

# From .png to mesh

A (very) fast guide to mesh a shape in Matlab and FreeFem++

SIMONE PARISOTTO

e-mail: [simone.parisotto@studenti.univr.it](mailto:simone.parisotto@studenti.univr.it)

## 1 Il problema

Si ha la necessità di riconoscere il bordo di una shape, nel nostro caso il Lago di Garda, per caratterizzare un dominio utile in diverse applicazioni a stampo matematico e fisico. Una volta ottenuto il bordo si procederà con il triangolare l'interno. Queste note vogliono proporre un metodo tra i tanti disponibili sul web.

## 2 L'approccio

Poiché il contorno della figura deve essere molto dettagliato, è necessario scaricare da un sito affidabile l'esatta mappa del Lago di Garda, scansionata da un satellite. Il sito scelto è il Geoportale Nazionale (<http://www.pcn.minambiente.it/viewer/>), il cui database è uno dei più completi sul web. Ingrandendo la cartina geografica nella zona prescelta, si sono eseguiti vari screenshot dal monitor (`mela+shift+4` in Mac OS X o `Stamp` in Windows e Linux) e si è scelto di salvare le varie immagini create (15 files) in scala 1:40000.

## 3 Il Merge

Per unire le varie immagini ci si può servire di un potente tool presente all'interno di Adobe Photoshop CS5 (`File -> Automatizza -> Photomerge`) con layout impostato su Automatico. Il risultato di questa operazione è la Figura 1, qui presentata in versione ulteriormente riscalata per una stampa più agevole.

## 4 La Post-Produzione

Sempre in Adobe Photoshop CS5, con lo **Strumento Selezione Rapida**, si può scontornare la figura usando tecniche frequentemente utilizzate nella fotografia digitale e basate sul riconoscimento dei colori: nel fotoritocco, infatti, queste pratiche sono di uso comune e prendono il nome di Post-Produzione (da qui il titolo di questa sezione).



Figura 1: Merge di 15 screenshots in scala 1:40000 (nel file originale).

Lo scopo è ricavare la zona blu dell'immagine, corrispondente all'acqua del lago: una volta che l'immagine è stata scontornata e salvata in un nuovo livello di Photoshop, è consigliato lavorare con lo **Strumento Gomma** attorno alla figura per eliminare le ultime impurità che potrebbero essere rimaste. Eliminata la zona pianeggiante e prealpina circostante, l'immagine è stata salvata tramite il comando **Salva per web e dispositivi**, in qualità **Bicubica** più nitida ed in formato **.png**, ad 8 bit in due colori (bianco e nero) per ottenere un miglior contrasto. L'immagine è stata inoltre salvata in formato quadrato (estendendo la zona trasparente attorno per evitare distorsioni nello script **MATLAB** proposto) con risoluzione 5000x5000 pixels, preservando la scala iniziale di 1:40000. La dimensione del file salvato è pari a circa 61 Kb. Il file immagine è stato chiamato **gardahr.png** ed è mostrato in Figura 2 (qui sempre riscalato per una stampa più agevole).

## 5 I Toolbox di Peyre

Per proseguire con il riconoscimento del bordo della figura si è fatto uso dei pacchetti **toolbox\_general**, **toolbox\_signal** e **toolbox\_graph**, scaricabili gratuitamente dalla pagina dell'autore Gabriel Peyre ([http://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/shapes.1\\_bendinginv.2d/](http://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/shapes.1_bendinginv.2d/)). Una volta installati questi toolbox, è possibile processare in **MATLAB** la figura.



Figura 2: Il Lago di Garda scontornato ed annerito per un miglior contrasto.

## 6 Risultati sul riconoscimento del bordo

La Figura 3, output del codice proposto, è composta da 5000 punti. I tools sviluppati da G. Peyre riescono nell'intento di calcolare le coordinate dei punti sul bordo della shape. In particolare, si vuole porre l'attenzione sul fatto che il bordo viene calcolato tramite interpolazione lineare (impostazione di default di `interp1`). Infine al bordo ricavato sono state applicate delle trasformazioni geometriche (una rotazione e una traslazione) per avere, nel plot, una immagine correttamente orientata rispetto al Polo Nord geografico.

