Risoluzione di un problema di diffusione-trasporto-reazione sul Lago di Garda con Elementi Finiti

Bonazzoli Marcella, Gaburro Elena, Visentin Francesco

Università degli studi di Verona
Seminario di Calcolo Scientifico

27 Giugno 2013

Introduzione

Lo scopo di questo progetto è di simulare la diffusione di una sostanza inquinante all'interno del Lago di Garda in diverse condizioni.

- L' equazione che descrive il fenomeno di diffusione-trasporto-reazione di questa sostanza nel tempo è:

$$\frac{\partial u}{\partial t} = d\,\nabla^2 u + c \cdot \nabla u + \rho\, u\,(1-u)\,(u-0.5)$$

- Le condizioni al bordo e le condizioni iniziali dipendono dalla particolare situazione presa in esame (esempi: sostanza inquinante scaricata in mezzo al lago, sostanza che continua ad affluire in modo costante dal Sarca, sostanza che esce dal Mincio...)

Introduzione

Per risolvere questa equazione abbiamo dovuto occuparci di:

- Manipolare l'equazione: discretizzazione in tempo e calcolo della formulazione debole dell'equazione data dal metodo di Newton
- Costruire una triangolazione del dominio (il lago di Garda)
- Imporre le condizioni al bordo

Abbiamo poi fatto diversi esperimenti:

- Variando le condizioni al bordo e quelle iniziali
- Modificando i parametri $d, c,$ e ρ
- Usando vari solutori di sistemi lineari, più spazi di elementi finiti e diverse tecniche per fare e raffinare la mesh...

L'equazione ... Discretizzazione nel tempo

Per prima cosa discretizziamo l'equazione nel tempo usando il metodo di Eulero Implicito:

$$\frac{u^{n+1} - u^n}{\Delta t} = d\,\nabla^2 u^{n+1} + c \cdot \nabla u^{n+1} + \rho\, u^{n+1}\,(1 - u^{n+1})\,(u^{n+1} - 0.5)$$

moltiplicando per Δt entrambi i membri e portando a sinistra otteniamo

$$u^{n+1} - u^n - \Delta t \left[d\,\nabla^2 u^{n+1} + c \cdot \nabla u^{n+1} + \rho\, u^{n+1}\,(1 - u^{n+1})\,(u^{n+1} - 0.5) \right] = 0$$

quindi per trovare la soluzione al tempo $n+1$, avendo quella al tempo n, sarà necessario trovare gli zeri della funzione

$$F(u) = u - u^n - \Delta t \left[d\,\nabla^2 u + c \cdot \nabla u + \rho\, u\,(1 - u)\,(u - 0.5) \right]$$

L'equazione ... Calcolo della matrice Jacobiana

Per trovare gli zeri di F scegliamo di utilizzare il Metodo di Newton. E' quindi necessario calcolare la matrice Jacobiana $J(u)$. Per migliorare l'efficienza dell'algoritmo calcoliamo direttamente la matrice applicata a un vettore v, cioè $J(u)v$ (usando le derivate di Gateaux):

$$J(u)v = v - 0 - \Delta t \Big[d\,\nabla^2 v + c \cdot \nabla v + \rho\,(1-u)(u-0.5)v - \rho\,u\,(u-0.5)v + \rho\,u\,(1-u)v \Big]$$

e dopo aver raccolto otteniamo

$$J(u)v = v - \Delta t \Big\{ d\,\nabla^2 v + c \cdot \nabla v + \rho \Big[(1-u)(u-0.5) - u\,(u-0.5) + u\,(1-u) \Big] v \Big\}$$

La struttura base dell'algoritmo è la seguente:

1: Sia u^0 la soluzione al tempo zero
2: **for** $n = 0, 1, 2 \ldots$ numero intervalli temporali **do**
3: Algoritmo di Newton per trovare u^{n+1}:
4: $u^r = u^n$ approssimazione iniziale per Newton
5: **while** norma $(\delta^r) >$ tol **do**
6: $J(u^r)\delta^r = -F(u^r)$
7: $u^{r+1} = u^r + \delta^r$
8: **end while**
9: $u^{n+1} = u^r$
10: **end for**

Ora quindi dobbiamo risolvere $J(u^r)\delta^r + F(u^r) = 0$ cioè ...

