Modelli Computazionali per Sistemi Complessi

Esercitazione 2

Argomenti fatti a lezione:

1. Discussione sull'algoritmo di minimizzazione LINK AI LUCIDI MOSTRATI IN LABORATORIO
2. Progetto Camelot lava_minimizzazione

Argomenti suggeriti per casa: LINK AL MATERIALE ONLINE

---

2. Progetto Camelot lava_minimizzazione

Ripartendo dalla prima esercitazione, creiamo all'interno della cartella progetti la cartella lava_minimizzazione con il comando

```
cd progetti
mkdir  lava_minimizzazione
cd  lava_minimizzazione
```

a questo punto siamo nella cartella lava_minimizzazione e dobbiamo scaricare dal sito del prof. Spataro i files che ci servono con il comando

```
wget http://www.mat.unical.it/spataro/teaching/modelli/istruzioni.txt
wget http://www.mat.unical.it/spataro/teaching/modelli/macrocell.c
wget http://www.mat.unical.it/spataro/teaching/modelli/bove_501_351_z.cmt
```

il primo file contiene le istruzioni per settare le variabili interne di camelot, il secondo è un file che va sempre incluso in qualsiasi progetto vogliamo fare, il terzo è una configurazione iniziale per il sottostato **quote** che è contenuto in questo progetto.

A questo punto apriamo il nostro editor di testo preferito, scriviamo il programma e lo salviamo col nome **lava_minimizzazione.cpt** nella cartella lava_minimizzazione

```
/*--------------------------------------------------------------------------*
 *                                                                          *
 * CAMELot/    Lava flows code - Release 1.0 - September, 21 2000          *
 *                                                                          *
 * File:   lava01.cpt                                                       *
 *                                                                          *
 * Authors:    Gino Mirocle Crisci        UNICAL - Dept. of Earth Sciences *
 *             Salvatore    Di Gregorio   UNICAL - Dept. of Mathematics    *
 *     Rocco        Rongo         UNICAL - Dept. of Earth Sciences *
 *     William      Spataro       UNICAL - Dept. of Mathematics    *
 *     Donato      D'Ambrosio     UNICAL - Dept. of Mathematics    *
 *                                                                          *
 * Purpose:     Simulation code for lava flows                              *
 *                                                                          *
 *--------------------------------------------------------------------------*
 */

# include <stdio.h>
# include <math.h>

#define TRUE 1
#define FALSE 0

cadef
{
   dimension 2;

   radius 1;

   state (float quote;
          float width,
                flow[4]);

   neighbor news[4]([0,-1]NORTH,[1,0]EAST, [-1,0]WEST,[0,1]SOUTH);

   parameter(prm_lato      50.0,
             prm_tick      30.0,
             prm_secstep 1440.0,
             prm_cratX    100.0,
             prm_cratY    100.0,
             prm_rate     100.0,
             prm_rall     0.1

   );
}//cadef



// Insert here your global variables inside the cell;


void  calc_width();
void  calc_flows();


// main
{
  if ((GetX==50)&&(GetY==50))
```

```
    update(cell_width, 10.);
  if ((step % 2)==0) // Two step algorithm
    calc_flows();
  else{
    calc_width();
  }//else

}//niam

/*************************************************************************
*********/
/* Calculate new lava width*/
/*************************************************************************
*********/

void calc_width()
{
  int   i;
  float new_width;

  new_width=cell_width;
  for (i=0; i<4; i++)
    new_width-=(cell_flow[i]-news[i]_flow[3-i]);
  update(cell_width,new_width);


} // calc_width



/*************************************************************************
*********/
/* Calculate outgoing flows*/
/*************************************************************************
*********/

void calc_flows()

{
  char eliminated[5]={FALSE,FALSE,FALSE,FALSE,FALSE},
       elim;
  int i,
      k,
      num_vic;
  float adherence,
        inamovibile[5],
        distribuibile,
        media,
        rall;

   inamovibile[0]=cell_quote;
   for (i=1; i<5; i++)
      inamovibile[i]=news[i-1]_width+news[i-1]_quote;
   distribuibile=cell_width;
  // calcola flussi
   do{
     elim=FALSE;
     media=distribuibile;
```

```
    num_vic=0;
    for (k=0; k<5; k++)
      if (!eliminated[k]){
        media+=inamovibile[k];
        num_vic++;
      }//if
    if (num_vic!=0)
      media/=num_vic;
    for (k=0; k<5; k++)
      if((media<=inamovibile[k])&&(!eliminated[k])){
        eliminated[k]=TRUE;
        elim=TRUE;
      }//if
  }while (elim);
  for (k=1; k<5; k++)
    if (!eliminated[k]){
      update(cell_flow[k-1],(media-inamovibile[k])*prm_rall);
    }//if
    else
      update(cell_flow[k-1],0.);

} //calc_flows
```

