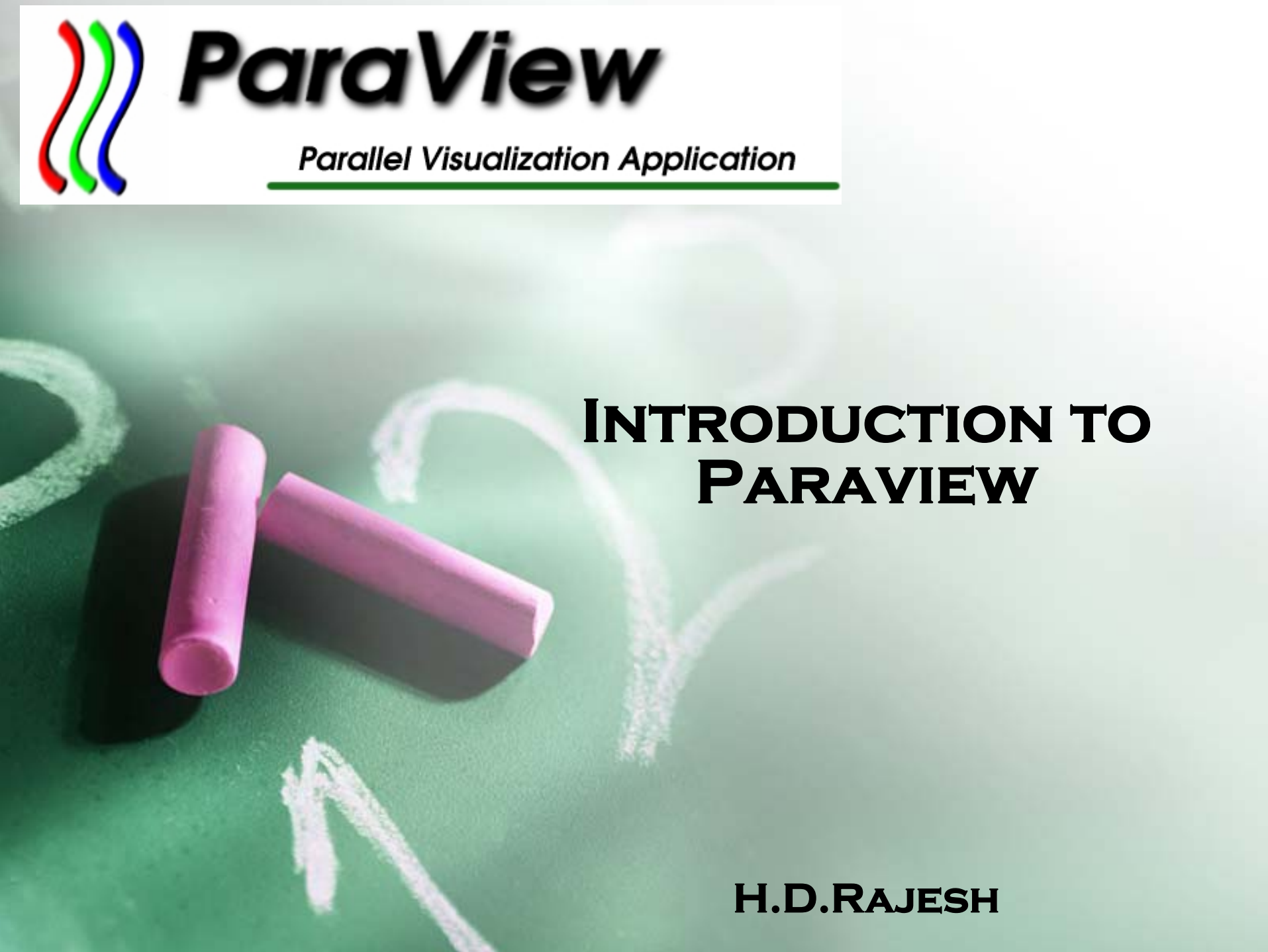


# Introduction to Paraview

H.D.Rajesh


1. Introduction
2. file formats
3. How to use

# Brief Overview

- Info: www.paraview.org
- http://www.paraview.org/wiki/paraview
- Open source ,multi-platform application (Linux, Mac, Windows..)
- Runs on single processor workstations ,multiple processor shared memory super computers or work clusters

- Based on VTK
- VTK: Visualization Tool Kit is an open source code for image processing, and visualization
- Written in c++, uses OpenGl for rendering
- ParaView supports:

  a) <u>Single-User Mode:</u>

  All processing done on the local machine

b) Client-Server Mode :

User interface: client

Data Processing: Server

Rendering: client/ Server

c) Client-Distributed Server Mode :

Data Processing, Rendering on Distributed Server

d) Client-Distributed Data-Distributed Render mode

data server :process the data

second set of servers :rendering

# File Formats

- PVD: (*.pvd) polygonal, image, rectilinear, structured grid or unstructured data sets
- VTK: (*.vtp *.vti *.vtr) XML based file format. polygonal, image, rectilinear, structured grid or unstructured data sets
- Parallel VTK: (*.pvtp, *pvti, *.pvtr) XML based file format d. spatial distribution of data , point to multiple VTK files.

