Introduction to AVS/Express

What is AVS/Express?

A development environment...

...for 3D visualization applications

...object oriented

...based on a visual programming paradigm

The LEGO DNA
What you will see today

Various exercises will guide you through the most important AVS/Express features
Three “theory interludes” will provide an inside view of AVS/Express functioning
It’s just a start-up, real learning happens when you have a concrete problem to solve
So, don’t be discouraged too fast!

Exercise 1
Start AVS/Express
Various AVS/Express editions

- **Developer Edition**
  - Everything you need
  - Modules to develop applications
  - Standard modules can be opened

- **Visualization Edition**
  - Some modules not available
  - Cannot create runtimes
  - Cost less
  - No differences if you are an End User

  - Tobias will explain them

Starting AVS/Express

- Launch AVS/Express (Linux):
  - $ ...setup env variables...$ express
  - (if problems: -nohw)

- Launch AVS/Express (Windows):
  - Start ➔ AVS/Express Collection ➔ Dev Edition

- Launch AVS/Express (Mac OS X):
  - Start the X/Server (Applications/Utilities) then $ express

- Load an example:
  - Library ➔ Examples ➔ Visualization ➔ Orthoslice

- Or start with an empty application:
  - File ➔ New Application ➔ Single-window DataViewer

  - Reset with the button.

Try to find the AVS/Express parts
Environment variables

- **XP_ROOT**
  - Where AVS/Express is installed
- **MACHINE**
  - The machine architecture (look under bin)
- **PATH**
  - $XP_ROOT/bin/$MACHINE
  - (Win add) $XP_ROOT/runtime/bin/$MACHINE
- **LD_LIBRARY_PATH** (Linux)
- **DYLD_LIBRARY_PATH** (Mac)
  - $XP_ROOT/lib/$MACHINE:
  - $XP_ROOT/lib/$MACHINE/missing
- **XP_LICENSE_SERVER**
  - (optional) lic.server.machine.ch:12345

Theory interlude 1

AVS/Express main concepts

AVS/Express components
### Viz techniques

- More than 500 visualization techniques available
- Readers available for a broad range of file formats
- AVS/Express prototyping support helps finding the most useful technique
- Any technique can be adapted to suit specific user requirements

### Network Editor: the build area

Applications are built dragging modules from the libraries to the Network Editor.

- An application is composed by modules connected together
- Colors of connection ports and lines are related to data type (only compatible types can connect)
- Each module starts when it receives all the needed data
- The user interface docks in the right place automatically
The result of this application
AVS/Express: behind the scene

Object Manager

Object references

= is assignment  => is object reference

Not only visualization: UI Kit
Some AVS/Express resources

- AVS homepage: http://www.avs.com/
- AVS forum: http://forum.avs.com/
- AVS/Express built-in examples
- Visualization techniques book (in the AVS/Express manuals)
- International AVS Center (IAC): http://www.iavsc.org/
- IAC training material: http://www.iavsc.org/training
- Other resources on: http://mariovalle.name/AVS/
- A user level AVS/Express course: http://mariovalle.name/AVS/introduction-to-xp.html

Exercise 1b
Try built-in examples

Try to find the AVS/Express parts

Go to: Libraries → Examples

Instantiate: Visualization → Orthoslice
By dragging to the empty area
Exercise 2

Electronic density slice
Workbook page 4

Exercise 3

Lobster internal density
Workbook page 26
Editors: interface to services

- Modules: User interfaces for loaded modules
- View: Rendering parameters
- Transform: Geometrical transformations
- Light: Setup, color and type
- Camera: Camera position and type
- Object: Selected object appearance
- Datamap: Colormap setup
- Graph: 2D charts parameters (if loaded)
- Print: Screen hardcopy
Object rendering properties

Select the graphical object

Modify its properties

Other viewer / object properties

- Alternate Object
  - Enable it for the selected object
- Lights
  - One or more lights
  - Directional, bidirectional, spot
- Camera
  - Depth cueing
  - Clipping planes
  - Perspective or orthogonal camera
- Mouse manipulator
  - Object, camera or light

Theory interlude 2

Mesh and Field types
Scatter data

From points to surfaces/volumes

Unstructured grids
Cell types for unstructured mesh

AVS/Express Field Data Type

Grid and Node Data

- Positions and data components
- Data components can be scalar and vector
Grid and Node Data

Connectivity Information

- cell is a geometric entity
- cell_sets are a collection of cells

Connectivity
Cell Data

- Field
  - Mesh
    - Grad
    - Cell Sets
  - Data
    - Node Data
    - Cell Data

- Cell data completes the definition

Cell Data

- NodeData1
  - CellData1
  - NodeData2
  - CellData2
  - NodeData3
  - CellData3

Unstructured Meshes

- The only type of mesh where cell sets are required
**Structured Meshes**

- Connectivity is implied by dimensions (all structured meshes have dimensions specified)

**Rectilinear Meshes**

- You only specify the grid (node) coordinates along each axes

**Uniform Meshes**

- You just provide grid (node) coordinates for minimum and maximum
Data import in AVS/Express

