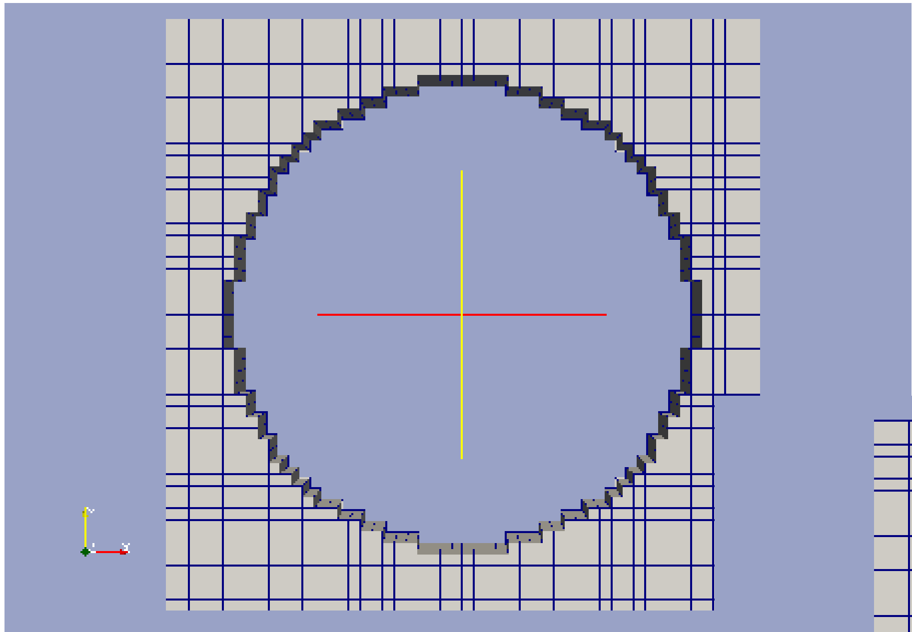
- The vertex-moving algorithm is as follows:
  1) First search for and move vertices of "GOOD" cells which have **exactly four** intersections on edges parallel to X,Y or Z.
  2) Move vertices of **not** "GOOD" CutCells in X,Y,Z direction along **edges** of cells.
  3) **Remove** "BAD" cells of which **all neighbors** are either CutCells or have PRPS=198.