Legacy VTK : (*.vtk ) polygonal, rectilinear, structured grid or unstructured data sets

Other formats: EnSight, EnSight Master Server, HDF5 (image only), VRML, PLOT3D, Stereo Lithography, BYU, POP Ocean, PDB (Protein Data Bank), XMOL, XDMF, raw binary, Exodus, SAF, AVS UCD, Meta Image, Gaussian Cube.

# Supported Data Sets:

- ImageData (.vti) —Serial vtkImageData (structured).
- PolyData (.vtp) — Serial vtkPolyData (unstructured).
- RectilinearGrid (.vtr) — Serial vtkRectilinearGrid (structured).
- StructuredGrid (.vts) —Serial vtkStructuredGrid (structured).
- UnstructuredGrid (.vtu) —Serial vtkUnstructuredGrid (unstructured).

- PImageData (.pvti) —Parallel vtkImageData (structured).
- PPolyData (.pvtp) — Parallel vtkPolyData (unstructured).
- PRectilinearGrid (.pvtr) — Parallel vtkRectilinearGrid (structured).
- PStructuredGrid (.pvts) — Parallel vtkStructuredGrid (structured).
- PUnstructuredGrid (.pvtu) —Parallel vtkUnstructuredGrid (unstructured).

# legacy VTK file formats

- Five basic parts :

1) File version and identifier: # vtk DataFile Version

2) Header: 256 characters maximum

3) Format: ASCII or BINARY

4) Dataset structure :geometry and topology

5) Dataset attributes :i. e., scalars, vectors, tensors, normal , texture coordinates, or field data.

# Legacy VTK example:

```
# vtk DataFile Version 2.0
gmv example
ASCII
DATASET UNSTRUCTURED_GRID
POINTS 4 double
0.0 0.0 0.0
1.0 0.0 0.0
1.0 1.0 0.0
0.0 1.0 0.0
CELLS 1 5
4 0 1 2 3
CELL_TYPES 1
9
POINT_DATA 4
VECTORS velocity double
1 -1 0  1 1 0  -1 1 0  -1 -1 0
SCALARS pressure double 1
LOOKUP_TABLE default
1.0
2.0
3.0
4.0
SCALARS streamkine double 1
LOOKUP_TABLE default
8.0
5.0
9.0
6.0
```

# XML File Formats

- two types :

a) Serial :single file ,run in single process

b) parallel :multiple processes executing in parallel

- dataset is broken into pieces

- piece is stored in a corresponding serial file type (structured or unstructured .etc)

-parallel file type does not contain data

-describes structural information and then

references other serial files

# XML serial File example (single piece)

```
<VTKFile type=" UnstructuredGrid"
  ...>
  <UnstructuredGrid>
  <Piece NumberOfPoints="#"
  NumberOfCells="#">
  <PointData>...</ PointData>
  <CellData>...</ CellData>
  <Points>...</ Points>
  <Cells>...</ Cells>
  </ Piece>
  </ UnstructuredGrid>
  </ VTKFile>
```

# XML parallel File example (single piece)

```
<VTKFile type=" PUnstructuredGrid" ...>
  <PUnstructuredGrid GhostLevel=" 0">
  <PPointData>...</ PPointData>
  <PCellData>...</ PCellData>
  <Appoints>...</ PPoints>
  <Piece Source=" unstructuredGrid0. vtu"/>
  ...
  </ PUnstructuredGrid>
  </ VTKFile>
```

. references to other serial files is stored, not data

# ParaView By Example

- >module load paraview/2.4.4
- >paraview
- Initial setups:
- Tabs
- General : set background color of choice
- Annotate:Set axis label color of choice
- Camera : default is fine!
- From main Menu:View>Application settings>Interface settings> autoaccept(put it on)

- Load a file:
- file>open data>2d.gmv
- 3 Tabs:
- 1.parameter
- 2.display
- 3.Information
- Go to>display
- Section view:choose scalar bar,label ID etc. (node field in GMV).
- Section color: Choose velocity pressure ..etc Edit color map(Data limits in GMV)
- Reset range:resets the changes made

- Section Displaystyle: choose surface,wireframe,points etc,rest leave default
- Section Actor control:Can be Done with Mouse control
- Go to tab Information:
- Information of Dataset.
- Mouse movements:
- Like GMV.
- Defaults are fine.
- To customize movements:
- Main menu-view>3D view properties>camera>camera control 2d/3d

- To come back to view dataset:
- Main menu-view>source
- Center of rotation:
- Right top most corner, 4 options
- Save image:
- save view image(jpg,png,bmp ....)
- Save data:
- file>save data( in pvd, or many other format)
- Temporary position storage:
- Main menu-view>3D view properties>camera>stored camera position-RIGHT CLICK ON BUTTON(6 stores)

- Print data:
- prints to ps file
- Save sessions state,geometry.. etc
- On file menu
- Loading the data in time steps:
- Open first file
- Parameters> time steps>add all >close
- Use time step slider

# Operations

- Filters are used
- In built sources are available in main menu-sources.
- The options in filters appear only  when perticular operation is possible .
- Most frequently used operations are visible in bottom panel
- Filters can be added to bottom panel by clicking Right most bottom of the bottom panel.

