ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Implementation of a Distributed System for the Solution of MultiDomain / MultiPhysics Problems

Ανάπτυξη Κατανεμημένου Συστήματος για Επίλυση Προβλημάτων Πολλαπλών - Χωρίων / Πολλαπλών - Φυσικών Μοντέλων

Διπλωματική Εργασία

Κωνσταντίνος Κ. Χαλκιάς

Επιβλέποντες καθηγητές: Παναγιώτα Τσομπανοπούλου
Επίκουρη Καθηγήτρια Π.Θ.

Εμμανουήλ Βάβαλης
Καθηγητής Π.Θ.

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Βόλος, Ιούλιος 2013

...a few words

- The modeling and simulation of complex physical systems often involves many components because
  1. the physical system itself has components of differing natures,
  2. parallel computing strategies require many (somewhat independent) components, and
  3. existing simulation software implies only to simpler geometrical shapes and physical situations.

- We propose ***IRTool***, a simulation environment which
  1. solves MultiDomain/MultiPhysics problems using relaxation methods
  2. has wide application,
  3. has increased flexibility,
  4. has high performance,
  5. offers parallelism and
  6. reuses existing software

Contents

- Computational approaches simulating large scientific problems
- Related Works
- Domain Decomposition
- Interface Relaxation (IR)
- GEO Method
- IRToolBox Implementation
- IRToolBox GUI
- User Guide / Numerical Experiments
- Conclusions

Computational approaches simulating large scientific problems

- *Simple Domain Decomposition*
  1. discretization of the geometrical domain using grids or meshes to create a large discrete problem
  2. grids or meshes are then partitioned to create a set of inter-connected discrete problems
  3. solves a boundary value problem by splitting it into smaller boundary value problems on subdomains
  4. iterates to coordinate the solution between adjacent subdomains.

+ problems on the subdomains are independent
+ suitable for parallel computing
+ Overlapping / non overlapping subdomains

- one or few unknowns per subdomain
- typically used as pre-conditioners

Computational approaches simulating large scientific problems

- *Schwarz Splitting*
  1. decomposes the geometrical domain into components with small overlap
  2. each component can then be solved independently
  3. the Schwarz alternating method is applied iteratively to compute the global solution
  4. iterates to coordinate the solution between adjacent subdomains.

+ problems on the subdomains are independent
+ suitable for parallel computing

- overlapping creates a serious complication in the Schwarz method
- Discovery of non overlapping methods

Computational approaches simulating large scientific problems

- *Interface Relaxation*

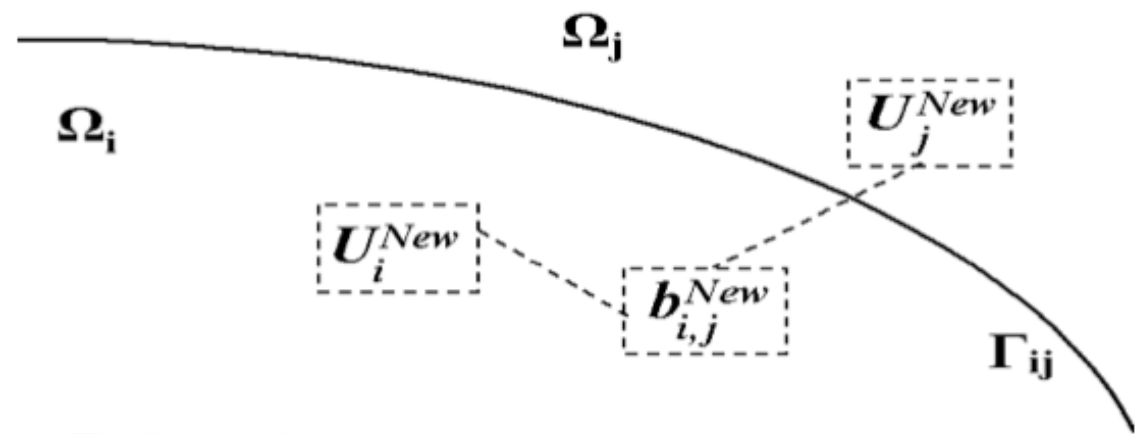
- A complex physical phenomenon consists of a collection of simple parts with each one of them obeying a single physical law locally and adjusting its interface conditions with neighbors.

  ~ *Southwell 1930*

- IR Procedure
  1. Guess solution values (and derivatives if needed) on all subdomain interfaces
  2. Solve all single PDEs exactly and independently on all the subdomain with these values as boundary conditions.
  3. Compare and improve the values on all interfaces using a relaxer
  4. Return to Step 2 until satisfactory accuracy is achieved.

+ Problems on the subdomains are independent
+ Suitable for parallel computing
+ Maximum Generality
+ Flexibility

IR Mechanism



Relaxation:

$$G_{ij}\left(U_i^{New}, U_j^{New}, \frac{\partial U_i^{New}}{\partial \eta}, \frac{\partial U_j^{New}}{\partial \eta} \right) = 0$$

<u>Differential Problem</u>

Du = f in Ω, Bu = c on δΩ (3.1),

D: elliptic, non linear differential operator

B : Condition operator on δΩ

*1.*This domain is partitioned into p open subdomains Ωi ,i=1,...,p such that $\Omega = \cup_{i=1}^{p} \overline{\Omega_i} \setminus \partial\Omega$ and $\cap_{i=1}^{p} \Omega_i = \emptyset.$

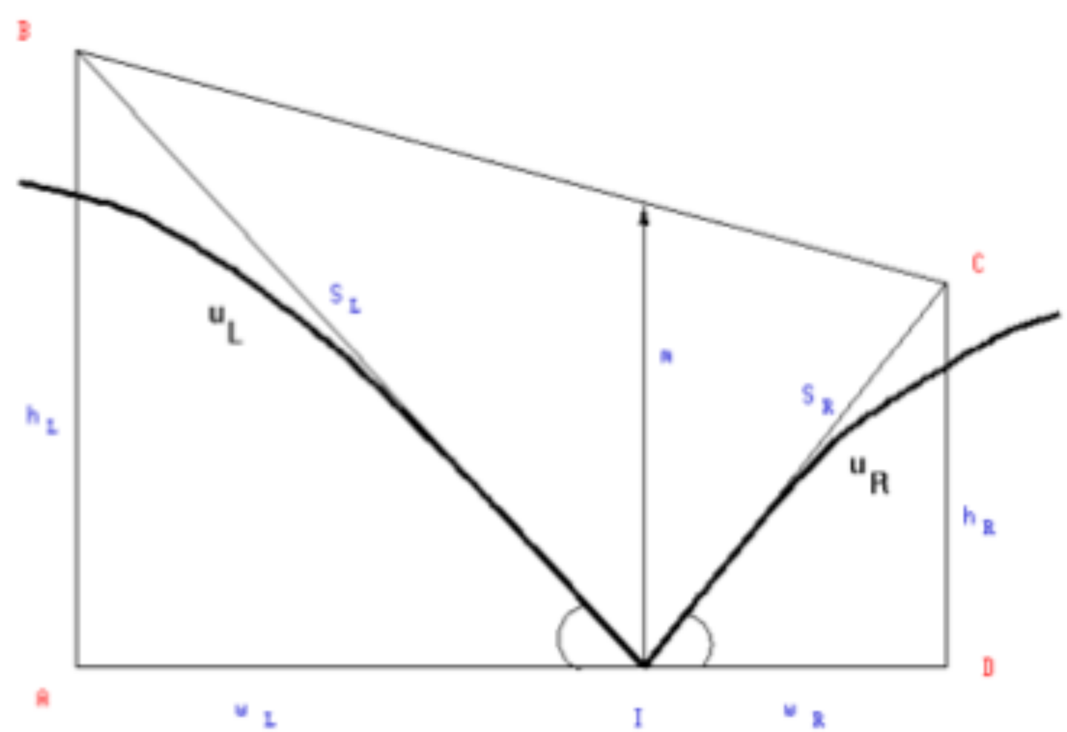
2. Replace (3.1) with

$$D_i u = f_i \; in \; \Omega_i, \qquad (3.2)$$
$$G_{ij} u = 0 \; on \; (\partial\Omega_i \cap \partial\Omega_j) \setminus \partial\Omega \quad \forall j \neq i, \qquad B_i u = c_i \; on \; \partial\Omega_i \cap \partial\Omega$$

GEO Method

- for k = 0,1,2, . . .
  - $ui^{(k+1)} = \frac{u_L^{(k)}+u_R^{(k)}}{2} - \frac{w_L w_R}{w_L+w_R}(\frac{\partial u_L^{(k)}}{\partial x} - \frac{\partial u_R^{(k)}}{\partial x})$ on each interface
  - $u^{(k+1)} =$ solve_pde$(ui^{(k+1)})$ in each subdomain

- One step method
- U: solution
- S: slope
- m: needed correction
- w: width
- One can intuitively view this as grabbing the function U at the I and stretching it up by m until its derivative becomes continuous



