An Overture Tutorial: Grid Generation and Solving PDEs

Bill Henshaw

Rensselaer Polytechnic Institute

For documentation or to download Overture goto:

overtureFramework.org

To get started type:

cd *to-the-Overture-directory*

source defenv

This will set the following Overture environment variables and aliases

```
setenv Overture /usr/common/acts/Overture/Overture.v26.d
setenv CG /usr/common/acts/Overture/cg.v26
setenv CGE /usr/common/acts/Overture/cg.v26.d
alias ogen ${Overture}/bin/ogen
alias plotStuff ${Overture}/bin/plotStuff
alias cgins ${CGE}/ins/bin/cgins
alias cgcns ${CGE}/cns/bin/cgcns
set primer = ${Overture}/primer
set sampleGrids = ${Overture}/sampleGrids
set cns = ${CG}/cns
set ins = ${CG}/ins
```

To check if everything is set-up correctly type:

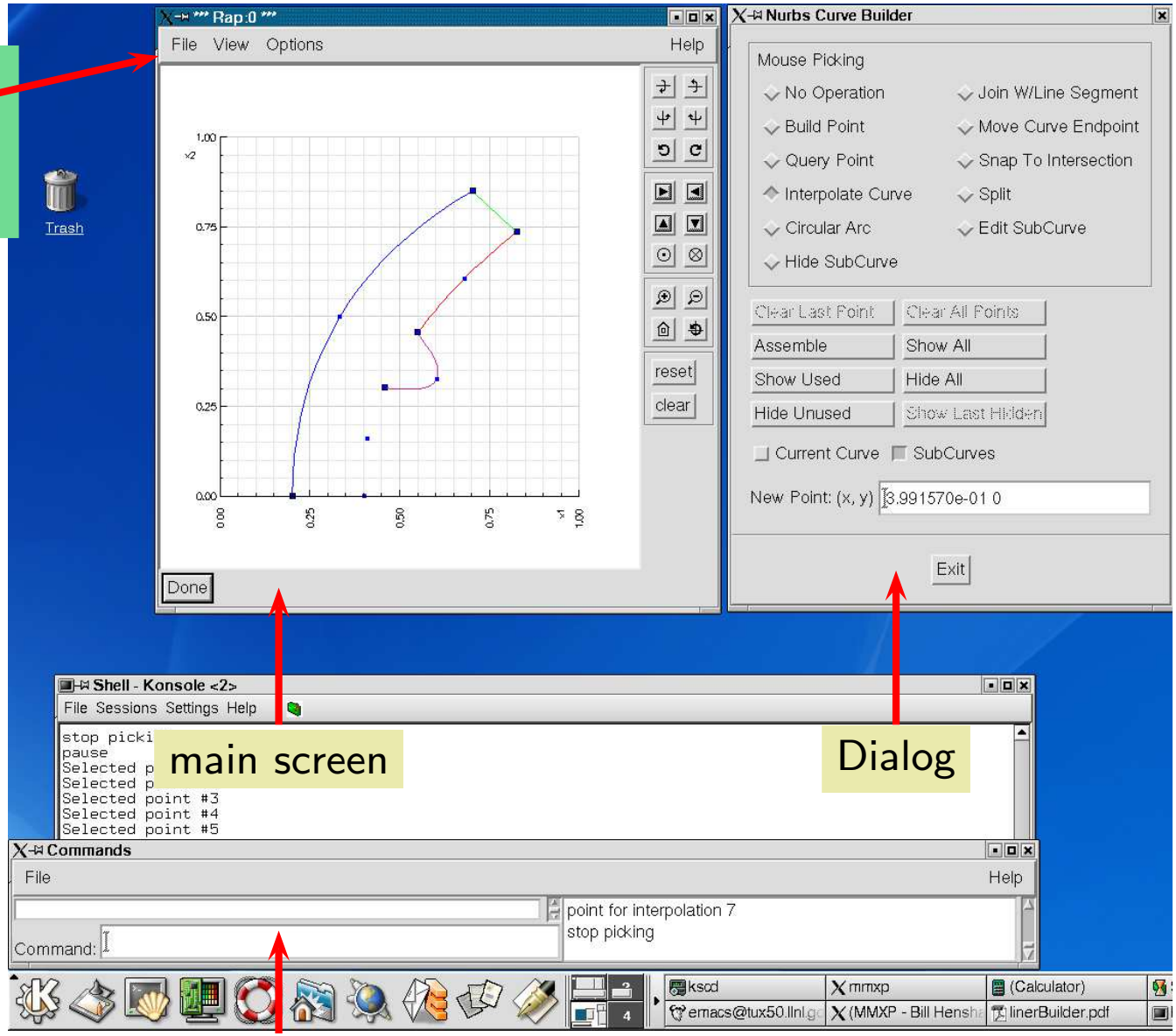
ogen

and the ogen graphics windows should appear. Choose `right-mouse` and `exit` to exit.

