

Compressibility of the surface currents created by nonlinear waves

Last reviewed: 6/7/2011 14:36

- Compressibility of a velocity field is a feature which influences the transport and mixing of various substances by fluid motion.
- When a flow is compressible, tracer particles tend to gather into patches
- Typically, compressibility in 3D Eulerian velocity fields is incapable of producing any patchiness
- If we consider only the surface 2D layer of a 3D field, things might be different
- Particles can actually “dive” into the third dimension
- This might rise the compressibility of the 2D surface flow, while the global 3D flow would still remain incompressible
- This phenomenon if verified, is expected to be the main cause of floaters patchiness in marine environment



First approach

Use velocity fields used by tracmass

- Velocity fields stored inside .snap1 files
- Encoded using Big Endian convention
- Only one existing FORTRAN program able to read them
- Not very well commented
- No clue about how the actual velocity field is stored (data structure)

```
#ifdef tempsalt
! the density
do i=1,imt
  do j=1,jmt
    if(kmt(i,j).ne.0) then
      kmm=kmt(i,j)
      do k=1,kmm
        kk=km+1-k
        tempb(k)=tem(i,j,kk,2)
!       saltb(k)=(sal(i,j,kk,2)-35.)/1000.
        saltb(k)=sal(i,j,kk,2)
        if(saltb(k).lt.0.) saltb(k)=0.
      enddo
      call statv(tempb,saltb,rhob,kmm)
      do k=1,kmm
        kk=km+1-k
        rho(i,j,kk,2)=rhob(k)
      enddo
    endif
  enddo
enddo
#endif
```

```
#endif
enddo
```



Second approach

26C010 9bb109cu

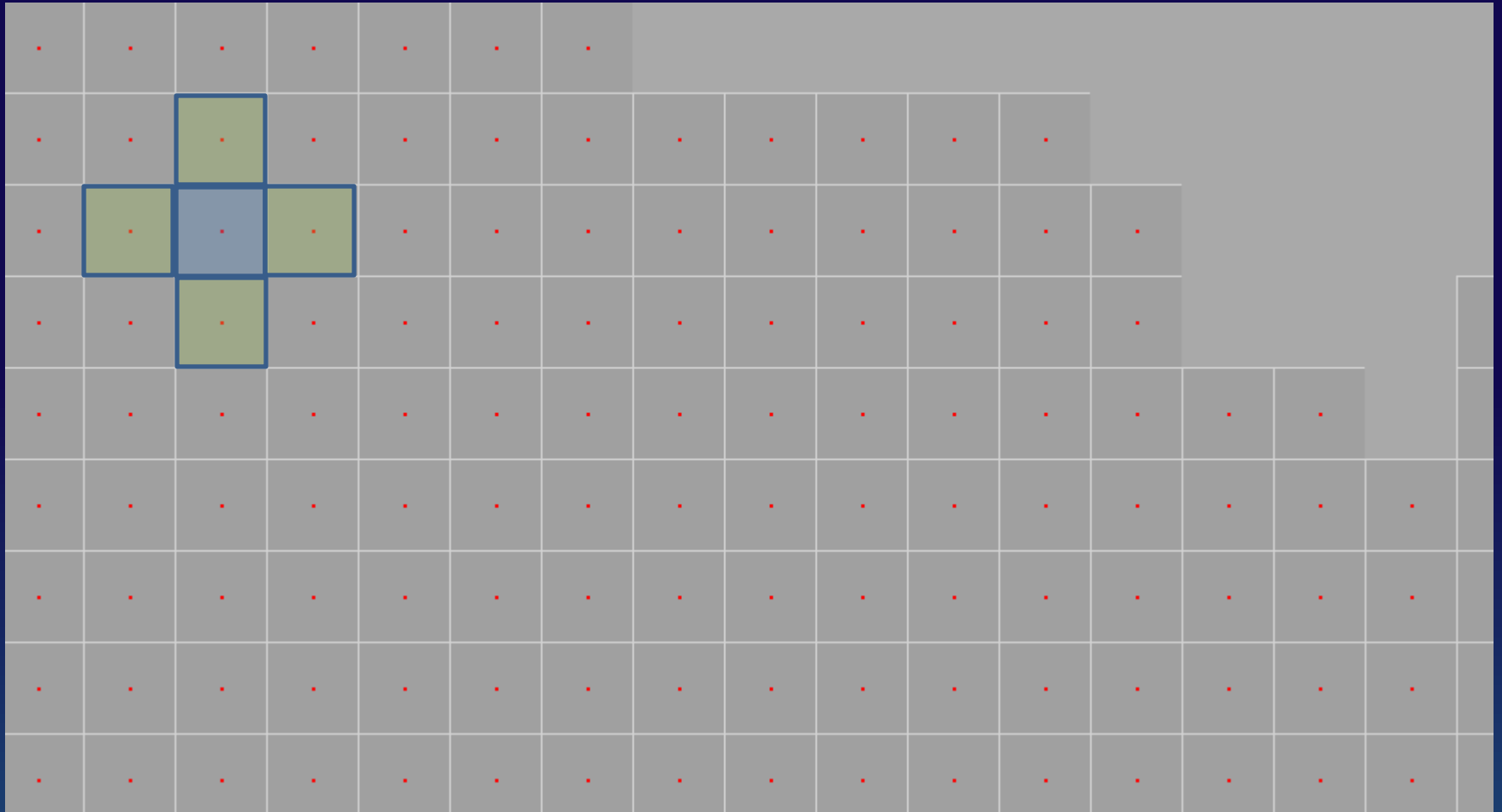
Use trajectories computed by TRACMASS

- TRACMASS knows how to deal with those .snap1 files
- Produces trajectories that are, in fact, the direct effect of the same vector fields
- Trajectories are stored in nicely formatted ASCII tables
- A lot of data of this type is already available
- Simulation runs use good naming conventions

