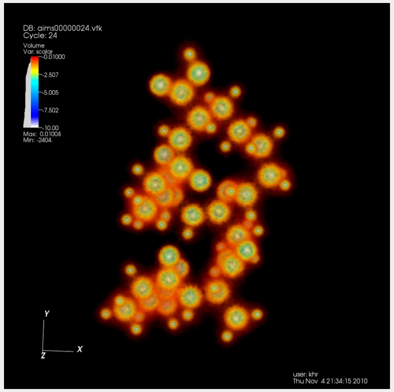
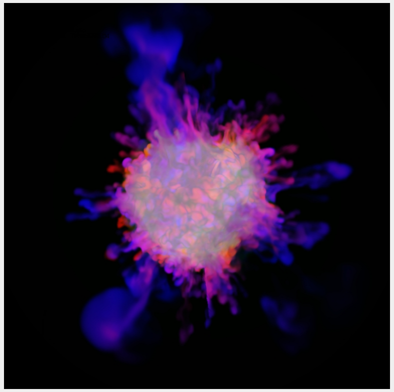
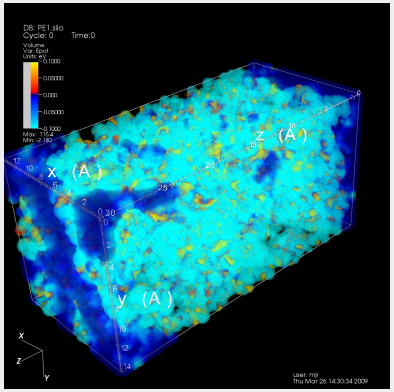
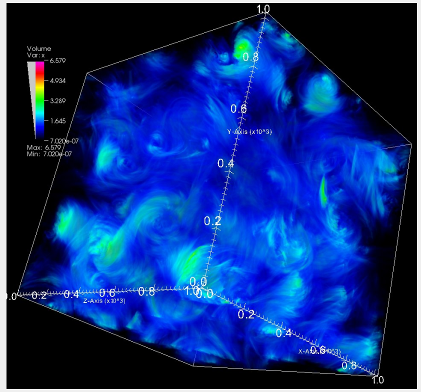
Introduction to VisIt

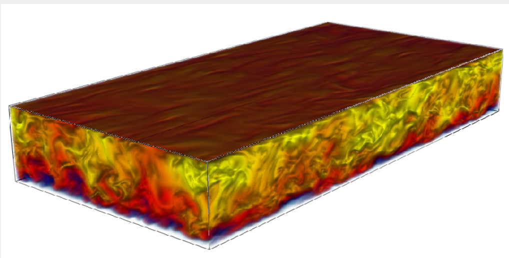
Markus Rampp
Computing Centre (RZG) of the Max-Planck-Society and IPP
markus.rampp@rzg.mpg.de

CINECA/PRACE Summer School of Scientific Visualisation
Bologna, June, 2012

Aims and claims of this lecture, main topics:

▷ why VisIt ? *decision-making aids from experience "in the field" (HPC)*

- sketch main features, capabilities (and deficiencies) of VisIt
- results & experiences from *real-world visualisation projects.*
  Our point of view:
  a) computing centre, HPC applications and visualisation support group
  b) scientific user
- *"what can be done ?"* ↷ *is it worth considering for my research ?*



by courtesy of V. Avsarkisov (TU Darmstadt)

▷ this is *not* primarily:

- about teaching *"how exactly this can be done"*
- a VisIt crash course / hands-on session (see `www.visitusers.org/index.php?title=Short_Tutorial`)
- about advertising VisIt (RZG has no interests in the VisIt business)

Overview

Outline:

1. tool overview & basic usage (GUI)
   - ▷ why VisIt ?
   - ▷ basic concepts
   - ▷ first impressions: a brief demo
2. advanced topics:
   - ▷ client-server mode: remote engine and data access
   - ▷ Python scripting
   - ▷ strategies for data-format handling
3. reference applications
   - ▷ show what can be done with VisIt
   - ▷ some explicit examples of VisIt usage
4. conclusions, discussion, Q&A

Part 1: Tool overview & basic functionalities *(VisIt 2)*

- Basic facts
- Why VisIt ?
- Concepts
- GUI

Tool overview

RECHENZENTRUM GARCHING

VisIt . . . is hard to google

- homepage: http://visit.llnl.gov/
- do not confound with VISIT - a Visualisation Toolkit http://www.fz-juelich.de/jsc/visit/

VisIt *(according to the VisIt homepage)* is . . .

- *a free [and open-source], interactive parallel visualization and graphical analysis tool for viewing scientific data on Unix and PC platforms [Windows, Mac OS].*
- *users can quickly generate visualizations from their data, animate them through time, manipulate them, and save the resulting images for presentations.*
- *VisIt can be used to visualize scalar and vector fields defined on two- and three-dimensional (2D and 3D) structured and unstructured meshes.*
- *VisIt was designed to handle very large data set sizes in the terascale range and yet can also handle small data sets in the kilobyte range*
- *originated from Lawrence Livermore National Laboratory (ASC/DOE)*
- *distributed project, developed by several groups: VACET (SciDAC), ASC, GNEP*

VisIt *(according to our own experience at RZG)* is . . .

- comprehensive
- well documented: web pages, manuals, WiKi
- well supported: responsive mailing lists, feedback from developers, user community: www.visitusers.org
- extensible: e.g. data-reader plugins easy to deploy
- widely used in the scientific community, installed at many computing centres
- easy to install (executables for Linux x86_32/x86_64, Windows XP/Vista/7, Mac OS X)
- quite mature but also still under active development (https://wci.llnl.gov/codes/visit/new.html):
  - May, 2012 - VisIt 2.5 released
  - Nov, 2011 - VisIt 2.4 released (2.4.1: Feb 2012, 2.4.2: Mar 2012)
  - Jun, 2011 - VisIt 2.3 released (2.3.1: Jul 2011, 2.3.2: Sep 2011)
  - Jan, 2011 - VisIt 2.2 released (2.2.1: Mar 2011, 2.2.2: Apr 2011)
  - Sep, 2010 - VisIt 2.1 released (2.1.1: Nov 2010, 2.1.2: Dec 2010)
  - May, 2010 - VisIt 2.0 released (2.0.1: Jun 2010, 2.0.2: Aug 2010)

  - Aug, 2009 - VisIt 1.12 released
  - Nov, 2008 - VisIt 1.11 released
  - Aug, 2008 - VisIt 1.10 released
  - [. . .]

VisIt is *not* . . .

- ▷ necessary (sufficient?) for analyzing low-dimensional data ($< 3D$)
  ↷ python, GNU-R, idl, matlab, . . .
- ▷ a complete replacement for comprehensive quantitative analysis
  ↷ python, GNU-R, idl, matlab, . . .
- ▷ a world-wide community effort for developing comprehensive repositories like GNU-R

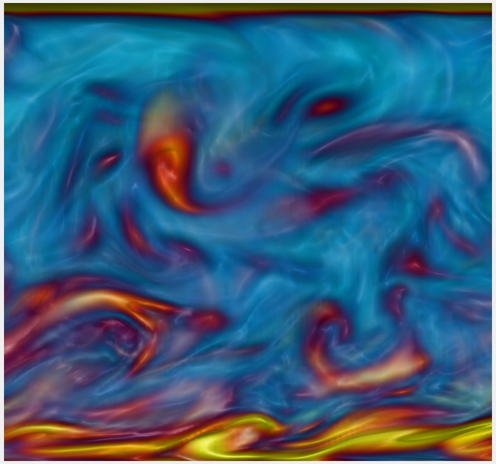
Alternative tools (for general-purpose visualisation of multi-D data)

- ▷ *ParaView*: very similar in concept and functionality
- ▷ *VAPOR*: tailored to — very efficiently — handle 3D data on cartesian grids
  note: VAPOR is *not* restricted to earth/climate data
- ▷ *Voreen*: comparable to VAPOR (no preprocessing necessary)
- ▷ *Amira/Avizo*: commercial, with domain-specific extensions, e.g. earth-mapped data
- ▷ . . .

Why VisIt at RZG ? — basic experiences:

Computing Centre's point of view:

- ▷ 2007: survey of freely available tools: VisIt, (ParaView)
- ▷ VisIt produced good results in short time
- ▷ flexible client-server architecture: allows running GUI, and rendering, data access on different machines
- ▷ continuing positive feedback from a (experienced and critical) scientific users from different domains
- ▷ a main workhorse employed by RZG visualisation team



by courtesy of V. Avsarkisov (TU Darmstadt)

Scientific user's point of view:

- ▷ steep (i.e. *efficient*) learning curve (knowledge transfer: RZG visualisation team ⟶ users)
- ▷ comprehensive set of standard functionalities
- ▷ data handling: many supported formats
- ▷ allows to produce publication-quality plots along the way
- ▷ promotes gradual transition from GUI-based, interactive work to Python scripting

Tool overview

Basic concepts

- ▷ *plots* (ways to render data): $> \mathbf{20}$
  - pseudocolor, volume-rendering, contour, vector, scatter, parallel coordinates, histogram, ...
- ▷ *operators* (ways to manipulate data, "filters"): $> \mathbf{40}$
  - isosurface, isovolume, clip, slice, project, (coordinate-)transforms, ...
- ▷ *file format readers* (ways to import data): $> \mathbf{80}$
  - HDF5/XDMF, NetCDF, (common CFD fmts), PDB, VTK, BOV, LAMMPS, GADGET, ...
- ▷ *queries* (ways to extract quantitative information): $> \mathbf{50}$
- ▷ *expressions* (ways to create derived quantities)
  - simple expression syntax: math, logical, relational, mesh, ...
  - recently added: python syntax ...

Other features/utilities (not fully covered here)

- ▷ *animation* of time-dependent databases
- ▷ *movie generation*: simple, keyframing
- ▷ *session management:* save/restore status of interactive session
- ▷ *it-situ visualisation:* instrumentation of simulation codes

Basic functionalities of the GUI: Plots & Operators

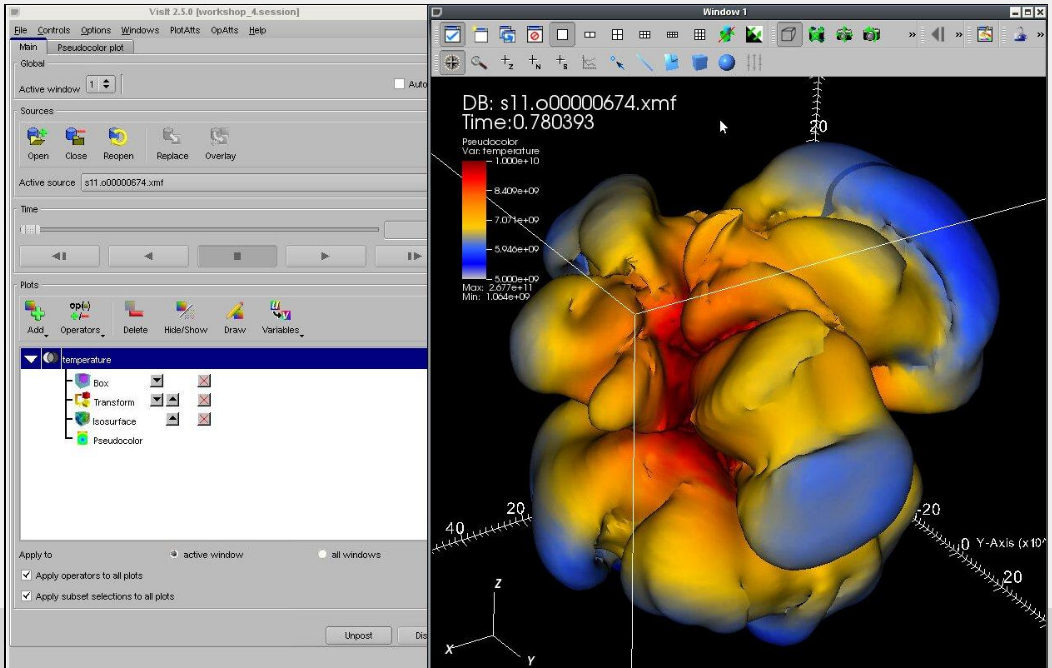
Intuitive concept:

- ▷ *Plots*: pseudocolor, volume, vector, ...
- ▷ and *Operators*: isosurface, transform (e.g. coordinate), clip, box, revolve, ...
- ▷ options for plots and operators (double-click or pull down menu)
- ▷ multiple (successive) operators per plot
  hint: reduce as early as possible, e.g.:
  1: box, 2: transform (not vice versa)
- ▷ multiple plots per window

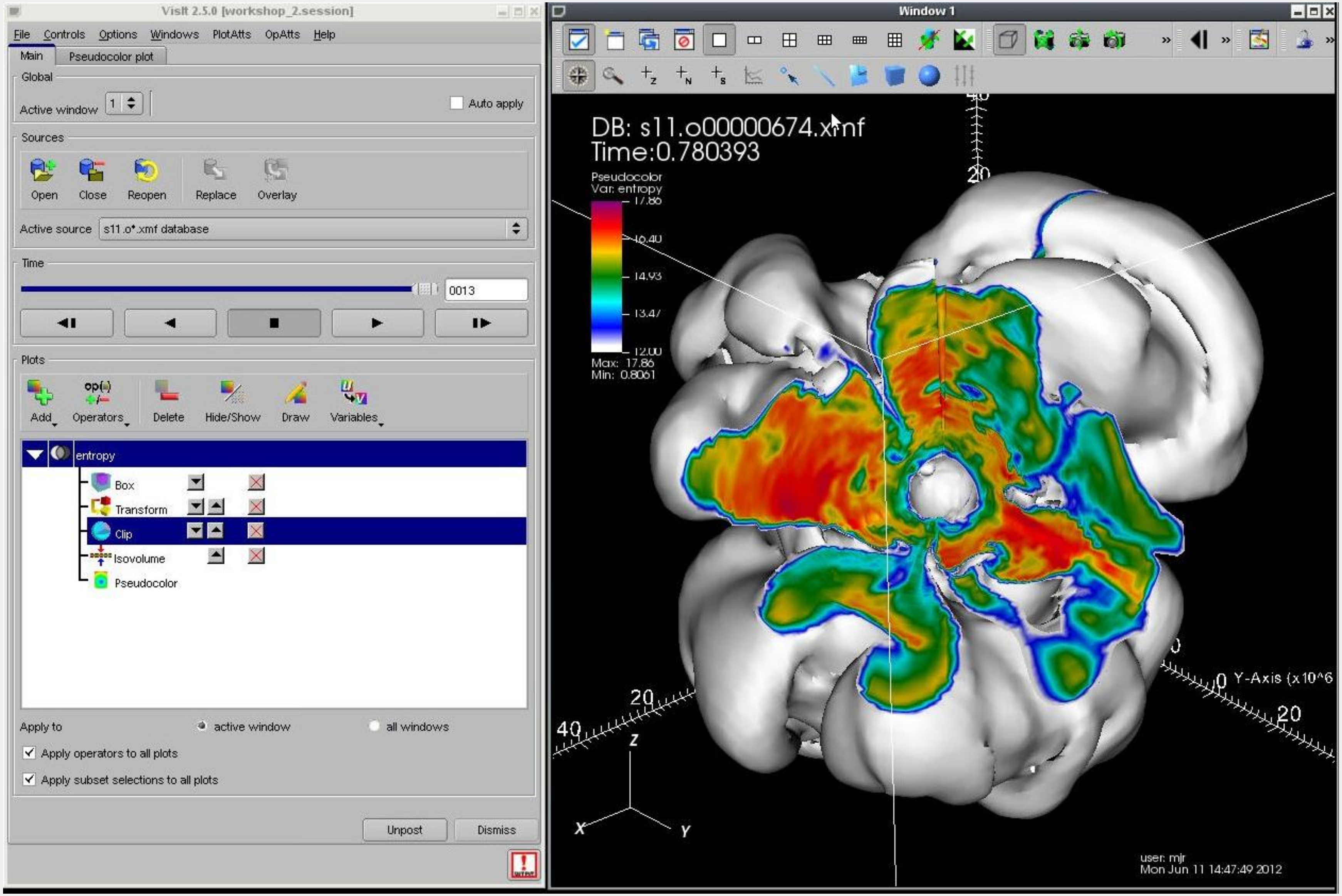
Example:

isosurface of a scalar variable $\boldsymbol{X}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, colored by the value of another scalar variable $\boldsymbol{Y}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ ($\boldsymbol{X}$, $\boldsymbol{Y}$ defined on the same mesh)

1. create pseudocolor plot for $\boldsymbol{Y}$
2. apply isosurface operator for $\boldsymbol{X}$ (instead of default: $\boldsymbol{Y}$)

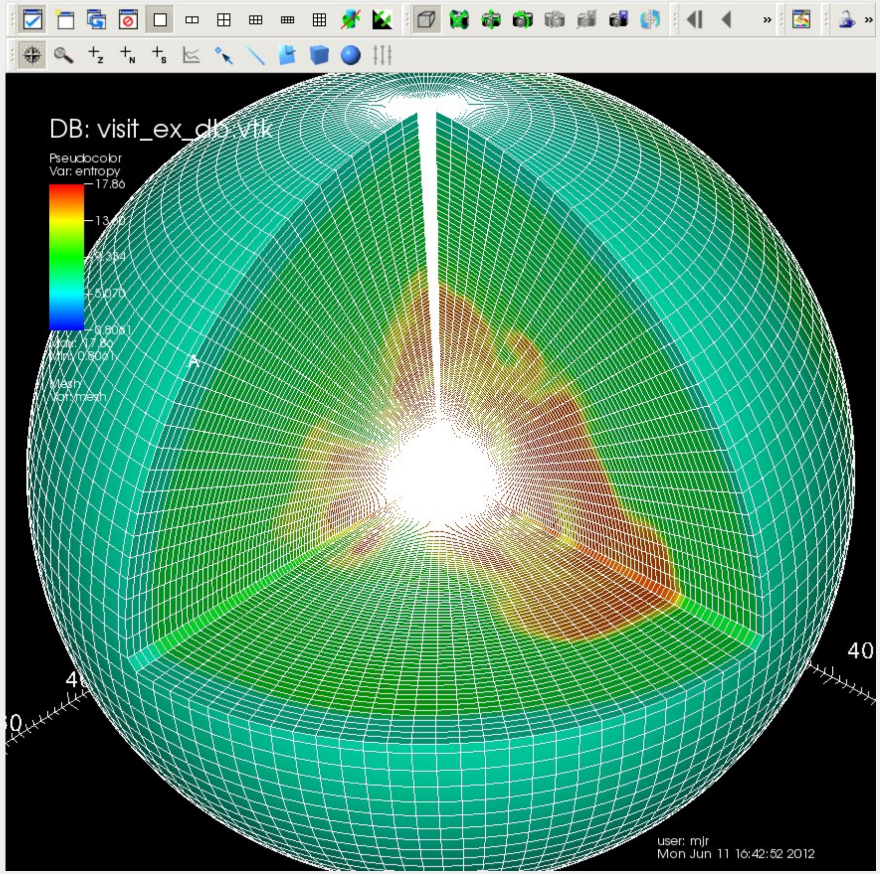


Basic functionalities of the GUI: Appearance



The demo dataset

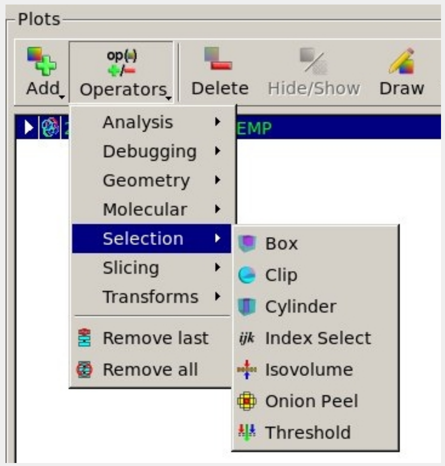
- ▷ example data taken from 3D Supernova simulations of Hanke et al. (2011)
- ▷ *Data and grid:*
  - spherical coordinates r, θ, ϕ
  - scalar variable $s(r, \theta, \phi)$
  - a subset ($0 \le r \le R_0$, single timestep) of the complete dataset ($400 \times 60 \times 120$ zones)

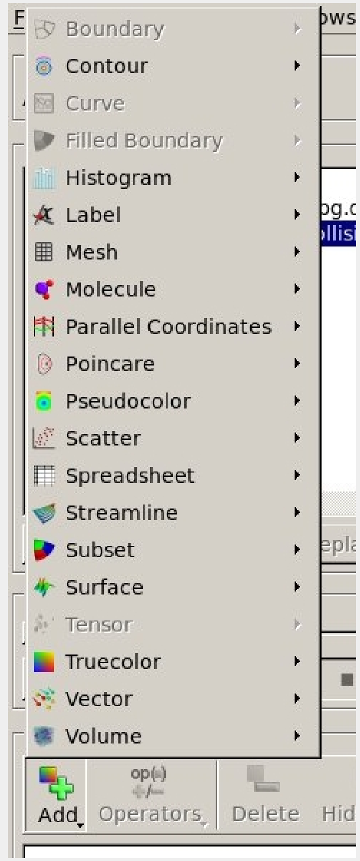


RECHENZENTRUM GARCHING

A brief demo covering:

- ▷ *plots:*
  - pseudocolor
  - volume: splatting, (ray-casting)
- ▷ *operators*
  - transform coordinates: "spherical" ⟶ cartesian.
  - isovolume
  - slice, clip, box
- ▷ *interaction tools*
  - navigate (default) 
  - zoom 
  - plane 
  - line 
  - (zone pick )
  - (node pick )





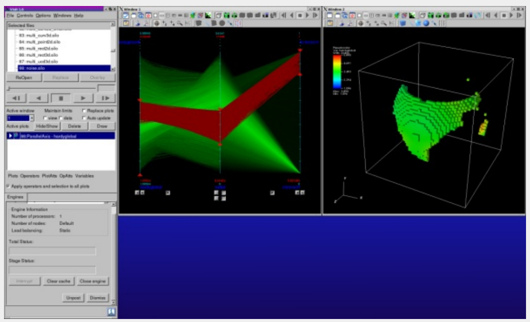
Part 2: Advanced usage *(VisIt 2)*

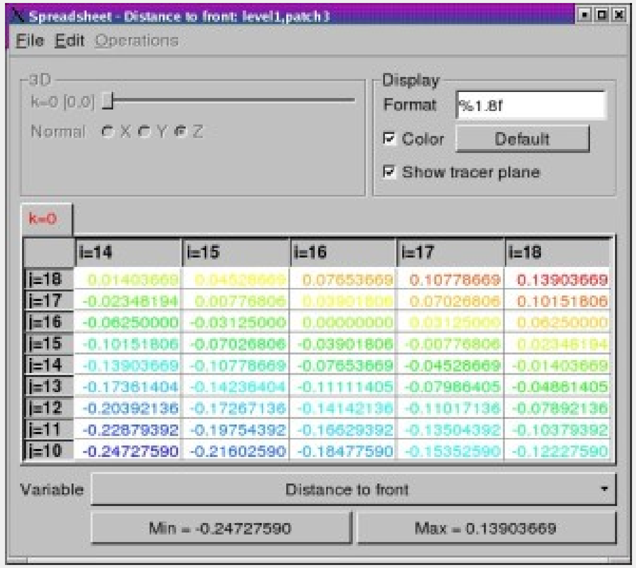
- advanced GUI features
- Python scripting
- data formats
- color tables
- client-server mode