Related Work

- SciAgent
  - Agent Based Framework
  - C & Java
  - Parallelism of the Interface Relaxation Methods on heterogenous workstations

IRToolBox Implementation

- Matlab's PDETool as "backbone".
  - \+ Flexible for the study and solution of PDE's.
  - \+ Faces multi-domain problems at the linear algebra level of study
  - \+ Open Source code for further development
  - \- Unable to solve multiDomain / multiPhysics problems with IR Methods
  - \- Interfaces cannot be treated properly and cannot be treated by other methods
  - \- Unable to treat each subdomain separately

IRToolBox Implementation

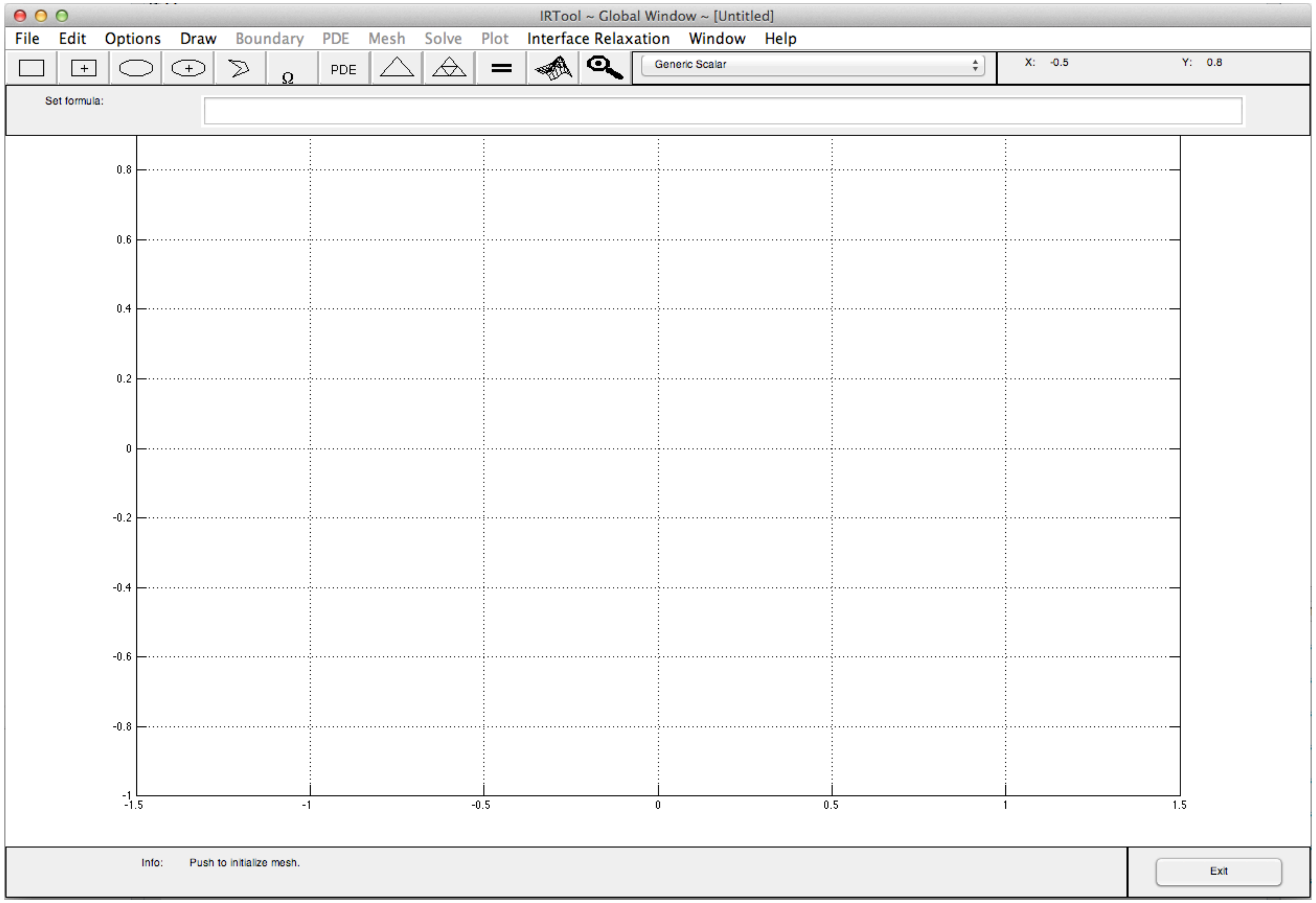
- IRTool
  - Uses existing of wide use scientific software
  - Deals with the interface segments
  - Treats each subdomain independently
  - Solves the global problem by contemplating parameters as
    - interface relaxation methods
    - initial guesses on the interface segments
    - tolerance (convergence).
    - all the parameters for the PDE problems

IRTool GUI

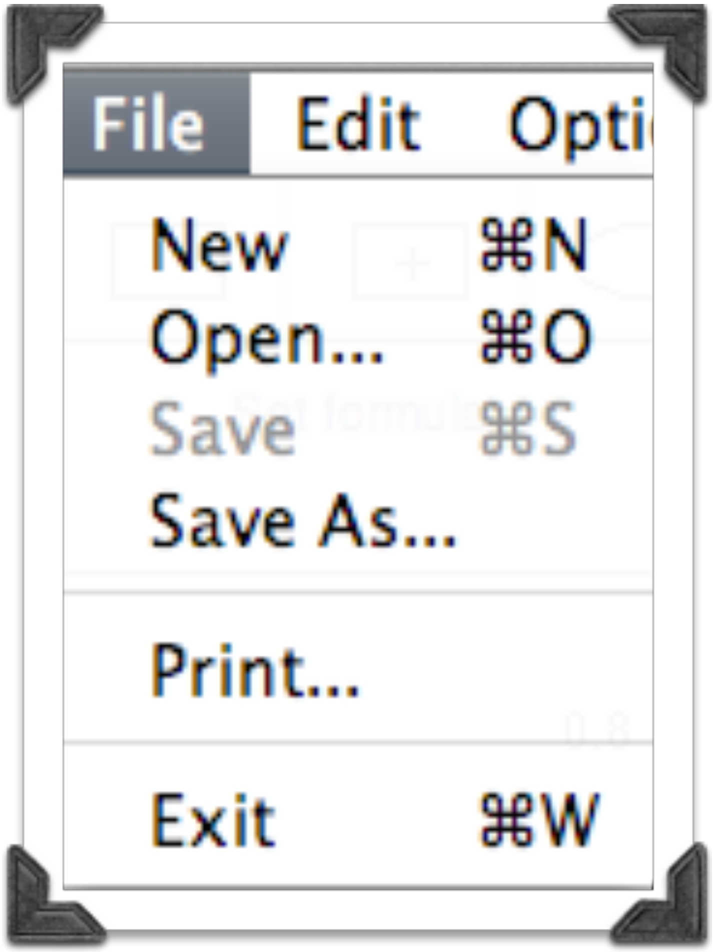
- Conforms to common pull-down menu standards
- Menu items followed by a right arrow lead to a submenu
- Menu items followed by an ellipsis lead to a dialog box
- Stand-alone menu items lead to direct action
- Icon buttons for quick and easy access
- Keyboard accelerators