Use trajectories computed by TRACMASS

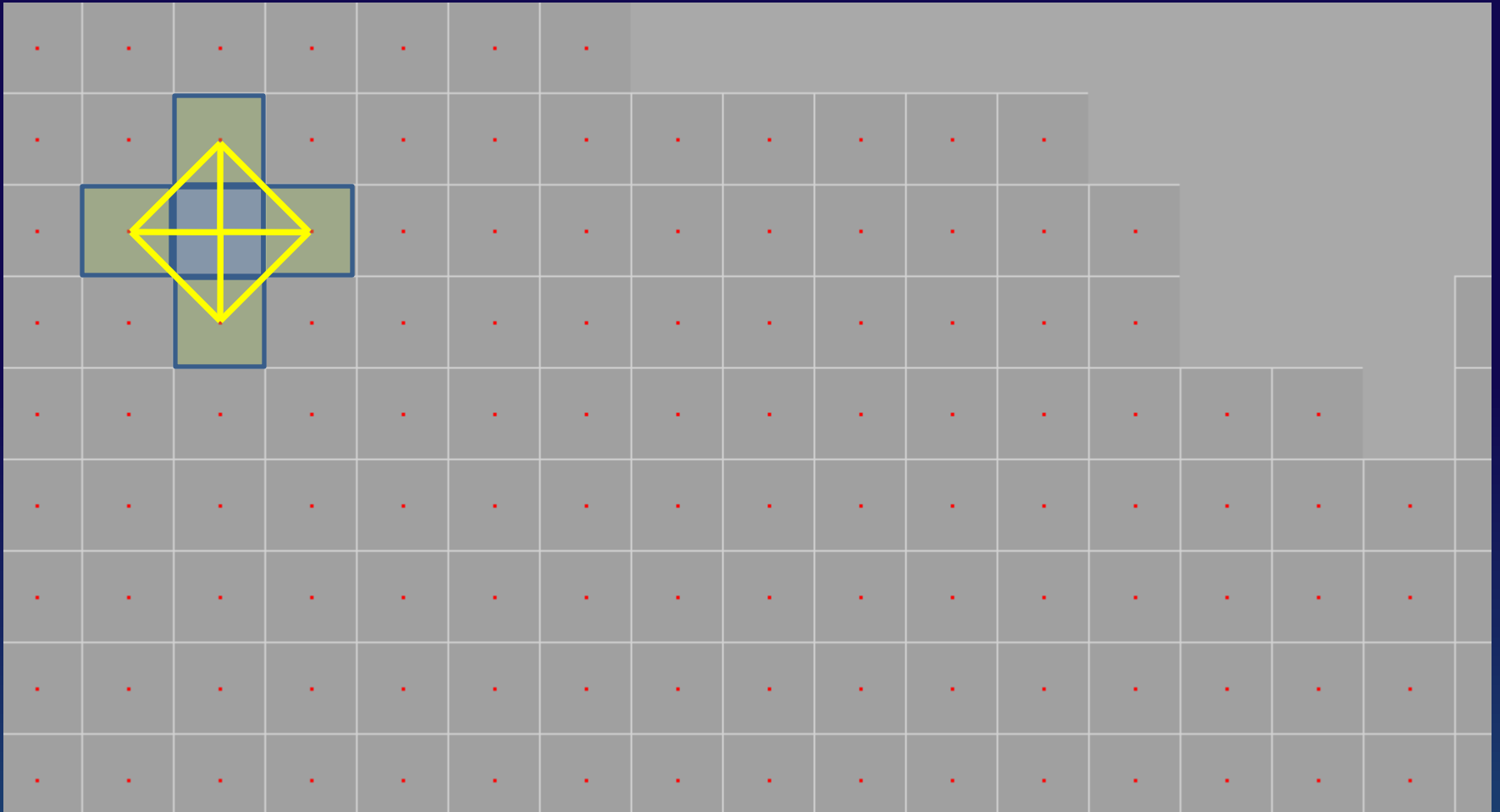


How do we use these trajectories?