- **Important feature**: CutCells always have **hexahedral form !**

# AGG SBC algorithm
# Example #1: 2D cylinder

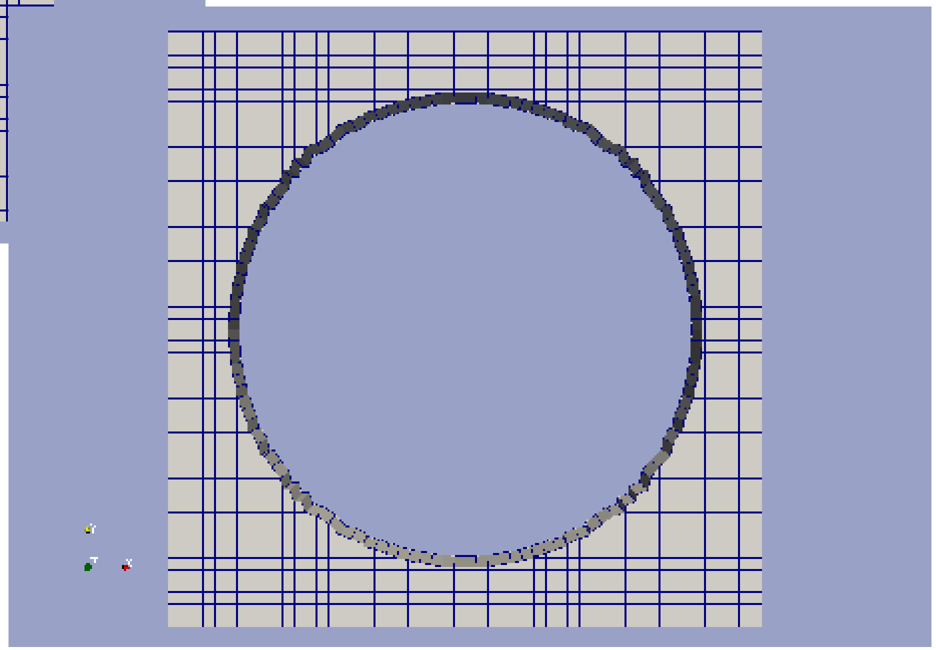






ß Whole Cells algorithm

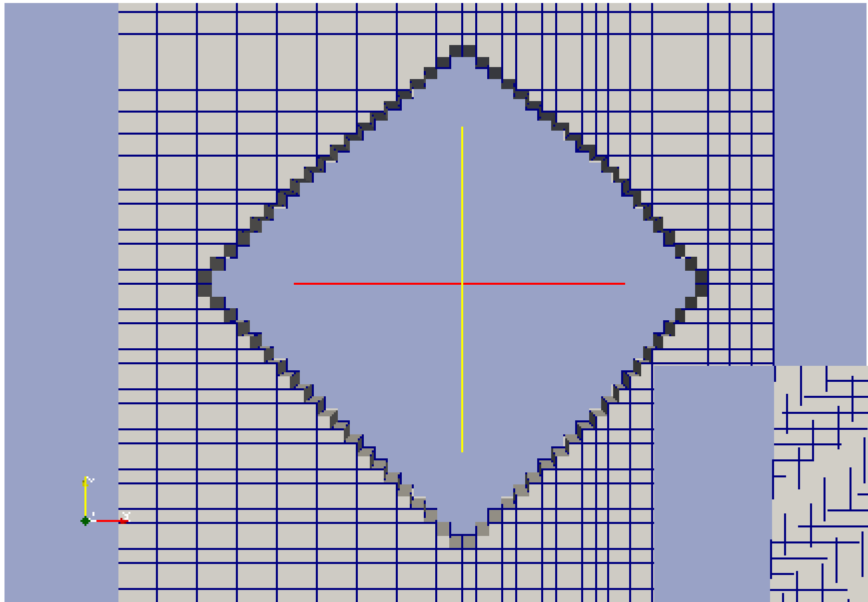
SBC algorithm à



