

Modelli Computazionali per Sistemi Complessi

Esercitazioni 3 e 4

Argomenti fatti a lezione:

1. Approfondimento riguardo il Progetto Camelot lava_minimizzazione
2. Progetto Camelot lago_arcavacata
 - o versione base a partire dalla lava...
 - o variante 1: il deposito dell'inquinante sul fondo
 - o variante 2: la corrente superficiale che sposta l'inquinante favorendo in alcuni casi e sfavorendo in altri il lavoro dell' algoritmo di minimizzazione
 - o variante 3: la dispersione dell'inquinante per contatto con la riva

Argomenti proposti per casa:

1. L'implementazione delle varianti sopracitate
NOTA: la 2 è la più difficile
-

2. Progetto Camelot lago_arcavacata

Traccia:

Inquinamento al "Lago di Arcavacata".

Un file binario, i cui elementi sono numeri interi di tipo short (2 byte), rappresenta tramite una matrice di dimensione 100x100 un'area con al centro un lago (0=acqua, 1=terra). Nella configurazione iniziale si suppone che la cella di coordinate (52, 13) rappresenti una sorgente di inquinamento per i primi $n \leq 100$ passi (specificati da un parametro) che immetta nel sistema una quantità di inquinante pari a $0 < q \leq 100$ (specificato da un parametro).

L'evoluzione del sistema prevede il diffondersi dell'inquinante secondo le modalità dell'algoritmo di minimizzazione, con la variazione specificata di seguito:

Variazione 1

Supporre che ad ogni passo parte dell'inquinante pari al 2-5%, specificato da un parametro, si depositi sul "fondo" del lago. Le celle di tipo terra non ricevono inquinante col meccanismo dell'algoritmo di minimizzazione.

Variazione 2

Supporre che ad ogni passo una corrente superficiale sposti il 5-15% dell'inquinante (specificato da un parametro) sulla cella vicina a Sud. Le celle di tipo terra non ricevono flusso, né a causa dell' algoritmo di minimizzazione, né a causa della corrente superficiale.

Variazione 3

Supporre che ad ogni passo le celle di tipo acqua cedano ad ogni cella di tipo terra con cui sono a contatto, una quantità pari al 5-15% di inquinante (specificato da un parametro). Le celle di tipo terra non ricevono inquinante col meccanismo dell' algoritmo di minimizzazione.

Ripartendo dalla lava riadattiamo il codice della scorsa esercitazione per modellare il lago di arcavacata!

```
cd progetti
mkdir lago_arcavacata
cd lago_arcavacata
cp ../lava_minimizzazione/lava_minimizzazione.cpt lago_arcavacata.cpt
```

a questo punto siamo nella cartella lago_arcavacata e dobbiamo scaricare dal sito 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/lago_100_100_short_source52x13.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 **lago** che è contenuto in questo progetto.

A questo punto apriamo con il nostro editor di testo preferito il file **lago_aracavacata.cpt** e lo modifichiamo nell'ottica della modellazione di un fenomeno legato allo studio degli inquinanti in lago anziché allo studio delle colate laviche.

Modifiche per passare dalla lava al lago di arcavacata:

- **definiamo** oltre a TRUE e FALSE anche **ACQUA** e **TERRA**, ovvero 0 e 1
- sparisce il sottostato quote ed è rimpiazzato dal **sottostato lago**
- lago è di tipo **short**
- **cell_lago** può quindi essere ==ACQUA oppure ==TERRA
- supporremo di usare il vicinato di **Moore**
- avremo bisogno di un sottostato chiamato **inquinante** di tipo **float**
- utilizzeremo la strategia del doppio-passo (con **switch step%2**) per lanciare la funzione `calcolaFlusso()` e la funzione `calcolaInquinante()`
- la funzione **calcolaFlusso()**
 - è l'algoritmo di minimizzazione! Dobbiamo solo convertire qualche nome come ad esempio
 - `if (cell_lago == ACQUA)`
 - `DIMVICINATO...`
 - `ecc...`
- la funzione **calcolaInquinante()**
 - è la funzione che distribuisce l'inquinante!
 - ricalca perfettamente la funzione `calc_width()` vista per la lava ma ovviamente dobbiamo riadattarla affinché distribuisca l'inquinante

fatto!!!