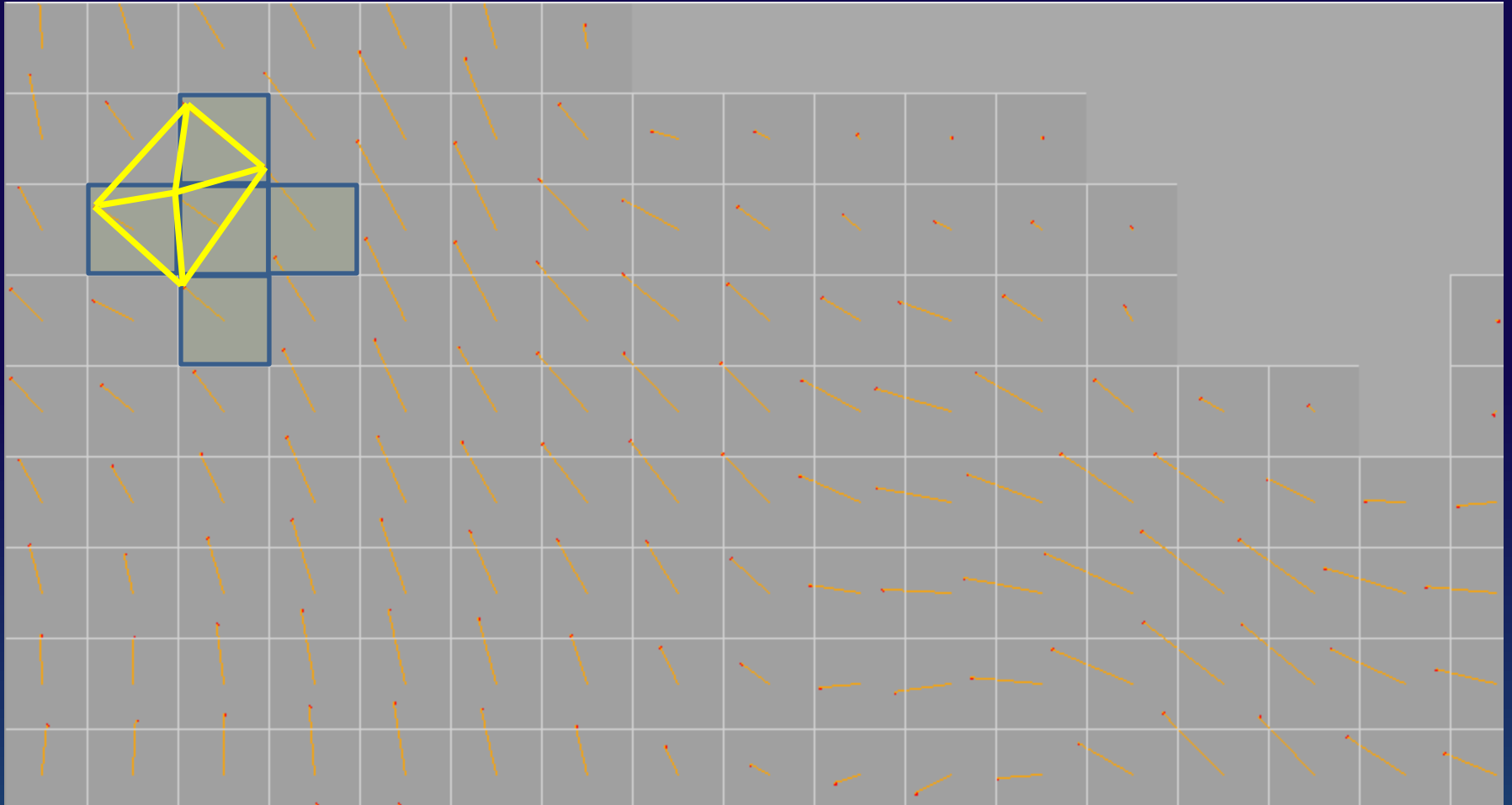
-Let us consider one grid cell, at timestep 0, and find its neighbors, according to some stencil

How do we use these trajectories?