Global Window

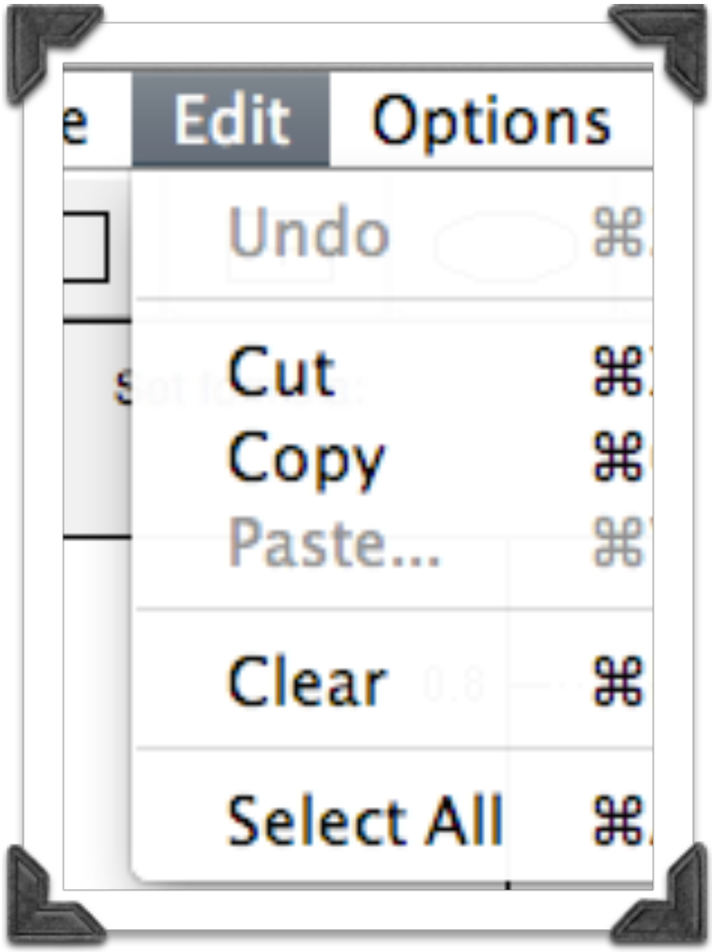
- Global Geometry is drawn



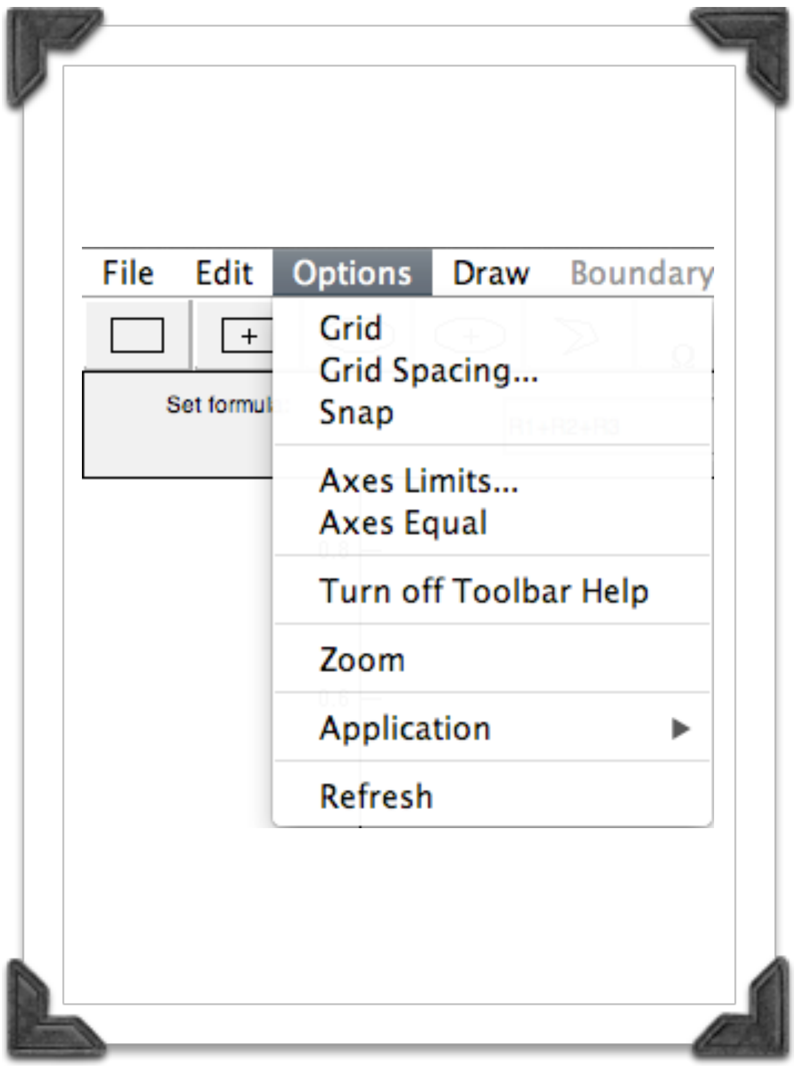
File Menu



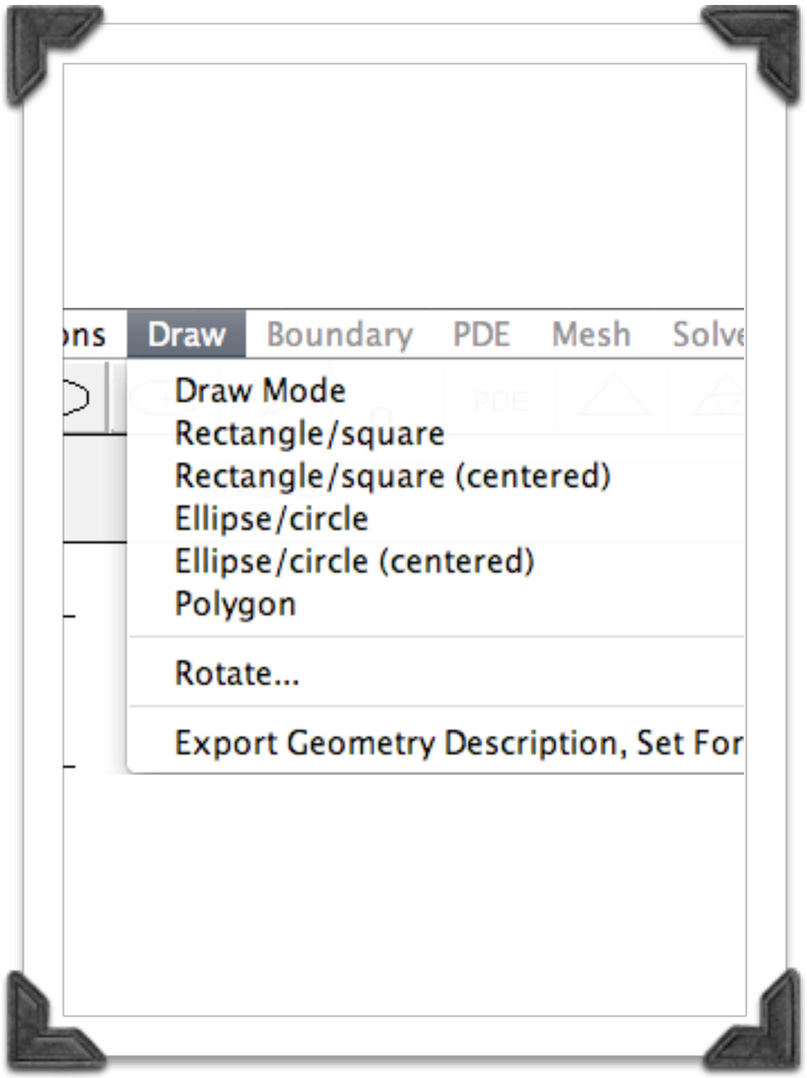
Edit Menu



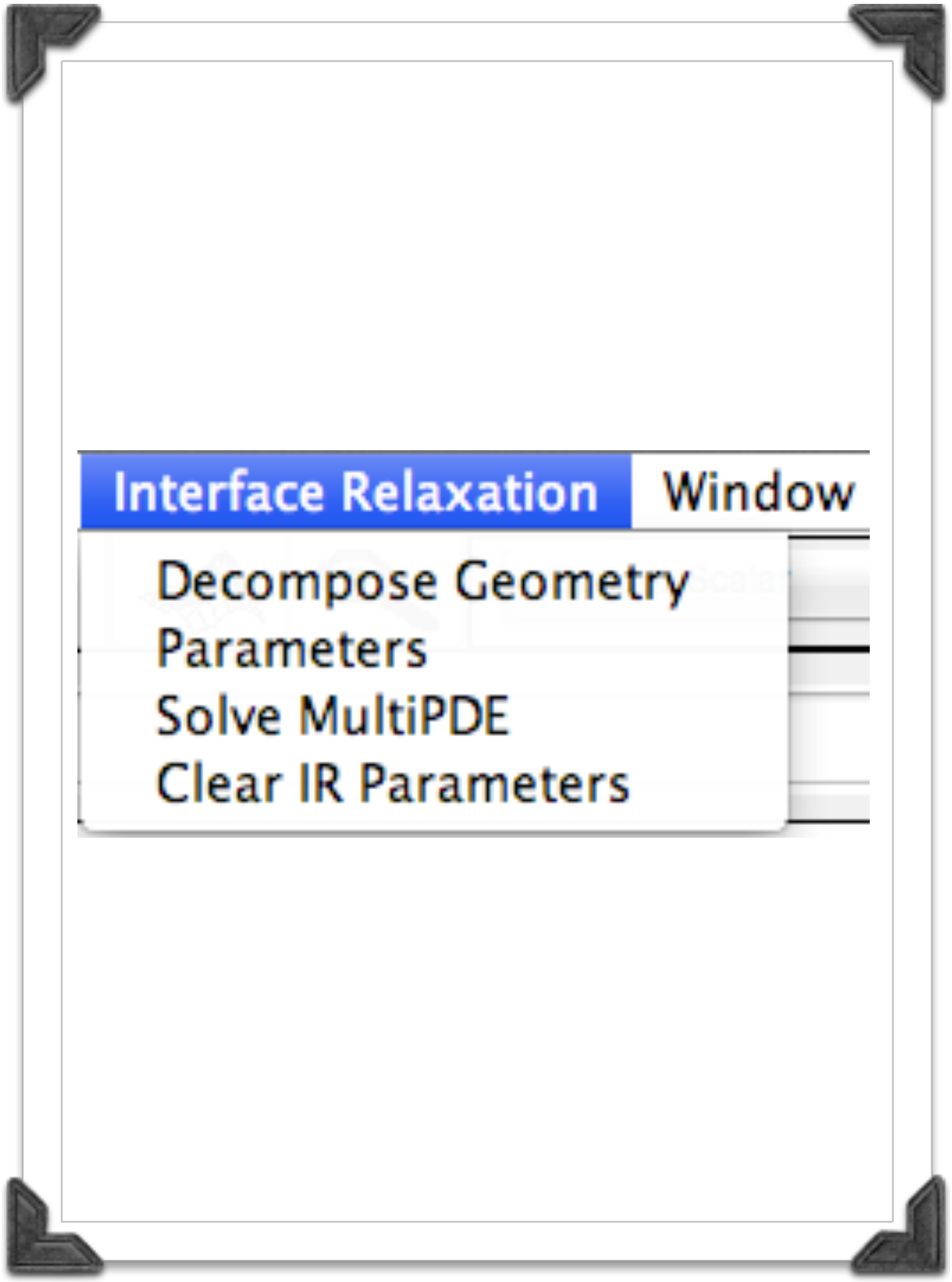
Options Menu