Overture Graphics Interface

File: hardcopy
View: clipping
Options: axes, colour-bar

Mouse Buttons
left : pick
middle : zoom
right : pop-up menu
shift-left : translate
shift-middle : rotate



Important Overture and CG programs and directories

Overture:

$Overture/bin/ogen : overlapping grid generator (alias *ogen*).
$Overture/bin/plotStuff : plot grids and show files (alias *plotStuff*).
$Overture/primer/ : holds the primer examples.
$Overture/sampleGrids/ : holds ogen scripts (*.cmd) for building many different grids.

CG:

$CGE/cns/bin/cgcns : the compressible Navier-Stokes solver (alias *cgcns*).
$CG/cns/runs/ : directories with scripts (*.cmd) for sample runs *new*.
$CG/cns/cmd/ : scripts (*.cmd) for running different scenarios (see Readme file).

$CGE/ins/bin/cgins : the incompressible Navier-Stokes solver (alias *cgins*).
$CG/ins/runs/ : directories with scripts (*.cmd) for sample runs *new*.
$CG/ins/cmd/ : scripts (*.cmd) for running different scenarios (see Readme file).

The setup is similar for **cgad**, the advection diffusion solver, **cgmx**, the electromagnetic solver, **cgsm** the solid mechanics solver, and **cgmp**, the multi-physics solver.

Running the primer examples

To run the primer example `mappedGridExample3` type:

$primer/mappedGridExample3

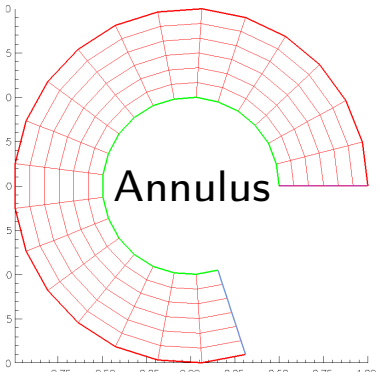
Then

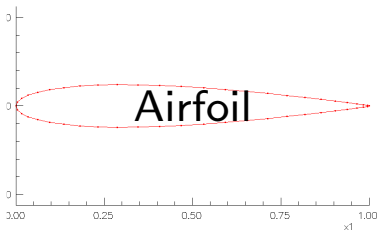
optionally type **nts=1000** to set the number of time steps.
type **exit** and the contour plotter dialog should appear.
choose the **exit** button at the bottom of the contour dialog to run the example.

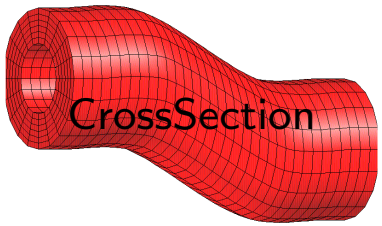
To look at the source file use your favourite editor:

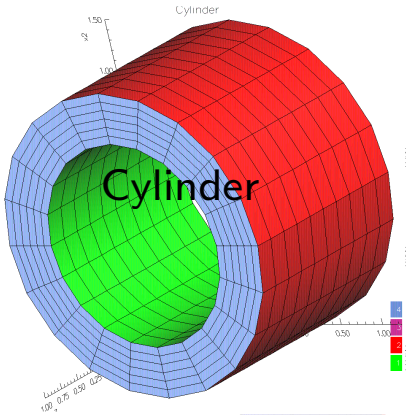
emacs $primer/mappedGridExample3.C

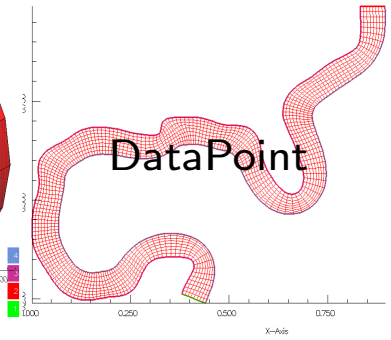
Overture Mappings

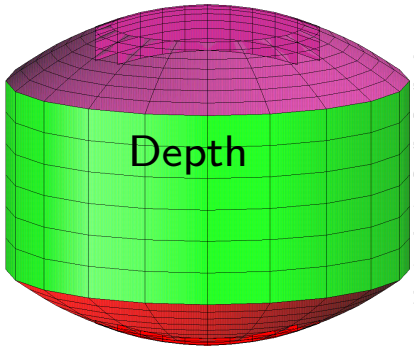


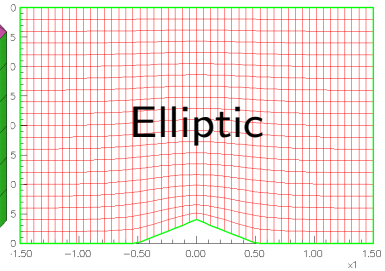


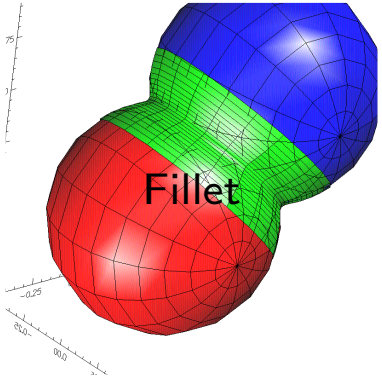


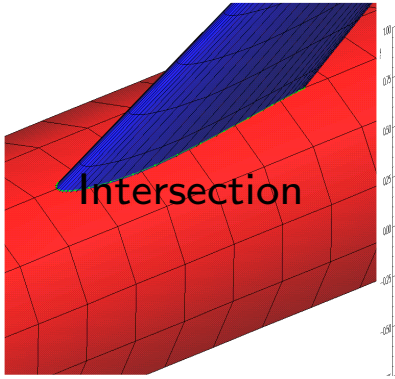


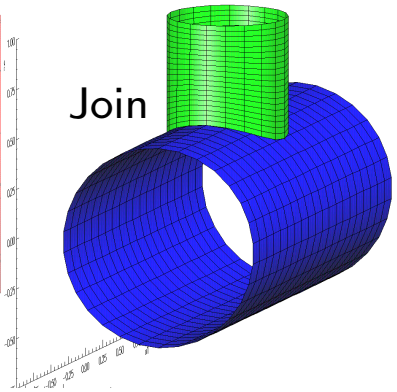


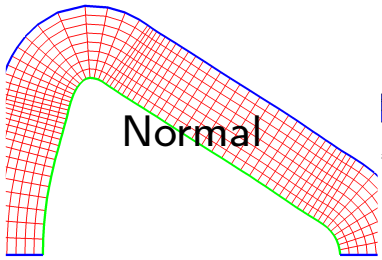


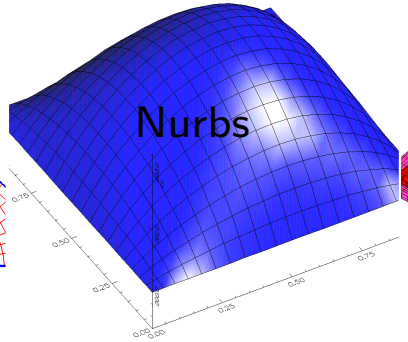


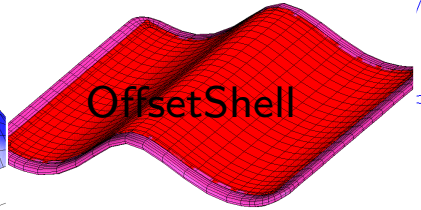


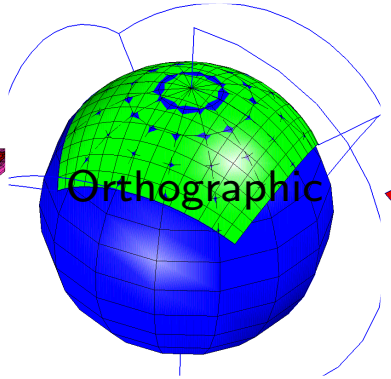


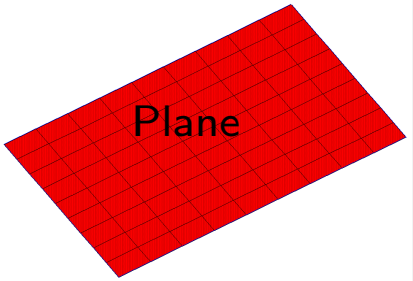




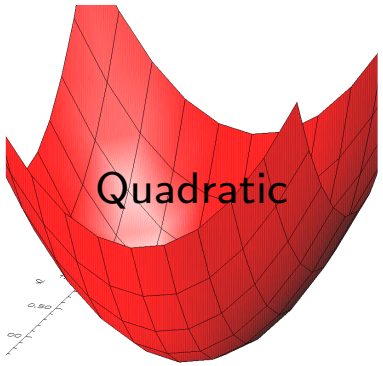


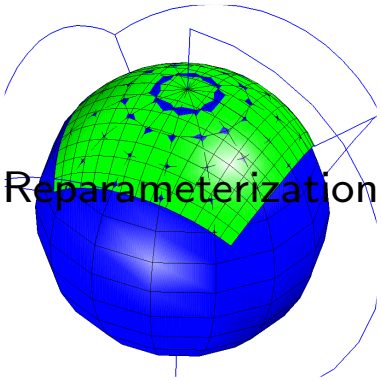


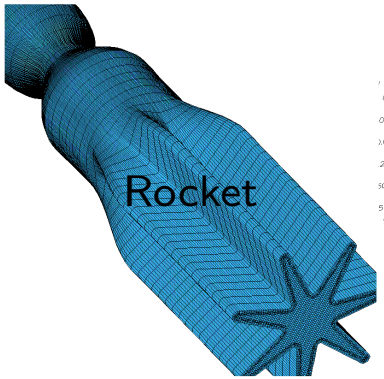


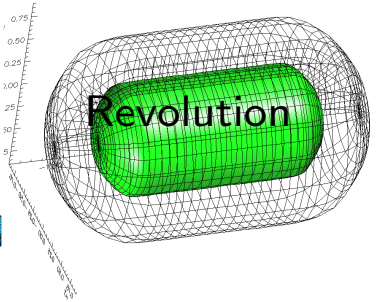