Con queste banali modifiche siamo passati dalla modellizzazione di una colata lavica alla modellizzazione della distribuzione di un inquinante in un lago. Questo deve necessariamente farci riflettere sulla potenza degli strumenti che usiamo!

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

```
camelot lago_arcavacata.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 **100 per larghezza (width) e 100 per l'altezza (height)**.

A questo punto salviamo la configurazione fin qui creata: **FILE-SAVE_CONFIGURATION...**

Così facendo vogliamo salvare un file dal nome **lago_arcavacata.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 lago_arcavacata.cnf** e lo rendiamo uguale alle seguenti righe:

```
100 100 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:

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

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

Fatto ciò lanciamo il nostro progetto in camelot col comando:

```
camelot lago_arcavacata &
```

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 **"lago"** 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 **lago** nella finestra che compare;

in fine facciamo doppio click sul file **lago_100_100_short_source52x13.cmt** che compare in una nuova maschera.

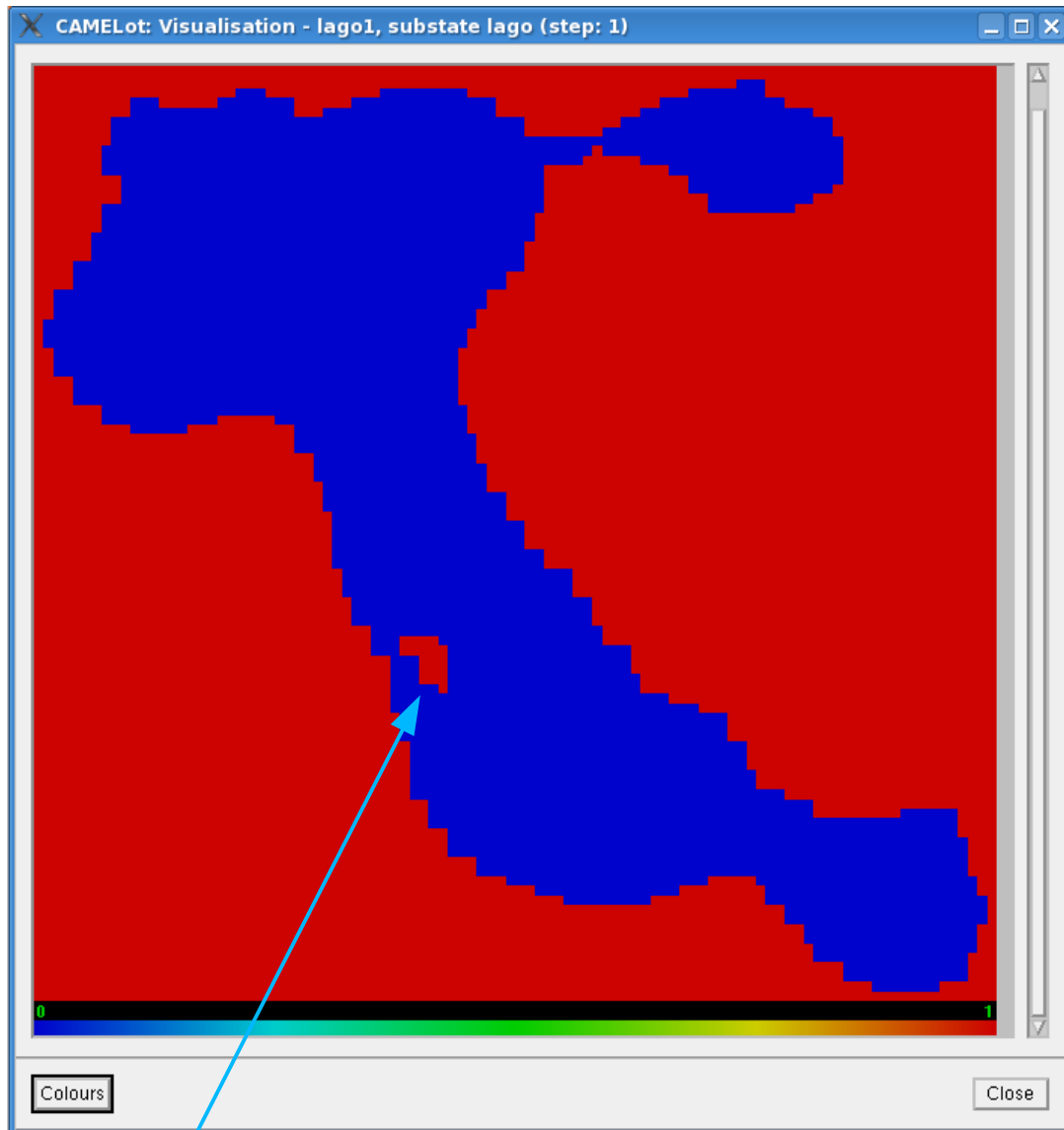
Siamo pronti finalmente a visualizzare il nostro progetto!!!