A questo punto possiamo aprire il nostro progetto lava_minimizzazione con camelot con il comando:

```
camelot lava_minimizzazione.cpt &
```

fatto ciò controlliamo che non ci siano errori sintattici nel programma: usiamo il bottone **Compile** di camelot. Il messaggio “CARPET compilation successfull” ci da l'ok a proseguire.

Fatto ciò il menu **CONFIGURE** di camelot diverrà attivo e lo utilizzeremo per configurare alcune variabili interne del nostro progetto (nello specifico andremo a indicare la dimensione dell'automa cellulare e i flag con cui compilare il progetto); usando il menu **CONFIGURE-AUTOMATA_DIMENSIONS** andremo a specificare 501 per larghezza (width) e 351 per l'altezza (height).

A questo punto salviamo la configurazione fin qui creata: **FILE-SAVE_CONFIGURATION...**
Così facendo è stato creato un file dal nome **lava_minimizzazione.cnf** che contiene i le informazioni riguardo il progetto. Ora chiudiamo camelot: **FILE_EXIT.**

Per evitare di usare la simulazione del terzo tasto del mouse andiamo a **modificare a mano il file lava_minimizzazione.cnf** e lo rendiamo uguale alle seguenti righe:

```
501 351 1
1
1
0
0
$MPIR_ROOT/bin/mpicc
-O -DCPT_INCLUDE_FILE=\"%s\" -DNO_XDR_READ -DNO_XDR_WRITE -I/opt/Modelli/Camelot-1.3/si_3_6_1 -L/opt/Modelli/Camelot-1.3/si_3_6_1/linux -I/usr/local/mpich-1.2.5/include -L/usr/local/mpich-1.2.5/lib
-lm
$MPIR_ROOT/bin/mpirun -np %d
```

il carattere nanoscopico con cui è scritto è dettato dal fatto che se la riga più lunga fosse spalmata su due righe, tutto sarebbe interpretato in modo errato da camelot. Ad ogni modo ecco la versione leggibile:

```
501 351 1
1
1
0
0
$MPIR_ROOT/bin/mpicc
-O -DCPT_INCLUDE_FILE=\"%s\" -DNO_XDR_READ -DNO_XDR_WRITE
-I/opt/Modelli/Camelot-1.3/si_3_6_1 -L/opt/Modelli/Camelot-1.3/si_3_6_1/linux
-I/usr/local/mpich-1.2.5/include -L/usr/local/mpich-1.2.5/lib
-lm
$MPIR_ROOT/bin/mpirun -np %d
```

NOTA: 501x351, questo AC ha dimensioni diverse rispetto a quello denominato mod2 (100x100).

è importante che il file sia costituito esattamente da 9 righe.

Fatto ciò lanciamo il nostro progetto in camelot col comando:

```
camelot lava_minimizzazione.cpt &
```

così facendo il file di configurazione prima editato è caricato in automatico; ora pigiamo prima **Compile** e poi **Build** in camelot

Camelot dovrebbe dirci **[OK]** dopo il comando Build se tutto è andato bene (il risultato di Compile è “CARPET compilation successfull”).