L'equazione $\ldots J(u^r)\delta^r + F(u^r) = 0$

$$\delta^r - \Delta t \Big\{ d\,\nabla^2\delta^r + c\cdot\nabla\delta^r + \rho\Big[(1-u^r)(u^r-0.5) - u^r(u^r-0.5) + u^r(1-u^r)\Big]\delta^r \Big\}$$
$$+ u^r - u^n - \Delta t\Big[\, d\,\nabla^2 u^r + c\cdot\nabla u^r + \rho u^r(1-u^r)(u^r-0.5)\Big] = 0$$

In questo modo abbiamo ottenuto un'equazione con dipendenza lineare dall'incognita δ_r e quindi possiamo scriverne la formulazione variazionale da passare a FreeFem. Moltiplichiamo perciò entrambi i membri per una funzione test φ e integriamo sul dominio Ω:

$$\int_\Omega \delta^r\varphi - \int_\Omega \Delta t\, d\,\nabla^2\delta^r\varphi - \int_\Omega \Delta t\, c\cdot\nabla\delta^r\varphi -$$
$$\int_\Omega \Delta t\,\rho\Big[(1-u^r)(u^r-0.5) - u^r(u^r-0.5) + u^r(1-u^r)\Big]\delta^r\,\varphi +$$
$$\int_\Omega u^r\varphi - \int_\Omega u^n\varphi - \int_\Omega \Delta t\, d\,\nabla^2 u^r\varphi - \int_\Omega \Delta t\, c\cdot\nabla u^r\varphi +$$
$$\int_\Omega \Delta t\,\rho\, u^r(1-u^r)(u^r-0.5)\,\varphi = 0$$

L'equazione $\ldots J(u^r)\delta^r + F(u^r) = 0$

Ora, applicando la formula di *Green* ai termini in cui compaiono i laplaciani, otteniamo:

$$\int_\Omega \delta^r \varphi + \int_\Omega \Delta t\, d\, \nabla \delta^r \nabla \varphi - \int_{\partial\Omega} \Delta t\, d\, \nabla \delta^r \cdot \hat{n}\, \varphi - \int_\Omega \Delta t\, c \cdot \nabla \delta^r \varphi -$$
$$\int_\Omega \Delta t\, \rho \left[(1-u^r)(u^r - 0.5) - u^r(u^r - 0.5) + u^r(1-u^r) \right] \delta^r \varphi +$$
$$\int_\Omega u^r \varphi - \int_\Omega u^n \varphi + \int_\Omega \Delta t\, d\, \nabla u^r \nabla \varphi - \int_{\partial\Omega} \Delta t\, d\, \nabla u^r \cdot \hat{n}\, \varphi -$$
$$\int_\Omega \Delta t\, c \cdot \nabla u^r \varphi - \int_\Omega \Delta t\, \rho u^r (1-u^r)(u^r - 0.5)\, \varphi = 0$$

Formula di Green:

$$\int_\Omega \nabla^2 u\, v = -\int_\Omega \nabla u \nabla v + \int_{\partial\Omega} \nabla u \cdot \hat{n}\, v \quad \text{con} \quad \nabla u \cdot \hat{n} = \frac{\partial u}{\partial \hat{n}}$$

Triangolazione del Dominio

Ora che siamo riusciti ad avere l'equazione nella forma adatta per farla risolvere a FreeFem++, dobbiamo occuparci del dominio.

Il dominio su cui vogliamo risolvere l'equazione è il Lago di Garda.

Il lavoro che abbiamo fatto per ottenere la triangolazione del dominio è suddiviso in tre parti:

- Creazione di un'immagine in bianco e nero a partire da un'immagine trovata su internet
- Estrazione del bordo
- Costruzione e raffinamento della mesh

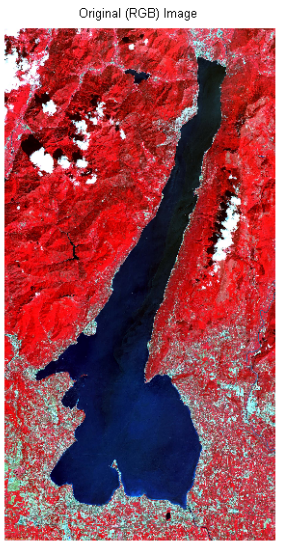
Per le prime due fasi abbiamo utilizzato MatLab e poi abbiamo creato la mesh sia con MatLab che con FreeFem++.

Triangolazione del Dominio

vs immagine in bianco e nero

- Il comando **imread** legge l'immagine a colori (a sx nella figura) e restituisce un array **I** di dimensione *NxMx*3, nel quale è indicata la concentrazione rispettivamente di rosso, verde e blu, in ogni pixel.
- Poi selezionando solo la componente rossa con $\mathbf{I} = \mathbf{I}(:, :, \mathbf{1})$ otteniamo un'immagine in scala di grigi (a dx), con sfumature più chiare dove c'era il rosso e più scure dove c'era il blu.

```
1: I = imread( ...
   ...'lago_sat.jpg');
2: I = I(:, :, 1);
```



Triangolazione del Dominio

vs immagine in bianco e nero

Ora vogliamo ottenere un'immagine in bianco e nero, dove *solo il lago* sia *bianco*.