# AGG SBC algorithm
Example #2: 2D rectangle

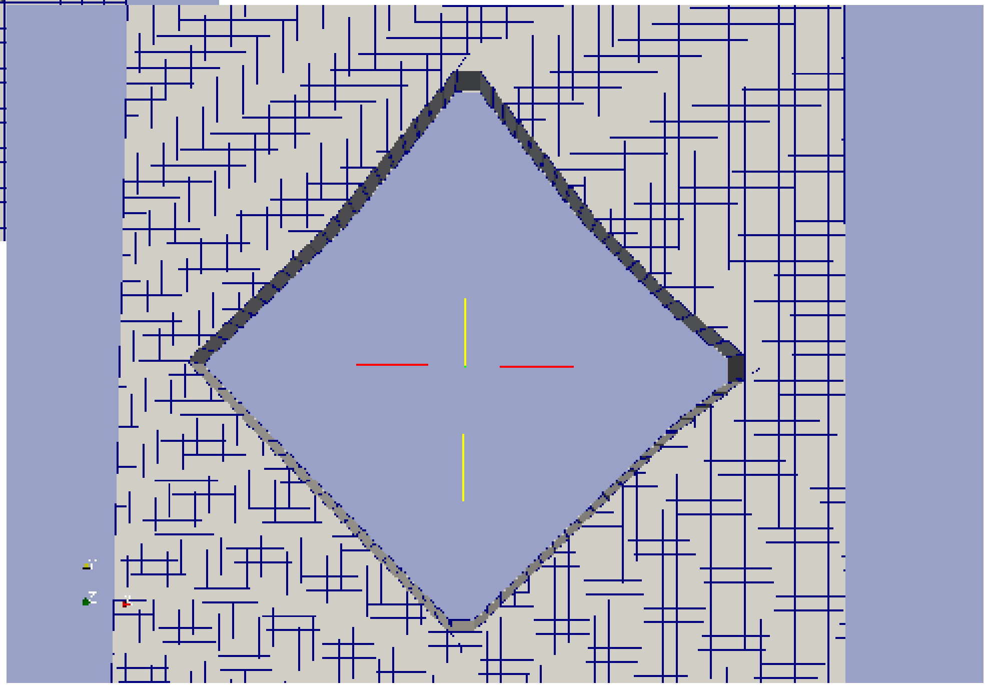






ß Whole Cells algorithm

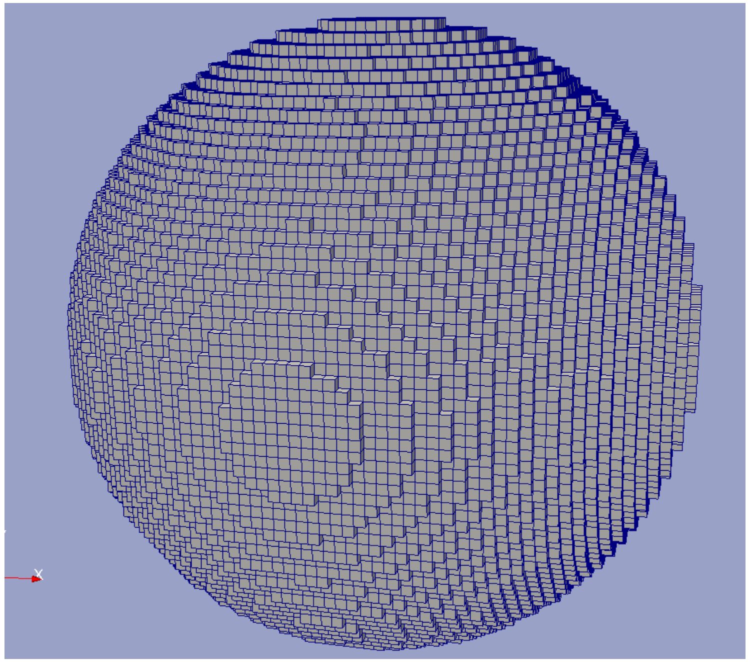
SBC algorithm à



# AGG SBC algorithm
# Example #3: 3D sphere

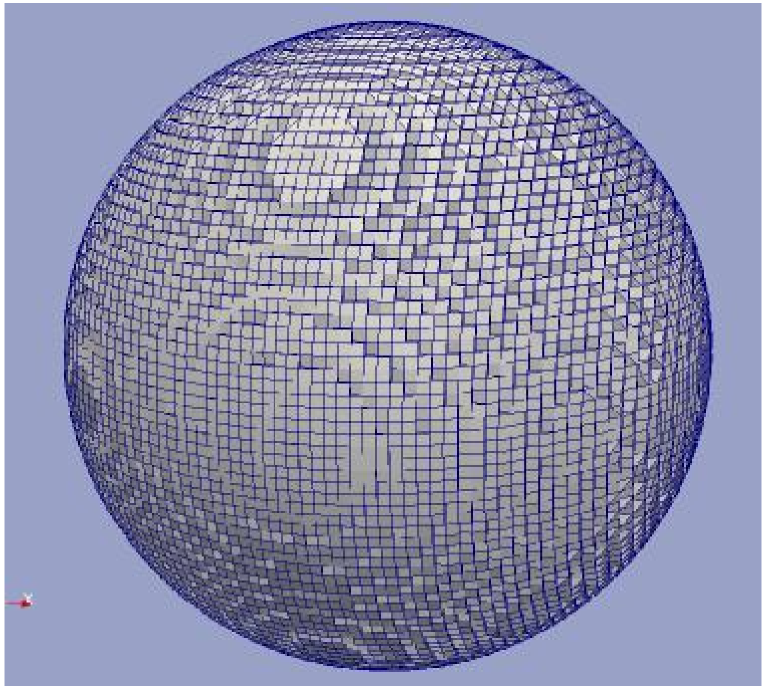