Siamo finalmente pronti per caricare la configurazione iniziale per il sottostato chiamato “**quote**” e procedere alla simulazione: premiamo **Run** in camelot e ci appare una nuova maschera, da questa scegliamo il menu:
STATE-INITIALIZE-SUBSTATE-FROM_FILE...

e selezioniamo il sottostato **quote** nella finesta che compare;

in fine facciamo doppio click sul file **bove_501_351_z.cmt** che compare in una nuova maschera.

Siamo pronti finalmente a visualizzare il nostro progetto!!!

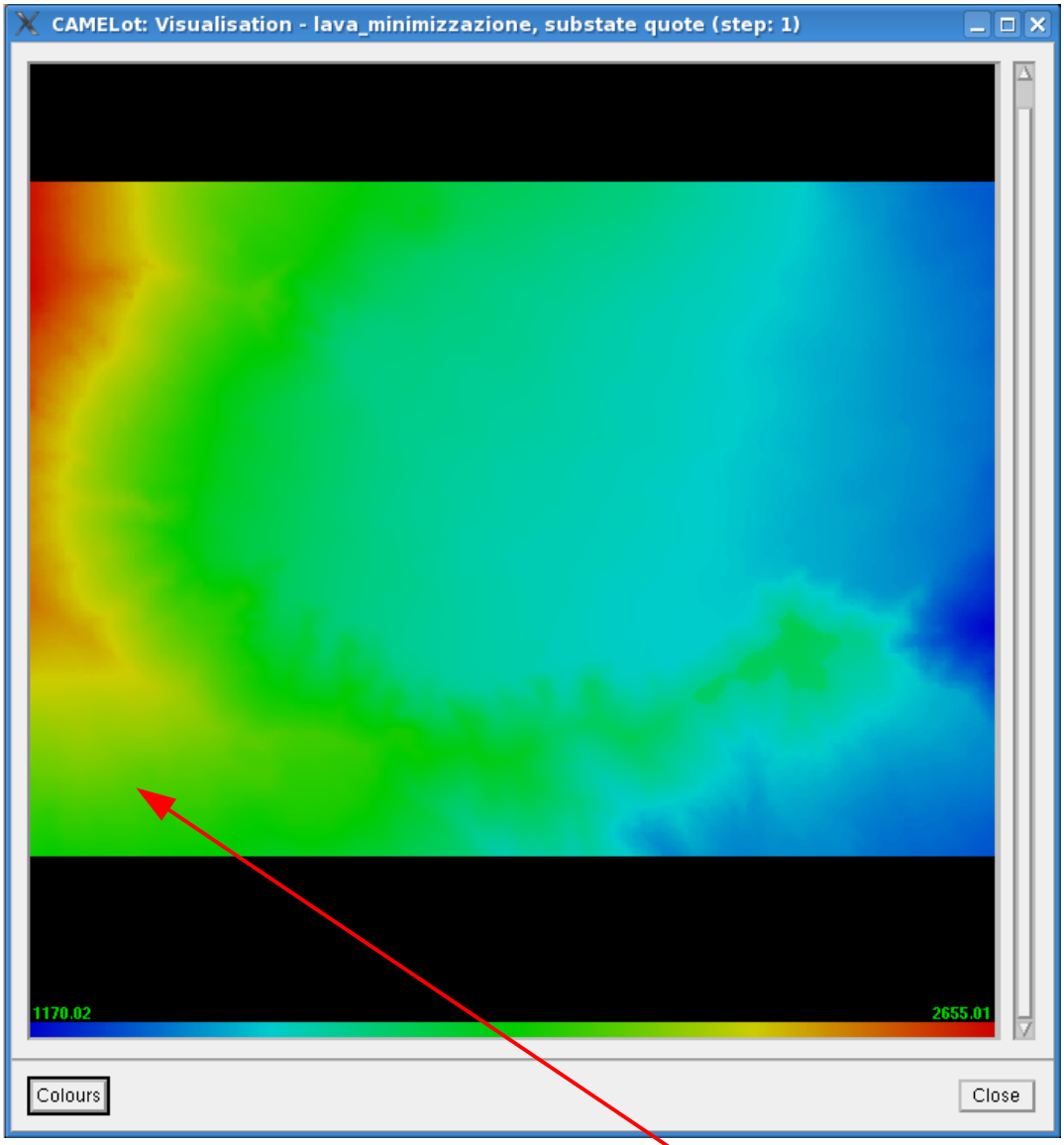
Nella finesta dal titolo **“CAMELot: Simulation – lava_minimizzazione”**

scegliamo il menu **SETUP-SET_NUMBER_OF_STEPS** ed introduciamo il valore **1** in modo che quando andremo a premere Go farà un passo per volta.