- Con **BW = im2bw(I, graythresh(I))** mettiamo un 1 dove la sfumatura è più chiara e uno 0 dove è più scura.
  Rimangono però troppi rumori.
- Quindi usiamo **BW = imfill(BW,'holes')**, che controllando le vicinanze di ogni punto trasforma, per esempio, un 1 in 0 se tutti i suoi vicini sono 0.
- Con **BW=** ~**BW** invertiamo gli 0 e gli 1 (per comodità nell'uso delle funzioni successive).

```
1: BW = im2bw(I, graythresh(I));
2: BW = imfill(BW,'holes');
3: BW = ~BW;
```

Triangolazione del Dominio

vs immagine in bianco e nero

Triangolazione del Dominio

vs immagine in bianco e nero

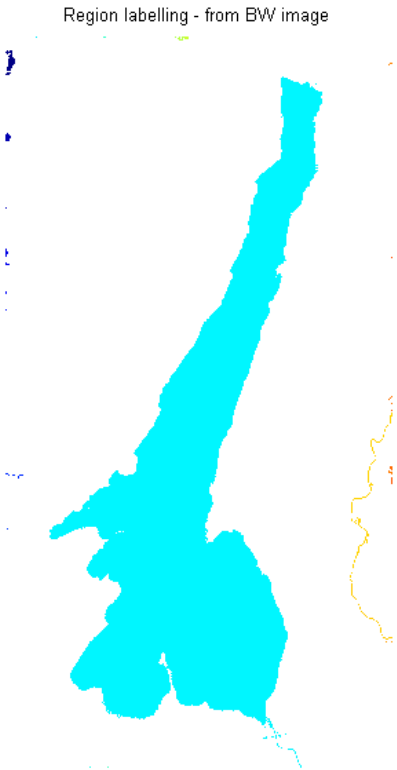
Per eliminare le regioni bianche non connesse con il lago:

- Diamo un'etichetta ad ogni regione bianca (nell'immagine ad ogni etichetta è associato un colore).
- Misuriamo l'area di ogni regione bianca (contando i pixel).
- Se l'area è inferiore a 7900 mettiamo a zero la sua etichetta.

```
1: [L, N] = bwlabel(BW);
2: areaTh = 7900;
3: for k = 1:N do
4:     s = regionprops(L==k,'Area');
5:     if(s.Area < areaTh)
6:     L(L == k) = 0;
7:     end
8: end for
```

Triangolazione del Dominio

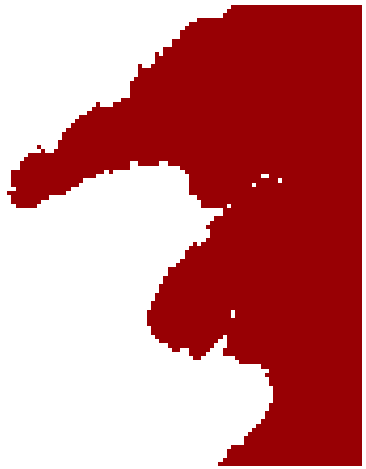
vs immagine in bianco e nero



Triangolazione del Dominio

vs immagine in bianco e nero

Però sono rimasti ancora sia il Sarca che il Mincio e inoltre i bordi non sono ben definiti.



Usiamo quindi:

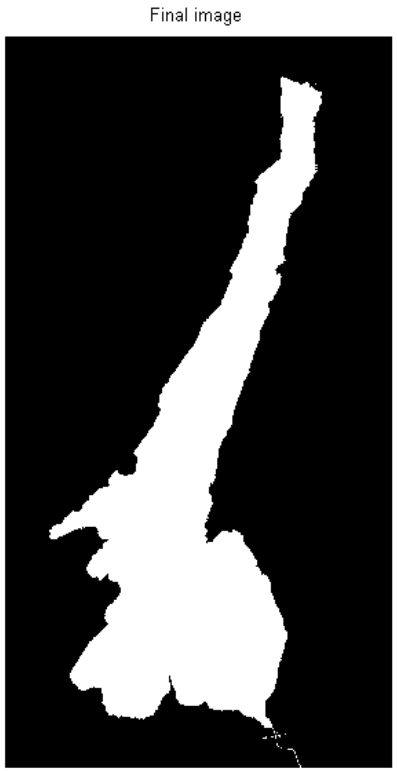
- Di nuovo **BW = imfill(BW,'holes')**
- E **imerode(BW,se)** che erode l'immagine attraverso la struttura circolare **se**.