## 7 Triangolare il dominio: MESH2D

I punti appartenenti al bordo della shape servono ora come dato iniziale per triangolare la shape al suo interno: saranno infatti i vincoli oltre i quali non avrà più senso triangolare e vicino ai quali avrà più senso avere una mesh più fine. Il tool proposto si baserà sulle triangolazioni di Delanuay, perché hanno delle buone proprietà computazionali. Si è usato il codice `MATLAB MESH2D - Automatic Mesh Generation`, di Darren Engwirda (scaricabile gratuitamente da <http://www.mathworks.com/matlabcentral/fileexchange/25555-mesh2d-automatic-mesh-generation>). Il tool è semplice da utilizzare ed adatto allo scopo iniziale, in quanto permette di triangolare il dominio, adattativamente, rispetto al bordo. Inoltre, se il dominio non fosse semplicemente connesso, questo tool sarà in grado di riconoscere i buchi e trattarli come un bordo interno, a patto di conoscere certe informazioni. Si invita il lettore

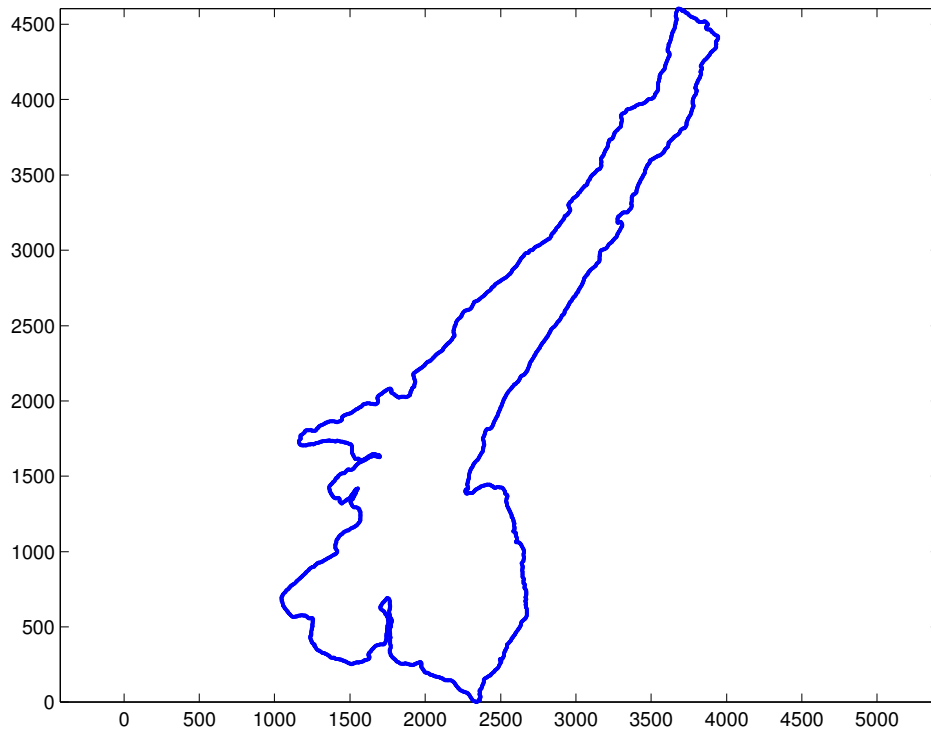


Figura 3: Il contorno del Lago di Garda.

a testare il file `meshdemo.m` e `mesh_collection.m`, soprattutto nell'esempio 16, all'interno del tool, per approfondirne le potenzialità. In alternativa si può guardare con attenzione al paper *A Simple Mesh Generator in MATLAB* di Per-Olof Persson e Gilbert Strang. Si sconsiglia l'uso dei codici di Peyre per questo passaggio in quanto computazionalmente onerosi e poco accurati.

## 8 Risultati sulla triangolazione

Il Lago di Garda triangolato viene presentato in Figura 4, mentre in Figura 5 viene presentato un particolare ingrandito della triangolazione ottenuta.

