TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

TEXAS

The University of Texas at Austin

ParaView

Visualization with ParaView

PRESENTED BY:

Joao Barbosa

Before we begin…

You can download ParaView and install on your computer

http://www.paraview.org/download/

Launching a VNC session on Maverick:

- From the command line
- From the vis.portal.tacc.utexas.edu

Data Types

- Supports a wide variety of data types
  - Structured grids
    - uniform rectilinear, rectilinear, and curvilinear
  - Unstructured grids
  - Polygonal data

- Time series support

Data Formats

- Supports a wide variety of data formats
  - File->Open->Files of Type)

- Users can write data readers to extend support to other formats

- Or conversion to the VTK format (or others)

Convert to vtk example

```
login3.maverick(55)$ head time300_part001.d
 0.3381826E+02  0.4178067E+02 -0.7625223E+04   -0.1482713390E+01
 0.2996829E+02  0.2668810E+02 -0.5915045E+04   -0.1029315675E+01
 0.3152208E+02  0.2927930E+02 -0.7144176E+04   -0.7941290576E+00
 0.3746464E+02  0.5067991E+02 -0.8226453E+04   -0.1569707201E+01
 0.2154900E+02  0.3764573E+02 -0.9470392E+04   -0.1322161129E+01
 0.4752824E+01  0.2128451E+02 -0.5226978E+04   -0.1837562219E+01
 0.2500508E+02  0.5073580E+02 -0.4899845E+04   -0.1796484516E+01
 0.2512511E+02  0.4947190E+01 -0.6735189E+04   -0.1520137835E+01
 0.4855696E+02  0.2542868E+02 -0.4856916E+04   -0.1442525579E+01
 0.3696677E+02  0.3556828E+02 -0.5594683E+04   -0.1563389487E+01
```

```
# vtk DataFile Version 2.0
Written using VisIt writer
BINARY
DATASET UNSTRUCTURED_GRID
POINTS 10 float
B6^R®A6^ÇÂ^T<89>ºB/*^D?¢ÂïÄúb^TB
r½AFfíÃã^<8f>@ãcAB?<99>ôÃ^_<82><8f>AÈßp@^G^PàÅ)5yAíÇ<9e>B+Å<86>Ãý^MÓB=BãAÈ~ßÅ^X^SøAl~^HAÐ^XÝÅ^^^MP@<81>mvB^W;ÈÅ^@ù
AýèÜB23=Ãý^MyCELLS 10 20
^@^@^@^A^@^@^@^@^@^@^@^@^A^@^@^@^A^@^@^@^A^@^@^@^B^@^@^@^A^@^@^@^C^@^@^@^A^@^@^@^D^@^@^@^A^@^@^@^E^@^@^@^A^@^@^@^F^@^@^@^A^@^@
^@^@^@^A^@^@^@^A^@^@^@^A^@^@^@^A^@^@^@^A^@^@^@^A^@^@^@^A^@^@^@^A^@^@^@^ACELL_DATA 10
POINT_DATA 10
SCALARS data float
LOOKUP_TABLE default
Å        <82><8e>Å^D bÅ^Cç^Cé¿ÎÈ§¿ø?×¿ô*æ¿qà<99>^@^@^@^@¿Ü<83>KÅ^H +VECTORS ptsvec float
B6^R®A6^ÇÂ^T<89>ºB/*^D?¢ÂïÄúb^TB
r½AFfíÃã^<8f>@ãcAB?<99>ôÃ^_<82><8f>AÈßp@^G^PàÅ)5yAíÇ<9e>B+Å<86>Ãý^MÓB=BãAÈ~ßÅ^X^SøAl~^HAÐ^XÝÅ^^^MP@<81>mvB^W;ÈÅ^@ù
AýèÜB23=Ãý^My
~
~
~
~
```

```
import sys
import vtk


file_name = sys.argv[1]



basefilename = file_name.split(".")
basefilename = basefilename[0]
newfilename = basefilename+".vtk"



ug = vtk.vtkUnstructuredGrid()
pts = vtk.vtkPoints()
vals = vtk.vtkFloatArray()
vals.SetNumberOfComponents(1)

f = open(file_name, "r")
k = 0
for ff in f:
    k = k + 1
    if k % 100000 == 0:
                print k
    ff = [float(i) for i in ff.strip().split()]
    pts.InsertNextPoint(ff[:3])
    vals.InsertNextValue(ff[3])

f.close()

ug.SetPoints(pts)
ug.GetPointData().SetScalars(vals)

wrtr = vtk.vtkXMLUnstructuredGridWriter()
wrtr.SetInput(ug)
wrtr.SetFileName(newfilename)
wrtr.Update()
~
```