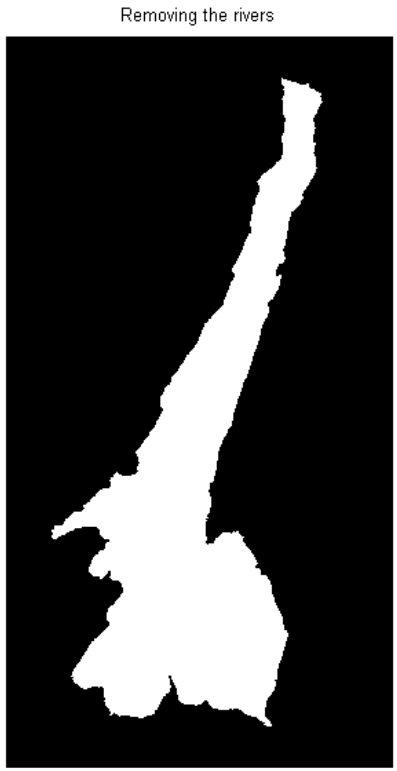
```
1: se = strel('disk',5);
2: BW = imerode(BW,se);
```

Ora abbiamo l'immagine da cui ricavare il bordo del nostro dominio! (quella in mezzo nella figura successiva)

Triangolazione del Dominio

vs immagine in bianco e nero

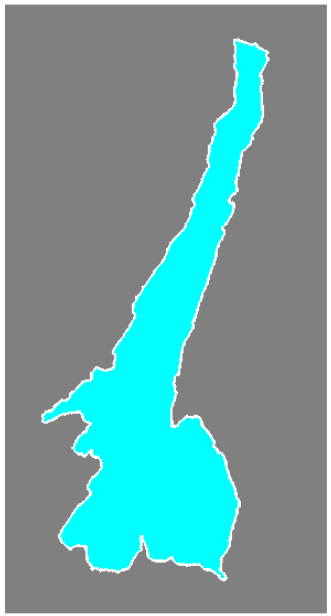




Triangolazione del Dominio

Estrazione del bordo

Ricaviamo i bordi con il comando
[B,L] = bwboundaries(BW,'noholes')
che ci permette di trovare i nodi di bordo e la lista dei lati



Ora per la costruzione e il raffinamento della mesh abbiamo 2 opzioni:

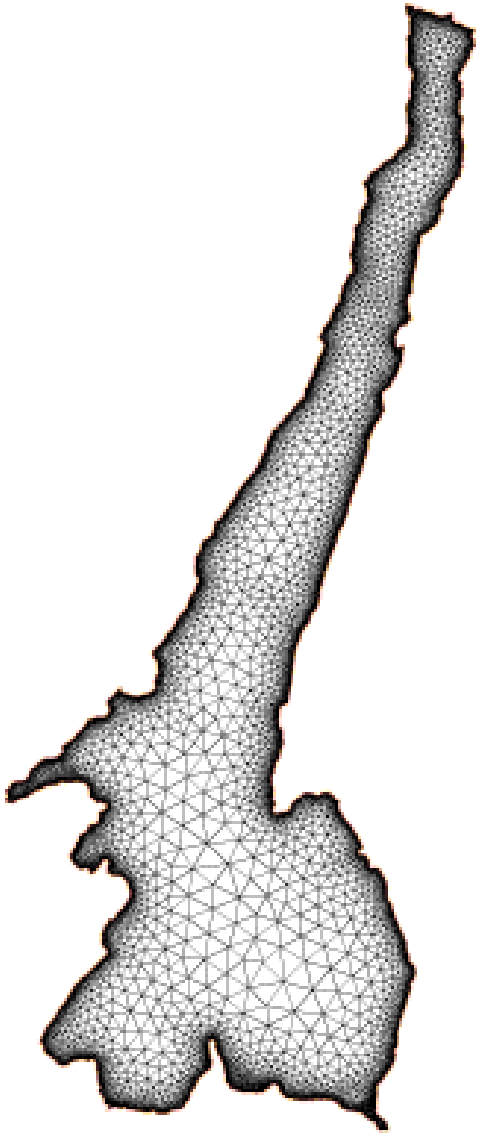
Fare direttamente la mesh in Matlab:

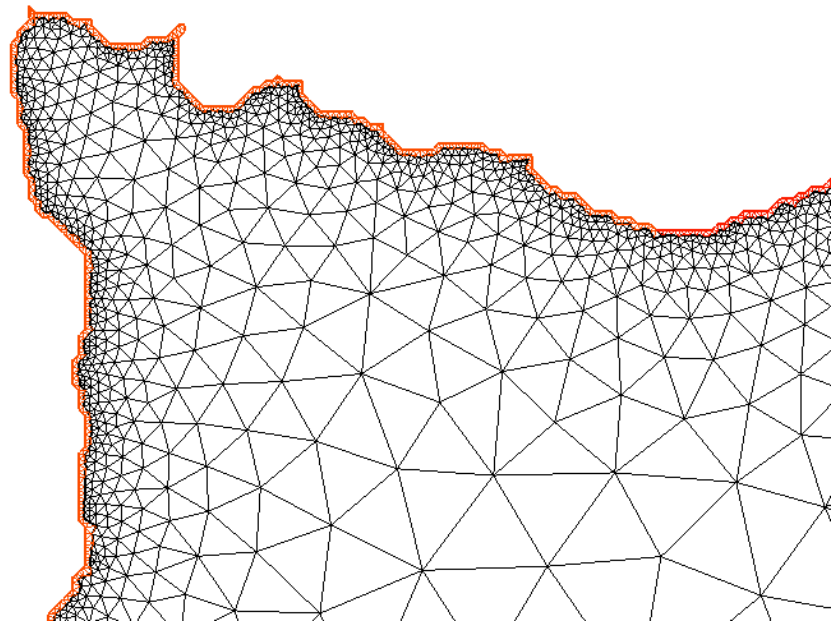
In questo caso si usa il comando **[p,t] = mesh2d(node, edgeLst)** (pacchetto esterno) che restituisce l'elenco dei vertici e dei triangoli da cui è possibile estrarre i dati necessari affinché FreeFem++ debba solo leggere la mesh già fatta (con **readmesh**) e già raffinata.

Questa mesh è molto accurata (i triangoli che la compongono non sono sbilanciati). Essendo così dettagliata il calcolo della soluzione è sia più accurato sia più lento!

Triangolazione del Dominio

Costruzione e Raffinamento della Mesh





Fare la mesh con FreeFem++ e successivamente raffinarla:

- Per prima cosa dobbiamo fornire a FreeFem++ il contorno del lago, ottenuto in Matlab, in forma parametrica.
  Perciò ci calcoliamo le rette passanti per ogni coppia dei nodi del bordo restituiti da MatLab.
- Poi usiamo il comando **border** nomebordo$(t = a, b)(x = x\{t\}, y\{t\})$.
- Infine con i comandi:

```
1: func geom= C1(n) + C2(n) + C3(n)...  ▹ Diciamo quanti punti prendere su ogni parte del contorno
2: mesh Th = buildmesh(geom, ...)       ▹ Costruiamo la mesh
```

Il risultato che si ottiene è il seguente:

Triangolazione del Dominio

Costruzione e Raffinamento della Mesh

Ora quindi è necessario raffinare la mesh, allo lo scopo di:

- Migliorare la forma dei triangoli: in modo che siano sbilanciati il meno possibile.
- Diminuire il numero dei triangoli dove possibile: cioè dove sia il dominio che la soluzione che ci si aspetta di trovare hanno una forma regolare.
- Infittirla ai bordi e dove la soluzione subisce bruschi cambiamenti.