Il tool proposto riporta anche alcune informazioni utili sulla mesh:

```

Time: 22.2800
Triangles: 45919
Nodes: 25776
Mean_quality: 0.9487
Min_quality: 0.5696

```

In particolare osserviamo che la mesh è una buona mesh in quanto il parametro `Min_quality` è maggiore di 0.5 mentre la qualità media, `Mean_quality`, è molto alta (il valore massimo che potrebbe assumere questo parametro è 1). Infine osserviamo

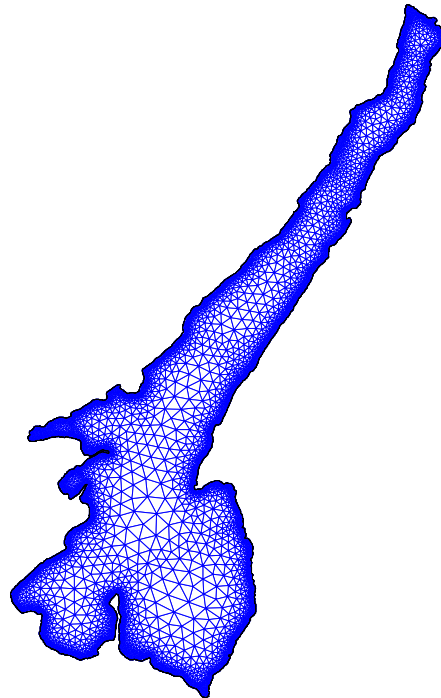


Figura 4: Lago di Garda triangolato.

che i triangoli che presentano una qualità inferiore a 0.7 sono solo 69 su 45919 triangoli calcolati.

## 9 Il codice Matlab utilizzato

Programma 1: GardaLakeBoundary.m

```
1 clear all
2 close all
3 clc
4
5 getd = @(p)path(p,path);
6
7 getd('toolbox.signal/');
8 getd('toolbox.general/');
9 getd('toolbox.graph/');
10
11 %% Points on boundary
12 % A 2D shape is a connected compact planar domain.
13 % It is represented as a black and white image corresponding
14 % to the indicator function of the set.
15 t_boundary = tic;
16
```