Draw Menu

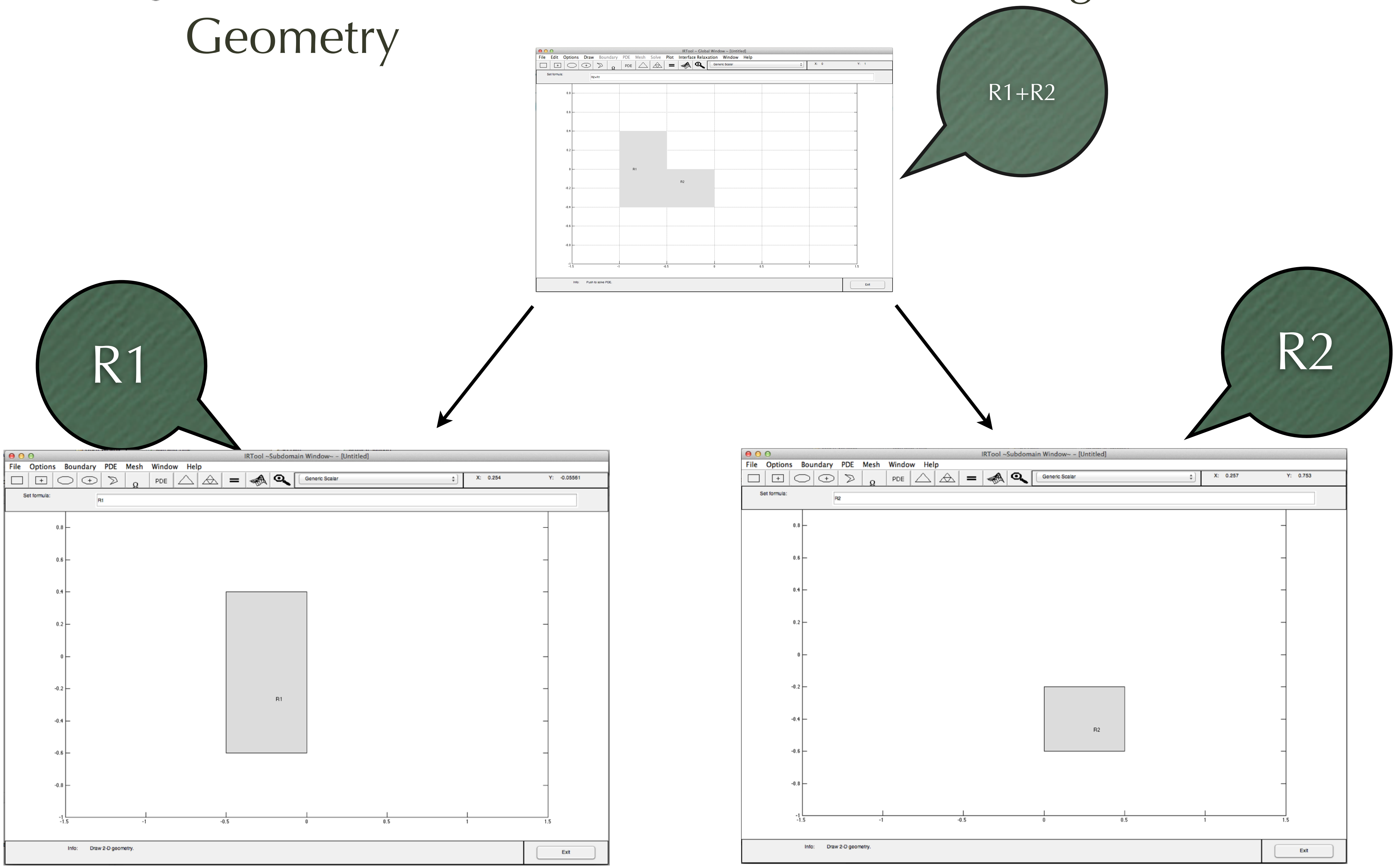


Interface Relaxation Menu

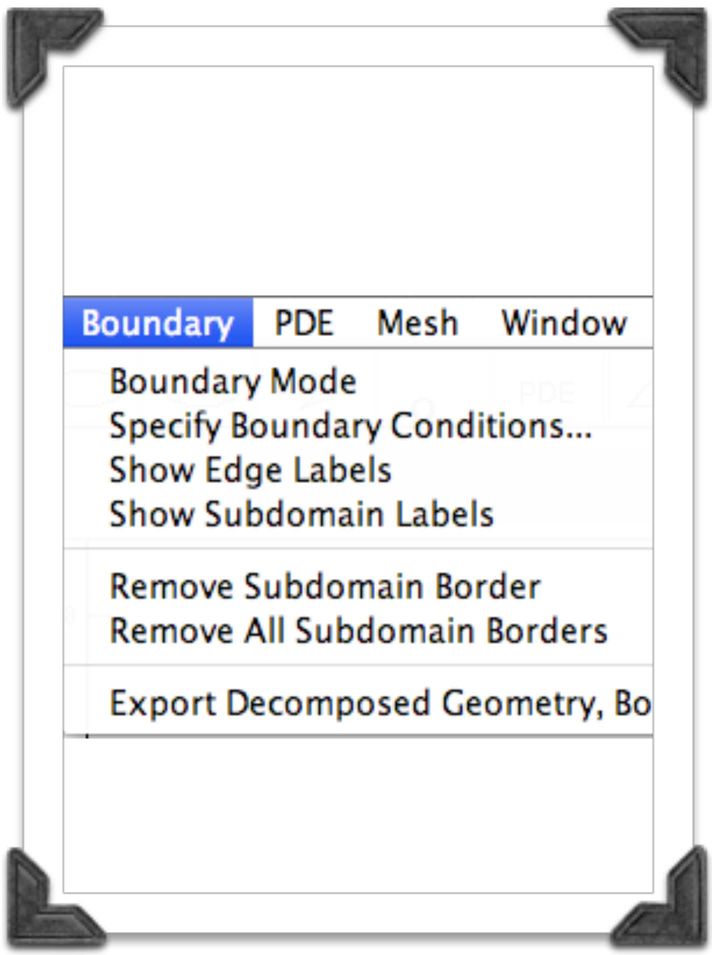


Subdomain Window

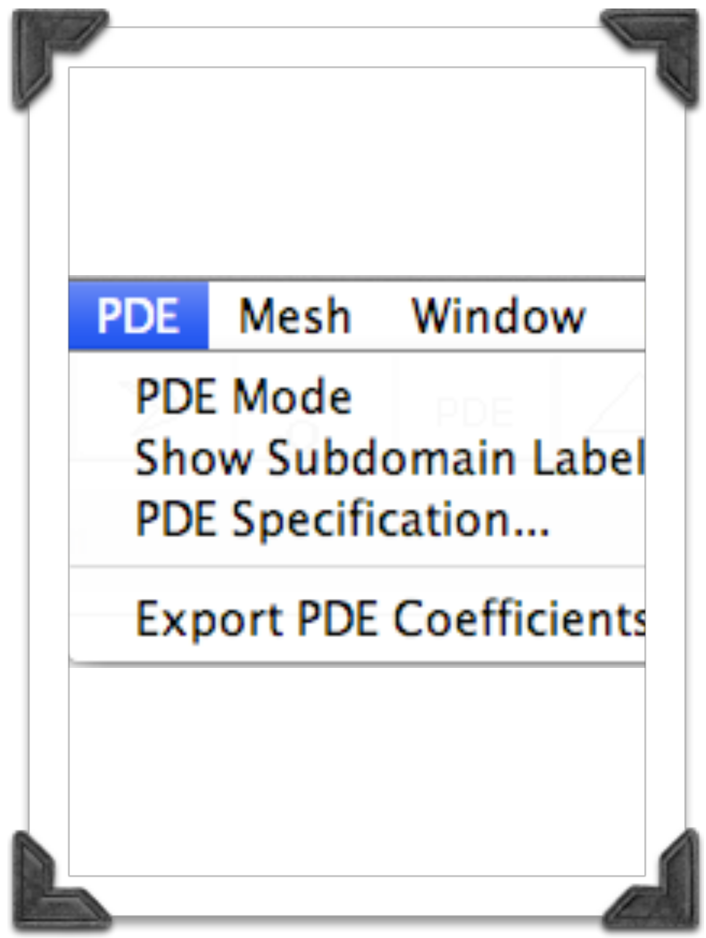
- Each one contains the subdomain of the global Geometry