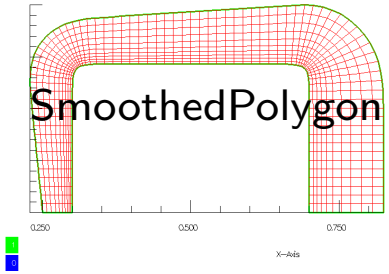
Overture Mappings

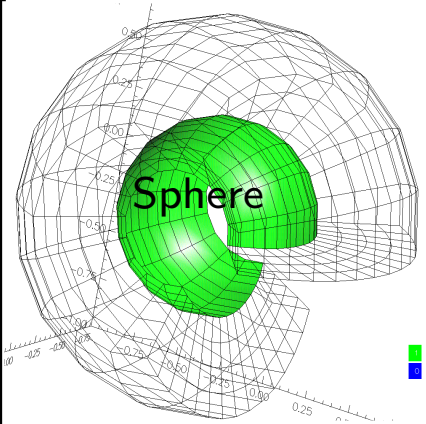


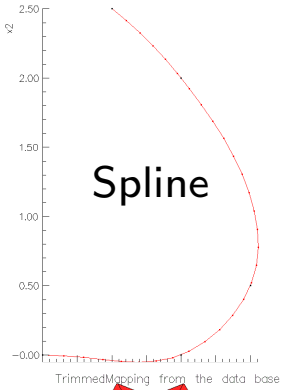


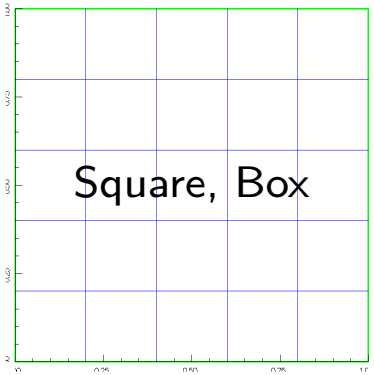


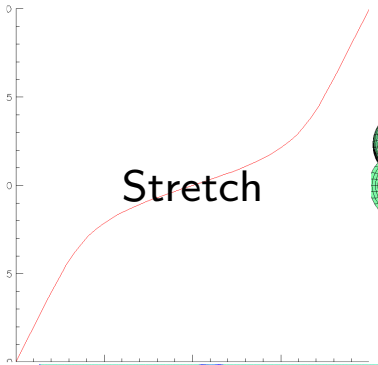


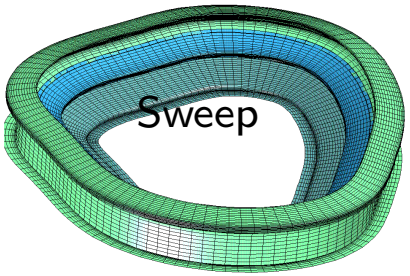


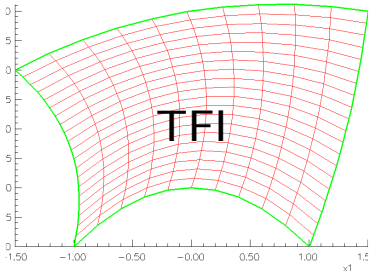


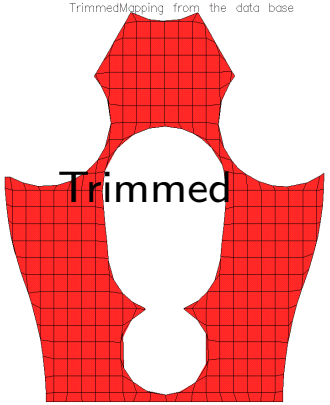


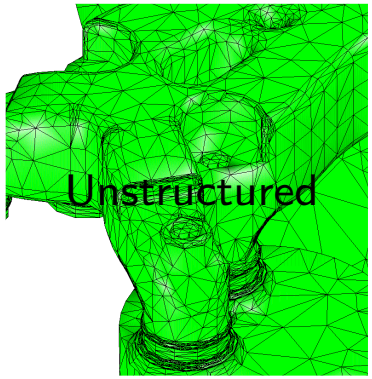


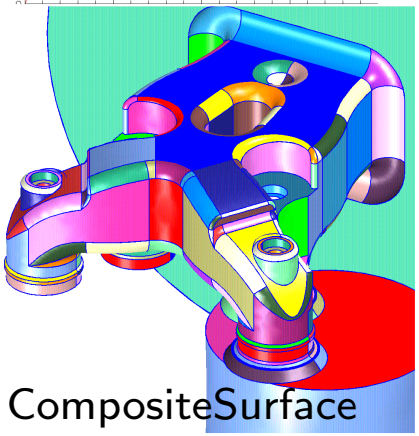












Overture geometry tools: creating mappings

Step 1 : **ogen** (*start ogen*)

Step 2 : **create mappings** (*from pop-up menu, right-mouse-button*)

Example 1: Build a square:

rectangle (*pop-up, under 2D Mappings*)

- **set corners: -2. 2. -2. 2.** (*type into text box and hit enter*)