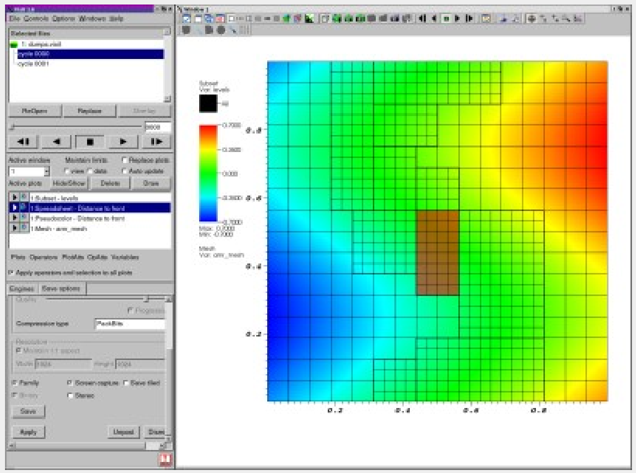
RECHENZENTRUM GARCHING

Advanced features:

- ▷ *expressions*: expression language
- ▷ *animation*: simple movie creation
- ▷ *sessions*: save/restore interactive sessions
- ▷ *movies*: keyframing
- ▷ *quantitative analysis*:
  - spreadsheet
  - zone-pick
  - queries
  - correlations of multivariate data

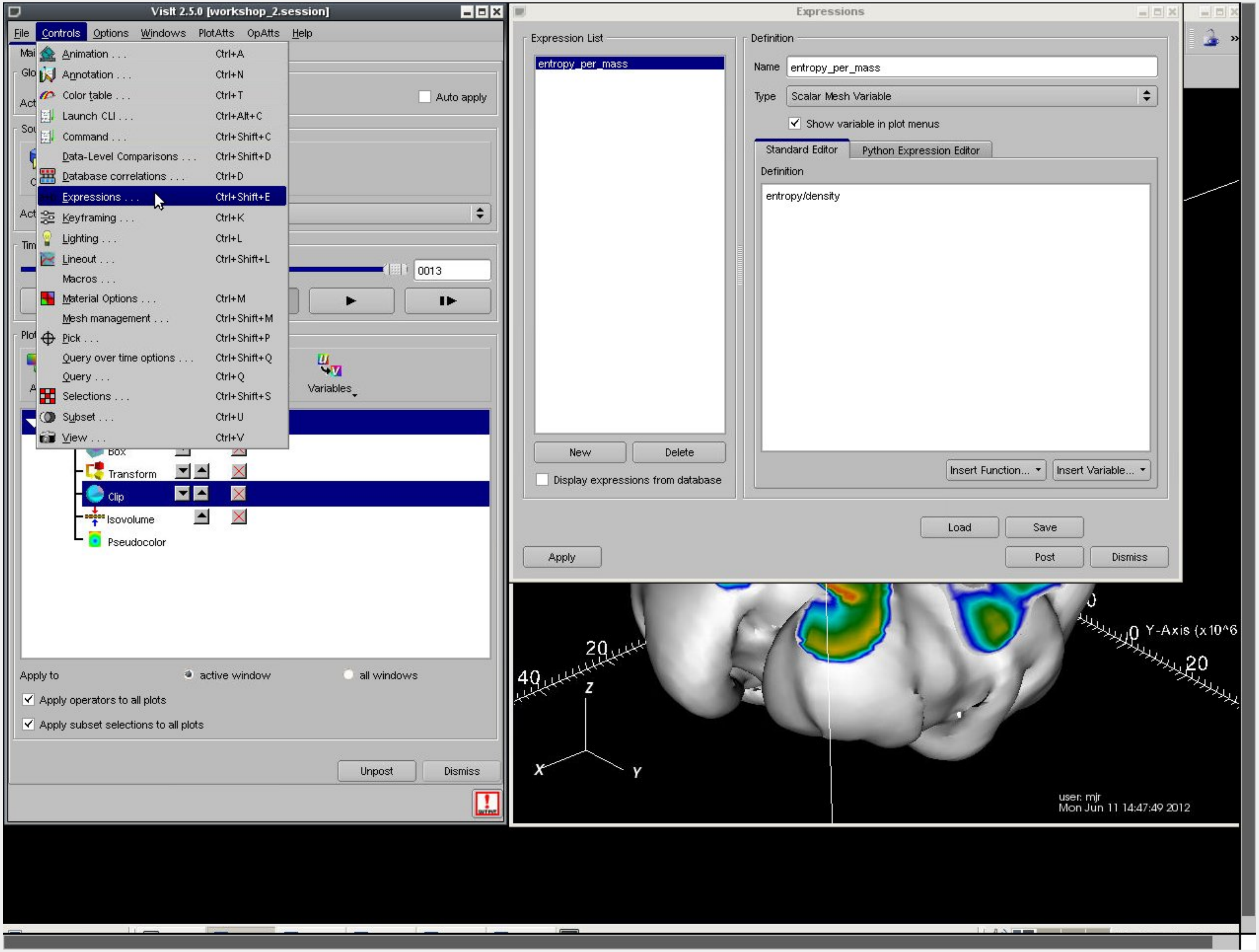


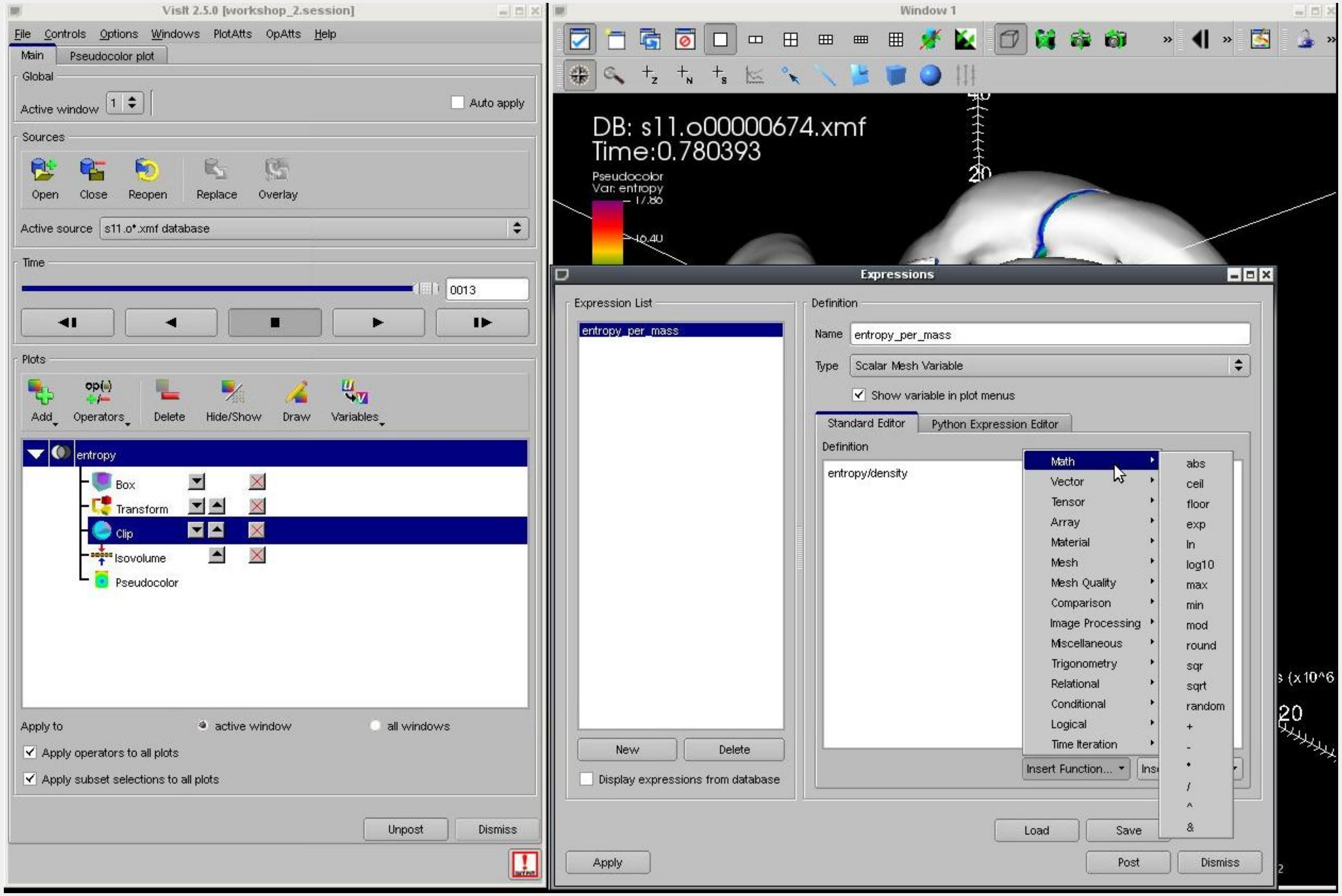




(https://wci.llnl.gov/codes/visit/screens3.html)

Advanced topics: Expressions





▷ allows simulation codes to dump only non-redundant physical quantities

▷ basic expression syntax: math, logical, relational, mesh, . . .

▷ new: Python expression editor

Advanced topics: Python scripting

Startup:

```
~>visit -cli -nowin -s example.py
```

List available functions:

```
~>visit -cli -nowin
Running: cli2.0.0 -nowin
Running: viewer2.0.0 -host 127.0.0.1 -port 5600 -noint -nowin
Python 2.6.4 (r264:75706, Mar 23 2010, 16:35:33)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-44)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> dir()
['ActivateDatabase', 'AddArgument', 'AddColorTable', 'AddOperator',
...
'GetQueryOutputValue', 'GetQueryOutputXML', 'GetQueryOverTimeAttributes', 'GetRenderingAttributes',
'GetSaveWindowAttributes', 'GetTimeSliders', 'GetUltraScript', 'GetView2D', 'GetView3D',
...
'SetPrinterAttributes', 'SetQueryFloatFormat', 'SetQueryOverTimeAttributes', 'SetRenderingAttributes',
'SetSaveWindowAttributes', 'SetTimeSliderState', 'SetTreatAllDBsAsTimeVarying', 'SetTryHarderCyclesTimes',
'SetUltraScript', 'SetView2D', 'SetView3D',
...
'__visit_script_file__', '__visit_source_file__', '__visit_source_stack__']
>>>
```

Query/modify objects with accessor methods GetXXX/SetXXX:

```
>>> GetView3D()
viewNormal = (0, 0, 1)
focus = (0, 0, 0)
viewUp = (0, 1, 0)
viewAngle = 30
parallelScale = 0.5
nearPlane = -0.5
farPlane = 0.5
imagePan = (0, 0)
imageZoom = 1
...
>>>
```

Prototypical example: "flyaround"

Python fragment for rotating an object (resp. animating the camera):

```
import math

OpenComputeEngine("localhost",("-np","4")) # open a (parallel) compute engine
OpenDatabase("localhost:/vizdata/mjr/HOTB/data/b0123dDZ_0656.silo") # open a single data file

AddPlot("Volume","Ni56") # volume plot for variable named "Ni56"

DrawPlots() # required once for proper View3D initialisation

c = GetView3D() # get a reference to the View3D object

s = SaveWindowAttributes() # instantiate a new WindowAttributes object
s.format = s.JPEG
s.width = 1024
s.height = 1024
s.screenCapture = 0
SetSaveWindowAttributes(s) # do not forget this for newly created instances

nsteps = 100
for i in range(0,nsteps):
    phi = 2*math.pi*(float(i)/float(nsteps-1))
    c.viewNormal = (math.cos(phi),math.sin(phi), 0.2)

    s.fileName = "flyaround_"+str(i)
    SaveWindow()

DeleteActivePlots()
CloseComputeEngine()
```