ß Whole Cells algorithm

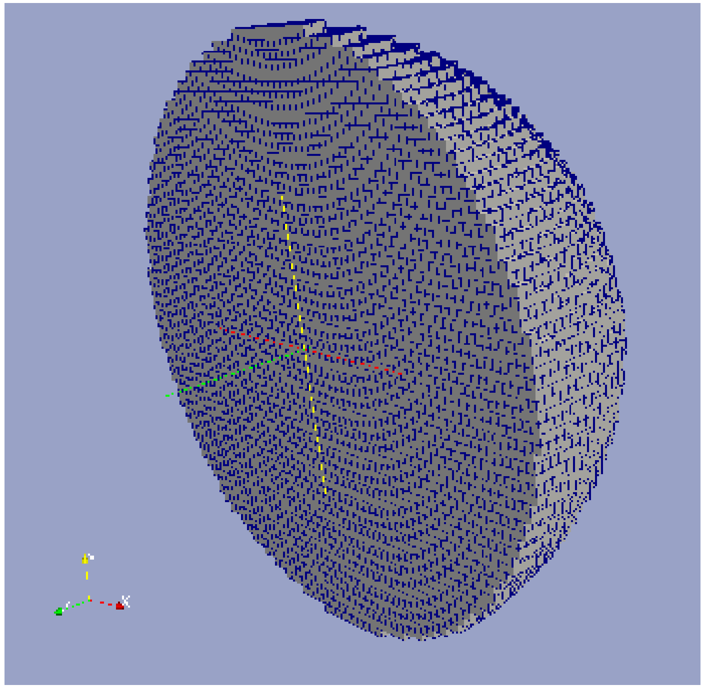
SBC algorithm à



# AGG SBC algorithm
# Example #3: 3D sphere

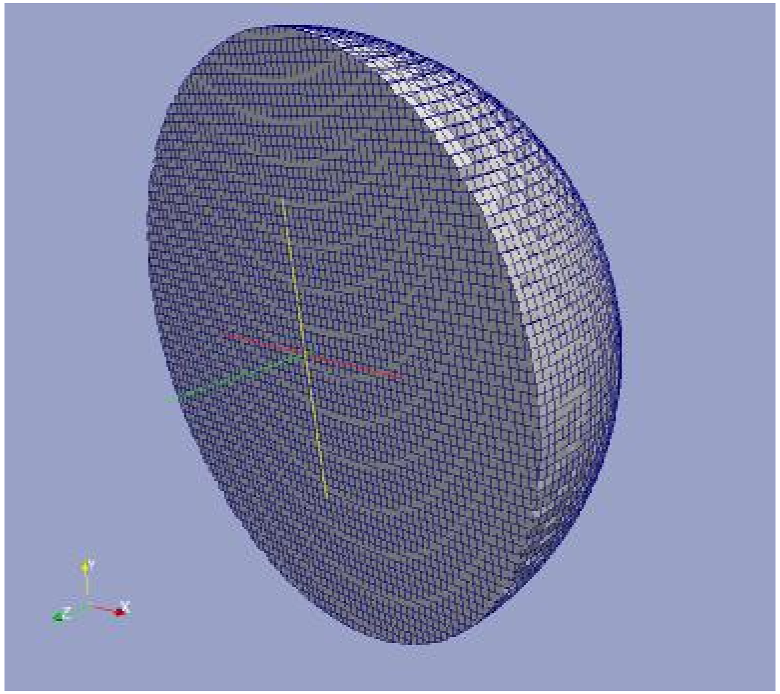
**Unstructured PHOENICS June, 2009**





ß Whole Cells algorithm