- **mapping parameters...** (*push button in dialog, will open new dialog*)
- **lines:** `32 32` (*type into text box and hit enter*)
- **boundary conditions:**`1 2 3 4` (*type into text box and hit enter*)
- **name:** `square` (*in dialog, type into the text box and hit enter*)
- **close** (*in dialog, at bottom, close dialog*)
- **exit** (*in Square Mapping dialog at bottom*)

Overture geometry tools: creating mappings

Other mappings that you can try:

airfoil : various airfoil shapes (NACA etc.)

annulus :

box : 3d box.

cylinder : 3d cylinder.

dataPointMapping : mapping defined by data points.

nurbs : Non-Uniform Rational B-Spline.

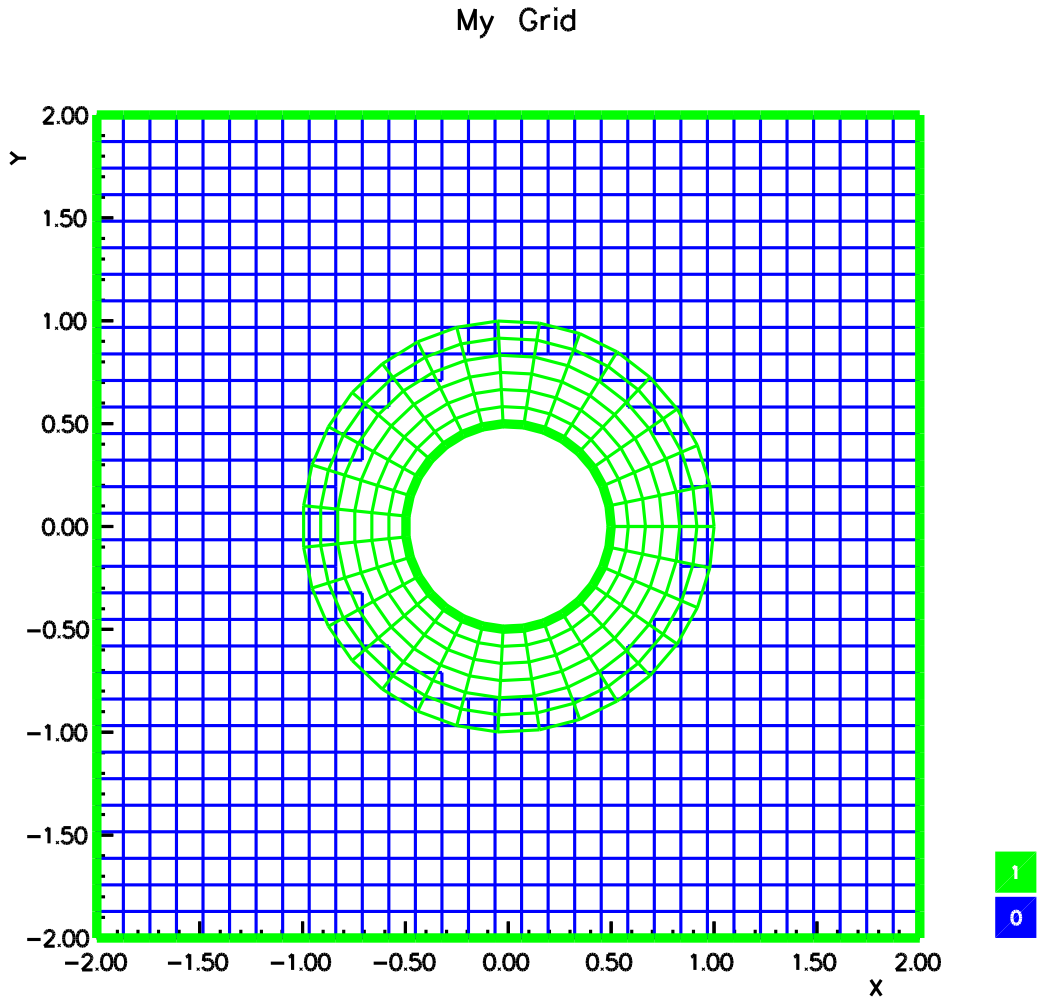
smoothedPolygon : a polygon with smoothed corners.