Visualization Algorithms

Supports a wide variety of visualization algorithms -> Filters

- Isosurfaces
- Cutting planes
- Streamlines
- Glyphs
- Volume rendering
- Clipping
- Height maps
- …

Special Features

- Supports derived variables
  - New scalar / vector variables that are functions of existing variables in your data set
- Scriptable via Python
- Saves animations
- Can run in parallel / distributed mode for large data visualization

ParaView Visualization Pipeline

All processing operations (filters) produce data sets

Can further process the result of every operation to build complex visualizations

- e.g. can extract a cutting plane, and apply glyphs (i.e. vector arrows) to the result
  - Gives a plane of glyphs through your 3D volume

ParaView

Toolbars
Pipeline Browser
Object Inspector
3D View
ParaView 4.2.0 64-bit
Time: 0
59
Representation
Extract Location
Pipeline Browser
builtin:
Properties
Information
Properties
Apply
Reset
Delete
Search ... (use Esc to clear text)
Properties
Display
View (Render View)
RenderView1

Paraview

Undo/Redo

Camera Controls

View Controls (undo/redo)

VCR Controls

Common Controls

Active Variable Controls

ParaView Demo

WRF weather forecast data set
(smallwrf.zip)

Paraview WRT Demo

Paraview WRT Demo

WRF weather forecast data set

- Rectilinear grid
- Multiple scalar and vector variables
- Time series

Can show:

- Clouds
- Wind
- Temperature
- …

ParaView WRT Demo

ParaView Lab

Getting Started

Download example data file

‘disk_out_ref.ex2’

https://vis.tacc.utexas.edu/training/

Right-click, Save link as…

Open ParaView

ParaView

Today we will:

- Create isosurfaces for a scalar variable
- Clip and slice the surfaces
- Use glyphs to display a vector field
- Use streamlines to show flow through a vector field
- Add slices to show variable values over a plane
- Create volume rendering

ParaView

- Open the file `disk_out_ref.ex2`

- Click `File -> Open`
- Select `disk_out_ref.ex2`
- Click `OK`
- Select ALL variables
- Click blue `Apply`
- Cylinder outline of dataset extent displayed

ParaView

- Open the file `disk_out_ref.ex2`

- Click `File -> Open`
- Select `disk_out_ref.ex2`
- Click `OK`
- Select ALL variables
- Click blue `Apply`
- Cylinder outline of dataset extent displayed

ParaView

Manipulate Representation and color

- Use the Active Variable Controls to change color from `Solid Color -> Pres`
- Use Representation toolbar to change representation `Surface -> Surface With Edges`
- Click on `Display` Panel

ParaView

Manipulate Representation and color

- Use the Active Variable Controls to change color from `Solid Color -> Pres`
- Use Representation toolbar to change representation `Surface -> Surface With Edges`
- Click on `Display` Panel

ParaView

Manipulate Representation and color

- Use the Active Variable Controls to change color from `Solid Color -> Pres`
- Use Representation toolbar to change representation `Surface -> Surface With Edges`
- Click on `Display` Panel

Manipulate Representation and color

- Review `Coloring`
- Review `Representation`
- Enable Color Legend by clicking on icon in toolbar
- Click `+Z` view button
- Explore dataset with mouse

Manipulate Representation and color

- Review `Coloring`
- Review `Representation`
- Enable Color Legend by clicking on icon in toolbar
- Click `+Z` view button
- Explore dataset with mouse

Manipulate Representation and color

- Review `Coloring`
- Review `Representation`
- Enable Color Legend by clicking on icon in toolbar
- Click `+Z` view button
- Explore dataset with mouse

Create isosurfaces

Select Contour in pipeline

In `Contour By`, select `Temp`

Enter `400`

Click `Apply`

Click the eye icon next to `disk_out_ref.ex2`

Change representation to `Wireframe`

Create isosurfaces

Select Contour in pipeline

In `Contour By`, select `Temp`

Enter `400`

Click `Apply`

Click the eye icon next to `disk_out_ref.ex2`

Change representation to `Wireframe`

Create isosurfaces

Select Contour in pipeline

In `Contour By`, select `Temp`

Enter `400`

Click `Apply`

Click the eye icon next to `disk_out_ref.ex2`

Change representation to `Wireframe`

Create isosurfaces

Select Contour in pipeline

In `Contour By`, select `Temp`

Enter `400`

Click `Apply`

Click the eye icon next to `disk_out_ref.ex2`

Change representation to `Wireframe`

Create isosurfaces

Select Contour in pipeline

In `Contour By`, select `Temp`

Enter `400`

Click `Apply`

Click the eye icon next to `disk_out_ref.ex2`

Change representation to `Wireframe`

Edit View Sources Filters Tools Macros Help
ParaView 3.14.1 64-bit
Time: 0
Wireframe
Layout #1
Information
Delete
Uncheck Selected Bloc
Materials Faces
Blocks
ck ID: 1 Type: HEX8 Size.
Y
Z
X