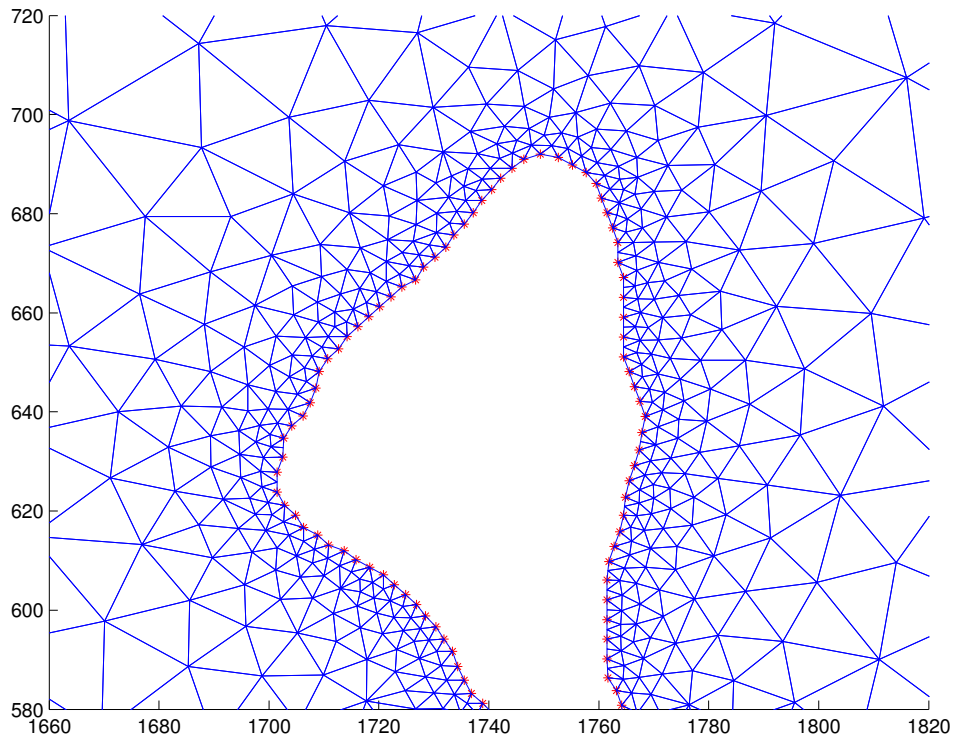


Figura 5: Particolare del Lago di Garda triangolato attorno alla penisola di Sirmione. In rosso si sono voluti evidenziare i punti di bordo.

```

17 n = 5000;
18
19 name = './figure/gardahr';
20
21 % Rescaling data in [0,1] (grayscale)
22 M = rescale( load_image(name,n) );
23
24 % Threshold to be sure to have a binary image.
25 M = perform_blurring(M,3);
26 M = double(M>.5);
27
28 % Compute the boundaries of the shape.
29 bound0 = compute_shape_boundary(M);
30
31 % Re-sample the boundary
32 nbound = 5000;
33
34 nbound0 = size(bound0,2);
35 t = linspace(1,nbound0,nbound+1); t(end) = [];
36 bound(1,:) = interp1( bound0(1,:), t );
37 bound(2,:) = interp1( bound0(2,:), t );
38 bound = bound';
39 t_elapsed_boundary = toc(t_boundary);
40

```

```

41 % Display the boundary.
42 figure;
43
44 % Rotate the boundary to avoid axis('ij')
45 theta = 90 * pi/180;
46 T = [cos(theta) sin(-theta);
47      sin(theta) cos(theta)];
48 b = [bound(:,1),bound(:,2)]*T;
49
50 % Correction to have positive coordinates
51 b(:,2) = b(:,2) - min(b(:,2));
52
53 figure
54 hh = plot(b(:,1), b(:,2), '.');
55 set(hh, 'LineWidth', .5);
56 %axis('ij');
57 axis square;
58 axis equal;
59 print -depsc -tiff -r300 ./figure/gardaboundary5000;
60 save('GardaLakeBoundary5000.mat','b');
61
62 % triangulation
63 [vertices,faces] = mesh2d(b);
64
65 save('GardaLakeVertices.mat','vertices');
66 save('GardaLakeFaces.mat','faces');
67
68 print -depsc -tiff -r300 ./figure/gardatriangulation;
69
70 % zoom
71 figure
72 hold on
73 hh = plot(b(:,1), b(:,2), 'r*');
74 set(hh, 'Markers', 5);
75 triplot(faces,vertices(:,1),vertices(:,2))
76 xlim([1660,1820])
77 ylim([580,720])
78 hold off
79 print -depsc -tiff -r300 ./figure/gardatriangulationzoom;

```

## 10 È possibile passare da Matlab a FreeFem++?

La domanda sorge spontanea: sapendo calcolare certe quantità in MATLAB è poi possibile passare queste variabili a FreeFem++ per eseguire altri tipi di calcoli? La risposta è affermativa seppur il metodo risulti alquanto forzato: bisogna far in modo che MATLAB restituisca un file `.edp` compatibile. Ed è proprio quello che è stato fatto nel Programma 2, dove si è salvato nel file `gardameshff.edp`, il codice generato dai comandi `fprintf`.

Programma 2: `gardamesh_mat2ff.m`

```

1 clear all
2 close all
3 clc

```

```

4
5
6 %% GARDA BOUNDARY
7 load('GardaLakeBoundary5000')
8
9 if exist('gardameshff.edp')
10     delete('gardameshff.edp');
11 end
12
13 fileID = fopen('gardameshff.edp','w');
14
15 % condition for a closing curve
16 b = [b;b(1,:)];
17
18 % findind where i can't divide
19 ix = find(abs(diff(b(:,1)')) < 10e-10);
20 iy = find(abs(diff(b(:,2)')) < 10e-10);
21
22 N = length(b);
23 C= '';
24 for i=1:N-1
25
26     % interval t=[tax,tbx]
27     tax = num2strsign(b(i,1)); tbx = num2strsign(b(i+1,1));
28     % interval t=[tay,tby]
29     tay = num2strsign(b(i,2)); tby = num2strsign(b(i+1,2));
30     % find the line connecting x1 to x2
31     x1 = b(i,1);
32     mx1 = -b(i,1);
33     x2 = b(i+1,1);
34     y1 = b(i,2);
35     my1 = -b(i,2);
36     y2 = b(i+1,2);
37
38     % conditions including "strange lines" (ex. vertical ones)
39     if ~any(i==ix)
40         m(i) = (y2-y1)/(x2-x1);
41         fprintf(fileID, 'border C%d(t=%s,%s){x=t; y=%6.2f*(t%s)%s;}\n', i
42             , tax, tbx, m(i), num2strsign(mx1), num2strsign(y1));
43         if i < N-1
44             is = int2str(i);
45             C = strcat('C', is, '(1)', C);
46         end
47     else
48         m(i) = (x2-x1)/(y2-y1);
49         fprintf(fileID, 'border C%d(t=%s,%s){x=%6.2f*(t%s)%s; y=t;}\n', i
50             , tay, tby, m(i), num2strsign(my1), num2strsign(x1));
51         if i < N-1
52             is = int2str(i);
53             C = strcat('C', is, '(1)', C);
54         end
55     end
56 end
57 is = int2str(N-1);

