

# PDE Toolbox in Matlab

Lucia Gastaldi

Dipartimento di Matematica,  
<http://lucia-gastaldi.unibs.it>

# Indice

- 1 Uso di PDE Toolbox con interfaccia grafica
  - Problema e definizioni
- 2 Esercizi
- 3 Comandi di Matlab per la risoluzione di PDE

# PDE Toolbox e avvio

IL Toolbox `pdetool` è il toolbox di Matlab per la risoluzione di equazioni alle derivate parziali in domini bidimensionali.

I problemi possono essere completamente definiti e risolti usando una interfaccia grafica (GUI) oppure usando solo parzialmente la GUI per specificare alcuni dati come la geometria e le condizioni al bordo.

## Avvio

L'interfaccia grafica si avvia con il comando `pdetool`

## Scelta dell'equazione differenziale

Scegliere nel menù **Options** fra le seguenti applicazioni:

- Generic Scalar (the default mode)
- Generic System
- Structural Mechanics Plane Stress
- Structural Mechanics Plane Strain
- Electrostatics
- Magnetostatics
- AC Power Electromagnetics
- Conductive Media DC
- Heat Transfer
- Diffusion

Nel seguito consideriamo il caso di default.

## Scelta dell'equazione differenziale

Nel menù **PDE** selezionare *PDE Specification* e assegnare i coefficienti e i dati come richiesto dalla forma dell'equazione differenziale prescelta.

I coefficienti possono essere funzioni di  $x$ ,  $y$  e per problemi nonlineari anche di  $u$ ,  $u_x$  e  $u_y$ .

Nelle espressioni usare le variabili  $x$ ,  $y$ ,  $u$ ,  $ux$ ,  $uy$  e tenere conto che verranno calcolate in un certo insieme di punti come vettori, quindi usare la notazione punto. Ad esempio se la funzione è  $xy$ , usare  $x.*y$ .

# Definizione dell'area di lavoro

## Settaggio - **Options**

- *Axes Limits ...*: per visionare qual è l'area che viene mostrata sullo schermo (default:  $x \in [-1.5, 1.5]$  e  $y \in [-1, 1]$ ) e per modificarla se serve.
- *Grid*: appare una griglia.
- *Grid spacing*: per cambiare la spaziatura della griglia (default: 0.5 nella variabile  $x$  e 0.2 nella  $y$ ).
- *Snap*: serve per posizionarsi automaticamente lungo la griglia (utile per disegnare regioni semplici).

## Definizione del dominio

I domini sono costruiti sommando o sottraendo *domini primitivi* come rettangoli, ellissi o poligoni.

### Domini primitivi

- *Rettangolo* Cliccare sul bottone con il + e poi spostare il mouse tenendo premuto il tasto destro. Si forma un rettangolo con centro nel punto dove si è cliccato all'inizio. Se **Snap** è attivo il centro è uno dei punti della griglia e il rettangolo ha lati lungo la griglia. Se si clicca sul bottone senza il +, il punto dove si è cliccato inizialmente coincide con il vertice in alto a sinistra
- *Ellissi* Si procede come per il rettangolo.
- *Poligono* Cliccare sul simbolo del poligono, poi si clicca in un punto dello schermo. Con ogni click ulteriore si forma un lato.

## Salvataggio dati del dominio

Per salvare il dominio per usi successivi, selezionare *Export geometry description* dal menù **Draw** e cliccare su OK nel riquadro. Vengono salvati i dati della geometria `gd`, le formule che descrivono come è ottenuto il dominio `sf`, e la matrice `ns` che mette in relazione le colonne di `gd` con i nomi delle variabili in `sf`.



## Come generare domini composti

Per costruire il dominio dato da un cerchio di raggio 1 a cui è tolta la parte nel primo quadrante si può procedere come segue:

- dare i seguenti comandi

```
pdecirc(0,0,1)  
pderect([0 1 0 1])
```

- Inserire la formula C1-SQ1 nel riquadro in alto **Set formula**

Cliccando sul menù **Boundary** verrà evidenziata la frontiera del dominio.

## Condizioni al bordo

Cliccare sul tasto  $\partial\Omega$  oppure selezionare *Boundary mode* dal menù **Boundary**.

Selezionare:

- un segmento del bordo alla volta cliccando su esso
- alcuni segmenti insieme tenendo premuto il tasto *Shift* e cliccando contemporaneamente sui segmenti desiderati
- tutto il bordo selezionando il tasto *Select All* dal menù **Edit**.

Per inserire le condizioni al bordo sulla parte di frontiera scelta, selezionare il tasto *Specify Boundary Conditions...* dal menù **Boundary**. Si apre un riquadro nel quale si può scegliere condizioni di Dirichlet o di Neumann.