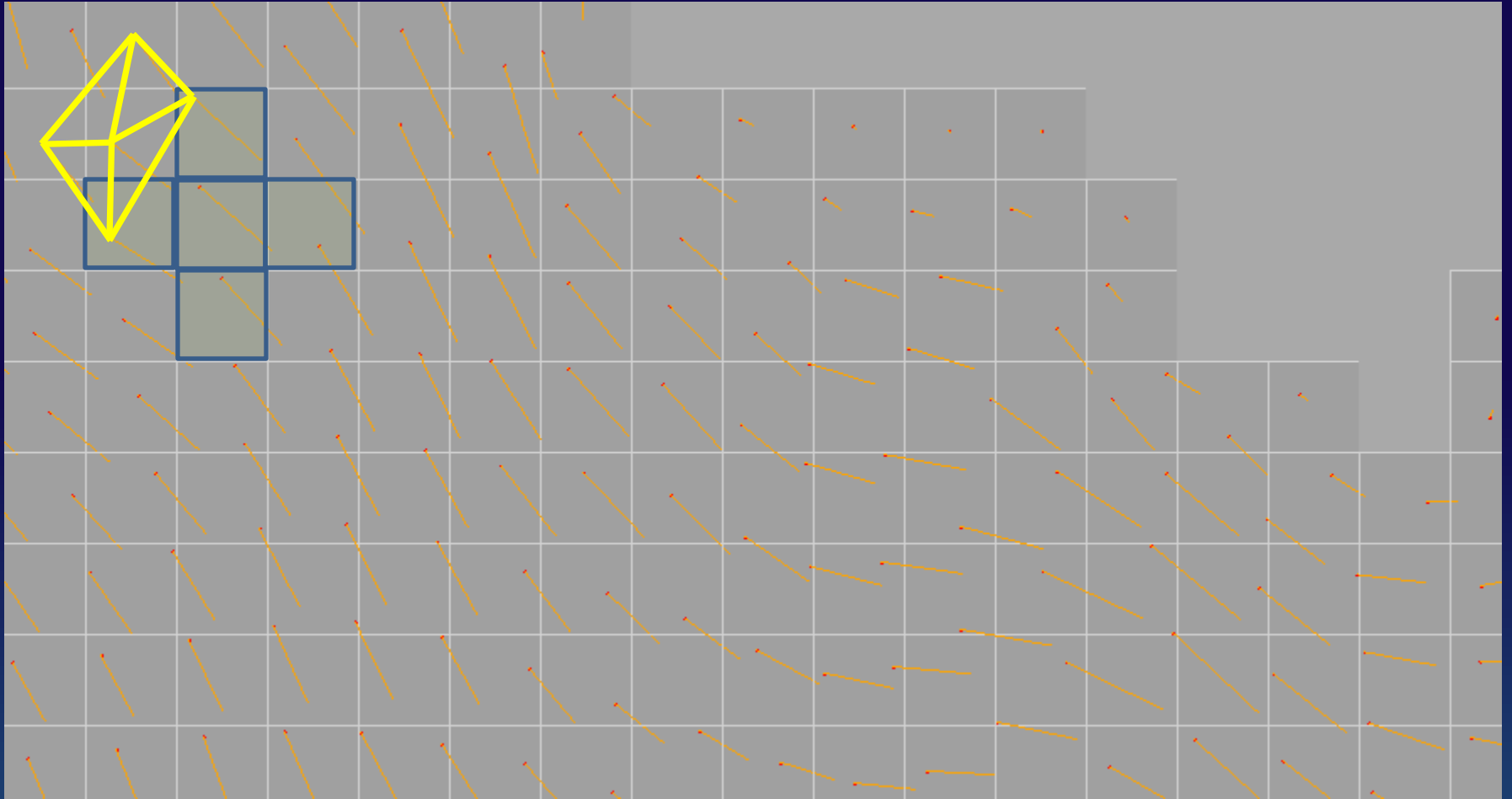
-We define a *Patch* as the summation of the areas of four triangles

How do we use these trajectories?



-If we have a look at next timesteps, we observe how this *Patch's* surface changes with time.

How do we use these trajectories?



-If we have a look at next timesteps, we observe how this *Patch's* surface changes with time.

- We know the coordinates of the tracers for each timestep, therefore we can calculate the area of a patch
- We can then express the compressibility property of a single grid cell by tendency of this area to increase or decrease through time:

$$\beta_l = \frac{\sum_{t=0}^{T-1} \frac{S_t}{S_{t+1}}}{T-1} \quad S_t = \left| (x_1^t - x_2^t) \cdot (y_1^t - y_3^t) - (y_1^t - y_2^t) \cdot (x_1^t - x_3^t) \right|$$

- And then express the global compressibility value by averaging over the total number of grid cells for which we have a local value

$$\beta_{tot} = \frac{\sum_{x=0, y=0}^{W, H} \beta_{[x,y]}^+}{W \cdot H}$$

Where W and H are the Weight and Height of the grid

- Therefore, concerning compressibility values, we can obtain:

- $\beta=1 \rightarrow$ Patch surfaces are constant over time
- $\beta>1 \rightarrow$ Patch surfaces are decreasing over time
- $\beta<1 \rightarrow$ Patch surfaces are increasing over time