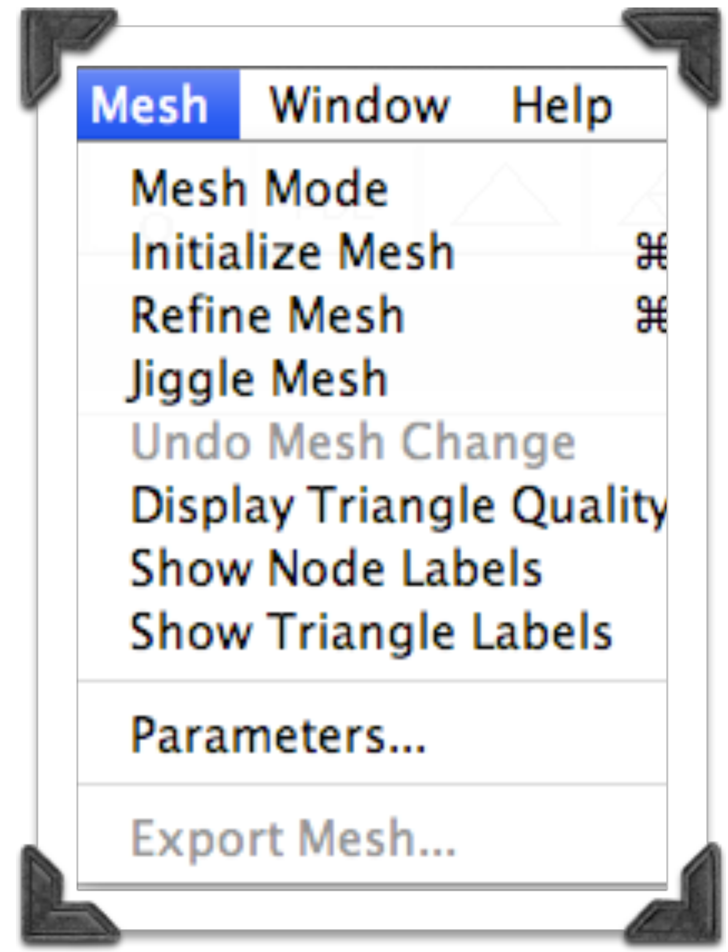
Boundary Menu



PDE Menu



Mesh Menu



User Guide / Numerical Experiments

Uniform Problem

- **Elliptic problem**

$$Lu(x,y) \equiv -\nabla^2 u(x,y) + \gamma^2 u(x,y) = f(x,y),\ (x,y) \in \Omega$$
$$u(x, y) = u^b(x, y),\ (x, y) \in \partial\Omega,$$

- The true solution *u(x,y)* is

$$u(x,y) = e^{y(x+4)} x(x-1)(x-0.9)y(y-0.5).$$

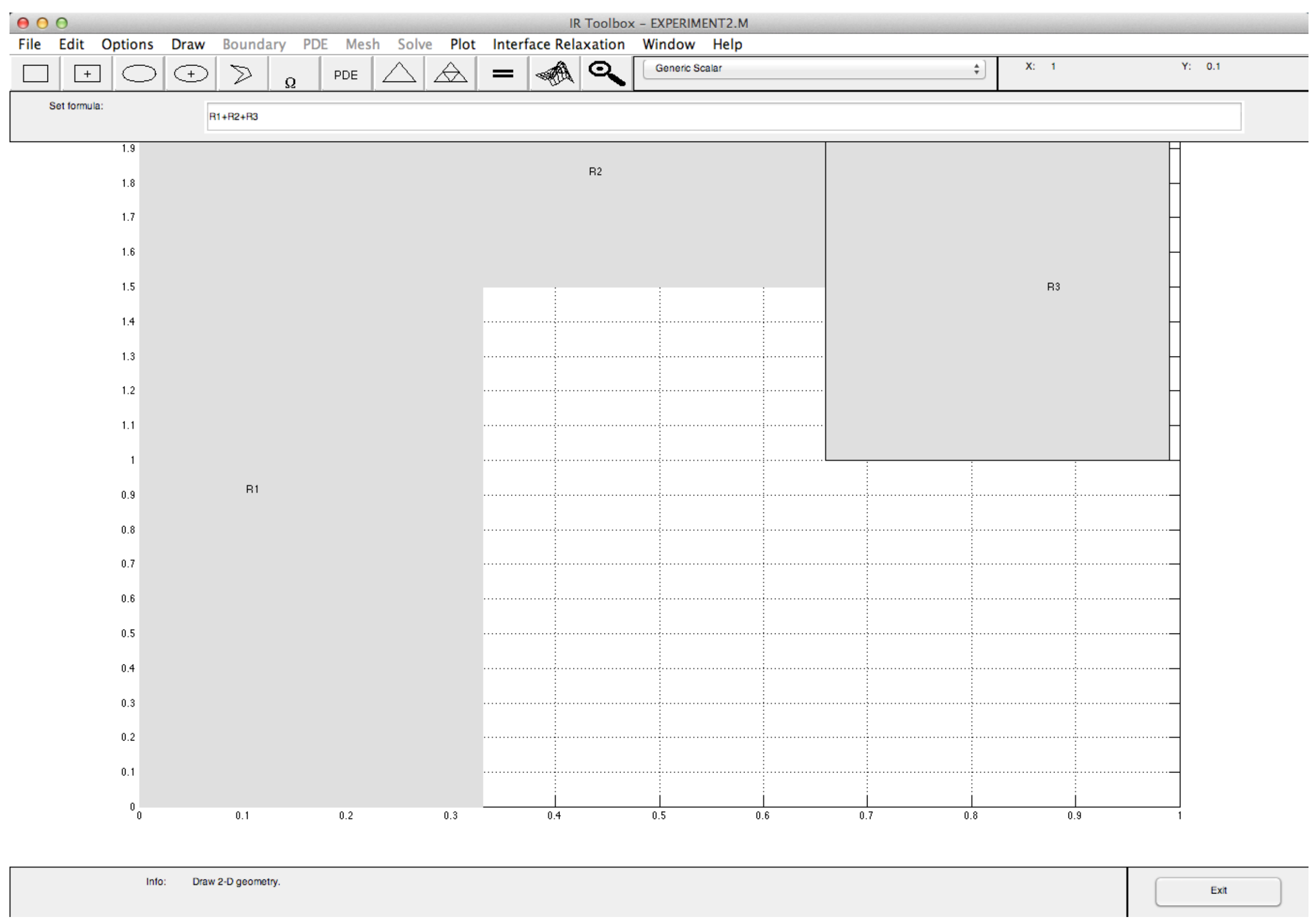
- Geometry
  - Three domains with interfaces on $x_1 = 1/3$ and $x_2 = 2/3$.
- Mesh Parameter
  - Hmax =0.05
- Interface parameter
  - omega is 0.03 and 0.04 for the first and second interface respectively.