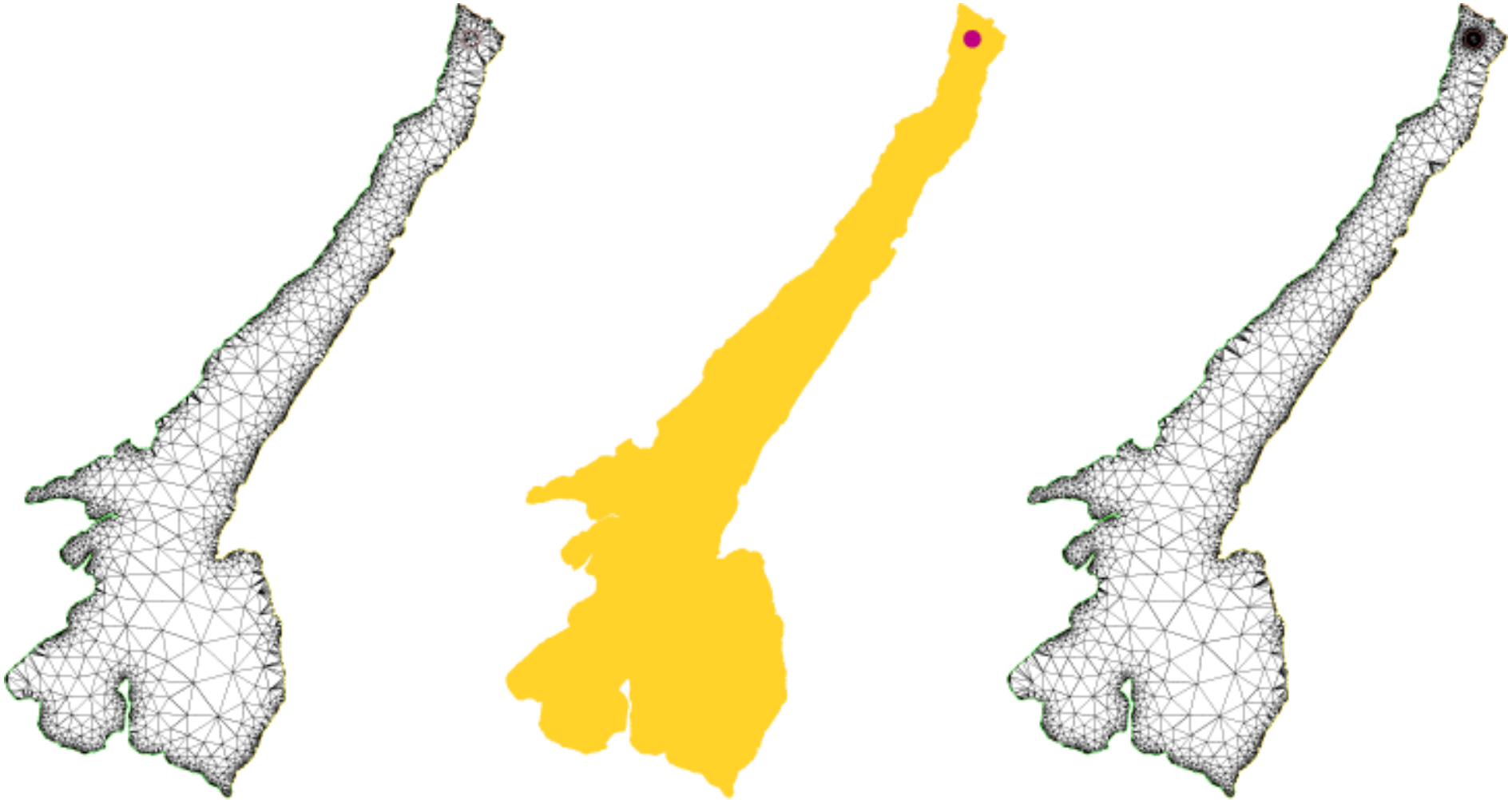
Per migliorare sempre più la mesh abbiamo usato le seguenti procedure ripetendole più volte.

Procedure per raffinare la mesh:

- **Adaptmesh**: è un comando di FreeFem++ che si occupa in modo automatico di raffinare la mesh.
- Abbiamo aggiunto un contorno **circolare all'interno del dominio** in corrispondenza della sostanza inquinante al tempo 0: così otteniamo una miglior definizione del dato iniziale.
- **Adaptmesh(Th, u0)** che adatta la mesh in riferimento alla soluzione iniziale u0 in modo che sia più fitta dove il gradiente della soluzione è più alto.

Triangolazione del Dominio

Costruzione e Raffinamento della Mesh



Condizioni al bordo

Per prima cosa vediamo come vanno imposte, in generale, le condizioni al bordo in FreeFem++, facendo riferimento ad un problema semplice, del tipo $-\nabla^2 u = f$

Condizioni di Neumann:

- Si trova la formulazione debole del problema
- Si sostituisce il valore della derivata normale, dato dalla condizione che si vuole imporre, nell'integrale di bordo che proviene dal laplaciano.

In particolare

- se la condizione è nulla: non si deve scrivere niente
- se la condizione non è nulla (su un certo bordo Γ) si scrive: **int1d(Th, Γ)(valore derivata normale su $\Gamma*$ funzione test φ)**

Condizioni al bordo

Condizioni di Dirichlet

- Si scrive la formulazione debole come se le funzioni test φ fossero nulle ai bordi (quindi senza scrivere gli integrali di bordo provenienti dai laplaciani)
- Si aggiunge (in fondo) alla formulazione del problema: **on(Γ, u = condizione di Dirichlet al bordo)**

Torniamo ora al nostro problema: ovvero alla formulazione debole di $J(u^r)\delta^r + F(u^r) = 0$ nel quale la funzione incognita è δ^r.

Nell'equazione ci sono due integrali di bordo:

$$\int_{\partial\Omega} \Delta t \; d\, \nabla\delta^r \cdot \hat{n}\, \varphi \quad \text{e} \quad \int_{\partial\Omega} \Delta t \; d\, \nabla u^r \cdot \hat{n}\, \varphi$$

Condizioni al bordo

Occupiamoci del **secondo**: $\int_{\partial\Omega} \Delta t \; d \, \nabla u^r \cdot \hat{n} \, \varphi$

- Facciamo subito notare che in questo compare solo u^r che è sempre nota (al passo $r = 0$ è la soluzione iniziale e, ad un generico passo dell'algoritmo di Newton, è la soluzione calcolata al passo precedente).
- Quindi la scrittura di questo integrale non è influenzata dalle condizioni al bordo, ma è semplicemente:
  $\text{int1d}(\text{Th}, \Gamma)(dt * d * (N.x * dx(uR) + N.y * dy(uR)) * phi)$