- To prove our point, we would like to obtain values of $\beta>1$

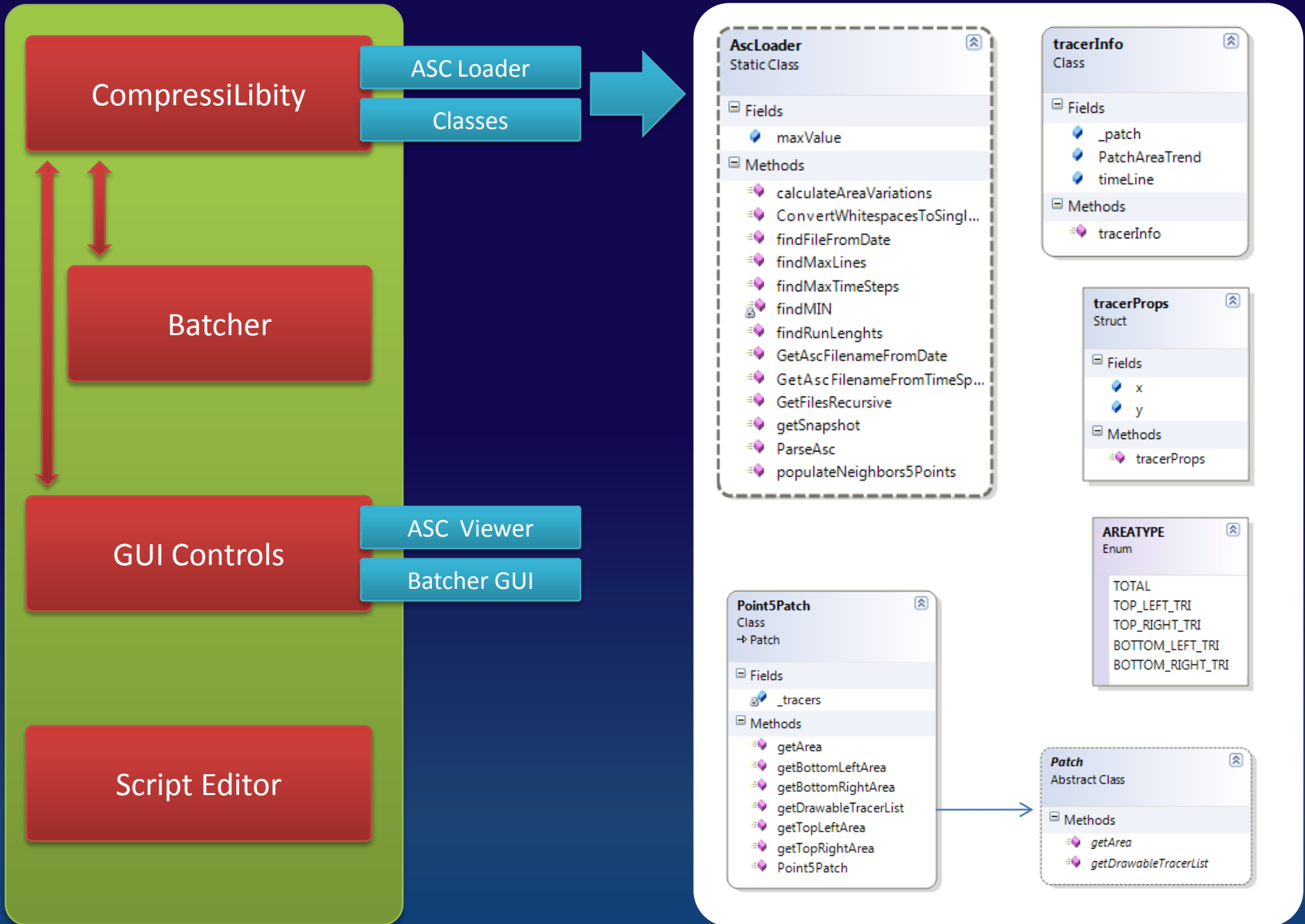




Implementation

Implementation

- The implementation of the method we've just seen has been done using Visual C#
- Flexible and transparent program, instead of a single script which does-it-all
- Suitable to be *used* by anyone with no programming skills
- Suitable to be *extended* by anyone with little programming skills
- Full code documentation / Use case tests
- Includes *modules*, a fully customizable *batch CLI*, an overlying *GUI* and a *Script Editor*
- Let's see all this goodness a bit more in detail!