Extract Surface

- Click `Filters -> Alphabetical -> Extract Surface`
- Click `Apply`

Extract Surface

- Click `Filters -> Alphabetical -> Extract Surface`
- Click `Apply`

Clip Surface

- Click `Filters -> Common -> Clip`
- `Show Plane` should be checked
- Click `Y Normal` and enter `-1` in second box under `Normal`
- Click blue `Apply`
- Unselect `Show Plane`

Clip Surface

- Click `Filters -> Common -> Clip`
- `Show Plane` should be checked
- Click `Y Normal` and enter `-1` in second box under `Normal`
- Click blue `Apply`
- Unselect `Show Plane`

Clip Surface

- Click `Filters -> Common -> Clip`
- `Show Plane` should be checked
- Click `Y Normal` and enter `-1` in second box under `Normal`
- Click blue `Apply`
- Unselect `Show Plane`

Clip Surface

- Click `Filters -> Common -> Clip`
- `Show Plane` should be checked
- Click `Y Normal` and enter `-1` in second box under `Normal`
- Click blue `Apply`
- Unselect `Show Plane`

disk_out_ref.ex2
ExtractSurface1
Contour1
Clip1

ParaView File Edit View Sources Filters Tools Macros Help
Pipeline Browser
builtin:
disk_out_ref.ex2
Contour1
ExtractSurface1
Clip1
Properties Display Information
Apply Reset Delete
Clip Type Plane
Show Plane
Origin
Normal
X Normal
Reset Bounds
Y Normal
Z Normal
Camera Normal
Center on Bounds
Inside Out
Layout #1

Slice Surface

- Click `disk_out_ref.ex2` in Pipeline Browser
- Click `Filters -> Common -> Slice`
- Click `Apply`
- Click eye next to `ExtractSurface1` and `Clip1` to hide clip plot
- Click on `Y Normal`
- Click `Apply`
- Change `Coloring` to Temp

Slice Surface

- Click `disk_out_ref.ex2` in Pipeline Browser
- Click `Filters -> Common -> Slice`
- Click `Apply`
- Click eye next to `ExtractSurface1` and `Clip1` to hide clip plot
- Click on `Y Normal`
- Click `Apply`
- Change `Coloring` to Temp

Slice Surface

- Click `disk_out_ref.ex2` in Pipeline Browser
- Click `Filters -> Common -> Slice`
- Click `Apply`
- Click eye next to `ExtractSurface1` and `Clip1` to hide clip plot
- Click on `Y Normal`
- Click `Apply`
- Change `Coloring` to Temp

Slice Surface

- Click `disk_out_ref.ex2` in Pipeline Browser
- Click `Filters -> Common -> Slice`
- Click `Apply`
- Click eye next to `ExtractSurface1` and `Clip1` to hide clip plot
- Click on `Y Normal`
- Click `Apply`
- Change `Coloring` to Temp

Create Streamlines

- Click `disk_out_ref.ex2` in Pipeline Browser
- Click `Filters -> Common -> Stream Tracer`
- Click `Apply`
- `Hide Slice`

Create Tubes

- Click `StreamTracer1` in Pipeline Browser
- Click `Filters -> Alphabetical -> Tube`
- Click `Apply`

Create Glyph of Vector Field

- Click `StreamTracer1` in Pipeline Browser
- Click `Filters -> Common -> Glyph`
- Select `V` under `Vectors` box
- Select `Cone` under `Glyph Type`
- `Select Coloring Temp`
- Click `Apply`
- Click on `Tube1` and change to `Solid Color`

Create Glyph of Vector Field

- Click `StreamTracer1` in Pipeline Browser
- Click `Filters -> Common -> Glyph`
- Select `V` under `Vectors` box
- Select `Cone` under `Glyph Type`
- `Select Coloring Temp`
- Click `Apply`
- Click on `Tube1` and change to `Solid Color`

Create Glyph of Vector Field

- Click `StreamTracer1` in Pipeline Browser
- Click `Filters -> Common -> Glyph`
- Select `V` under `Vectors` box
- Select `Cone` under `Glyph Type`
- `Select Coloring Temp`
- Click `Apply`
- Click on `Tube1` and change to `Solid Color`