Initialization of IRToolbox

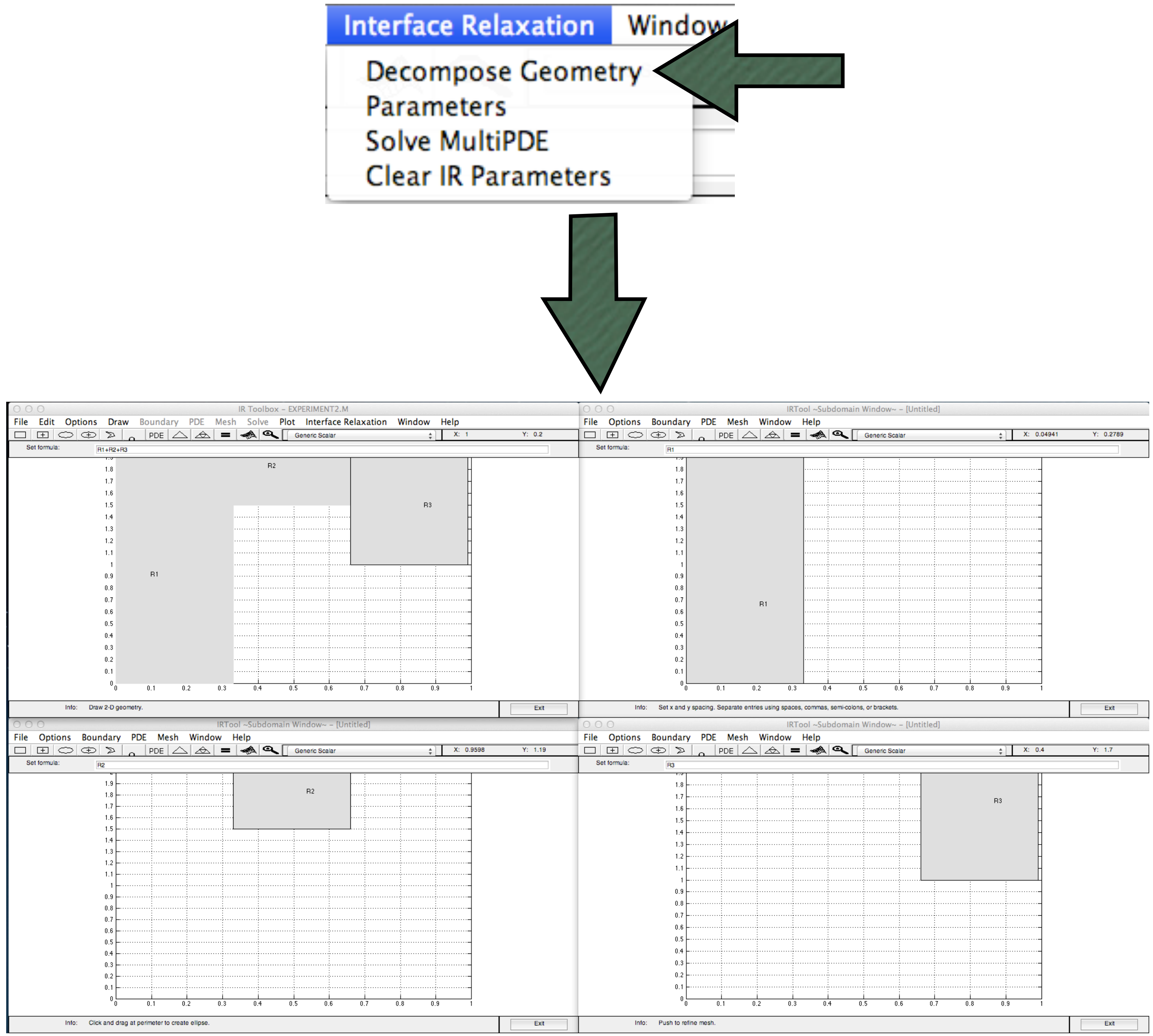
1. *Type "irtool" in Matlab's command window.*
2. *Global Window Opens*

Draw Mode

1. We can draw the global geometry

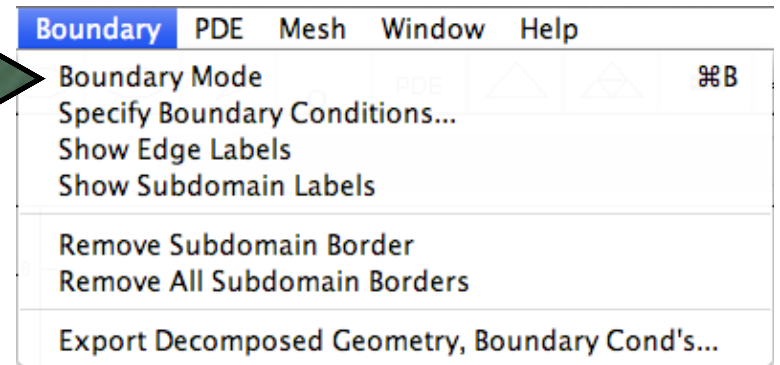


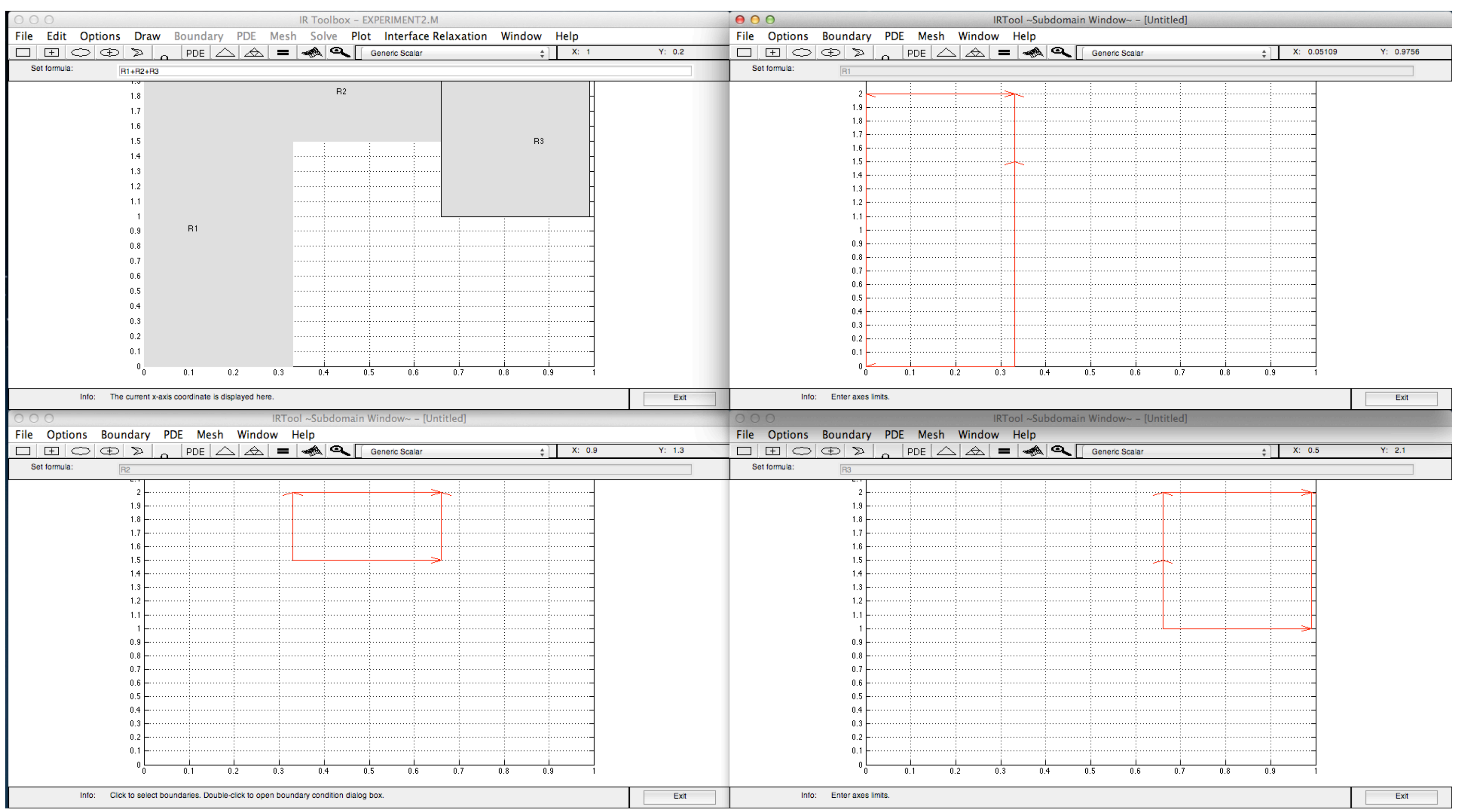
Decompose Global Geometry to Subdomains



Boundary Mode (1/2)

1. Choose Boundary Mode on Each Subdomain

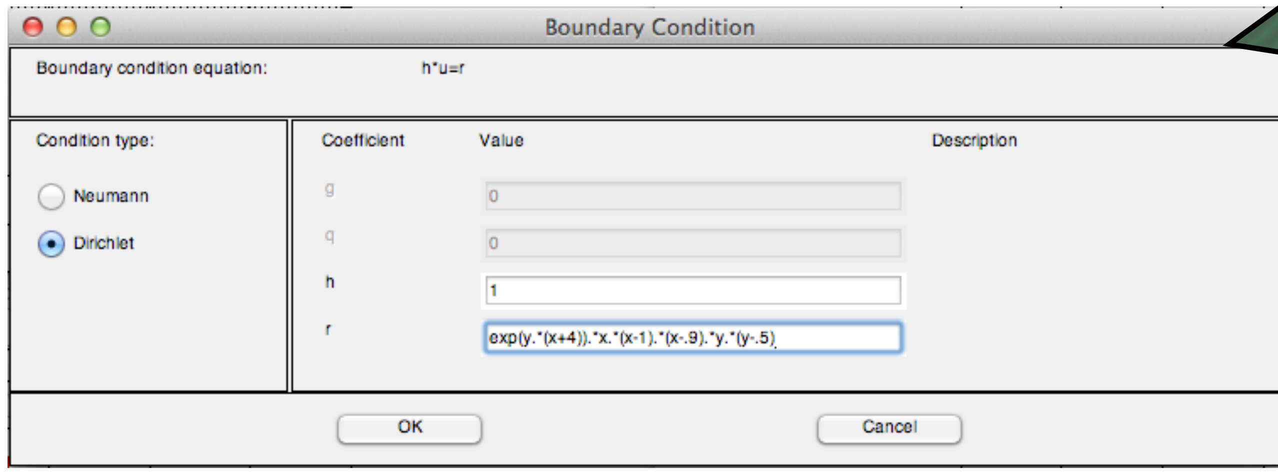




Boundary Mode (2/2)