- **Dirichlet:**  $h*u=r$   $h$  è un coefficiente scalare, mentre  $r$  può essere una funzione.
- **Neumann:**  $n*c*\text{grad}(u)+q*u=g$  dove  $n$  è la normale al bordo,  $c$  è il coefficiente legato al termine del secondo ordine dell'equazione alle derivate parziali,  $q$  è un coefficiente scalare,  $g$  può essere una funzione.

**N.B.** Le funzioni si assegnano come espressioni calcolabili.

## Dati per lavoro successivo

Una volta generata la geometria e assegnate le condizioni al bordo, si possono esportare i dati selezionando *Export Decomposed Geometry, Boundary Cond* dal menù **Boundary**. In questo modo si creano due array:

- *g decomposed geometry data*
- *b* dati al bordo.

A volte può essere utile generare la geometria decomposta indipendentemente dai dati al bordo e questo può essere fatto con il comando

```
g = decsg(gd,sf,ns);
```

## Definizione della griglia di calcolo

Nel menù **Mesh** selezionare *Parameters* per scegliere (se si vuole) un valore massimo per i lati degli elementi (il valore **Inf** fornisce la mesh più rada). Per generare la mesh iniziale cliccare sul bottone con il triangolo oppure selezionare *Initalize Mesh* dal menù **Mesh**.

## Salvataggio della griglia di calcolo

Per salvare la mesh in una forma che può essere usata in un programma, selezionare *Export Mesh*. Vengono generati tre array:

- **p** vertici dei triangoli, coordinate dei vertici.
- **e** lati dei triangoli.

Le prime due righe contengono gli indici dei vertici iniziali e finali, la terza e la quarta contengono i valori del parametro iniziale e finale, la quinta contiene la numerazione del lato, la sesta e la settima contengono i numeri dei sottodomini sinistro e destro rispettivamente.

- **t** matrice dei triangoli.

Le prime tre righe contengono la numerazione dei vertici in senso antiorario, mentre la quarta contiene il numero del sottodominio.

Questi dati servono se si vuole confrontare la soluzione discreta con quella esatta e calcolare l'errore in diverse norme.

## Raffinamento della mesh

Cliccare sul bottone con il triangolo suddiviso oppure selezionare *Refine Mesh* dal menù **Mesh**.

La soluzione numerica dell'equazione differenziale può essere ottenuta in maniera adattativa (cioè scegliendo la forma dei triangoli in modo che si adattino alla soluzione) selezionando nel menù **Solve** il tasto *Solve Parameters* e poi *Adaptive Mode*.

## Risoluzione e rappresentazione grafica della soluzione

Selezionare *Solve PDE* dal menù **Solve** oppure il tasto =.  
Per avere a disposizione la soluzione nel workspace, selezionare *Export Solution*.

Sono disponibili diversi tipi di grafici. Selezionare *Parameters* dal menù **Plot** oppure il penultimo tasto con la superficie.

# Esercizio 1

$$\begin{cases} -\Delta u = 2x(1-x) + 2y(1-y) & \text{per } (x, y) \in \Omega = [0, 1] \times [0, 1] \\ u = 0 & \text{su } \partial\Omega \end{cases}$$

La soluzione è:  $u(x, y) = x(1-x)y(1-y)$ .

Calcolare l'errore nella norma del massimo usando 3 diversi livelli di raffinamento.

Per calcolare l'errore occorre determinare il valore della soluzione nei punti della mesh. Usare il vettore  $\mathbf{p}$  che fornisce le coordinate dei punti.



## Esercizio 2

$$-\Delta u + (x^2 + y^2)u = 1$$

$$\text{per } (x, y) \in \Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{su } \partial\Omega$$

## Esercizio 3

$$\begin{cases} -\varepsilon \Delta u + u = 1 & \text{per } (x, y) \in \Omega = [-1, 1] \times [-1, 1] \\ u = 0 & \text{su } \partial\Omega \end{cases}$$

Per  $\varepsilon = 0.1, 0.01, 0.001$  calcolare la soluzione con diversi livelli di raffinamento uniforme della mesh.

Usare l'opzione *Adaptive Mode* e confrontare le diverse soluzioni.

## Esercizio 4

$$\begin{cases} -\Delta u = 1 & \text{per } (x, y) \in \Omega \\ u = 0 & \text{su } \partial\Omega \end{cases}$$

essendo  $\Omega$  il dominio a forma di L

$$\Omega = [-1, 1] \times [0, 1] \cup [0, 1] \times [-1, 0].$$

Usare l'opzione *Adaptive Mode* e confrontare le diverse soluzioni.