Ora scegliamo il bottone **Visualise**, scriviamo **1** nel box che ci chiede ogni quanti step visualizzare,

segliamo il sottostato **quote**; questo ci mostra la morfologia del vulcano Etna, in

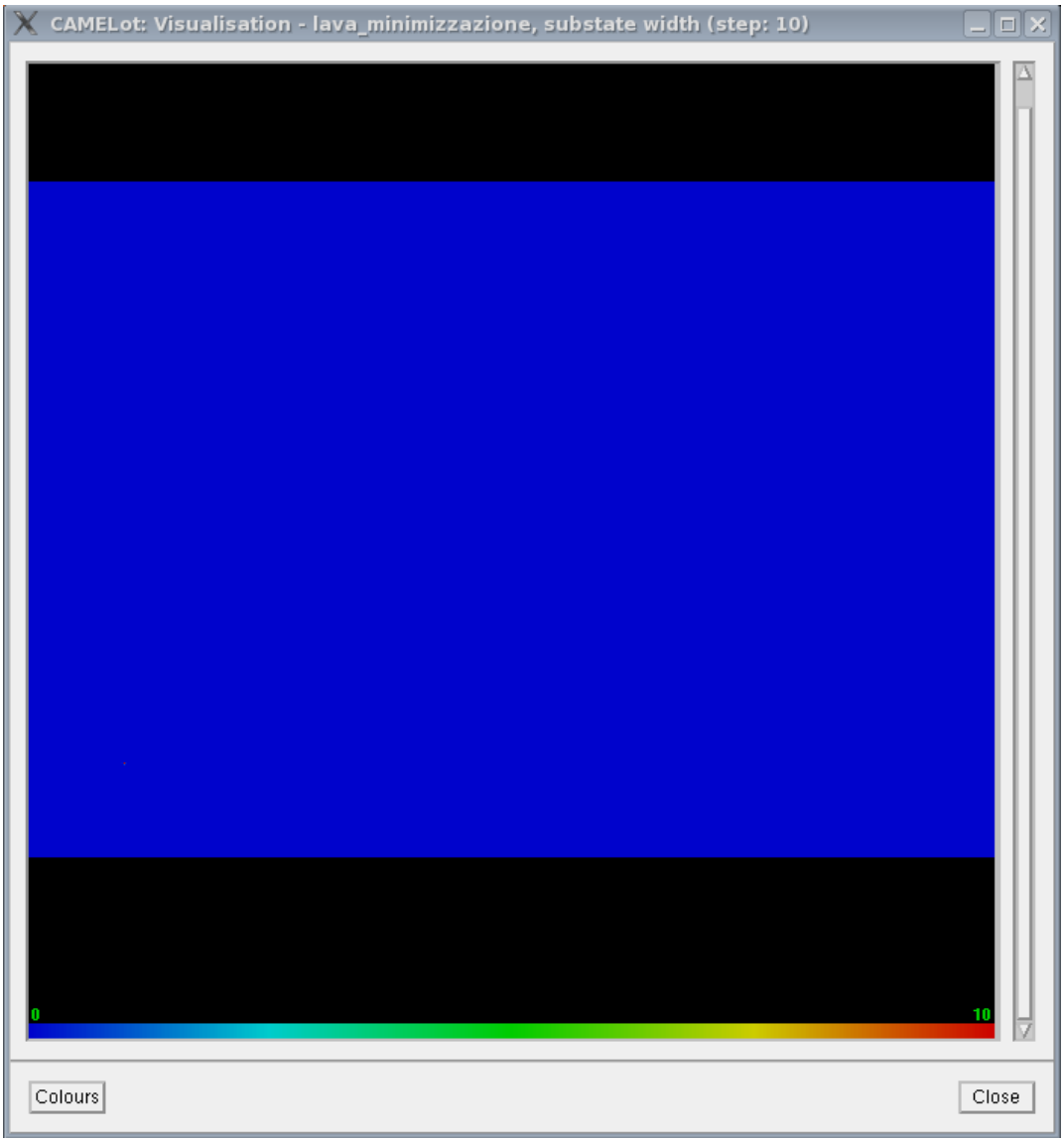
particolare si tratta della valle del bove