Nella finestra dal titolo “**CAMELot: Simulation - lago_arcavacata**”

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,

scegliamo il sottostato **lago**; questo ci mostra la morfologia dell'isola di arcavacata... la famosa isola di arcavacata! ;-)



c'è la terra in rosso tutto intorno, il lago di arcavacata in blu, e l'isola di arcavacata qui

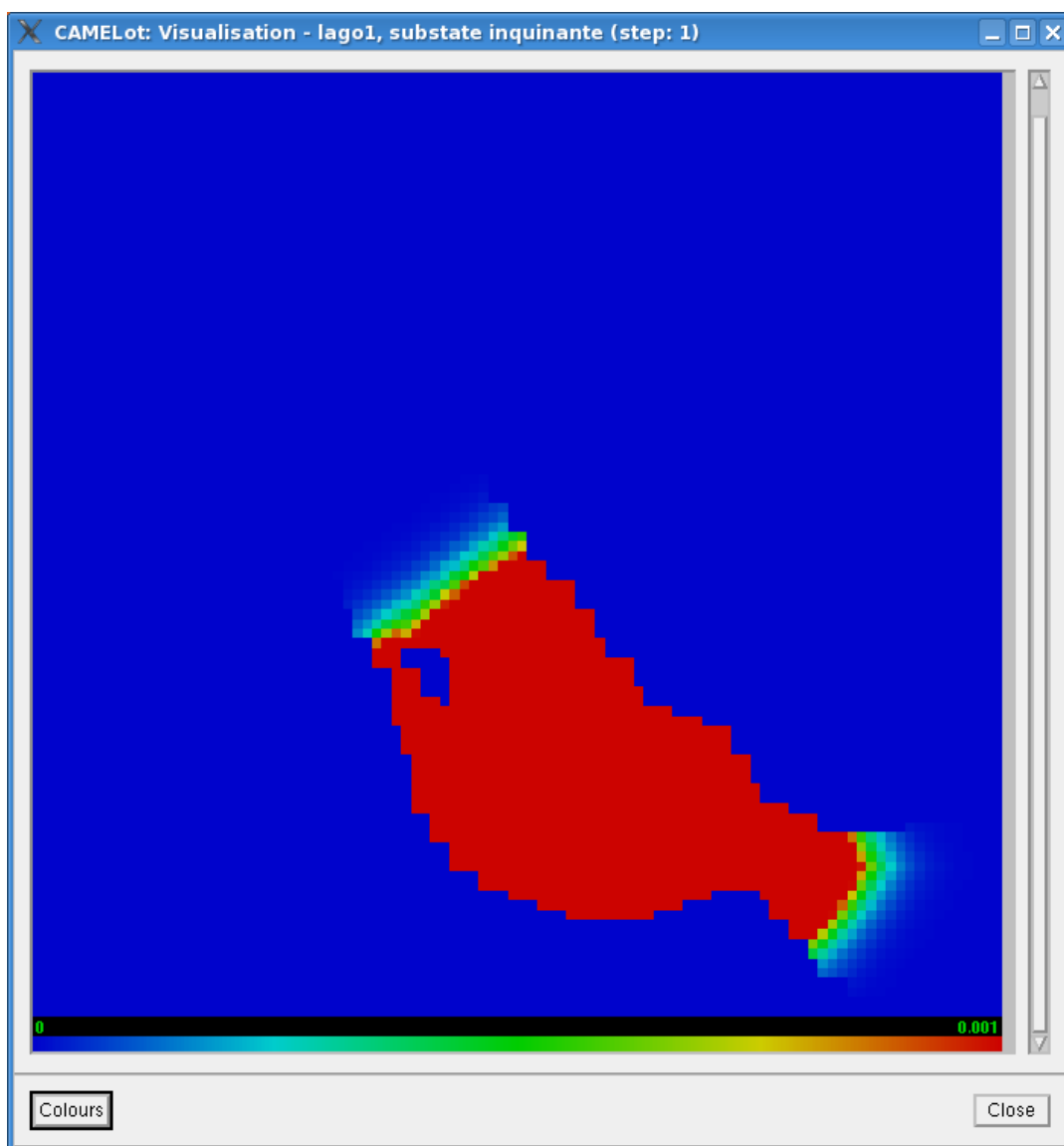
Ora possiamo chiudere questa visualizzazione mediante il bottone **Close**