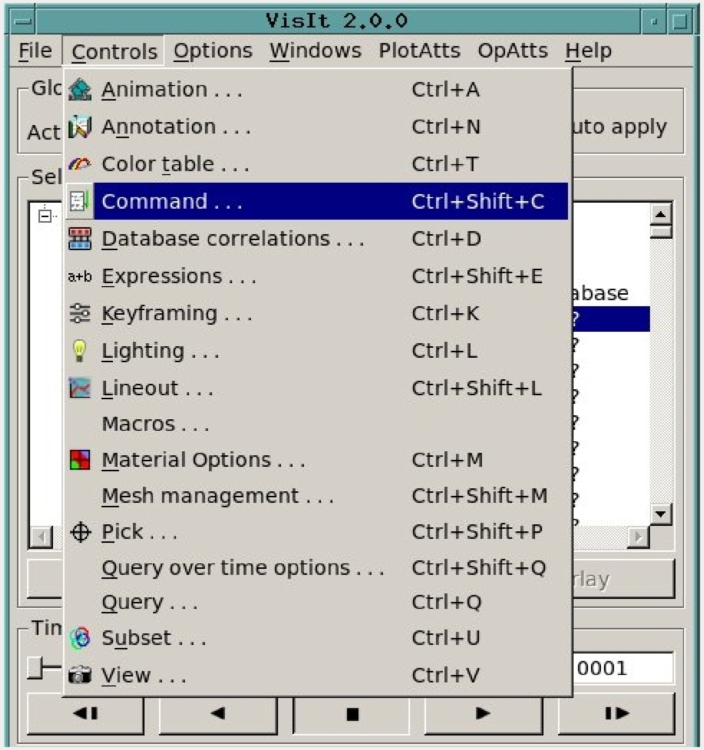
▷ see the following examples at http://visitusers.org/index.php?title=Category:Scripting

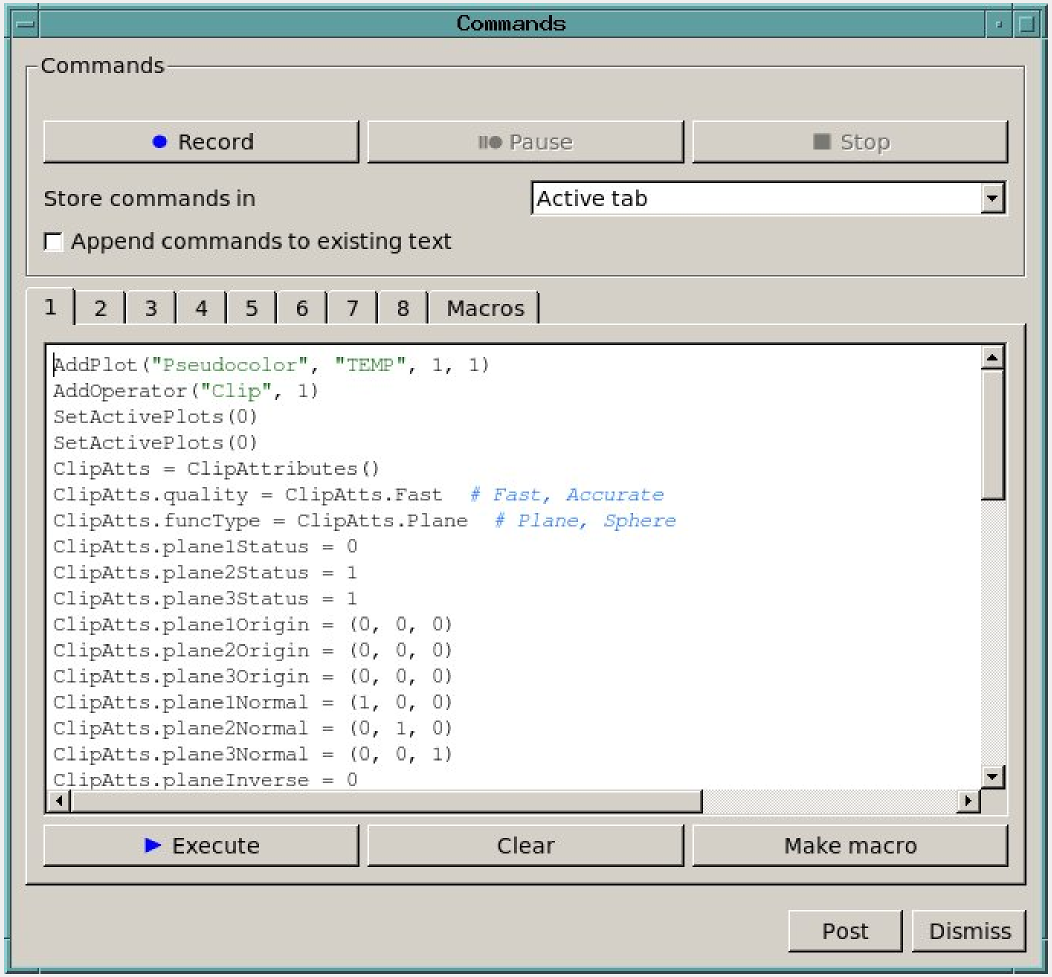
"Visit-tutorial-python-fly"

"Fly through"

"Python examples", *"Creating a color table"*, . . .

Hints for getting started with Python scripting





- ▷ use the dialog Controls → Command . . .
- ▷ . . . to interactively and iteratively create Python code
- ▷ paste code into your text/Python editor
- ▷ cf. the *VisIt Python Interface Manual*
  https://wci.llnl.gov/codes/visit/manuals.html

Advanced topics: Colortables

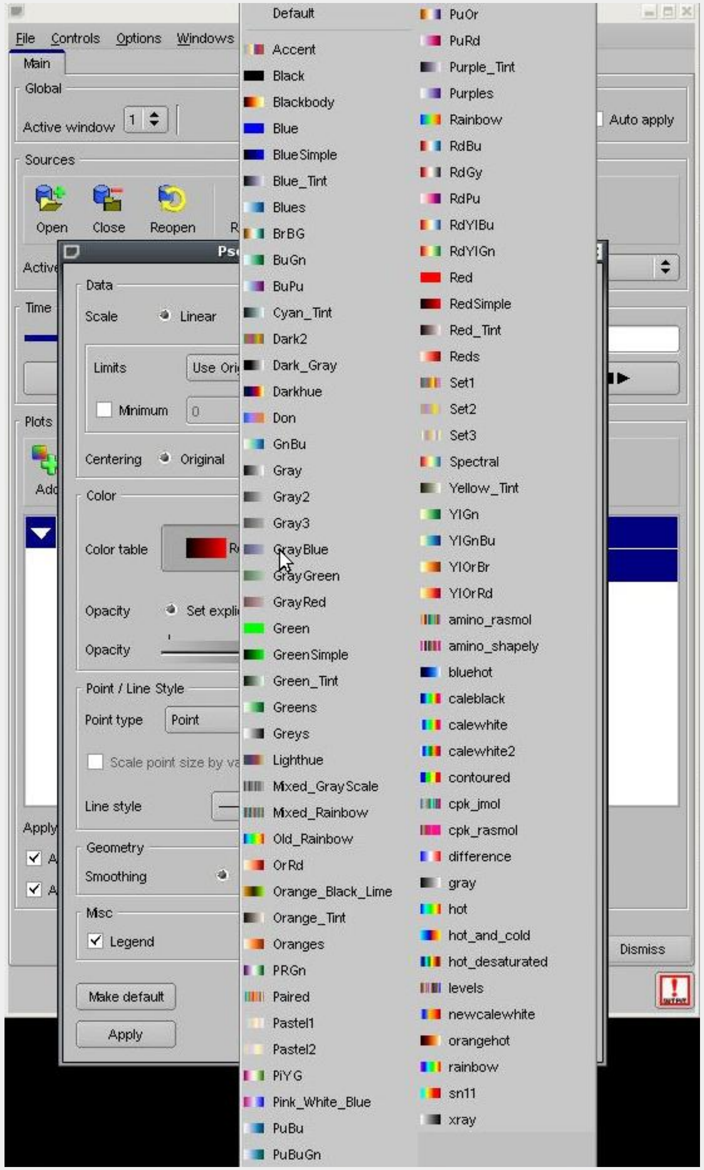
▷ latest VisIt 2.5 release added many new colortables

▷ *new*: useful preview icons for chosing colortable in plot options

▷ VisIt allows to create/manipulate colortables:

- *interactively:* by manipulating existing colortables
- *programmatically:* via Python interface
- *externally:* via editing colortable files

▷ additional colortables can be stored in userspace



RECHENZENTRUM GARCHING

- colortables in XML format defined by control points: position $\in [0, 1]$, (RGB colors, alpha) $\in ([0, 255]^3, [0, 255])$

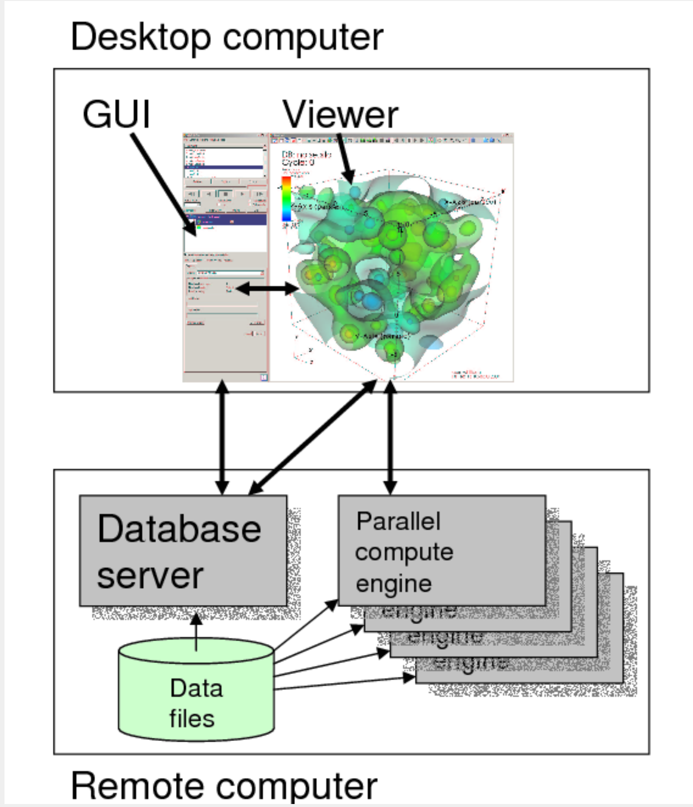
```
<?xml version="1.0"?>
<Object name="ColorTable">
    <Field name="Version" type="string">2.0.0</Field>
    <Object name="ColorControlPointList">
        <Object name="ColorControlPoint">
        </Object>
        <Object name="ColorControlPoint">
            <Field name="colors" type="unsignedCharArray" length="4">0 0 127 255 </Field>
            <Field name="position" type="float">0.333</Field>
        </Object>
        <Object name="ColorControlPoint">
            <Field name="colors" type="unsignedCharArray" length="4">0 127 255 255 </Field>
            <Field name="position" type="float">0.666</Field>
        </Object>
        <Object name="ColorControlPoint">
            <Field name="colors" type="unsignedCharArray" length="4">255 255 255 255 </Field>
            <Field name="position" type="float">1</Field>
        </Object>
    </Object>
</Object>
```

- simple XML format facilitates conversion or creation (e.g. Python script for converting Amira/Avizo colortables)

```
#! /usr/bin/env python

# Program: ctconvert.py
# Creator: Jeremy Meredith
# Date:    February 19, 2009
#
# Convert sampled color tables from one of a few input formats into
# VisIt's format, choosing an optimal selection of some number of
# control points.  (The number of control points is chosen by the
# user, though something between 5 and 10 does well for many
# common types of color table creations.)
#
# It currently supports already sampled color tables in Amira/Avizo
# formats.  It could easily support other sampled color table types,
[...]
```

Concepts

- start GUI on PC, access data and CPU/GPU power on other, "remote" server
- enables parallel rendering (MPI based)
- requires source installation (MPI) on the remote rendering platform (not trivial !)
- configuration via host-profiles (GUI, or pre-configured by computing center)
- setup for visualisation cluster:
  - display node (GPU) + render nodes (CPU, GPU)
- can be used as a (proprietary) "remote-rendering" solution:
  - install and run GUI on your laptop ...
  - ... to access and render data at the computing centre
  - probably requires VPN (firewall)



taken from: http://visit.llnl.gov/

Advanced topics: strategies for handling data formats

Explicit data conversion

+ quick (& dirty) programming: copy/paste from I/O statements in simulation code
+ allows some basic postprocessing and/or data reduction
- data duplication
- dirty programming

▷ which format? ↷ `Silo` (VisIt's "proprietary" data format), `HDF5`, `VTK`, ...