- Various formats directly readable
- Rd_Txt_Columns
- Database (ODBC / Oracle)
- AVS Field file format
- File import modules
- Custom readers

Formats directly supported

- AVS Field
  - Uniform, rectilinear, structured
- AVS UCD
  - Unstructured
- HDF5, Plot3D, netCDF, CGNF
- Standard CFD formats
- Text
  - Need a table_to_scatter/table_to_uniform module
- DXF, Polygon, Triangle, AVS Geom
- Geometries
- Images
  - BMP, GIF, JPEG, PBM, SGI RGB, TIFF

Import method choices

<table>
<thead>
<tr>
<th>Simple</th>
<th>Various directly readable formats</th>
<th>Simplest solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rd_Txt_Columns</td>
<td>Textual file. OK for Scatter or Line meshes</td>
<td></td>
</tr>
<tr>
<td>Database (ODBC and Oracle)</td>
<td>Read tables from a database</td>
<td></td>
</tr>
<tr>
<td>AVS Field file format</td>
<td>Convert data to FLD. Support uniform, rectilinear and structured meshes</td>
<td></td>
</tr>
<tr>
<td>File import modules</td>
<td>Read whatever you want and then combine them with field mappers</td>
<td></td>
</tr>
<tr>
<td>Custom readers</td>
<td>No constraints, but programming and knowledge of the FLD API needed</td>
<td></td>
</tr>
</tbody>
</table>
Most important user data import

AVS Field file format

Rd_Txt_Columns

The Read_Field Module

# AVS field file
# this is a header file for a structured field
#
# ndim = 3
dim1 = 40
dim2 = 32
veclen = 5

# data = float
field = irregular
label = density x-momentum y-momentum z-momentum stagnation

variable 1 file=./blntfinq.bin filetype=binary skip=28
variable 2 file=./blntfinq.bin filetype=binary skip=163868
variable 3 file=./blntfinq.bin filetype=binary skip=327708
variable 4 file=./blntfinq.bin filetype=binary skip=491548
variable 5 file=./blntfinq.bin filetype=binary skip=655388

coord 1 file=./blntfinx.bin filetype=binary skip=12
coord 2 file=./blntfinx.bin filetype=binary skip=163852
coord 3 file=./blntfinx.bin filetype=binary skip=327692

Default extension is:.fld

Various examples under express in data/field

Note: FORTRAN unformatted data can be read by using, for example:
variable 1 file=for0004.dat filetype=unformatted skip=32 stride=1

The Read_Field Module

# AVS field file
# this is a header file for a uniform field
#
ndim = 3
dim1 = 64
dim2 = 64
dim3 = 64
veclen = 1

# data = byte
field = uniform

variable 1 file=hydrogen.dat filetype=binary skip=3

Note: FORTRAN unformatted data can be read by using, for example:
variable 1 file=for0004.dat filetype=unformatted skip=32 stride=1
Rd_Txt_Columns example

- Import file conc.dat using Rd_Text_Columns (skip 1 line)
- Transform it into a scatter field using table_to_scatter_field (columns: x, y, z, value)
- Enhance visibility using: glyph+Diamond3D (normalize)

Time dependent data

- File list
- Multistep FLD
- Multistep UCD
- Animator
- Image Capture

Time dependent field

- AVS/Express extends the AVS/Express field ASCII description format to support time dependent (multistep) data. (Binary file input is not extended.)
- These extensions could be used to define steps in the data at which node data, and/or coordinate data can vary. The cell type, and the nspace and ndim values for the field cannot change.
- A string can be specified for each time step. The string will be displayed when the step is displayed. For example:
  - (hardcoded) time value=30.0sec
  - (from file) time file=cylinder.12 filetype=ascii offset=1
Time dependent field

% AVS field file
% ndim = 3
dim1 = 50
dim2 = 20
veclen = 1
data = float
field = irregular
label = vorticity
time file=cylinder/cylinder.1 filetype=ascii offset=1 variable 1 file=cylinder/cylinder.1 filetype=ascii skip=1 coord 1 file=cylinder/cylinder.mesh filetype=ascii skip=1 coord 2 file=cylinder/cylinder.mesh filetype=ascii skip=1E0
EOT

time file=cylinder/cylinder.2 filetype=ascii offset=1 variable 1 file=cylinder/cylinder.2 filetype=ascii skip=1 EOT

time file=cylinder/cylinder.3 filetype=ascii offset=1 variable 1 file=cylinder/cylinder.3 filetype=ascii skip=1 EOT

time file=cylinder/cylinder.4 filetype=ascii offset=1 variable 1 file=cylinder/cylinder.4 filetype=ascii skip=1 EOT
...(etc.)