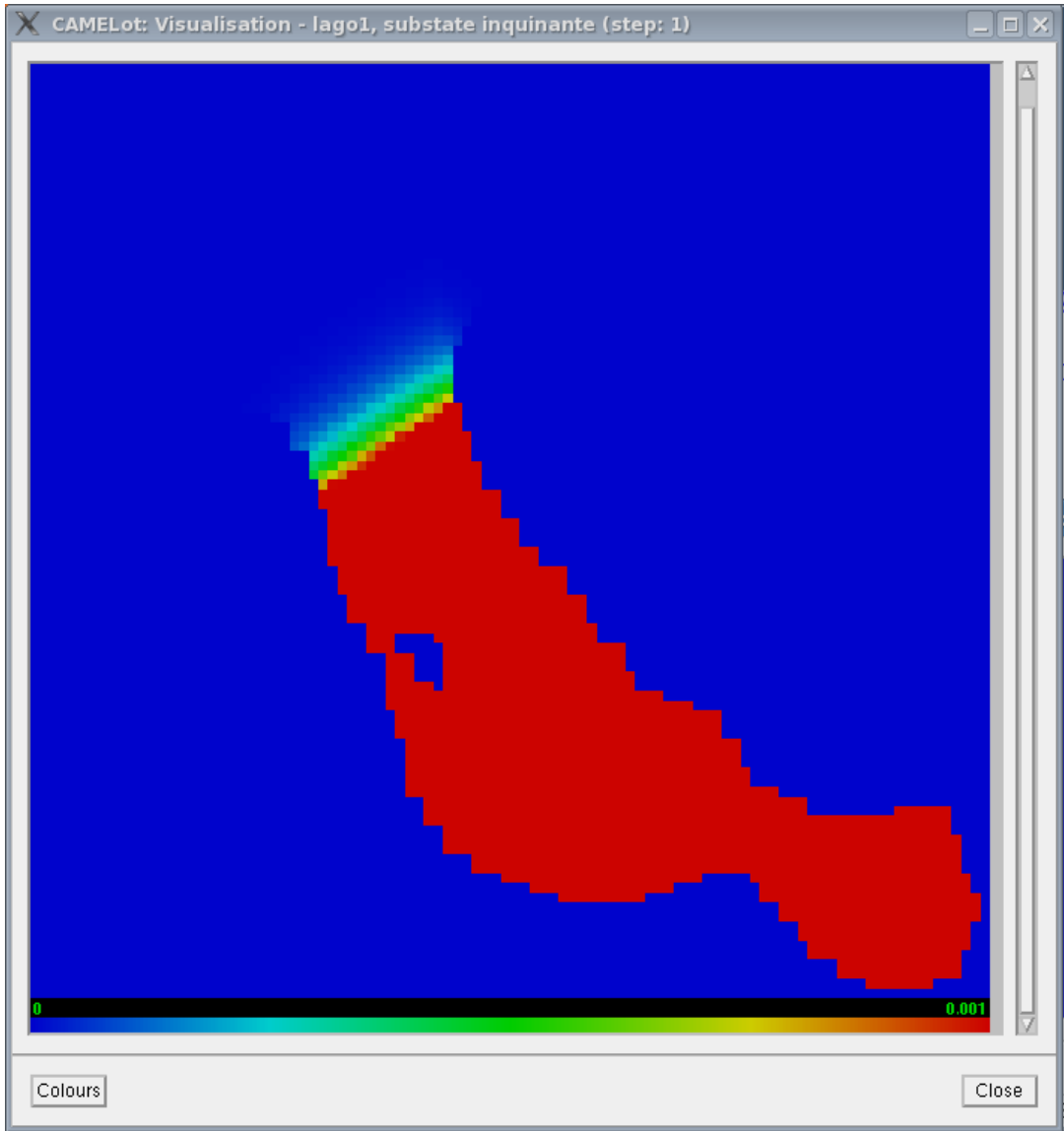
Andiamo nella finestra dal titolo **“CAMELot: Simulation - lago_arcavacata”** e scegliamo dal menu: **SETUP-MIN/MAX_SUBSTATE_VALUES** e scegliamo il sotto **stato inquinante**; a questo punto specifichiamo **Min pari a 0 e Max pari a 0.001**