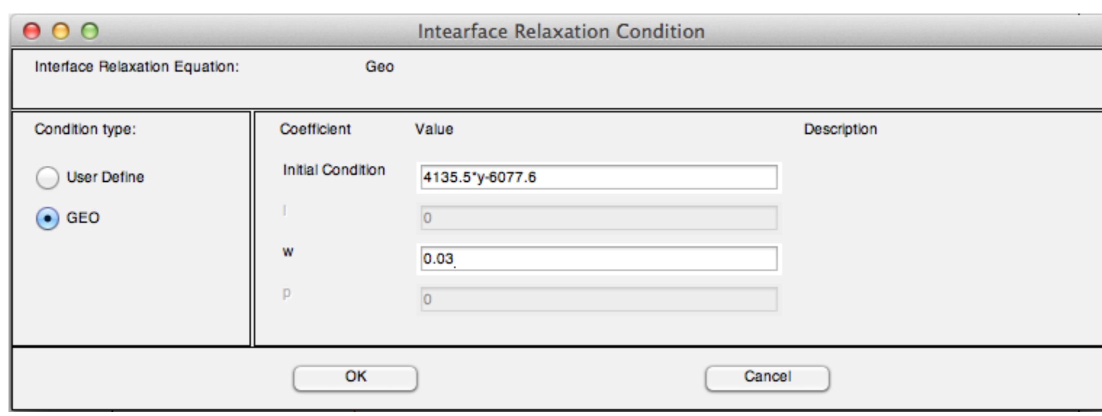
1. Define Boundary conditions
   - The true solution *u(x,y)* is

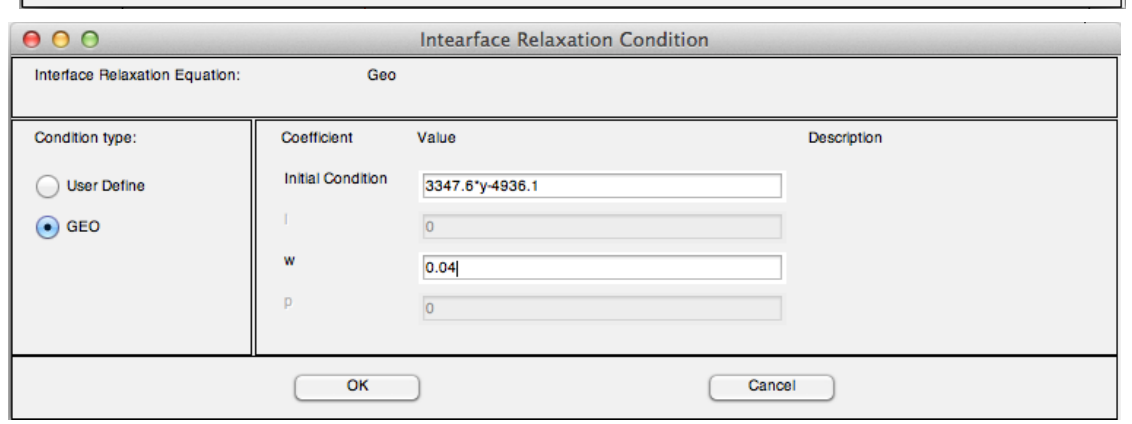
$$u(x,y) = e^{y(x+4)}x(x-1)(x-0.9)y(y-0.5).$$



Double click on boundaries

2. Define Interface Relaxation Conditions
   - Omega is 0.03 & 0.04 on the two interfaces

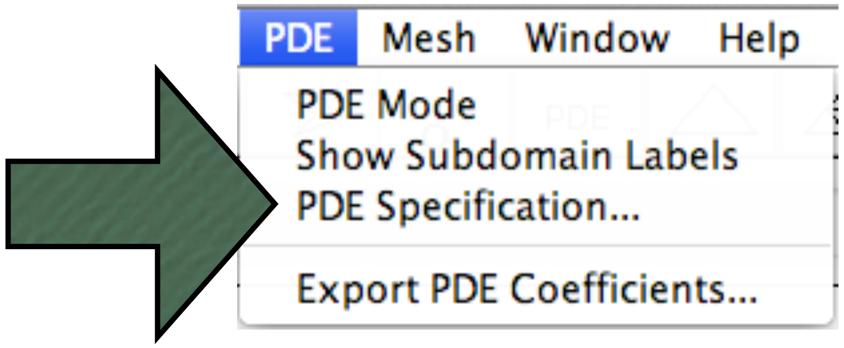




Double click on interfaces

PDE MODE

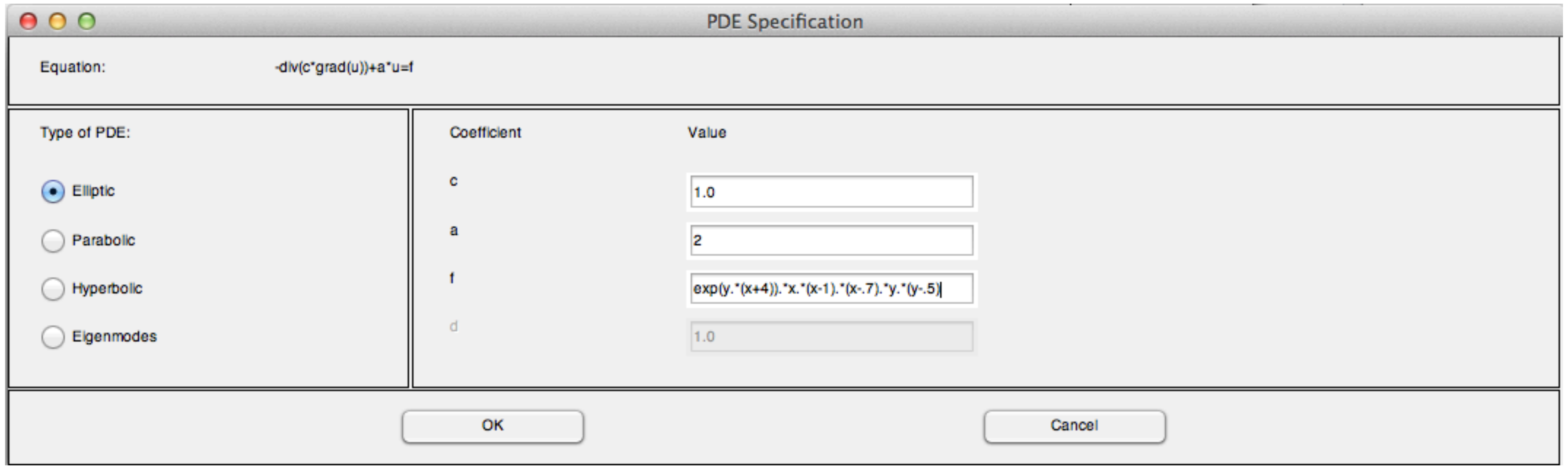
1. Choose PDE Specification from PDE Menu