- 2d contour:(only for scalars!!)
- display>choose any scalar(pressure)>
- Filter>contour
- Choose input
- Choose the same scalar as the previous one(pressure)
- Number of values(slide bar) select 20-30
- Generate
- Hide surface,click on eye mark on the first data line ---- the contours will appear.

- Contours in GMV style:
- Make the data surface visible (eye mark)
- Click on first pipeline(contour)
- Go to display tab
- Color section- choose property
- Change actor color of choice

- Vector contours:
- filter>calculator
- sqrt(velocity0^2+velocity1^2+velocity2^2)
- Go to information tab-- find result
- filter>contour
- input=calc, scalars= result
- Follow the same step as before

- Vector visualization:
- filter>glyph
- Make data invisible
- Choose glyph shape(cone,line, arrow....)
- Vary scale factor,
- Vary number of glyphs
- To choose other parameters :
- Go to display tab,choose velocity,pressure,vorticity etc

- Data Analysis:
- 1.prob
- 2.pick
- 3.data analysis
- Prob:
- filter>prob
- Choose point/line
- 1.save data in csv format
- 2.or hide surface, and save as jpg, png etc
- Pick
- Same as prob but additional cell centered values

- Data analysis:
- Both pick and prob
- Mesh quality:
- filter>mesh quality
- Shrink
- Shrink to centroid
- Stream tracer
- Same as stream function
- Wrap
- Scalar
- vector(has to be computed using calc)

- Threshold:
- filter>threshold
- Select scalars(pressure, stream,...)
- Go to display tag,select corresponding scalar
- Vary the glider to change threshold
- Transform,Reflection:
- Does the same operation as name
- Multiple Pipelines:
- More filters will appear

- Eg:1
- filter>stream tracer
- filter>tube/ribbon
- Vary the diameter in parameters
- Eg:2
- filter>stream tracer
- Filter> linear extrusion
- Vary the fields in parameters tab

- Cut/clip:
- filter>cut/clip
- Click on eye to view clip
- Move the slider
- Go to display tab to change the field
- Number of values determines cut plane numbers
- Change the cut plane in X,Y,Z direction using cut function buttons—x-normal, y-normal,z-normal

- **Animation:**
- file>548.vtk
- Parameters tab>time steps
- Add all(better to choose last file first)
- Move the slide bar to initial sate
- View>keyframe animation
- Choose frame equal to the files(random is OK)
- Current frame = 0
- Choose the source(imp!!), go back
- Press red button to start.(animation line at bottom)

- Keep timestep slider to “0” in
- Take a keyframe(next button)
- Move the slider to “1”(next timestep file)
- Take a keyframe
- Do the same thing for all files
- Stop animation(red button again)
- View:
- Press play.
- Advanced view:
- View>keyframe animation
- Choose real time /sequence change duration/sequence

- Saving:
- press the last button
- 1.save as jpg=> All key frames saved
- 2.Save in AVI/MPEG
- Advanced Animation:
- View>keyframe animation
- Press F6( the keyframe slider pops up)
- Track section:
- Choose source and property
- Active key properties will appear
- Choose properties of choice

- Animation of contours:
- Select the position of the data
- View>keyframe animation
- Choose frame of choice
- Current frame = 0
- Play mode --> sequence
- Select source as the contour(imp!)
- View>source(going back)
- Watch for contour range
- Delete the old contour value

. And choose all the contour values within the contour range. All the keyframes should be taken within this range.

Eg:

if range is -1.3 to 5.9

5 contour values for 5 keyframe is chosen as

-1.3, -1.2, 1.1, 1.4, 2.4

-set New value to initial state(slider)

-set the number of values(no. Of contours)

-vary the range

-click generate

-get the keyframes at different range and save.

- Mouse movement animation:
- Just take the keyframes at different position

# GMV converter :

- Paste the file in pp2d folder(with indat2df)
- Compile
- g++ -o gmvconverter gmvconverter.cpp
- ./gmvconverter *.gmv *.vtk
- Limitation:
- 1.Only tested/Works for GMV format published in Web(std)
- 2.polygons,materials ignored
- 3.Velocity and any number of variables
- 4.Vertex based(Not cell based)
- Writing it out much easier!!

# Comparison to GMV

- Advantages:
- 1.Handles large data sets
- 2.Faster
- 3.Many filters
- 4.Higher quality
- Disadvantages:
- Not specific to featflow(geography ,medical data..)
- Higher memory
- May not have all functions that of GMV

# Thank You!