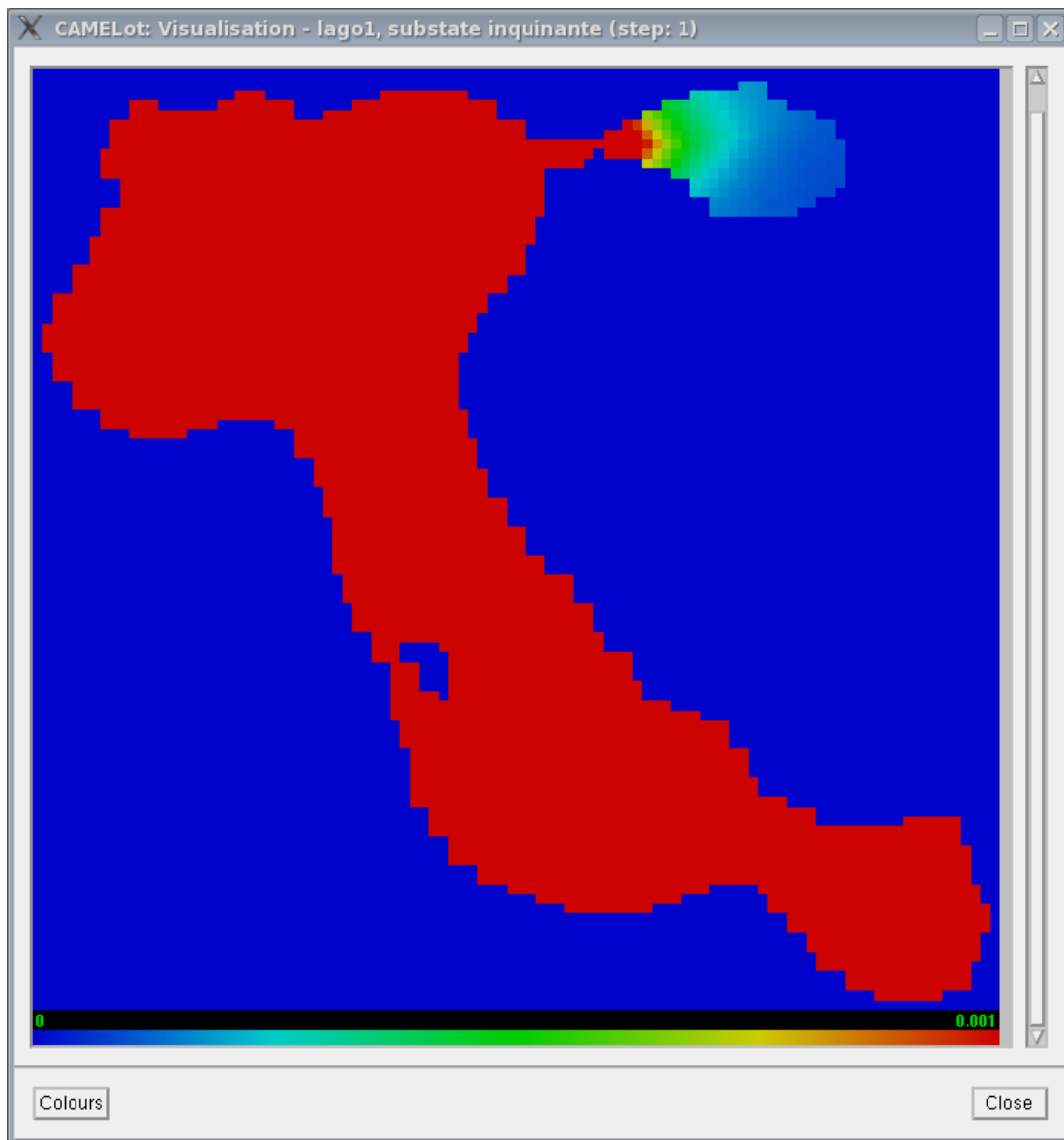
A questo punto siamo pronti per andare a vedere come evolve il nostro automa cellulare! In particolare siamo interessati al “percorso” che fa l'inquinante! Dopo aver chiuso la rappresentazione del lago, scegliamo di nuovo il bottone **Visualise**, scegliamo **1** come passo di visualizzazione ed scegliamo il **sottostato inquinante**; a questo punto lanciamo la simulazione con il tasto **Loop** e stiamo a guardare come evolve.