Development of a VisIt database plugin

+ no data duplication, no additional preprocessing step
+ plugin is dynamically loaded (code can reside under `$HOME`)
- development requires C programming and compilation against a VisIt installation
- not portable to other tools

Adaptation of simulation code I/O

+ no data duplication, no additional preprocessing step
+ can promote portability to other tools (depending on chosen format)
- implications for software management

▷ which format ? ↷ `HDF5` (requires `XDMF` or alike), `VTK`

Part 3: Reference applications

- Background: RZG/MPG visualisation infrastructure
  (PRACE context: cf. presentaton by P. Melis/L. Calori)

- Selection of reference applications

Max-Planck-Society (MPG)

▷ more than 80 institutes all over Germany, devoted to fundamental research:

- The Chemistry, Physics and Technology Section
- The Biology and Medicine Section
- The Humanities Section

▷ 2 major HPC centers:

- DKRZ, Hamburg (climate and earth sciences)
- RZG, Garching (materials and bio sciences, plasma physics, astrophysics)

Rationale for centralized visualisation:

▷ a *necessity for a HPC centre* rather than an optional service

- huge amounts of output data produced by HPC simulations
- transfer of raw data for local analysis & visualisation no more possible
- even dumping the RAM is becoming prohibitive due to I/O constraints
  ↷ *in-situ* visualisation (not covered here)
- visualisation requires HPC-like resources (specialized hardware, housing, . . .)
- requires substantial expertise on methods, software, . . . ↷ sustainability issues

Mission of RZG:

- ▷ main targets: interactive, remote data exploration & analysis, presentation, . . .
- ▷ support for adaptation and instrumentation of simulation codes
- ▷ guidance for selection, adoption and usage of analysis & visualisation software
- ▷ central software and hardware infrastructure for remote visualisation
- ▷ dedicated support for individual (particularly demanding) visualisation projects

Main conceptual challenges:

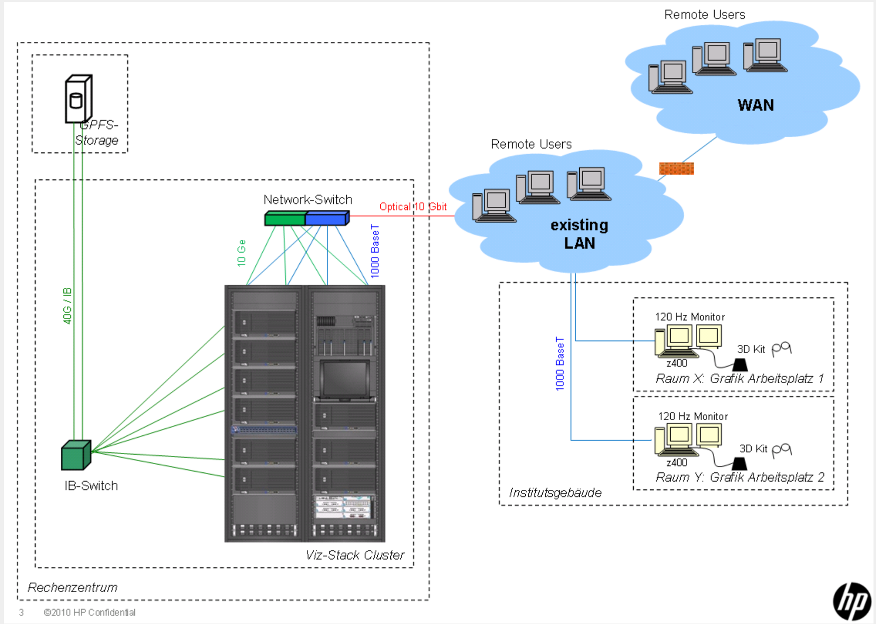
- ▷ broad range of disciplines in MPG: Plasmaphysics, Astrophysics, . . . , comp. Biology
  - ↷ many different scientific contexts
  - ↷ variety of simulation codes: "home-grown", commercial, open-source, third-party, . . .
  - ↷ non-standardized, heterogeneous data structures and formats
  - ↷ "legacy" analysis pipelines, . . .
- ▷ massive datasets from HPC simulations:
  (*massive:* amount of raw data, memory requirements, complexity)
  - ◦ multidimensional (3D + time), multi-variate data
  - ◦ "unusual" grids: meshless data, special curvilinear coordinates, . . .

Focus:

- ▷ enable our (geographically dispersed) scientific users to perform complex visualisation tasks *without special technical prerequisites (software, hardware)* ↷ *remote* visualisation

Hardware overview (Hewlett Packard)

- ▷ 5 "standard" visualisation nodes each equipped with:
  - 2 Intel quadcore CPUs: 8 cores, 144 GB RAM
  - 2 NVidia FX 5800 graphics cards
- ▷ 1 "high-end" visualisation node:
  - 4 Intel hexacore CPUs, 24 cores, 256 GB RAM
  - 2 NVidia FX 5800 graphics cards
- ▷ 1 login node: `viz00.rzg.mpg.de`
- ▷ dedicated disk system (GPFS, ≃ 30 TB)
- ▷ GPFS filesystem /ptmp of HPC systems mounted
- ▷ 2 graphics workplaces (active stereo) in RZG offices



Software stack (HP, open source)

- ▷ SLES 11 (RZG standard cluster setup), VizStack middleware (GPUs, X-servers, . . .)
- ▷ Web-based reservation system (HP)
- ▷ Remote rendering solution: VirtualGL/TurboVNC

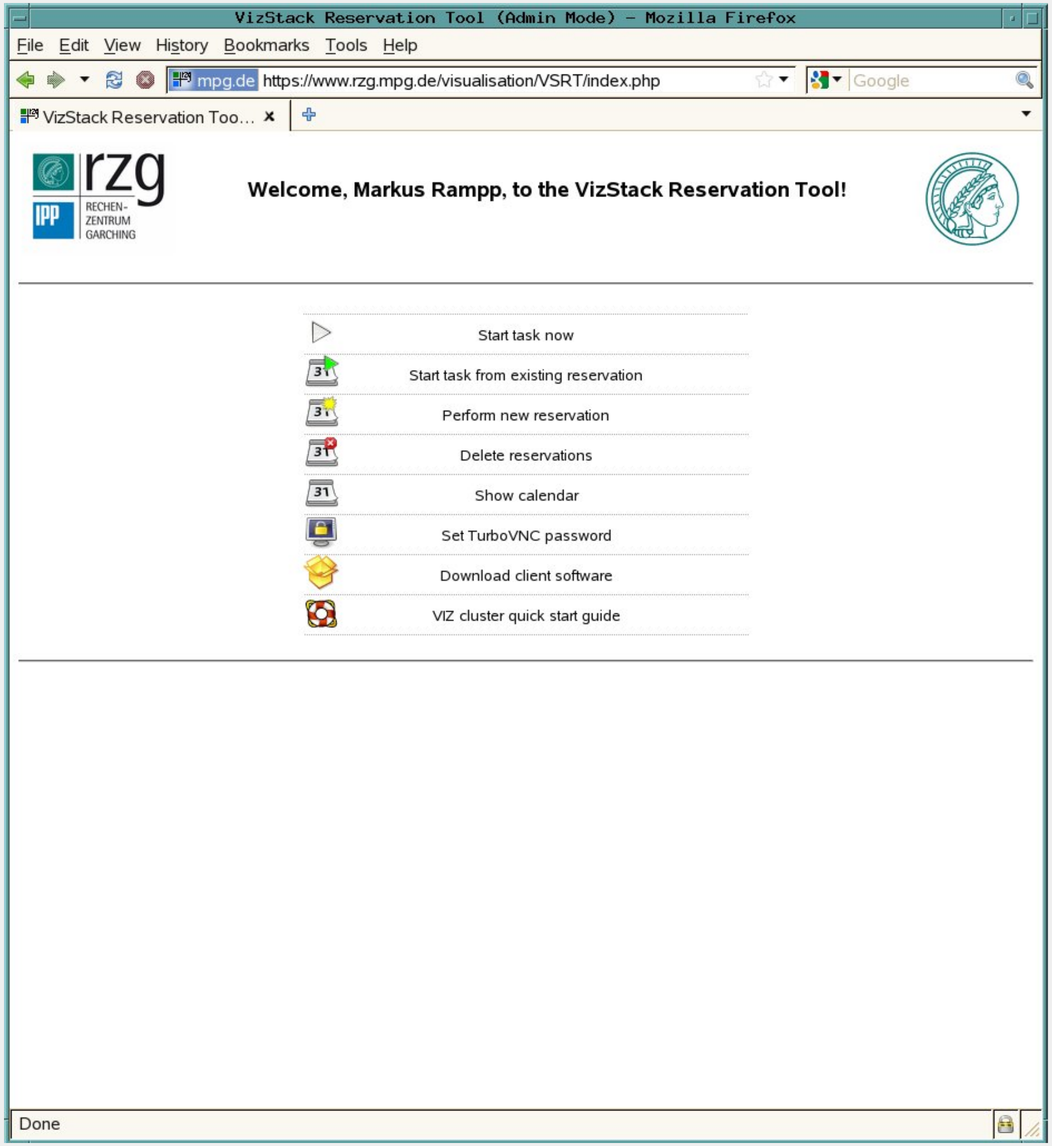
Hardware

Graphics workplace (with active stereo)

- ▷ remote stereo rendering (not a graphics workstation !)
- ▷ 120 Hz display (rhs.) with shutter glasses + control monitor (lhs.)