sphere :

spline : cubic spline.

Using ogen to build an overlapping grid from scratch (1).

Here is the circle-in-a-channel grid that we will build from scratch: (from `sampleGrids/cic.cmd`)



Using ogen to build an overlapping grid from scratch (2).

Step 1 : **ogen** (*start ogen*)
Step 2 : **create mappings** (*from pop-up menu, right-mouse-button*)

Steps 3: Build a square:

rectangle (*pop-up, under 2D Mappings*)
- **set corners: -2. 2. -2. 2.** (*type into text box and hit enter*)
- **mapping parameters...** (*push button in dialog, will open new dialog*)
- **lines:** 32 32 (*type into text box and hit enter*)
- **boundary conditions:**1 2 3 4 (*type into text box and hit enter*)
- **name:** square (*in dialog, type into the text box and hit enter*)
- **close** (*in dialog, at bottom, close dialog*)
- **exit** (*in Square Mapping dialog at bottom*)

Using ogen to build an overlapping grid from scratch (3).

Steps 4: Build an annulus:

Annulus (*pop-up, under 2D Mappings*)
- **mapping parameters...** (*push button in dialog, will open new dialog*)
- **lines: 33 7** (*text box, hit enter*)
- **boundary conditions: -1 -1 1 0** (*-1=periodic, 0=interp., 1=physical*)
- **name: annulus** (*text box, hit enter*)
- **close** (*in dialog, at bottom, close dialog*)
- **exit** (*in dialog at bottom*)

Using ogen to build an overlapping grid from scratch (3).

Steps 5: Construct the overlapping grid:

exit this menu (popup, at bottom, exit create mappings)
generate an overlapping grid
    square (*choose names of mappings to use in the grid*)
    annulus
  compute overlap (*compute holes and interpolation points*)
exit
save a grid (*save grid to a file*)
`myGrid.hdf` (*file name*)
`cic`
exit

Running ogen demos (the overlapping grid generator)

To run ogen with the a demo script type one of :

ogen $sampleGrids/valveDemo (2d valve)
ogen $sampleGrids/valvePortDemo (3d valve and port)
ogen $sampleGrids/shapesDemo (2d hybrid grid)

At the pause, choose:

mouse or main-window buttons : rotate and translate the plot.
continue : to advance to the next pause .
break : to break out of the demo script.

Flow past a cylinder with cgins (incompressible Navier-Stokes solver)

To run cgins with the `cylinder.cmd` script, type

cgins $ins/cmd/cylinder.cmd

Then choose one of:

continue : to advance to the next output time.
plot component : to plot different solution components.
streamlines : to plot streamlines (choose `erase first`).
grid : to plot the grid (choose `erase first`).
movie mode : to run and plot.
break : to break from movie mode.
final time 50. : to increase the final time.

NOTE: If cgins cannot find the cilc.hdf grid file, you can generate it with:

ogen noplot $sampleGrids/cilc.cmd

Falling bodies with cgins (incompressible Navier-Stokes solver)

To build the grid used by this example type:

ogen noplot $sampleGrids/twoDrop.cmd

To run cgins type

cgins $ins/cmd/twoDrop.cmd

Then choose one of:

continue : to advance to the next output time.
movie mode : to run and plot.
plot component : to plot different solution components.
streamlines : to plot streamlines (choose `erase first`).
grid : to plot the grid (choose `erase first`).
break : to break from movie mode.