Create Glyph of Vector Field

- Click `StreamTracer1` in Pipeline Browser
- Click `Filters -> Common -> Glyph`
- Select `V` under `Vectors` box
- Select `Cone` under `Glyph Type`
- `Select Coloring Temp`
- Click `Apply`
- Click on `Tube1` and change to `Solid Color`

Create Glyph of Vector Field

- Click `StreamTracer1` in Pipeline Browser
- Click `Filters -> Common -> Glyph`
- Select `V` under `Vectors` box
- Select `Cone` under `Glyph Type`
- `Select Coloring Temp`
- Click `Apply`
- Click on `Tube1` and change to `Solid Color`

ParaView

Edit Color Map

Click on `Glyph` in Pipeline Browser

Under `Display -> Coloring -> Edit`

Click `Folder Icon`

Select `BLUE...HSV`

Click `OK`

Click `Close`

ParaView

- Under `Display -> Coloring -> Edit`
- Click Favorites (Folder icon with a Heart)
- Select `Blue to Red`

ParaView

Edit Color Map

Click on `Glyph` in Pipeline Browser

Under `Display -> Coloring -> Edit`

Click Favorites (Folder icon with a Heart)

Select `Blue to Red` Click `Close`

ParaView

Edit Background Color

Click on `View (Render View)` at bottom of Properties panel

Click Color Black

ParaView

Create Volume Rendering

Click `disk_out_ref.ex2` in Pipeline Browser

Under `Representation` select `Volume`

Show `disk_out_ref.ex2` if hidden

Hide `Clip1`

Properties -> Display -> Edit

Click `Choose Preset`

Choose `Black, Orange, .. RGB`

Modify Function

Click `Close`

ParaView

Start and Stop Trace

Click `Edit` -> `Delete All` (or Save your state before deleting: `File -> Save State`)

Click `Tools -> Start Trace`

Create a contour of Temperature (`Filters -> Common -> Contour`)

Click `Tools -> Stop Trace`

Why:

Preparing large render jobs

Automation. Reproducible views

Often easier and more direct way to interact with the GUI

Python “trace” good way to learn

Paraview “ Tools->Python Shell”

“Tools->Start Trace” and “Stop Trace”

in conjunction with

http://www.paraview.org/ParaView3/Doc/Nightly/www/py-doc/

ParaView

Start and Stop Trace

Click `Edit` -> `Delete All` (or Save your state before deleting: `File -> Save State`)

Click `Tools -> Start Trace`

Create a contour of Temperature (`Filters -> Common -> Contour`)

Click `Tools -> Stop Trace`

ParaView

Start and Stop Trace

Click `Edit` -> `Delete All` (or Save your state before deleting: `File -> Save State`)

Click `Tools -> Start Trace`

Create a contour of Temperature (`Filters -> Common -> Contour`)

Click `Tools -> Stop Trace`

Paraview quick demo

Tools->Start Trace

File->Open or File->”Load State”

Set up your desired view

Save out an image, move camera, save another image etc

Tools->Stop Trace

Paraview quick demo

```
RenderView1.CameraViewUp = [0.09543959367965343, 0.9234144845905065, -0.3717485354466269]
RenderView1.CameraPosition = [1.9286515104666857, 18.50177992113957, 46.53312519465689]
RenderView1.CameraClippingRange = [30.801335755472167, 64.07468538767495]

WriteImage('/home1/01891/adb/test2.png')


RenderView1.CameraViewUp = [0.023684855325377813, 0.8401066890459779, -0.5419038463126282]
RenderView1.CameraPosition = [31.296910589202966, 20.53710987684372, 33.28630820715742]
RenderView1.CameraClippingRange = [27.63131449767485, 71.00282572900147]

WriteImage('/home1/01891/adb/test3.png')
```

This is one way to add “keyframes” but could also do it programmatically…

Paraview quick demo:

Add something like this and run again:

```
framecnt=0
cam = GetActiveCamera()
num_runs=100
for i in range(0,num_runs):
  frac = float(i)/float(num_runs)
  if (frac < 1):
    cam.Azimuth(-90.0/(float(num_runs)/5.0))
  else:
    pass
  file = "newcam" + '%03d.jpg' % framecnt
  WriteImage(file)
  print "saved image: " + file
  framecnt += 1
```

```
ffmpeg –i newcam%03d.jpg newcam.mpg
```

Questions?

More tutorials available:

http://www.paraview.org/Wiki/The_ParaView_Tutorial