Se abbiamo scritto bene il programma l'inquinante dovrebbe diffondersi a partire dal basso in tutte le direzioni, fino a stabilizzare il proprio livello. Senza variazioni, l'automa non evolve oltre: il lago è inquinato e così rimane.

vediamo le fasi dell'evoluzione dell'AC...







A questo punto passiamo alle variazioni! In modo da vedere come questo lago di arcavacata intriso di inquinante possa evolvere in modo alternativo.

Variazione 1

Supporre che ad ogni passo parte dell'inquinante pari al 2-5%, specificato da un parametro, si depositi sul "fondo" del lago. Le celle di tipo terra non ricevono inquinante col meccanismo dell'algoritmo di minimizzazione.

- Introduciamo un'ottimizzazione a capo della funzione `calcolaFlusso()` ci assicuriamo che tutto sia fatto solo e soltanto se siamo in ACQUA! Poiché sulla terra non ci sono flussi
- la funzione `calcolaInquinante()` diventa `calcolaInquinanteESuoDeposito()`
- introduciamo i seguenti parametri:
 - `passi_inquinamento` 100.0
 - `q_inquinante` 50.0
 - `rallentamento` 0.7
 - `fattore_deposito` 0.03
- pseudocodice della funzione `calcolaInquinanteESuoDeposito()`
 - definisco una variabile chiamata `nuovo` e di tipo `float`
 - se sono in acqua allora
 - `nuovo` è uguale alla mia quantità di inquinante meno la mia quantità di inquinante moltiplicata per il fattore di deposito
 - aggiorno la mia quantità di inquinante con il valore della variabile chiamata `nuovo`

Il significato di questa variazione è quello di fare "sparire" l'inquinante dal pelo dell'acqua portandolo sul fondo.