Ora possiamo chiudere questa visualizzazione mediante il bottone **Close**, ma non prima d'aver fatto caso a come è fatta la morfologia in questo punto...
si tratta di una zona con una forte pendenza verso il basso e con un promontorio a monte... forti di questa osservazione, dove ci aspettiamo che vada una quantità di lava che si trova in questo punto? Se abbiamo scritto bene il programma la lava dovrebbe scendere, in modo da assecondare la pendenza. Ciò, ovviamente, avviene grazie all'algoritmo di minimizzazione di Di Gregorio e Serra che in base ai livelli di quota e spessore distribuisce le quantità di lava.

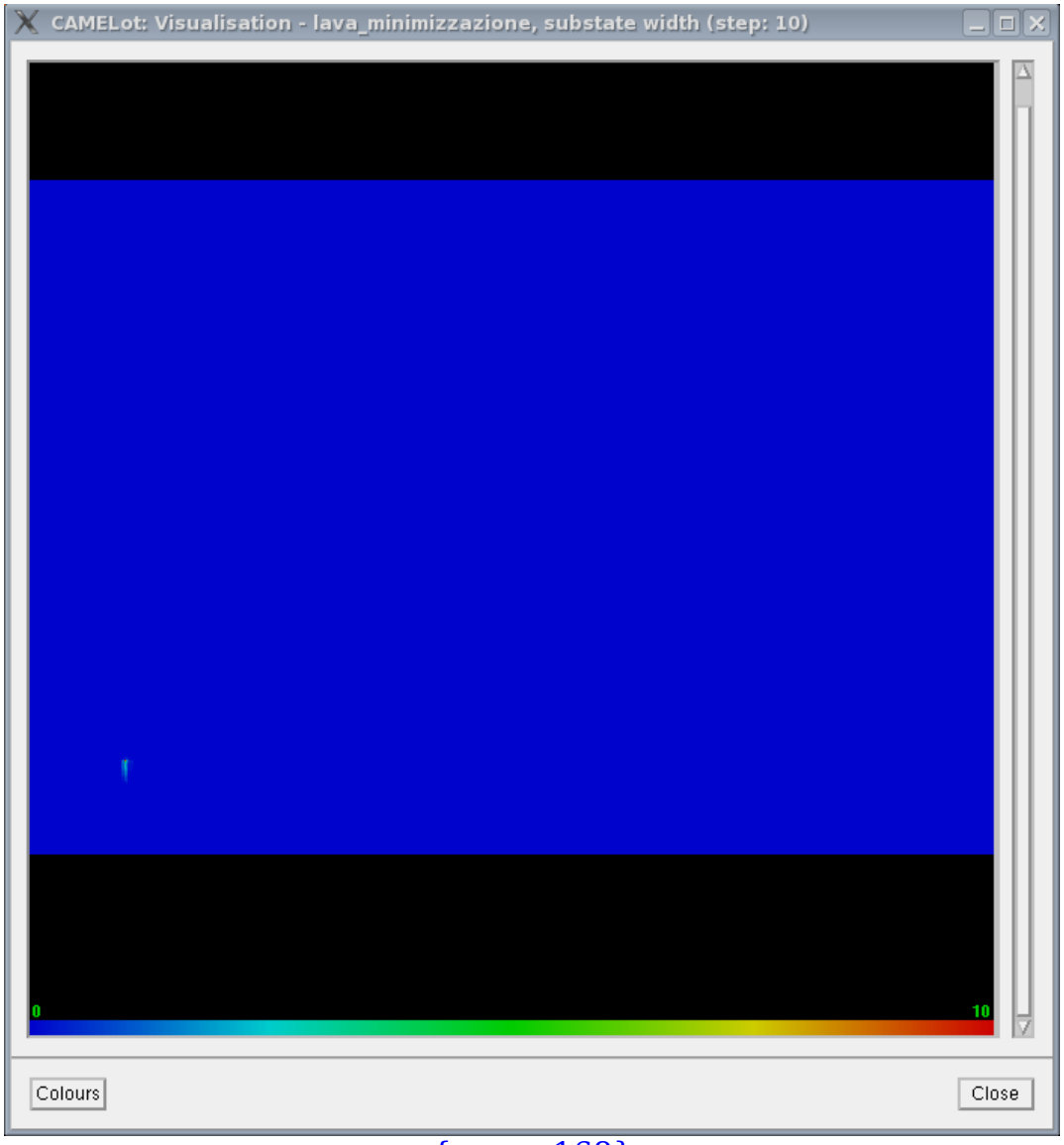
A questo punto siamo pronti per andare a vedere come evolve il nostro automa cellulare! In particolare siamo interessati al percorso che fa la lava nel caso in cui dovesse crearsi un cratere nel punto sopraindicato dalla freccia: dopo aver