Time dependent field modules

Read_Field
interp_time_step
extract_time_step
set_global_minmax

Time Data example

Exercise 4

Visualize 2D data
Workbook page 42
**Exercise 5**

Visualize 3D vector data

Workbook page 55
Exercise 6

Visualize volume data

Workbook page 77
Exercise 7
Visualize UCD data
Workbook page 85
Theory interlude 3

AVS/Express customization

1. Creating an application (if it is saved as runtime it does not need an AVS/Express installation to work)
2. Modifying standard modules
3. Grouping modules into macros (use DV* modules)
4. Writing V code directly
5. Integrating C / C++ / Fortran code

AVS/Express project structure

AVS Workshop - Mario Valle - 10/05/2012
avsenv and XP_PATH

- In the avsenv file various environment variable could be defined, but one is critical: XP_PATH
  - CACHE_SIZE=256
  - MSMSSERVER=C:\CSCS\msms_bin\msms_win32.exe
  - XP_PATH=C:\CSCS\$XP_ROOT
- XP_PATH defines the search order for all the Express pieces

\[ v\text{\templ.v} \rightarrow C:\CSCSlib \rightarrow C:\Express \]

\[ v\text{\templ.v} \rightarrow v\text{\templ.v} \]

Beware!

Save Application ≠ Save Project
Use Workspaces

Modified standard isosurface module to sweep the whole range of values
Create a user interface

Add a computing module

- Create a project
- Start the Add Module Wizard
- Remember to do a Save Project!
- Add your code to the skeleton
- Compile (express.dsw or make -f express.mk)
- Exec the new bin\pc\express or bin/linux/express

Add Module Wizard (1 of 3)
Add Module Wizard (2 of 3)

Add Module Wizard (3 of 3)

Interface between V and code

Generated by the wizard
Exercise 8

Modify isosurface module

- Read the AVS Field format file `hydrogen.fld`
- Show data limits with `bounds`
- Extract a slice with `orthoslice (axis 2)`
- Step from 0 to 63 using a `Loop`
- Connect the counter to `ortoslice.OrthoSliceParam.plane`

Exercise 9

Create small (active) user interface
Create active user interface

- Create a SingleWindowsApp
- Add a UImod_panel panel
- Insert a UIslider and a UIDial
- Connect their values
- What happens moving slider and dial?
- Bonus: UI Builder ➔ Layout Editor

Movie creation

- Animator module (under Main ➔ Data IO)
- Starts with the definition of key frames
- The module Image capture takes the frames and build MPEG or AVI movies

End of a nice day…

Thank you!
Remember, I’m mvalle@cscs.ch
Exercise

- Create a modulo `MyAdd` in C
- Create a modulo `MyAdd1` in C++ (two float `p1` and `p2` and float output `out`)
- Test it
- Find the differences between the two languages
- Enjoy! This is the last exercise

AVS/Express components

Data Viewer

Page 34
Visualization techniques

More than 500 techniques available
Support prototyping to find the most useful technique
Any technique can be adapted to suit specific user requirements

User data import

Various formats directly readable
Rd_Txt_Columns
Database (ODBC / Oracle)
AVS Field file format
Custom readers

Interactivity

Probe
ClickSketch
Edit Mesh
Draw Line
Draw Cursor
GDpick_process
Results presentation

- Print
  - normal and Vector Postscript
- Image
- Movies (MPEG)
- VRML
- Custom Writers

Exercise

- Try to use the visualization technique examples provided by AVS/Express
- For example try:
  - Library → Examples → Visualization → Orthoslice
- Open the example to look at its internal structure

My first network

- Application Type (Single, Multi, Default)
- Modules
- Connection ports (popup with info and help)
- Port colors
- User interface
- Save the application
Exercise

- Read file teapot.geo (Geom format)
- Visualize result
- Interact with the graphical object
- Save application as ex1.v
- Delete application and reload from file
- Bonus: add a fading background

Exercise

- Move the lights
- Change light type
- Which are the differences between:
  - Rotate object
  - Rotate lights
  - Rotate camera?

Most important user data import

- Rd_Txt_Columns
- AVS Field file format
Exercise

- Import file conc.dat using Rd_Text_Columns (skip 1 line)
- Transform it into a scatter field using table_to_scatter_field (columns: x, y, z, value)
- Bonus: glyph+Diamond3D (normalize)

Exercise

- Read the AVS Field format file splash.fld (open it to see the format)
- Create a surface plot using surf_plot
- Change the vertical scale
- Add bidirectional light
- Change datamap (bonus: ColormapEditor)
- Add Legend

Exercise

- Read the AVS UCD format file example2.inp
- Build vectors on the nodes: combine_vect
- Glyph + Arrow1 (scale 1e-5)
- Offset to deform the cube
- magnitude to color the deformed cube
Exercise

- Read the AVS Field format file hydrogen.fld
- Extract a slice using orthoslice (axis 2)
- Build a surface with surf_plot
- Read mandrill.x using Read_Image
- Spread it on the surface with texture_mesh

Movie creation

- Animator module (under Main → Data IO)
- Starts with the definition of key frames
- The module Image capture takes the frames and build MPEG or AVI movies

Exercise

- Load application ex1.v
- Add an animator module
- Move the graphical object and mark some key frames. Close the loop
- Playback
- Save as an MPEG movie ex1.mpg