VizStack Reservation System (VSRT)

- ▷ https://www.rzg.mpg.de/visualisation/VSRT/
- ▷ authentication with RZG Kerberos
- ▷ main menu for
  - starting visualisation sessions
  - reservation management
  - calendar
  - VNC passwords
  - client software
  - documentation

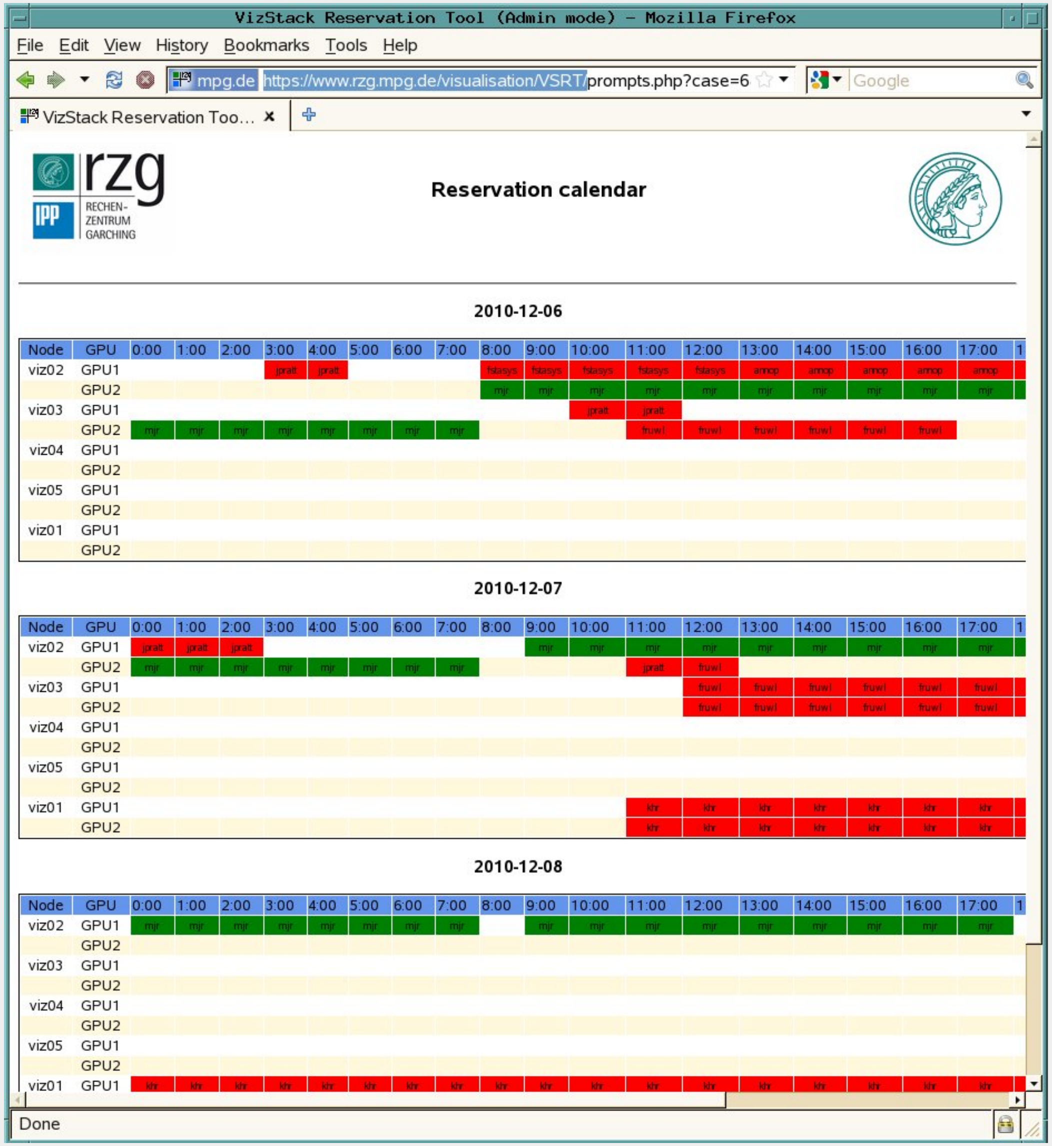


VizStack Reservation System (VSRT)

- reservation calendar:

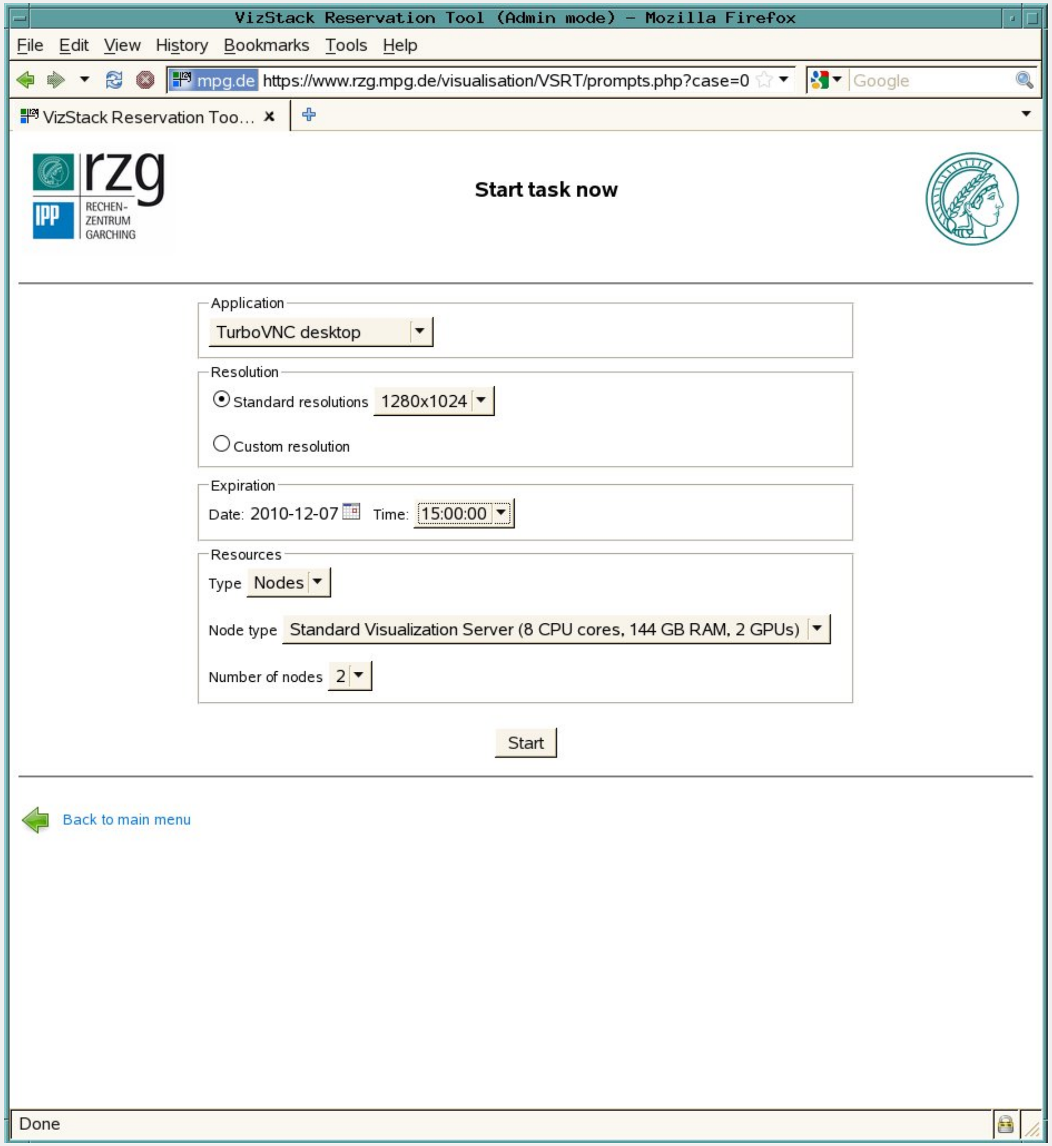
  green: my reservations

  red: other users



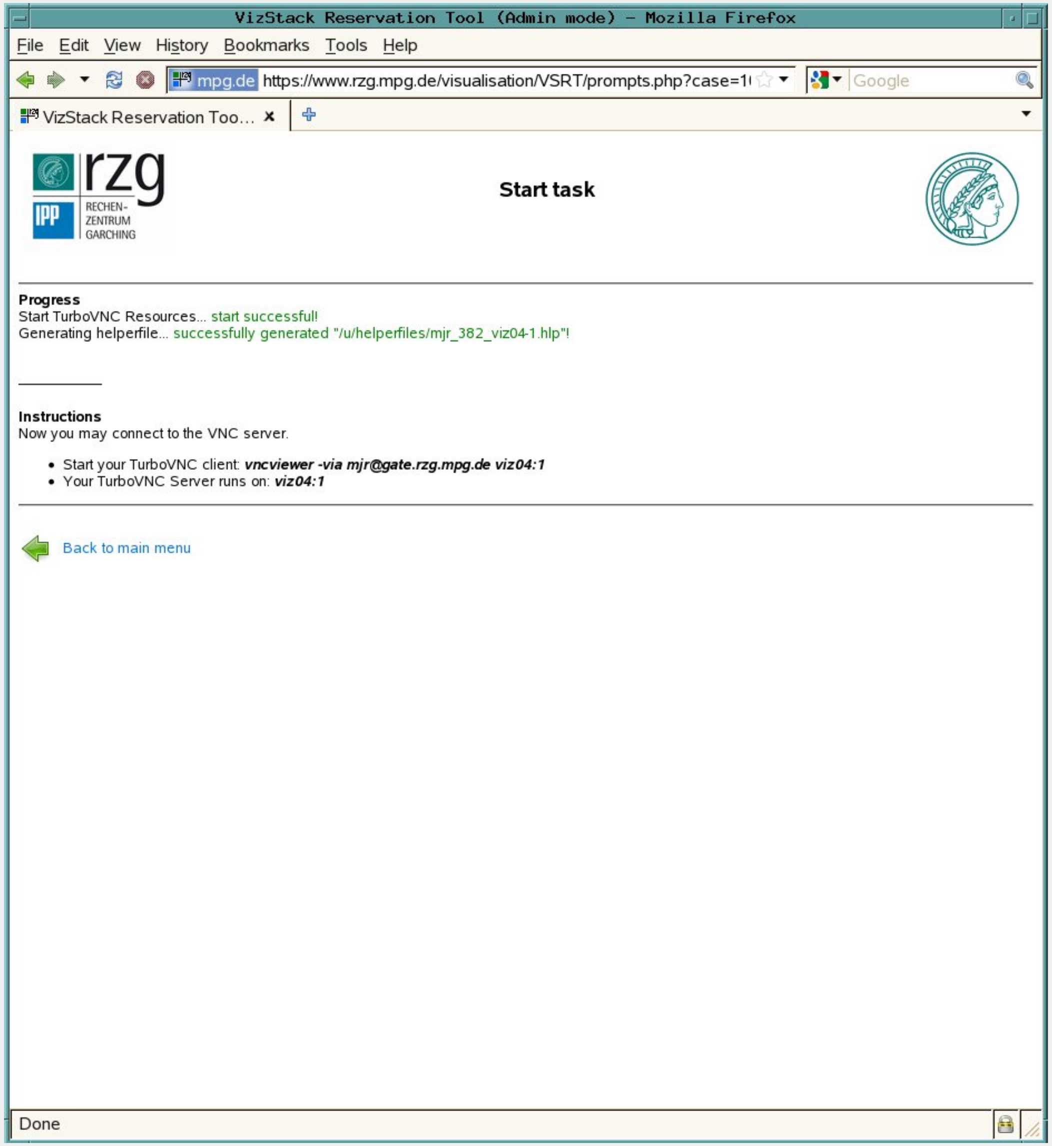
VizStack Reservation System (VSRT)

- ▷ start session
  - immediately (provided resources are available). . .
  - . . . or based on an *"existing reservation"*
- ▷ start remote visualisation application with a few clicks
- ▷ each session is mapped to (at least) 1 GPU
- ▷ additional nodes (+GPUs) can be added/reserved
- ▷ note: sessions cannot be extended beyond selected expiration date (*work in progress*)



VizStack Reservation System (VSRT)