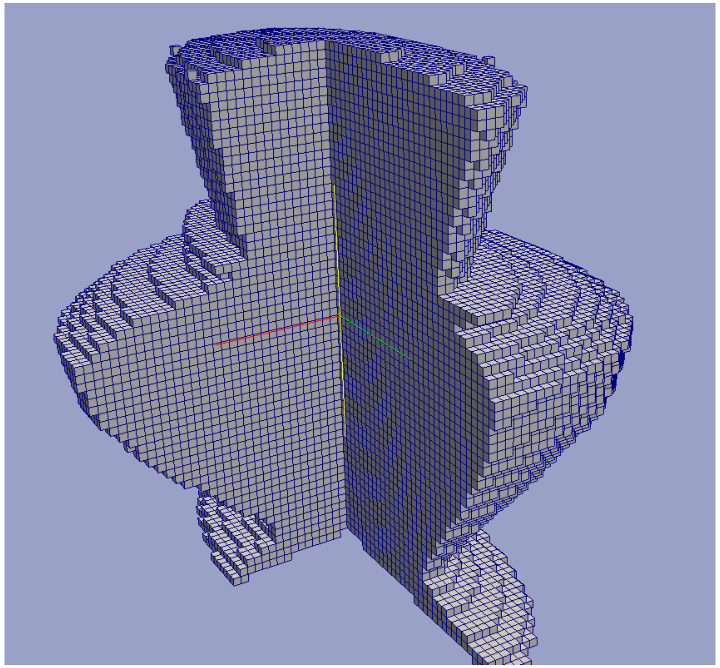
SBC algorithm à



# AGG SBC algorithm
# Example #3: 3D bottle

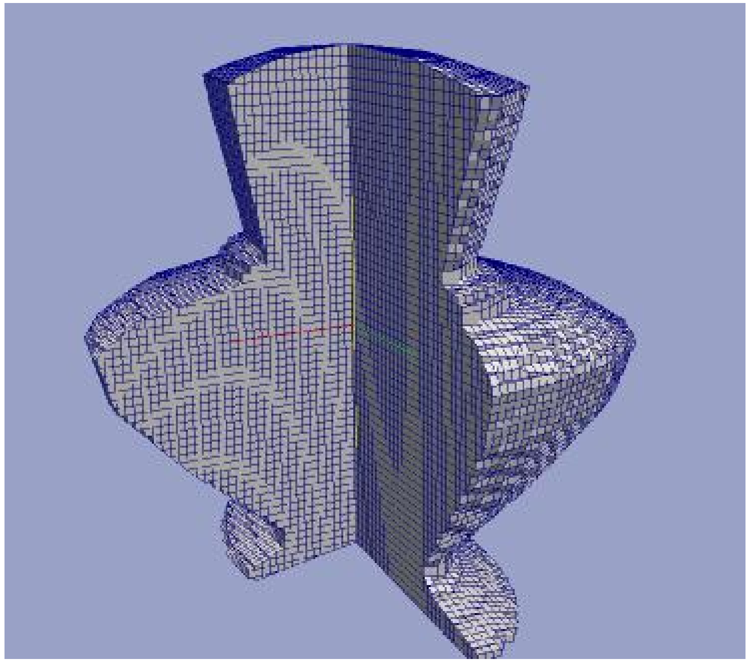
**Unstructured PHOENICS June, 2009**

**PHOENICS User Meetings, 2009**



ß Whole Cells algorithm