CompressiLibity

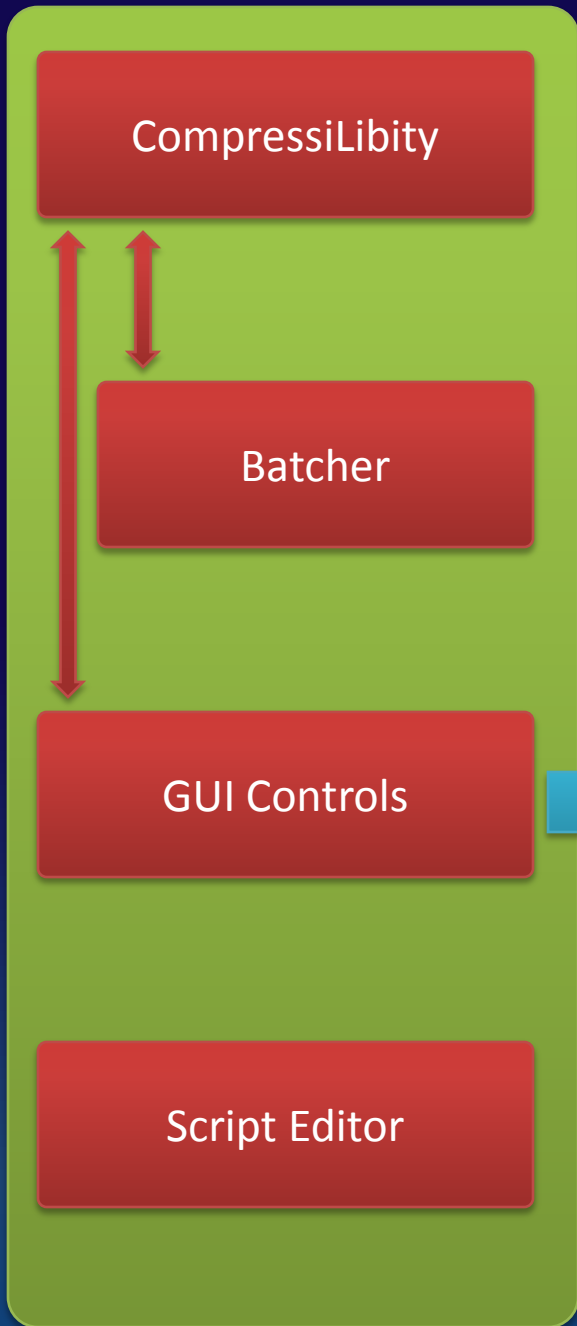
Batcher

GUI Controls

Script Editor

```
> Input files will be searched in c:\users\judgext\desktop\1991\  
> Output files will be in c:\users\judgext\desktop\outputs\output2\  
> Will process file for day 1/2/1991  
> Output file will be named 1.bmp  
> Statistics will be written in stats.txt  
> 1 files matching criteria found.  
Processing file data1991tw4_2_run.asc...  
ASCLOADER.MAXVALUE=1.529913  
  
***** JOB COMPLETED <2709.155 millisecs>  
  
c:\Users\JudgeXT\Desktop\Compressibility\Batcher\bin\Debug>batcher -i c:\users\judgext\des  
ktop\1991\ -o c:\users\judgext\desktop\outputs\output2\ -D 1/3/1991 -M 2.bmp -WRITESTATS s  
tats.txt  
  
##### ASC Batch Analysis tool #####  
# a tool written by Andrea Giudici (judge@area51staff.it) #  
# Tallinn University of Technology 2011 #  
#####  
> Input files will be searched in c:\users\judgext\desktop\1991\  
> Output files will be in c:\users\judgext\desktop\outputs\output2\  
> Will process file for day 1/3/1991  
> Output file will be named 2.bmp  
> Statistics will be written in stats.txt  
> 1 files matching criteria found.  
Processing file data1991tw4_3_run.asc...
```

```
*** On the command line, just type batcher -? followed by  
a combination of these parameters:  
  
-? Displays this screen  
-i [DIR] Specifies the input directory (Mandatory).  
Includes subdirectories.  
-o [DIR] Specifies the output directory (Mandatory)  
-tw [INT] Sets the time window (Default: 4)  
-A Processes all the files in input Directory and sub directories  
-R [DATE1][DATE2] Processes the files in the input folder regarding the time  
interval between [DATE1] and [DATE2]  
-D [DATE] Processes the files in the input folder regarding the specified [DATE]  
-tw [INT] Specifies the size of the timewindow for the current simulation  
-D [DATE] Processes the files in the input folder regarding the specified [DATE]  
-N [FILE] Sets a name for the output file [FILE]  
-WRITESTATS [F] Writes statistics to the specified [F] file  
-L [INT] Processes different sub patches, as follows:  
1 TOTAL  
2 TOP_LEFT_TRI  
3 TOP_RIGHT_TRI  
4 BOTTOM_LEFT_TRI  
5 BOTTOM_RIGHT_TRI  
  
All dates need to be specified in DD/MM/YYYY format.
```



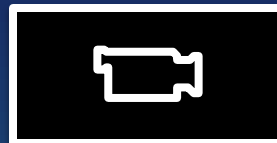
- This component allows us to explore graphically the trajectories of tracers

- Qualitatively assess the quality of a simulation before using it to extract further results

- Can be embedded within any* Windows application, from editors like Visual Studio, Delphi, MATLAB and others.

- All the underlying code is completely hidden to the user

- Interactive and customizable



CompressiLibity

Batcher

GUI Controls

Script Editor

```
c:\users\judgext\desktop\compressibility\batcher\bin\debug\  
  
    batcher -i c:\users\judgext\desktop\1991\ -o  
c:\users\judgext\desktop\outputs\ -R 1/1/1991 1/31/1991 -N 1jan.bmp -  
    WRITESTATS stats.txt  
  
    batcher -i c:\users\judgext\desktop\1991\ -o  
c:\users\judgext\desktop\outputs\ -R 2/1/1991 2/28/1991 -N 2feb.bmp -  
    WRITESTATS stats.txt  
  
    batcher -i c:\users\judgext\desktop\1991\ -o  
c:\users\judgext\desktop\outputs\ -R 3/1/1991 3/31/1991 -N 3mar.bmp -  
    WRITESTATS stats.txt  
  
    batcher -i c:\users\judgext\desktop\1991\ -o  
c:\users\judgext\desktop\outputs\ -R 4/1/1991 4/30/1991 -N 4apr.bmp -  
    WRITESTATS stats.txt
```