Condizioni al bordo

Per quanto riguarda il **primo**: $\int_{\partial\Omega} \Delta t \; d \, \nabla \delta^r \cdot \hat{n} \, \varphi$

- Se vogliamo che la soluzione u sia nulla su alcuni bordi, vuol dire che anche $\delta^r = u^{r+1} - u^r$ deve essere nulla su quei bordi: on $(\Gamma, \delta^r = 0)$
- Se vogliamo che la soluzione u sia costante su alcuni bordi, vuol dire che $\delta^r = u^{r+1} - u^r$ deve essere comunque nulla su quei bordi.
- Se vogliamo che la soluzione u abbia derivata normale nulla su alcuni bordi, poiché $\delta^r = u^{r+1} - u^r$, si ha

$$\frac{\partial \delta^r}{\partial \hat{n}} = \frac{\partial u^{r+1}}{\partial \hat{n}} - \frac{\partial u^r}{\partial \hat{n}} \Rightarrow \frac{\partial \delta^r}{\partial \hat{n}} = 0 - 0 = 0.$$

Quindi non dobbiamo scrivere nessun termine.

Ovviamente affinché tutto funzioni la condizione iniziale deve essere coerente con le condizioni imposte al bordo!

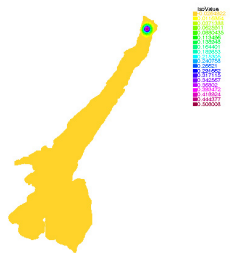
Risultati ottenuti

Sostanza in alto

Vediamo ora quali sono i risultati che abbiamo ottenuto.
Iniziamo col considerare il caso in cui

- La **sostanza è penetrata da poco nel lago**, quindi si trova nella parte superiore
- Ha smesso di entrare: quindi Dirichlet nullo sul Sarca e su tutte le altre parti del bordo (tranne il Mincio)
- La sostanza può uscire in modo uniforme dal Mincio: quindi Neumann nullo sul Mincio

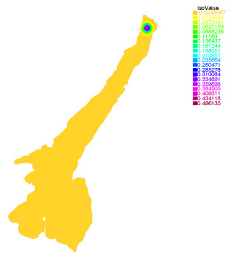
La mesh adattata alle condizioni iniziali è proprio quella vista nelle slide precedenti.
Abbiamo usato lo spazio di elementi finiti $P1$ e come solutore GMRES.
Parametri $d = 10000$, $c = [100, 500]$, $\rho = 0.1$



Risultati ottenuti

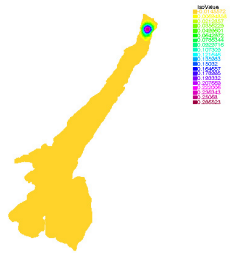
Sostanza in alto

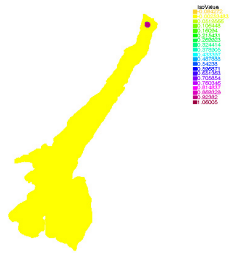
Usando come spazio degli elementi finiti $P2$ non notiamo sostanziali cambiamenti, però i calcoli richiedono un tempo maggiore.



Tornando allo spazio $P1$, proviamo a variare qualche parametro:

- Manteniamo d e ρ come prima e proviamo con $c = [0, 500]$ (la sostanza raggiunge il Mincio senza rimanere nel golfo di Desenzano).
- Oppure con parametri un po' più realistici $d = 10$, $c = [100, 500]$, $\rho = 0.1$. La sostanza un po' si diffonde, infatti il livello più alto scende, ma il fenomeno è molto lento.



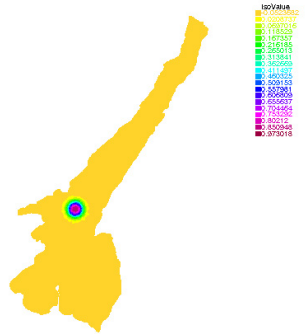


Risultati ottenuti

Sostanza al centro

Consideriamo ora il caso in cui la **sostanza è stata scaricata in mezzo al lago**, quindi mettiamo condizioni di Dirichlet nulle su tutti i bordi tranne sul Mincio, sul quale imponiamo Neumann nullo, in quanto la soluzione può uscire in modo uniforme.

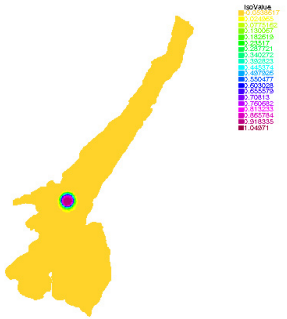
- Con $d = 10000$, $c = [-30, 100]$ e $\rho = 0.1$, siccome il fattore che descrive il trasporto è molto inferiore alla diffusione, la sostanza scende verso il Mincio ma si diffonde molto nel lago.