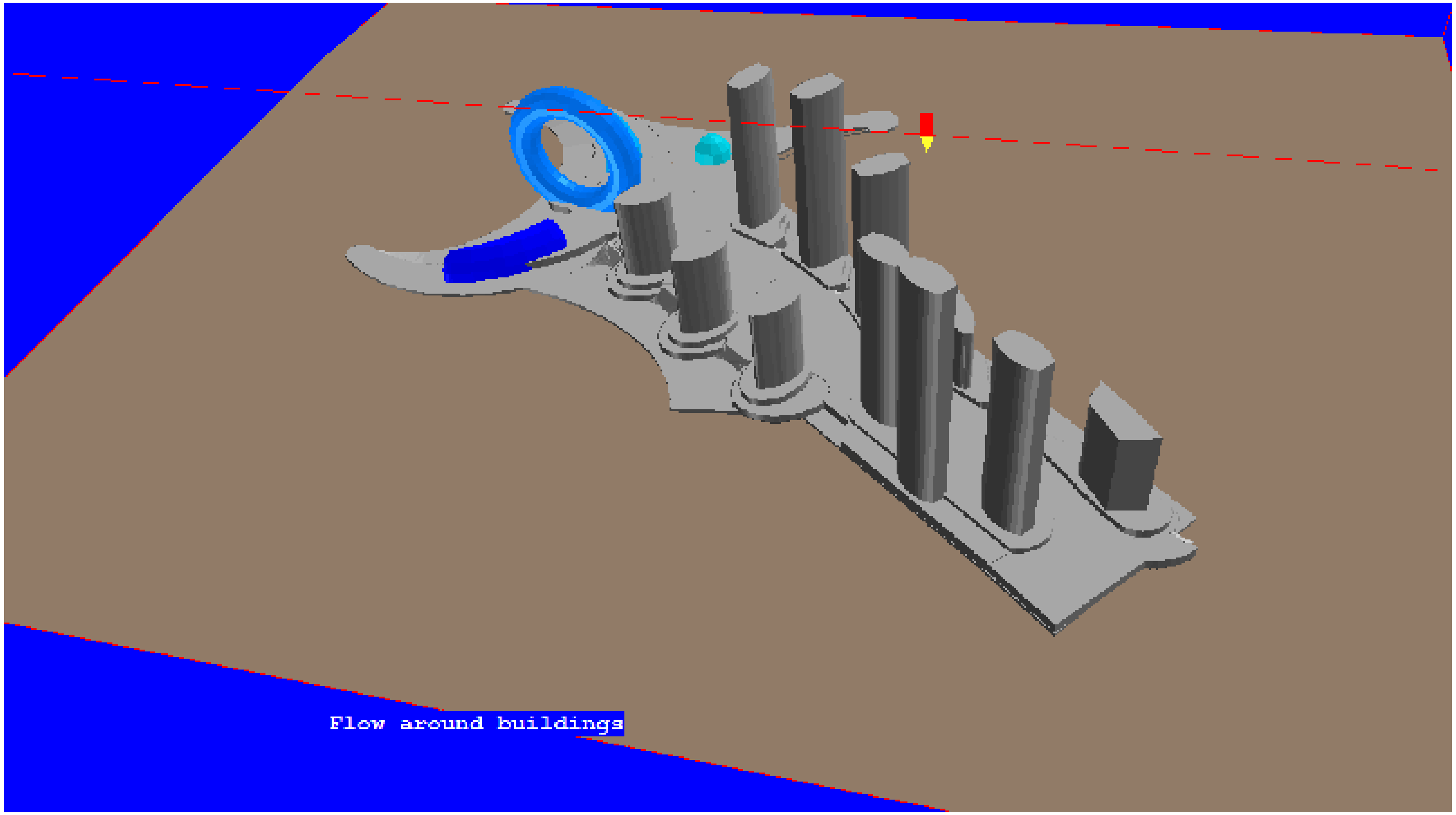
SBC algorithm à



# Finally:
# a glimpse of the future







and of how AGG will handle it.

Unstructured PHOENICS June, 2009

PHOENICS User Meetings, 2009

## Boundary Faces