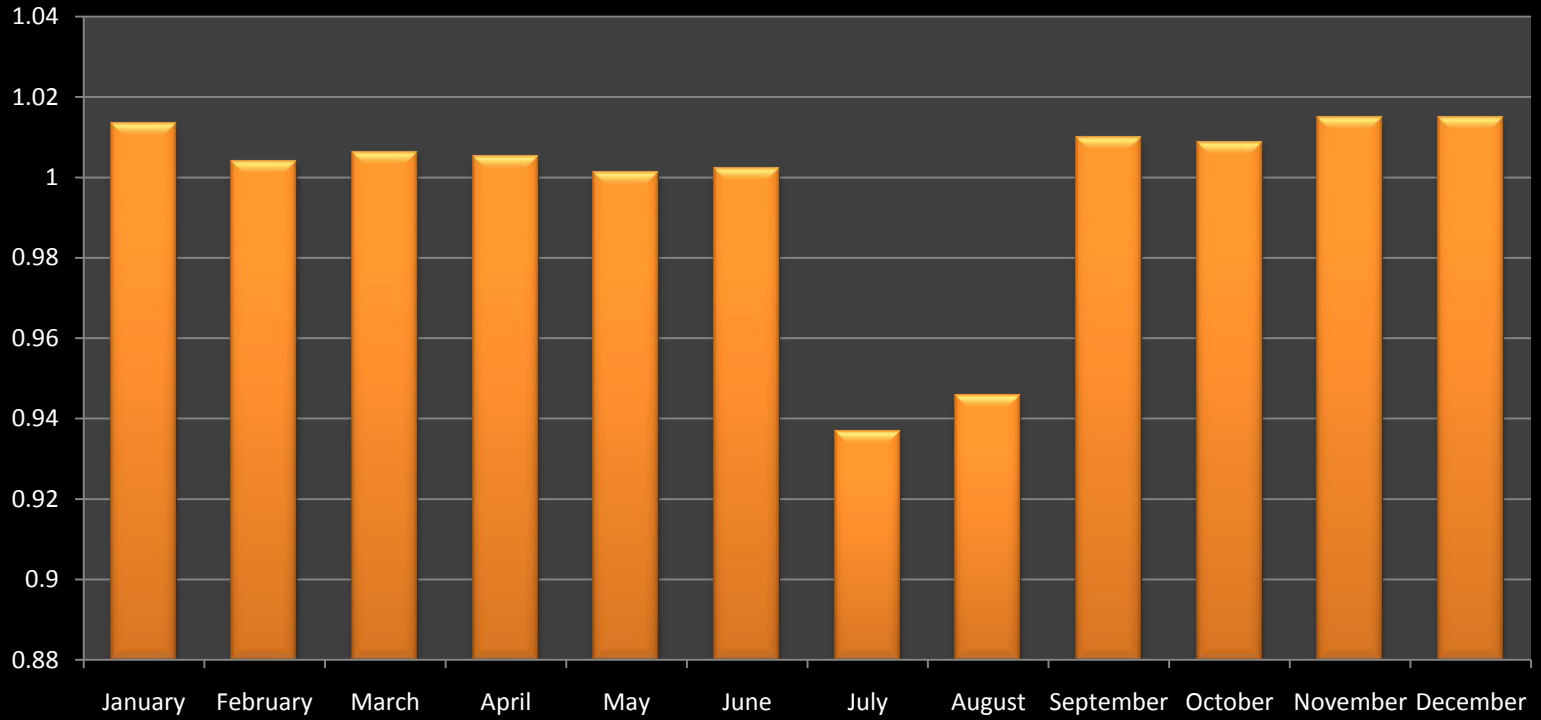
- Script creation tool to be used with Batcher
- Still working on it