```



```

58 C = strcat('C',is,'(1)',C);
59
60 nbvx = 50000;
61 plotstr = strcat('plot(',C,'%s,wait=true);\n');
62 fprintf(fileID,'plot(%s,wait=true);\n',C)
63 fprintf(fileID,'mesh Th = buildmesh (%s,nbvx=%d,fixeborder=1);\n',C,
        nbvx);
64 fprintf(fileID,'Th = adaptmesh (Th,keepbackvertices=0,iso=1);\n');
65 fprintf(fileID,'plot(Th,wait=true,ps="gardafreefem.eps");\n')
66 fprintf(fileID,'savemesh(Th,"gardameshff.msh");');

```

## 11 E da FreeFem++ Matlab ?

Ovviamente, anche per questa domanda ci si aspetta una risposta affermativa. Usando il toolbox `FreeFem to Matlab`, scaricabile gratuitamente da <http://www.mathworks.in/matlabcentral/fileexchange/26833?product=PD>, è possibile maneggiare in MATLAB tutti i risultati ottenuti con FreeFem++.

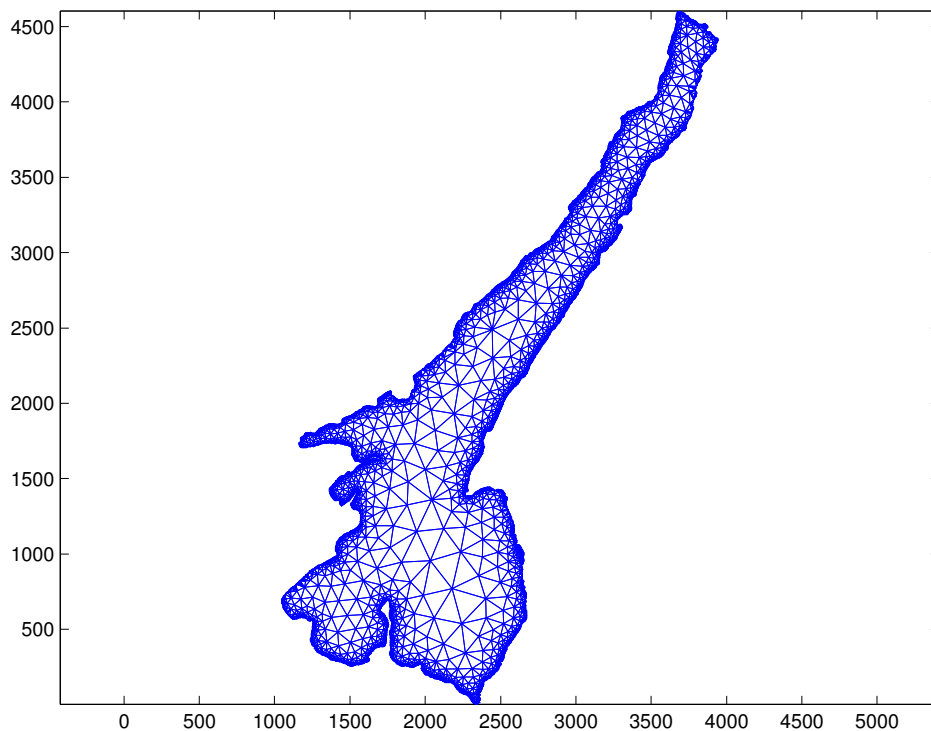


Figura 6: Lago di Garda triangolato in FreeFem++.