Variazione 2 (NOTA: un po' complessa)

Supporre che ad ogni passo una corrente superficiale sposti il 5-15% dell'inquinante (specificato da un parametro) sulla cella vicina a Sud. Le celle di tipo terra non ricevono flusso, né a causa dell'algoritmo di minimizzazione, né a causa della corrente superficiale.

- Introduciamo un'ottimizzazione a capo della funzione `calcolaFlusso()` ci assicuriamo che tutto sia fatto solo e soltanto se siamo in ACQUA! Poiché sulla terra non ci sono flussi
- la funzione `calcolaFlusso()` diventa `calcolaFlussoECorrenteSuperficiale()`
- introduciamo i seguenti parametri:
 - `passi_inquinamento` 100.0
 - `q_inquinante` 50.0
 - `rallentamento` 0.7
 - `fattore_spostamento_sud` 0.1
- pseudocodice della funzione `calcolaFlussoECorrenteSuperficiale()`
 - alla fine della funzione, dentro il controllo per vedere se siamo in acqua
 - calcoliamo l'inquinante residuo nella cella centrale come `cell_inquinante` meno la quantità $(media-qi[i]) * rallentamento$ che si trova in ogni cella non eliminata
 - a questo punto calcoliamo la corrente
 - se a sud abbiamo acqua e non abbiamo eliminato la cella a sud allora
 - aggiorniamo la componente verso sud del vettore flussi con $(media-qi[4]) * rallentamento + inquinante_residuo * fattore_spostamento_sud$
 - se a sud abbiamo acqua e abbiamo eliminato la cella a sud allora
 - aggiorniamo la componente verso sud del vettore flussi con $inquinante_residuo * fattore_spostamento_sud$
 - altrimenti aggiorniamo la componente verso sud del vettore flussi con *0.0*

Variazione 3

Supporre che ad ogni passo le celle di tipo acqua cedano ad ogni cella di tipo terra con cui sono a contatto, una quantità pari al 5-15% di inquinante (specificato da un parametro). Le celle di tipo terra non ricevono inquinante col meccanismo dell'algoritmo di minimizzazione.

- Introduciamo un'ottimizzazione a capo della funzione `calcolaFlusso()` ci assicuriamo che tutto sia fatto solo e soltanto se siamo in ACQUA! Poiché sulla terra non ci sono flussi
- la funzione `calcolaFlusso()` diventa `calcolaFlussoEFlussoVersoTerra()`
- introduciamo i seguenti parametri:
 - `passi_inquinamento` 100.0
 - `q_inquinante` 50.0
 - `rallentamento` 0.7
 - `fattore_spostamento_terra` 0.15
- pseudocodice della funzione `calcolaFlussoEFlussoVersoTerra()`
 - alla fine della funzione, dentro il controllo per vedere se siamo in acqua
 - calcoliamo l'inquinante residuo nella cella centrale come `cell_inquinante` meno la quantità $(media_qi[i]) * rallentamento$ che si trova in ogni cella non eliminata
 - a questo punto calcoliamo il flusso verso terra
 - per tutte le celle del vicinato, se `news[i-1]_lago` è TERRA allora
 - aggiorniamo la componente `[i-1]` del vettore flussi con
*`inquinante_residuo*fattore_spostamento_terra`*

L'effetto visuale di come evolve l'AC con ognuna di queste 3 modifiche è davvero interessante ed imprevedibile, specie se si pensa che tutto nasce esclusivamente da semplici interazioni locali...

vedere per credere...