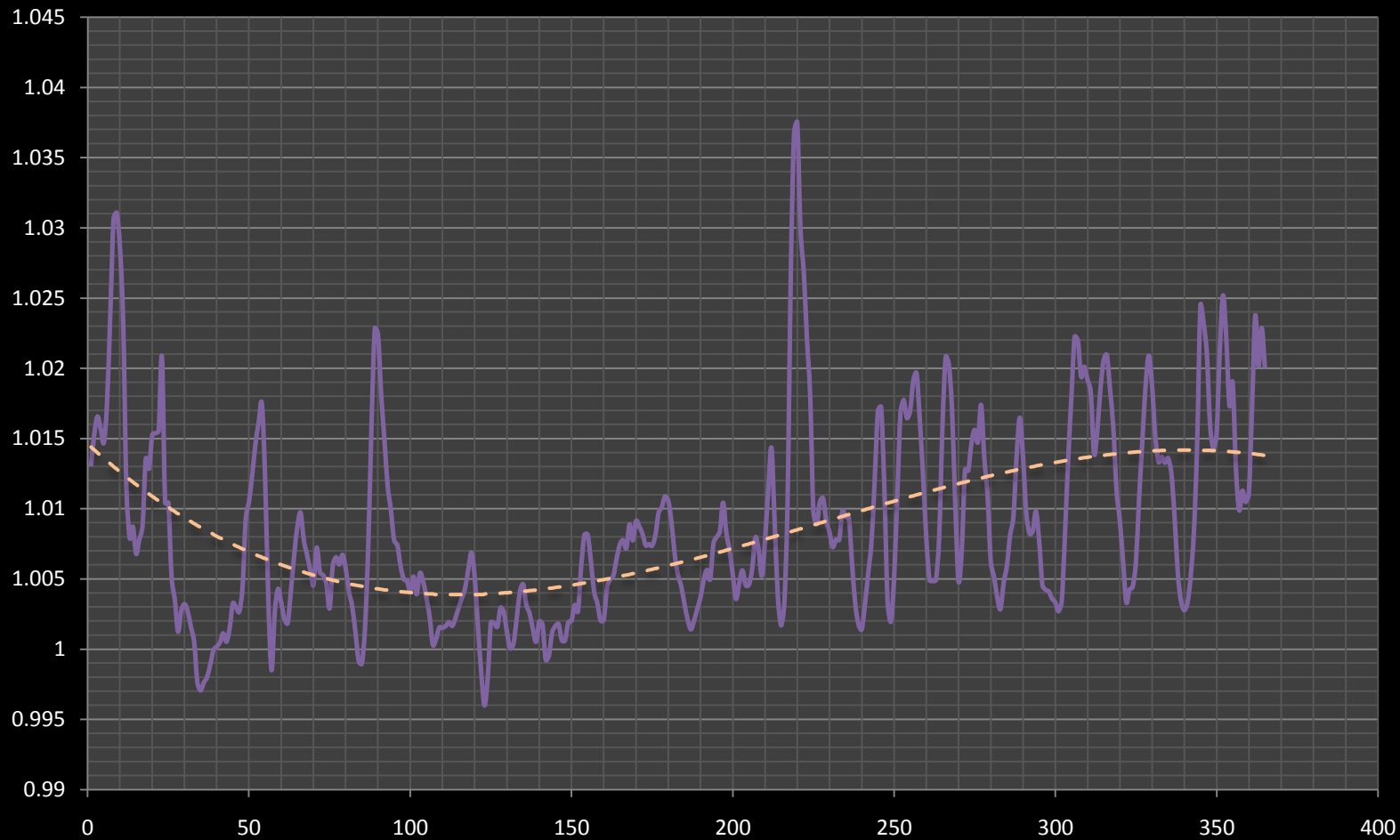
Some early results

2016 69117 162012

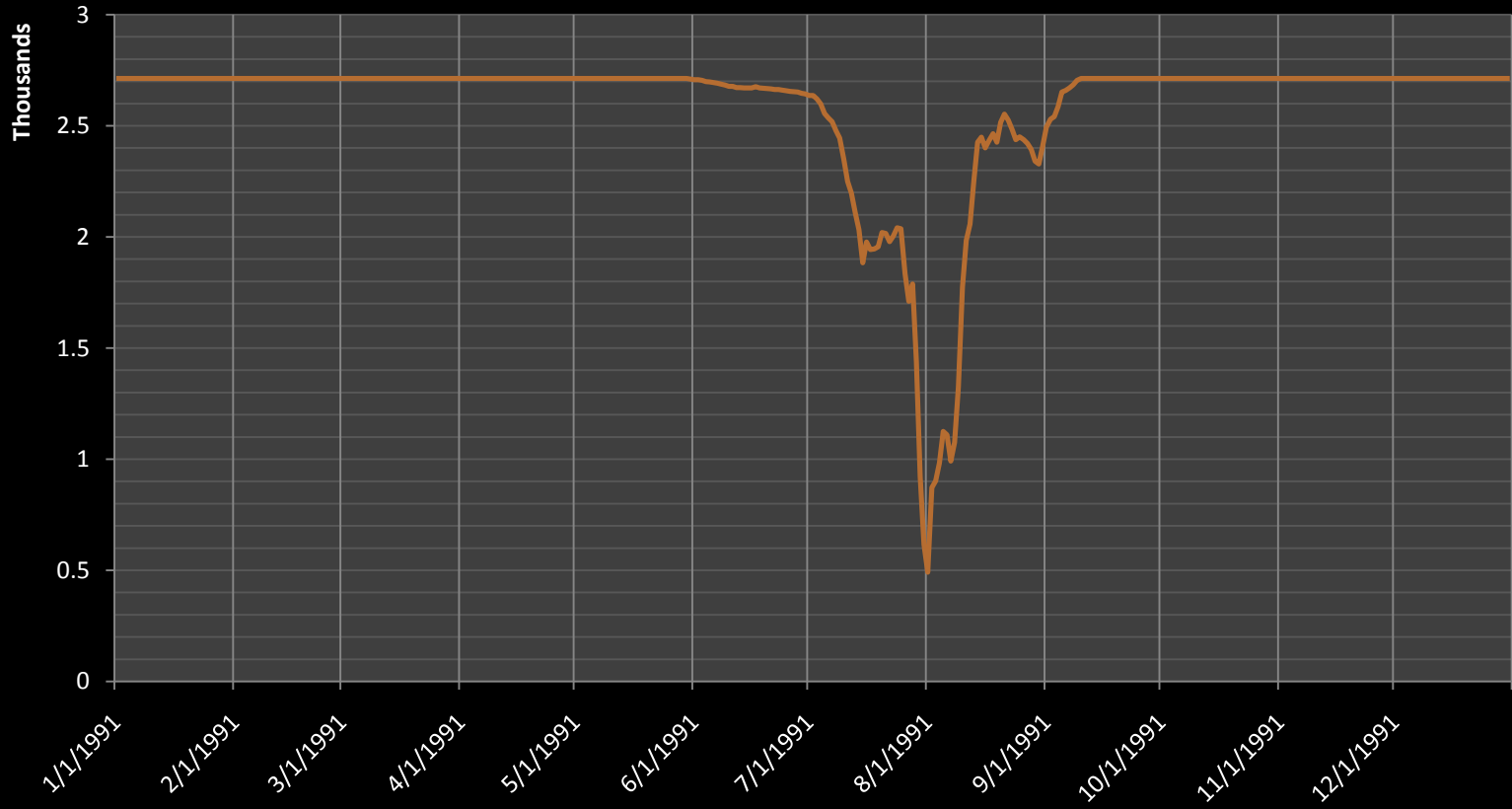
Compressibility value averaged by month



Compressibility value averaged per day



Number of simulated tracers



As always, there's space for improvements:

- Find/Fix bugs (#Tracers, division by zero, etc.)
- Perform analysis considering different parameters (time intervals, stencil of patches, patch definition methods, seasonal changes)
- Review complexity of algorithm (bottlenecks of $O(N^2)$, can be easily reduced to $O(N \log(N))$)

Any question?