chiuso la rappresentazione della morfologia, scegliamo di nuovo il bottone **Visualise**, scegliamo **10** come passo di visualizzazione ed scegliamo il **sottostato width** che rappresenta lo spessore di lava; a questo punto lanciamo la simulazione con il tasto **Loop** e stiamo a guardare come evolve.



{passo 10}

Se pur minuscolo già si vede, nella posizione di interesse indicata dalla freccia nella pagina precedente, che vi è un puntino rosso! Questo indica una differenza di spessore in quanto blu e rosso sono i due estremi nella gradazione utilizzata dal visualizzatore di camelot.



{passo 168}

questo è quello che vediamo dopo un centinaio di passi!

Ma cosa andremo a visualizzare dopo un migliaio di passi???

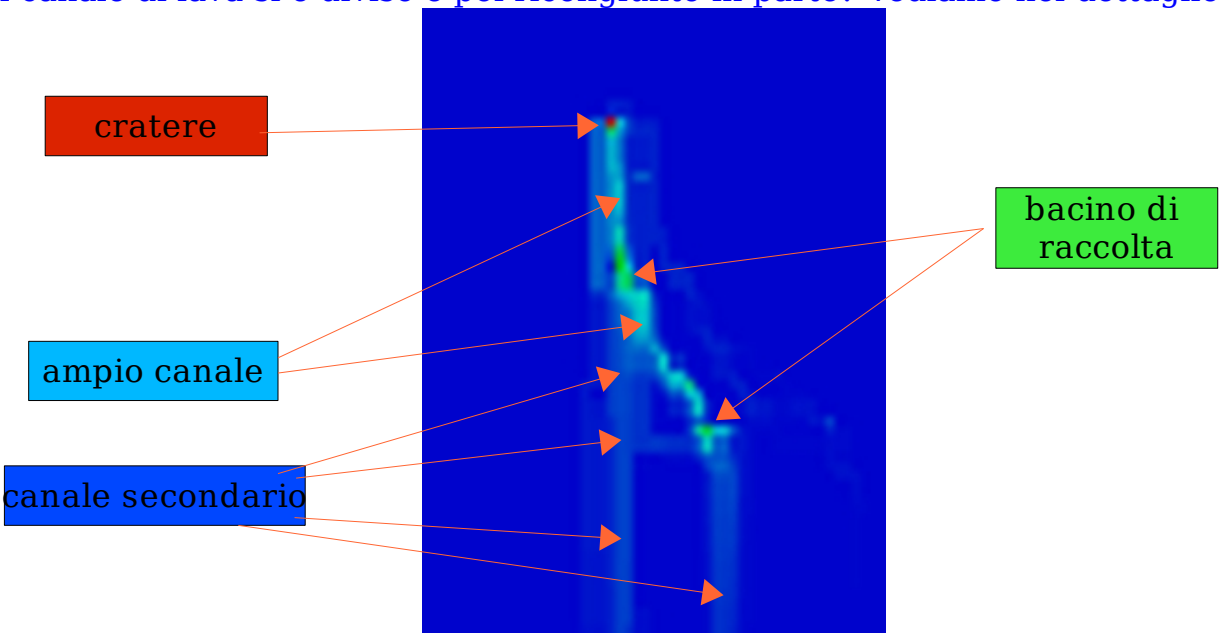
Sempre e soltanto un flusso verticale verso la zona di massima pendenza???