2. Enter PDE parameters for each subdomain

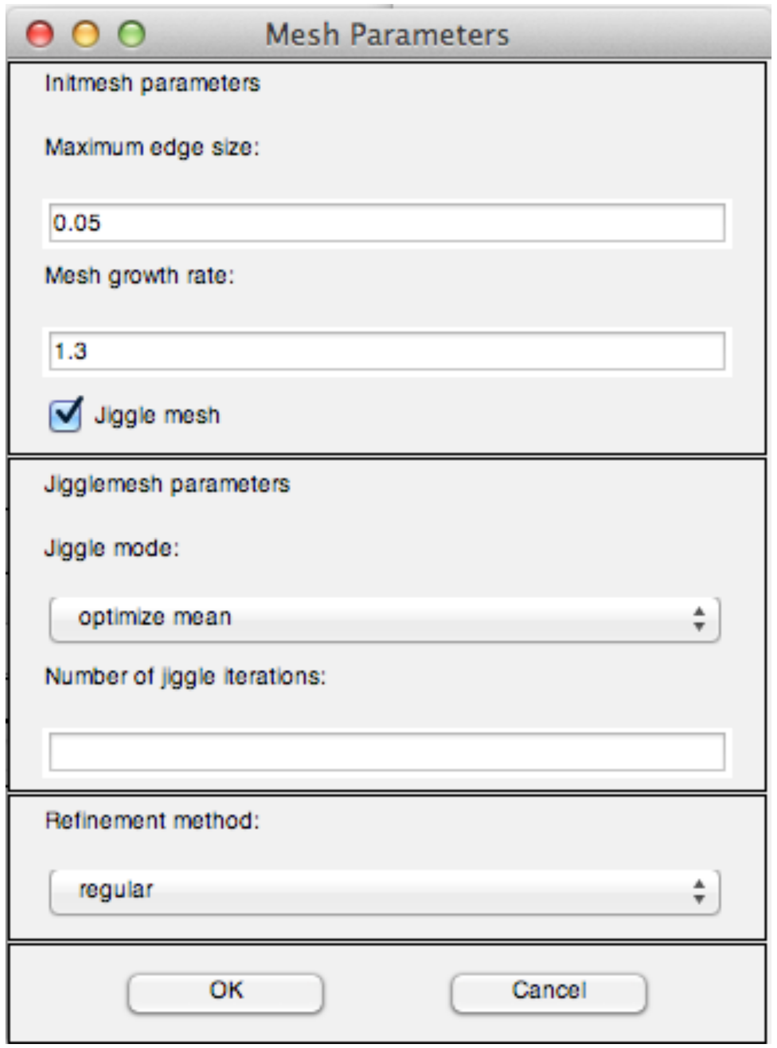
$$Lu(x,y) \equiv -\nabla^2 u(x,y) + \gamma^2 u(x,y) = f(x,y),\ (x,y) \in \Omega$$

$$u(x, y) = u^b(x, y),\ (x, y) \in \partial\Omega,$$



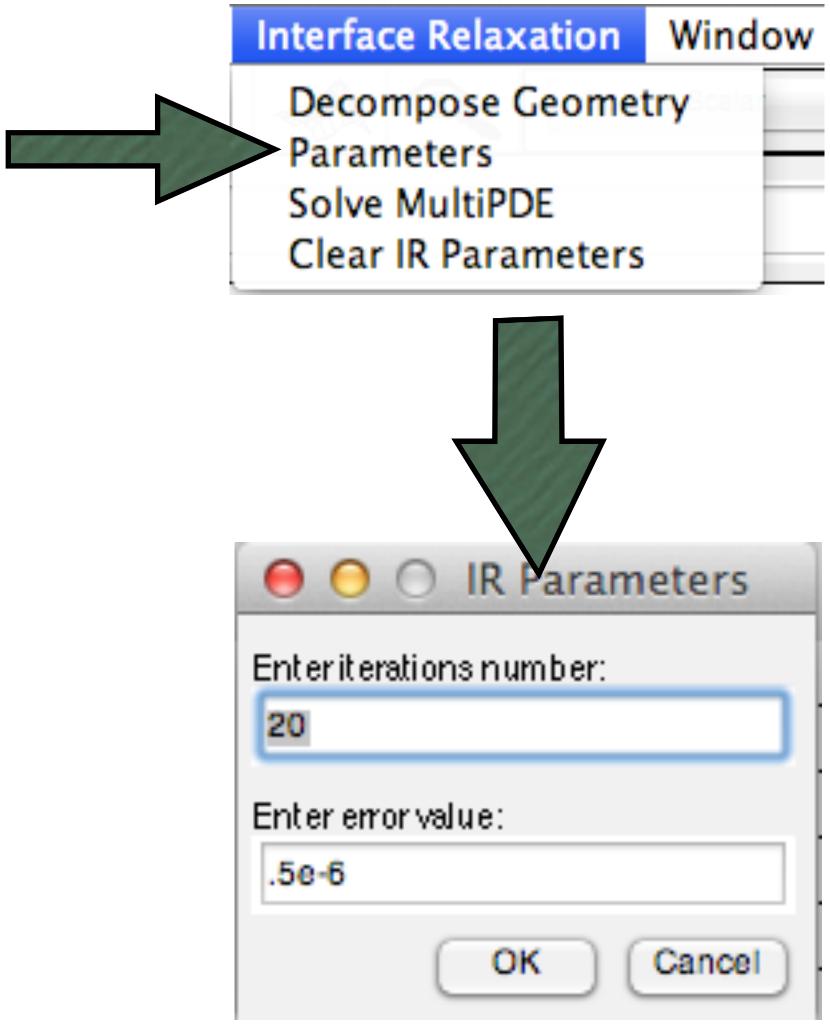
Mesh Mode

1. Enter Mesh parameters
   - Hmax = 0.05 on each subdomain



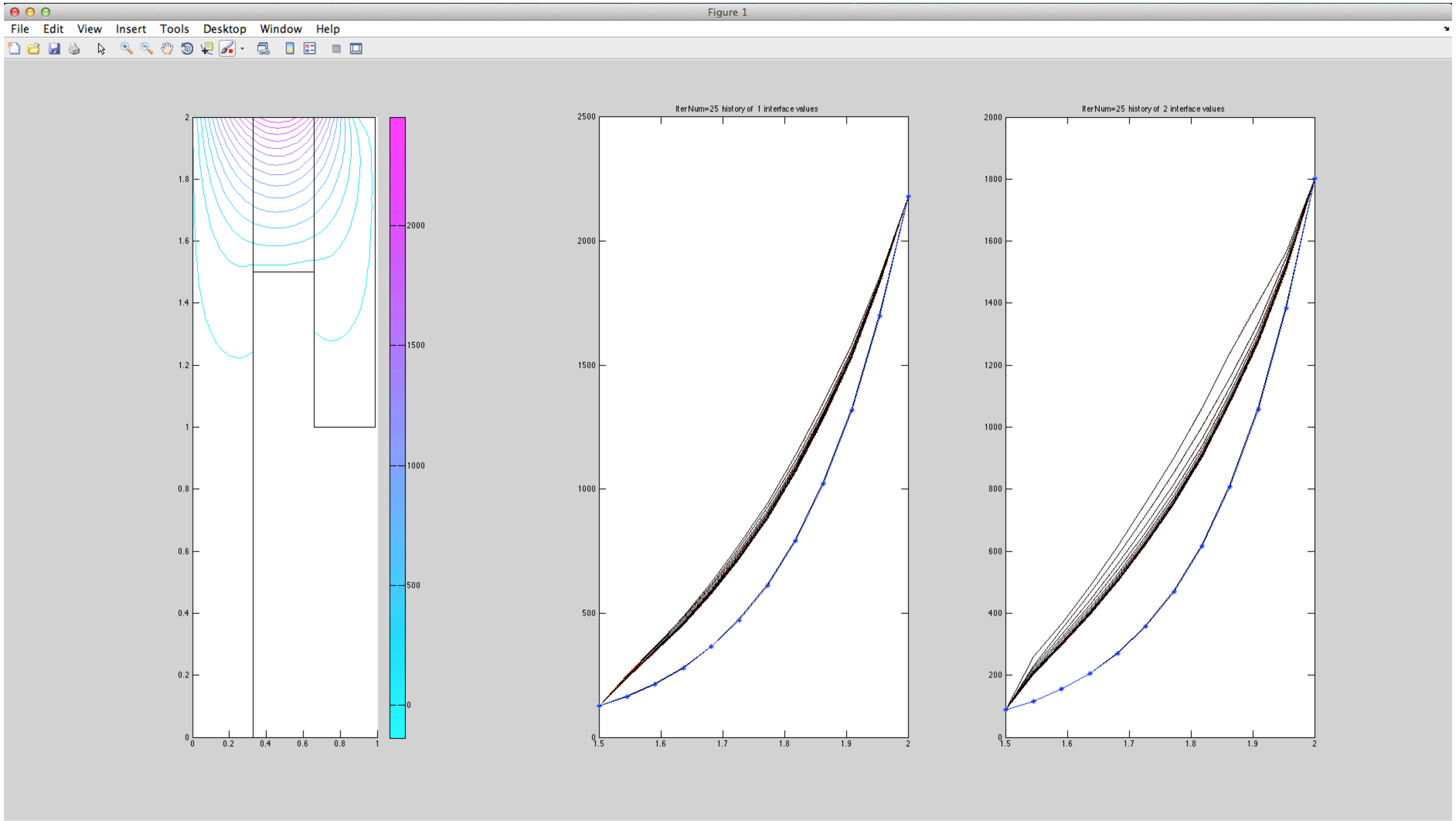
Specify Interface Relaxation Method parameters

- Number of iterations =20
- Maximum Error = .5e-6



Solve MultiPDE

1. Choose Solve MyltiPDE from the Interface Relaxation Menu



Conclusions

- Our approach enjoys the following approaches:
  1. Problem simplification.
     - Simpler local physical rules acting on simpler geometries.
     - convenient abstraction & closer to real world.
  2. Reduction in software development time.
     - Reuse of legacy scientific software.
  3. Numerical efficiency.
     - Use the most appropriate numerical method for each particular subproblem.

Questions?

Thank You