# Come risolvere un problema differenziale usando i comandi

## Due modi

- Risolvere il problema con la programmazione a oggetti.  
Le istruzioni si trovano nella documentazione seguendo il percorso:  
Partial Differential Equation Toolbox  
PDE Problem Setup  
PDE Problem Setup Basic  
Solve Problem Using PDEModel Objects
- Risolvere il problema mediante i comandi di Matlab  
Le istruzioni si trovano nella documentazione seguendo il percorso:  
Partial Differential Equation Toolbox  
PDE Problem Setup  
PDE Problem Setup Basic  
Solve PDES Programmatically

## Salvare i dati in un file

Per salvare i dati della geometria e delle condizioni al bordo per usarli in un secondo tempo si procede come segue:

- Esportare la geometria decomposta e le condizioni al bordo selezionando *Export Decomposed Geometry, Boundary Conds* dal menù **Boundary**
- Salvare la geometria in un file che si chiama *prob1g.m* con il comando `fid = wgeom(g, 'prob1g');`
- Salvare i dati relativi alle condizioni al bordo in un file che si chiama *prob1b.m* con il comando `fid = wbound(b, 'prob1b');`

## Risoluzione usando i comandi invece dell'interfaccia grafica

Si vuole risolvere il problema:

$$-\operatorname{div}(c \mathbf{grad} u) + au = f$$

```
[p,e,t] = initmesh('prob1g');    inizializza la griglia  
[p,e,t] = refinemesh('prob1g',p,e,t);    raffina la griglia  
c= '1'; a= '0';    coefficienti equazione differenziale  
f = '8 - 16.*(x.^2 + y.^2)';    funzione data  
u = assempde('prob1b',p,e,t,c,a,f);    fornisce la  
soluzione  
pdesurf(p,t,u)    per fare il grafico della soluzione.
```

## Comandi utili - initmesh

```
[p,e,t] = initmesh(g) oppure [p,e,t]=initmesh('problg')
[p,e,t] = initmesh(g,'PropertyName',PropertyValue,...)
```

Variabile	default	descrizione
Hmax	numerico	Maximum edge size
Hgrad	numerico, tra 1 e 2 (1.3)	Mesh growth rate
Init	'on','off' ('off')	Edge triangulation
Jiggle	'off', 'mean', 'minimum', 'on'	Muove i nodi per migliorare la qualità della mesh con il metodo specificato.
JiggleIter	numeric (10)	Maximum iterations

## Comandi utili - assemblaggio

`[K,M,F]=assema(p,t,c,a,f);`

assembla la matrice di stiffness  $K$ , la matrice di massa  $M$  e il termine noto  $F$ .

`[Q,G,H,R]=assemb('squareb',p,e);`

assembla le matrici  $Q$  e  $H$  e i vettori  $G$  e  $R$

$Q$  deve essere sommata alla matrice del sistema algebrico e contiene i contributi derivanti dalle condizioni al bordo di tipo misto.

$G$  va sommato al termine noto e contiene il contributo per le condizioni di Neumann e di tipo misto.

$H*u=R$  rappresenta le condizioni di Dirichlet.

`[K,F1,B1,UD]=asempde(b,p,e,t,c,a,f)` assembla le matrici del problema eliminando le variabili corrispondenti alle condizioni di Dirichlet dal sistema.

$UN=K \setminus F1$  fornisce la soluzione nei punti che non sono di Dirichlet.

La soluzione del problema si ottiene con il comando  $U=B1*UN+UD$ .



## Comandi utili - risoluzione e visualizzazione

`u = assempde(b,p,e,t,c,a,f);` oppure

`u = assempde('prob1b',p,e,t,c,a,f);`

assembla le matrici e il termine noto e risolve il problema

$-\text{div}(c * \text{grad}(u)) + a * u = f$ , sulla mesh descritta da `p`, `e`, `t`,

con le condizioni al bordo date da `b` o contenute nel file `prob1b.m`.

`[u,p,e,t] = adaptmesh(g,b,c,a,f)` oppure

`[u,p,e,t]=adaptmesh(g,b,c,a,f,'PropertyName',PropertyValue,)`

risolve il problema generando le mesh in modo adattativo.

`pdeplot(p,e,t)` plotta la mesh descritta da `p`, `e`, `t`.

`pdesurf(p,t,u)` plotta la superficie in 3-D surface corrispondente alla soluzione `u`.

`pdeplot(p,e,t,'xydata',u,'zdata',u,'mesh','on','colormap','jet')`

plotta la soluzione `u` come superficie insieme alla mesh