## 12 Esperimento e risultati

A questo punto, si è provato a fare una triangolazione in FreeFem++ usando come bordo i punti calcolati nella sezione 6. Il risultato è stato restituito a MATLAB da cui

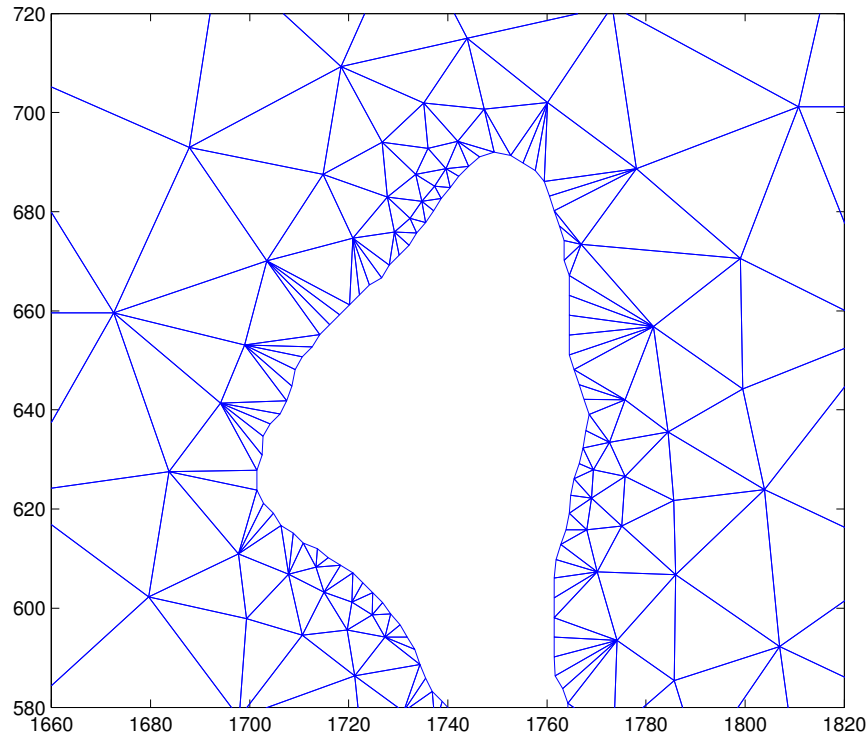


Figura 7: Particolare del Lago di Garda triangolato in FreeFem++ attorno alla penisola di Sirmione.

abbiamo salvato l'output qui proposto. Le Figure 6 e 7 mostrano, rispettivamente, una possibile triangolazione per il dominio dato (si veda il manuale del software per ulteriori approfondimenti) e l'ingrandimento nella zona di Sirmione.

Si riportano per completezza anche i parametri di output calcolati da FreeFem++ e il semplice codice MATLAB utilizzato per importare la mesh (salvata da FreeFem++ con l'estensione `.msh`).

```
-- mesh: Nb of Triangles = 94994, Nb of Vertices 49998
number of required edges : 0
-- adaptmesh Regulary: Nb triangles 11661 ,
  h min 0.296975 , h max 251.706
  area = 3.26355e+06 , M area = 4267.42 ,
  M area/( |Khat| nt) 0.845141
  infiny-regulaty: min 0.00155759 max 6.27906
  anisomax 36.1481, beta max = 106.036 min 0.372861
-- mesh: Nb of Triangles = 11661, Nb of Vertices 8340
number of required edges : 0
times: compile 0.306633s, execution 1.31563s, mpirank:0
```

Programma 3: `example_meshff2mat`

```
1 clear all;
2 close all
3 clc
4
5 [points seg tri]=importfilemesh('gardameshff.msh');
6 triplot(tri(1:3,:),points(1,:),points(2,:));
7 axis square;
8 axis equal;
9 print -depsc -tiff -r300 ../figure/gardamesh_ff;
10
11 xlim([1660,1820])
12 ylim([580,720])
13 print -depsc -tiff -r300 ../figure/gardamesh_ffzoom;
```