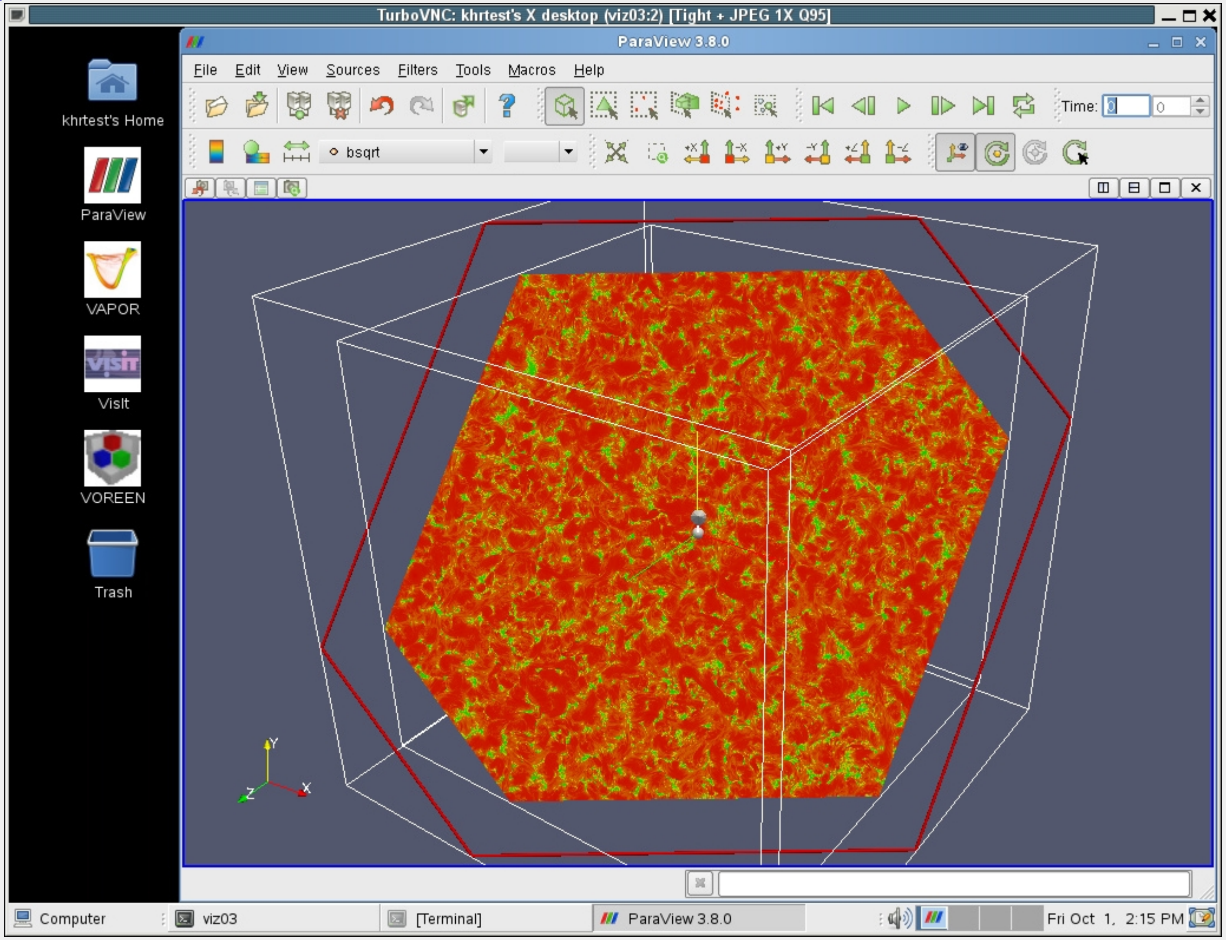
- ▷ start task, e.g.:
  `vncviewer -via gate.rzg.mpg.de viz04:1`



RECHENZENTRUM GARCHING

Remote desktop (via TurboVNC)

- a standard Gnome desktop in a separate window
- desktop icons for main applications
- preconfigured according to session properties (number of GPUs, CPUs)



Visualisation software

Software for interactive data visualisation and analysis

- ▷ VisIt
- ▷ Paraview
- ▷ VAPOR
- ▷ Voreen
- ▷ IDL
- ▷ Avizo (formerly Amira)

Libraries and Tools

- ▷ SILO: a library for reading and writing a wide variety of scientific data to binary files (cf. VisIt).
- ▷ VTK
- ▷ NCAR Graphics
- ▷ Grace (Xmgrace)
- ▷ gnuplot
- ▷ GNU R

Special-purpose Software

- ▷ Splotch: a (non-interactive), parallel ray tracer for SPH data.
- ▷ Blender: an open source, cross platform suite of tools for 3D creation.
- ▷ POV-Ray: a freeware multi-platform ray-tracing package.
- ▷ VMD (Visual Molecular Dynamics): a molecular graphics software.

Application support

Documentation

- ▷ http://www.rzg.mpg.de/visualisation/

Project support

- ▷ dedicated support for visualisation projects at different levels:
  - from basic "first level" support to comprehensive visualisation and analysis tasks
  - requires (considerable) insight to scientific domain
  - several completed and ongoing projects, in close collab. with the users/scientists: http://www.rzg.mpg.de/visualisation/scientificdata/projects
- ▷ contact: visualization@rzg.mpg.de

Training

- ▷ courses (http://www.rzg.mpg.de/visualisation/scientificdata/presentations)
  - K. Reuter: *RZG-Services zur Visualisierung wissenschaftlicher Datensätze*, DV-Treffen der Max-Planck-Institute, Göttingen, Sep 15, 2010
  - K. Reuter: *Scientific Visualisation Services at RZG*, Seventh GOTiT High Level Course, Garching, Oct 19, 2010
  - M. Rampp: *Introduction to VisIt*, LRZ course on "Visualisation of Large Data Sets on Supercomputers", 2010 – 2011
  - overview talks at Max-Planck-Institutes: MPA, Garching (2009), FHI, Berlin (2011), . . .

Reference applications

General

see the gallery at the VisIt homepage: http://visit.llnl.gov/

RZG projects using VisIt (in close collab. with research groups)

▷ application domains:

- *Plasmaphysics*: MHD turbulence simulations for nuclear fusion research (IPP)
- *Stellar astrophysics*: Core-Collapse Supernova simulations (MPA)
- *Molecular dynamics*: Materials research for plasma-wall-interaction (IPP)
- *CFD*: DNS simulatons of turbulent flows (pipe, Taylor-Couette, ...)

▷ data structures/grids:

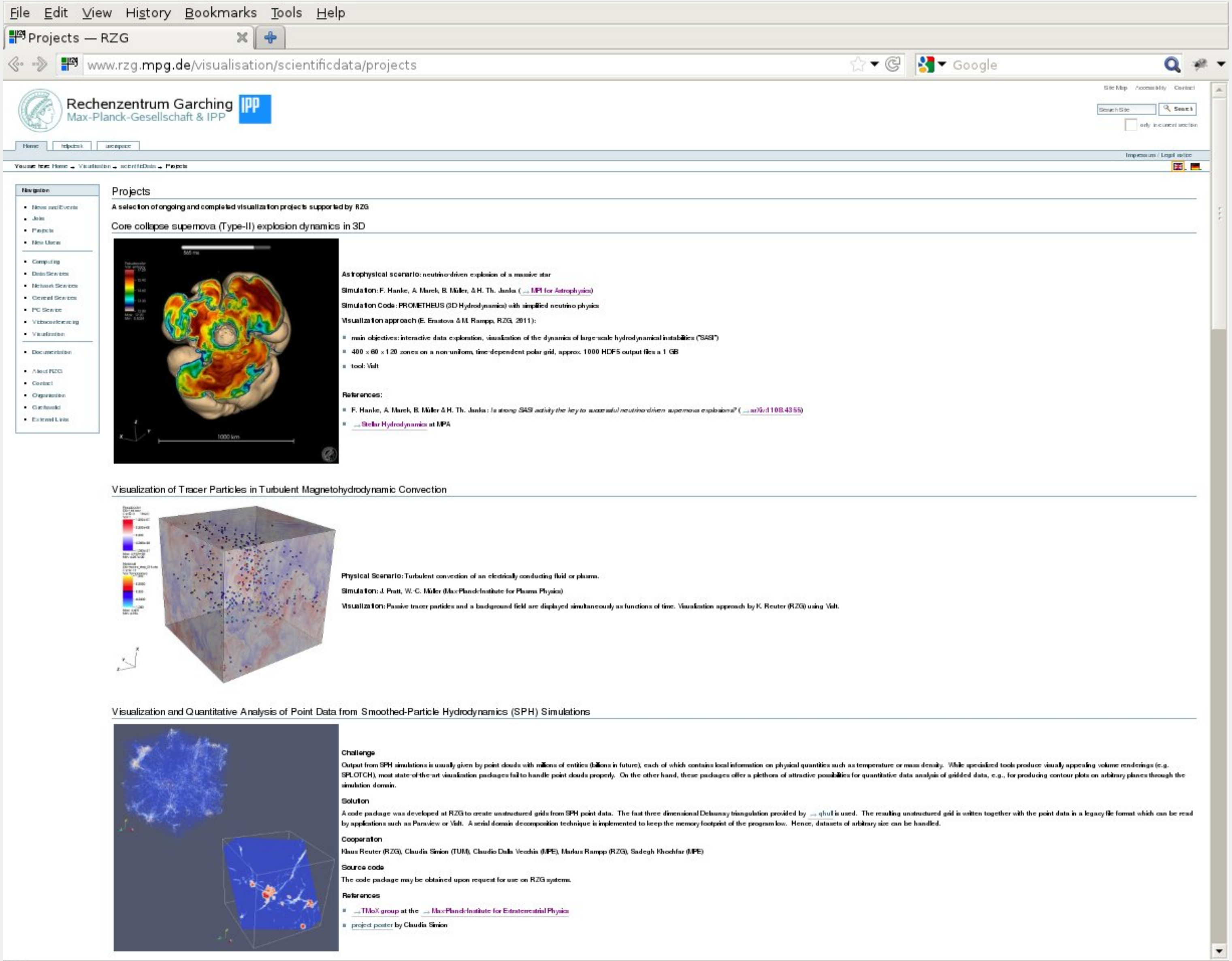
- regular: cartesian, polar (2D, 3D), block-structured ("Yin-Yan")
- irregular: (mapped) point clouds

▷ data sizes, dimensions:

- up to $\mathbf{2048^3}$ (cartesian), $\mathbf{1000 \times 180 \times 360}$ (polar)
- up to $\simeq \mathbf{10^6}$ (particles in 3D), $\simeq \mathbf{10^7}$ (nodes in 3D unstructured mesh)
- all: multi-variable (scalar, vector), time-dependent

see also: http://www.rzg.mpg.de/visualisation/scientificdata/projects

MPG/RZG reference applications



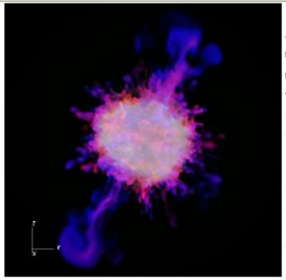
MPG/RZG reference applications

File Edit View History Bookmarks Tools Help

Projects — RZG

www.rzg.mpg.de/visualisation/scientificdata/projects

Google



Astrophysical scenario: 3D-Simulations of Mixing Instabilities in Type-II Supernova Explosions

Simulation: N. Hammer, H.-Th. Janka, E. Müller (→ MPI for Astrophysics)

Simulation Code: PROMETHEUS

Visualization approach (M. Rampp, 2009/2010):

- main objectives: exploration, quantitative analysis and visualization of the dynamics and morphology of the nuclear composition
- rectilinear (polar) grids with 500x180x360 zones per timestep
- tool: VisIt ("multi-channel" volume rendering, isosurfaces, 2D-slices)

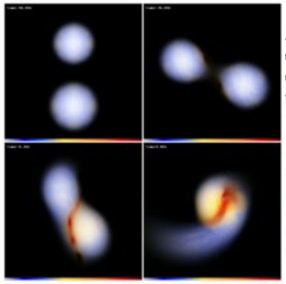
Results:

- download movie (8 MB mpeg4 avi)

References:

- N. Hammer et al., *Three-Dimensional Simulations of Mixing Instabilities in Supernova Explosions*, *Astrophysical Journal* 714 1371 (2010) doi: → 10.1088/0004-637X/714/2/1371 (→ arXiv:0910.5169)
- Press releases → *How a supernova obtains its shape* (MPA), → *Death of a star in three dimensions* (MPG), 05/2010

Merger of a white dwarf binary system



Astrophysical scenario: Merger of a white dwarf binary system as a Type Ia supernova progenitor

Simulation: R.Pakmor et al. (→ MPI for Astrophysics)

Simulation Code: GADGET

Visualization approach (E. Erastova, M. Rampp, 2009/2010):

- approx. 2 Million SPH particles per timestep
- tool: Splotch (ray-casting tailored to visualization of SPH data)

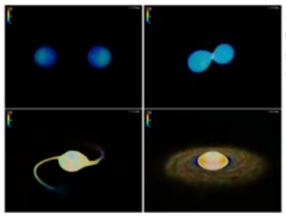
Results:

- download movie (1 MB mpeg4 avi)

References:

- R.Pakmor et al., → Sub-luminous type Ia supernovae from the mergers of equal-mass white dwarfs with M~0.9 M_solar, Nature, 7, 2010
- Press releases → "Violent explosions in space" (MPA), → "Tanz in die Katastrophe" (MPG)

Merging Neutron Stars



Astrophysical scenario: Dynamic merger phase of a binary neutron star system

Simulation: A.Bauswein & H.-Th.Janka (→ MPI for Astrophysics)

Simulation Code: relativistic smoothed particle hydrodynamics (SPH) code by → Oechslin et al.