Per fortuna no!

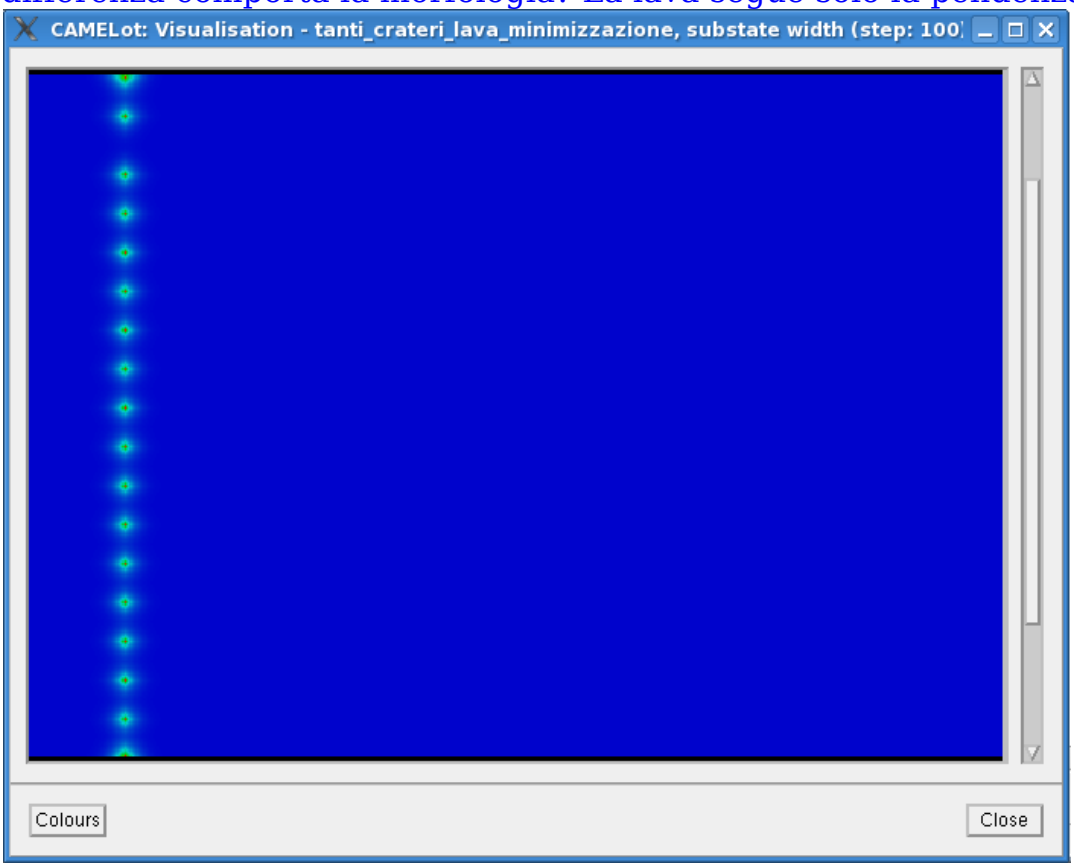
Le sorprese non sono poche quando di lavora con gli automi cellulari...

ecco cosa visualizzeremo al passo 1193:

il canale di lava si è diviso e poi ricongiunto in parte? Vediamo nel dettaglio...



Quanta differenza comporta la morfologia? La lava segue solo la pendenza?



sopra: 18 crateri in pianura
sotto: 18 crateri nella valle del bove