Shock hitting a cylinder with AMR using with cgcns (Euler equations)

To build the grid used by cgcns type:

ogen noplot $sampleGrids/cicArg.cmd -factor=2 -interp=e

To run cgcns with the `cicShockg.cmd` script (adaptive mesh refinement), type:

cgcns $cns/cmd/cicShockg.cmd

or (to run with 3 refinement levels of factor 2 to time $t = 1.4$, plotting every $.1$):

cgcns $cns/cmd/cicShockg.cmd -l=3 -r=2 -tf=1.4 -tp=.1

Then choose one of:

continue : to advance to the next output time.
plot component : to plot different solution components.
grid : to plot the grid.
contour : enter the contour plotter and choose `wire frame`.
movie mode : to run and plot.
break : to break from movie mode.
final time 50. : to increase the final time.

Deforming elastic sphere using cgsm

To build the grid type:

ogen noplot $sampleGrids/sphere -interp=e -factor=2

To run cgsm type (all on one line):

cgsm $sm/cmd/sphereEigen -g=spheree2.order2 -diss=0.5 -tp=.05 -vClass=2 -nMode=2 -mMode=1 -go=halt -dsf=.05

Then choose one of:

continue : to advance to the next output time.
movie mode : to run and plot.
break : to break from movie mode.
contour : to plot components of the solution (choose 'erase' first)

Conjugate heat transfer using cgmp (cgins + cgad)

To build the grid type:

ogen noplot $sampleGrids/diskArray -factor=1 -interp=e -nCylx=2 -nCyly=2

To run cgmp type (all on one line):

cgmp $mp/cmd/io.cmd -g=diskArray2x2ye1.order2.hdf -nu=.05 -kappa=.01 -tp=.05 -solver=yale

Then choose one of:

continue : to advance to the next output time.
plot component : to plot different solution components.
plot domain : to choose active domain for contour or grid commands.
contour : enter the contour plotter.
grid : to plot the grid.
movie mode : to run and plot.
break : to break from movie mode.

FSI: incompressible flow past a flexible beam (cgins)

See the cg/ins/runs/beamInAChannel directory (Readme).

To build the grid type:

ogen -noplot $ins/runs/beamInAChannel/beamInAChannelGrid -interp=e -factor=2

To run type:

cgins $ins/runs/beamInAChannel/beamInAChannel -g=beamInAChannelGride2.order2.hdf -numElem=21 -tf=5. -tp=.05 -rhoBeam=1. -E=2. -addedMass=1 -ampProjectVelocity=1 -nu=.05 -ad2=1 -rampInflow=1 -cfls=10. -numberOfCorrections=1 -nis=6 -psolver=yale -BM=FD -go=halt

Then choose one of:

continue : to advance to the next output time.
plot component : to plot different solution components.
plot domain : to choose active domain for contour or grid commands.
contour : enter the contour plotter.

grid : to plot the grid.

movie mode : to run and plot.

break : to break from movie mode.

Scattering of an electromagnetic wave using cgmx

To build the grid type:

ogen -noplot $sampleGrids/cicArg -order=4 -interp=e -factor=2

To run cgmx type (all on one line):

cgmx $mx/cmd/cic.planeWaveBC -g=cice2.order4.hdf

Then choose one of:

continue : to advance to the next output time.
movie mode : to run and plot.
break : to break from movie mode.
contour : to plot components of the solution (choose 'erase' first)

Using plotStuff for post-processing

The plotStuff program can be used to plot grids generated by ogen and to plot results saved in *show files*.

To plot a grid type:

plotStuff $sampleGrids/valve.hdf (2d valve)
plotStuff $sampleGrids/building3.hdf (3d buildings)

To plot results in the show file generated by cgins (from a previous demo) type

plotStuff cylinder.show

Then to plot contours choose:

contour
exit

and then choose one of:

next to plot the next solution.
component: to plot a different component.
show movie to plot a movie.