Risultati ottenuti

Sostanza al centro

- Scegliendo invece $c = [-50, 1000]$, essendo maggiore il fattore che descrive il trasporto, la sostanza invade meno il lago. Si noti l'uso della mesh adattata alla soluzione ad ogni passo (scelta dovuta al fatto che i triangoli della mesh iniziale in questa zona sono molto grandi).



Abbiamo provato a risolvere i vari problemi con diversi solutori, ma il risultato e il tempo impiegato non cambiano.

Risultati ottenuti

La sostanza entra dal Sarca

Supponiamo ora che la sostanza provenga dal Sarca e continui a riversarsi in modo costante nel Garda. Imponiamo quindi:

- Condizioni di Dirichlet nulle sui bordi del lago (fiumi esclusi)
- Condizione di Dirichlet costante sul Sarca (la sostanza entra in modo costante)
- Condizione di Neumann nulla sul Mincio (la sostanza può uscire in modo uniforme)

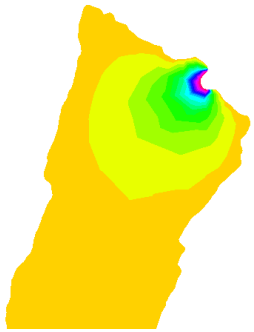
Usiamo come soluzione iniziale la soluzione del seguente problema

$$
\begin{aligned}
-\nabla^2 u &= 0 \\
u &= 1 \quad \text{sul Sarca} \\
u &= 0 \quad \text{sul resto del contorno}
\end{aligned}
$$

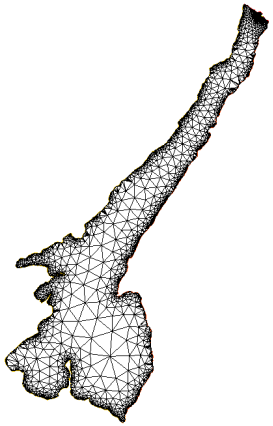
Risultati ottenuti

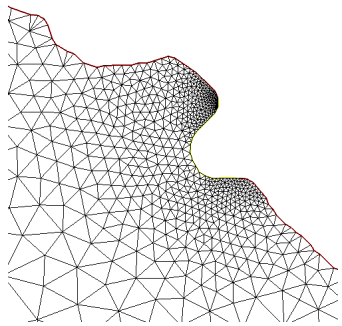
La sostanza entra dal Sarca

Soluzione iniziale



Mesh adattata alla soluzione:





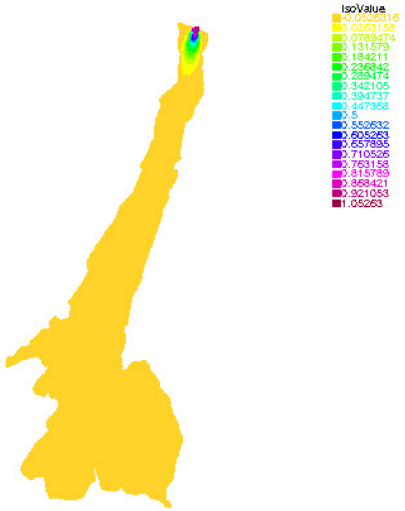
Risultati ottenuti

La sostanza entra dal Sarca

- Scegliamo come parametri:

$$d = 10000,\ c = [800, 4000],$$
$$\rho = 0.1$$

- La soluzione all'inizio si diffonde come previsto.
- Da un certo punto in poi però non si riesce più a notare alcun cambiamento.
- Abbiamo provato con entrambe le mesh e variando molto i parametri iniziali; quello presentato è il miglior risultato.

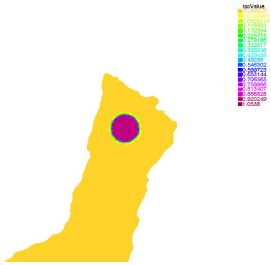


Risultati ottenuti

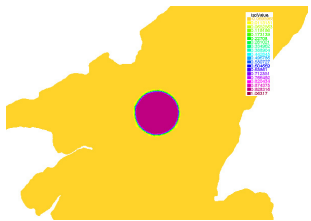
Reazione elevata

Vediamo cosa succede se aumentiamo il parametro ρ :

- Con la sostanza nella parte superiore del lago:
  $d = 100,\ c = [0.01, 0.05],\ \rho = 50$



- Con la sostanza in mezzo al lago:
  $d = 1000,\ c = [-0.01, 0.1],\ \rho = 500$



Conclusione