Deforming Grid Example

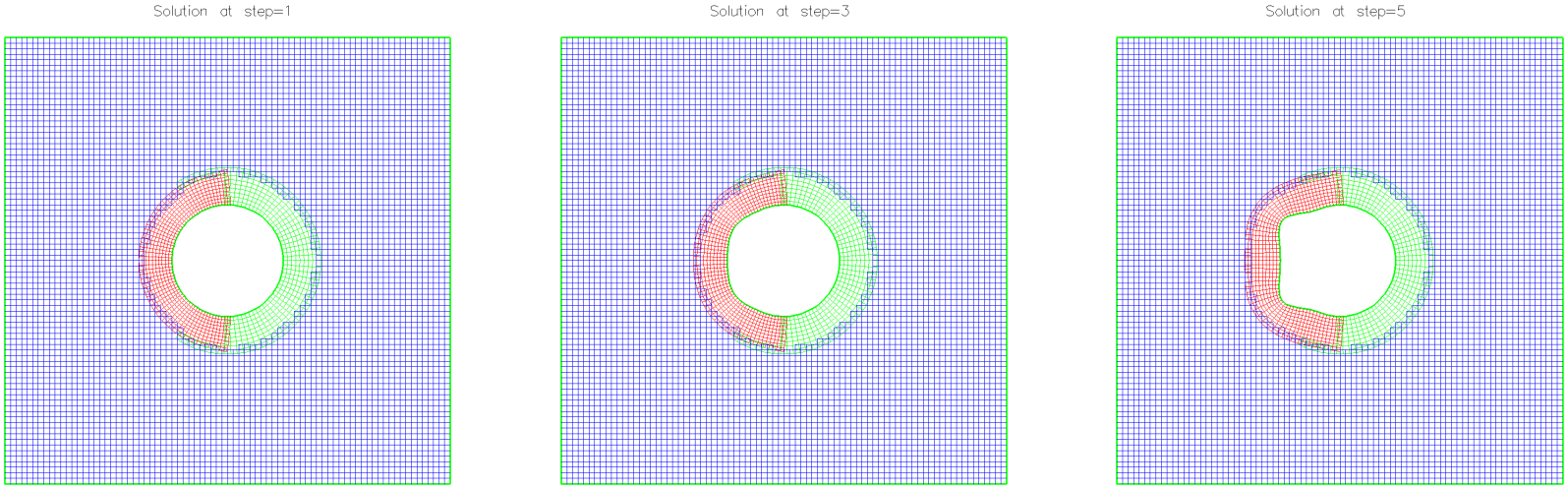
Step 1: Build the grid for the deforming grid example:

ogen noplot $sampleGrids/iceCircle

Step 2: Run the deform program

$primer/deform -numSteps=50

Step 5: Choose **exit** at the bottom of the *grid Plotter* dialog window to take a step, then repeat.



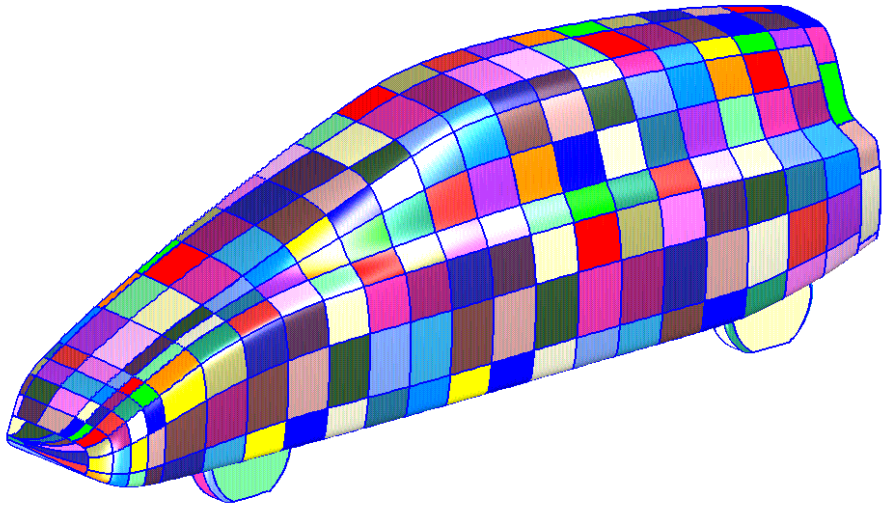
CAD Example: build an overlapping grid for a car (1)

Step 1: Use the **rap** program to read a CAD geometry for a car (IGES file), remove the wheels, and build a water-tight triangulation:

cp $Overture/sampleGrids/asmo.igs . (*get a copy of the CAD file)*
$Overture/bin/rap $sampleGrids/asmoNoWheels.cmd

Step 2: Use the **mbuilder** program to build grids for the car body using the hyperbolic grid generator:

$Overture/bin/mbuilder $sampleGrids/asmoBody.cmd



CAD Example: build an overlapping grid for a car (2)

Step 3: Read in the CAD surfaces that define the front wheel and build grids for the front wheel:

$Overture/bin/mbuilder $sampleGrids/asmoFrontWheel.cmd

Step 4: Read in the CAD surfaces that define the back wheel and build grids for the back wheel:

$Overture/bin/mbuilder $sampleGrids/asmoBackWheel.cmd

Step 5: Read the body and wheel grids generated in previous steps and construct an overlapping grid with ogen:

ogen $sampleGrids/asmo.cmd
plotStuff asmo.hdf (*view the grid*)