Visualization approach (M. Rampp, 2009):

- approx. 500000 SPH particles per timestep
- tools: Splotch (ray-casting tailored to visualization of SPH data), VisIt

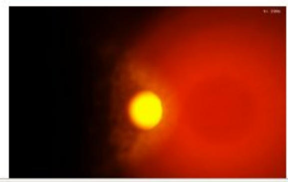
Results:

- for snapshots and movies see → MPA Research Highlight 11/2009

References:

- A.Bauswein & H.-Th.Janka : → Are Neutron Stars Strange?, MPA Research Highlight 11/2009
- A. Bauswein, R. Oechslin and H.-Th. Janka, *Discriminating Strange Star Mergers from Neutron Star Mergers by Gravitational-Wave Measurements*, Phys.Rev.D81 024012, 2010 → (arXiv:0910.5169)

Impact of Type Ia Supernovae on Binary Companions



Astrophysical scenario: Impact of type Ia supernova on its main sequence binary companion

Simulation: R.Pakmor (→ MPI for Astrophysics)

Simulation Code: GADGET

Visualization approach (M. Rampp, R.Bruckschen, RZG, R.Pakmor, 2009):

- approx. 9 Million SPH particles per timestep
- tool: Splotch (ray-casting tailored to visualization of SPH data)

Results:

- download movie (2.9 MB mpeg4 avi)

References:

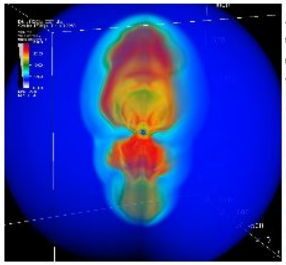
MPG/RZG reference applications

File Edit View History Bookmarks Tools Help

Projects — RZG

www.rzg.mpg.de/visualisation/scientificdata/projects

Core collapse supernova (Type-II) explosion



Astrophysical scenario: neutrino-driven explosion of a massive star

Simulation: A.Marek, H.Th.Janka (→ MPI for Astrophysics)

Simulation Code: VERTEX (Boltzmann neutrino radiation hydrodynamics in 2D)

Visualization approach (M. Rampp, RZG, 2008/2009):

- main objectives: interactive data exploration, visualization of the dynamics of large-scale hydrodynamical instabilities ("SASI")
- 500 x 128 zones on a non-uniform, time-dependent polar grid
- tool: VisIt (parallel ray-casting)

Results:

- download movies

References:

- A. Marek & H.Th. Janka : → Delayed neutrino-driven supernova explosions aided by the standing accretion-shock instability. Astrophysical Journal 694, 664 (2009).
- → Press Release Cluster of Excellence "Universe" (Feb 2, 2009)
- → Stellar Hydrodynamics at MPA

Turbulence in Fusion Plasmas

Physical scenario: fluctuations in a thermonuclear fusion plasma (tokamak geometry)

Simulation: M.Püschel, F.Jenko (→ IPP)

Simulation Code: GENE (gyrokinetic turbulence)

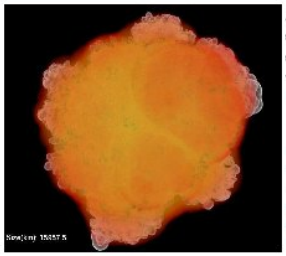
Visualization approach (R.Bruckschen & J.Mejia, RZG, 2007):

- main objectives: animation and interactive analysis of time-dependent data sets (3D impression, dynamics), development of a tailored visualization toolkit
- mapping of flux-tube coordinates to real space (torus) sampled by 1280 x 480 x 96 grid points
- out-of-core shader-based method
- point-based pseudo volume rendering
- Java application (JOGL API) for portability

References:

- Helmholtz-University Young Investigators Group → "Theory and ab-initio simulation of plasma turbulence"

Thermonuclear Supernova (Type-Ia) explosion



Astrophysical scenario: thermonuclear deflagration of a white dwarf star

Simulation: F.Röpke, W. Hillebrandt (→ MPI for Astrophysics)

Simulation Code: SUCCESs (finite volume hydrodynamics + level-set method for thermonuclear burning fronts)

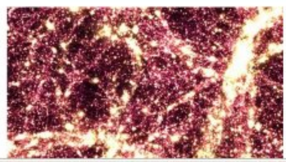
Visualization approach (R.Bruckschen, RZG, 2007):

- main objectives: combined volume rendering of the complete time-dependent dataset
  - stellar structure: scalar density field on the moving, rectilinear 1024^3 grid
  - deflagration front: level set renderend as a semitransparent cloud of points
- out-of-core rendering with hybrid display (point based/volume rendering)
- interactive preview, movie mode with high fidelity transparency

References:

- F K Röpke, R Bruckschen : → Thermonuclear supernovae: a multi-scale astrophysical problem challenging numerical simulations and visualization, New Journal of Physics 10, 2008
- Emmy Noether Junior Research Group → "Type Ia Supernova explosions" at MPA
- download movie (41 MB avi)

The → "Millenium simulation" (formation and evolution of structure in the universe)



Astrophysical scenario: formation and evolution of structure in the universe

Simulation: V.Springel, S.White (→ MPI for Astrophysics)

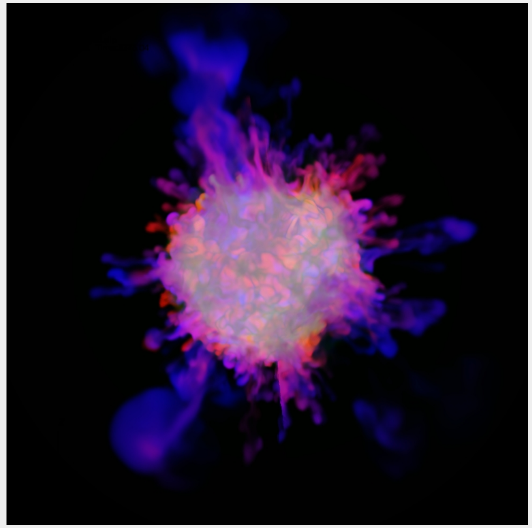
Simulation code: → GADGET-2 (cosmological N-body/SPH), ~ 10^{10} particles in 3D

Visualization approach (R.Bruckschen, RZG, 2006):

- main objectives: interactive exploration ("fly-through") of individual time-slices at full resolution
- visibility-based out-of-core method using a spatial subdivision scheme
- quantization based on local bounding boxes (low error)
- GPU shader based de-quantification and rendering

Application: Core-collapse Supernova

Simulations by N. Hammer, H.-Th. Janka & E. Müller (MPA)



- ▷ supernova explosion of $15\ M_\odot$ star
- ▷ instabilities & mixing of heavy elements
- ▷ first 3D simulations of long-term evolution

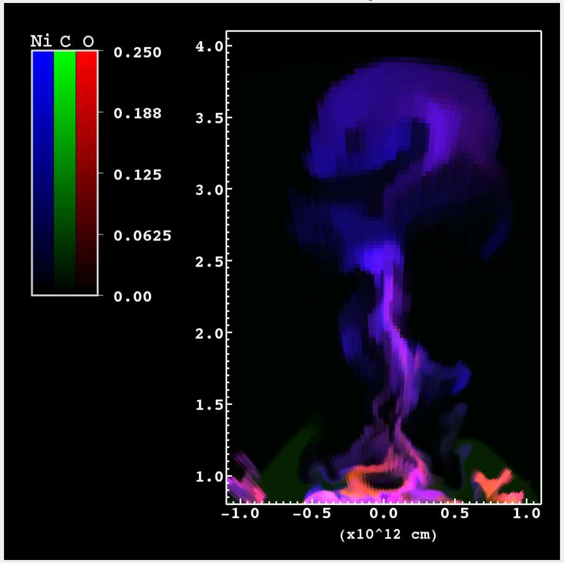
  Hammer et al., ApJ 714, 1371 (2010)
- ▷ simulation code: PROMETHEUS/HOTB (3D hydrodynamics, finite-volume, PPM)

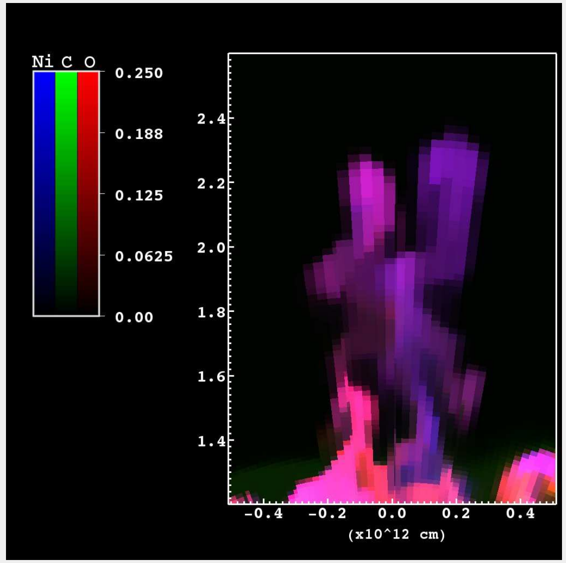
Visualisation approach (M. Rampp)

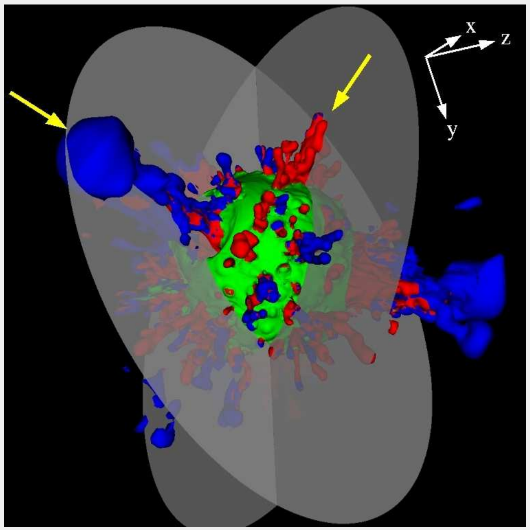
- ▷ data: $500 \dots 1000 \times 180 \times 360$ zones on non-uniform, polar grid
- ▷ ≈ 700 output files
- ▷ proprietary output format: converted to silo format (simple FORTRAN code)
- ▷ "multi-channel" volume-rendering: non-standard use-case for VisIt
  - elements Ni^{56}, O^{16}, C^{12} "shine" in blue, green, red
  - composite RGB image of individual volume renderings
- ▷ heavy use of VisIt's Python scripting interface
- ▷ nice results with stereo rendering

Application: Core-collapse Supernova

Quantitative analysis (plots taken from Hammer et al., ApJ 714, 1371, 2010)







- ▷ multiple isosurfaces (morphology)
- ▷ select slice planes
- ▷ analyze different scalar fields in selected slice planes (chem. composition mapped to RGB)

↷ stretches VisIt to its limits

Application: Core-collapse Supernova

Methods

Volume rendering

- ▷ operators: box, transform ("spherical to cartesian")
- ▷ plots: splatting (cheap, for quick exploration), "compositing" (HQ ray casting)
- ▷ individual image (file) for each of the 3 scalar variables
- ▷ RGB image composition (external: ImageMagick)

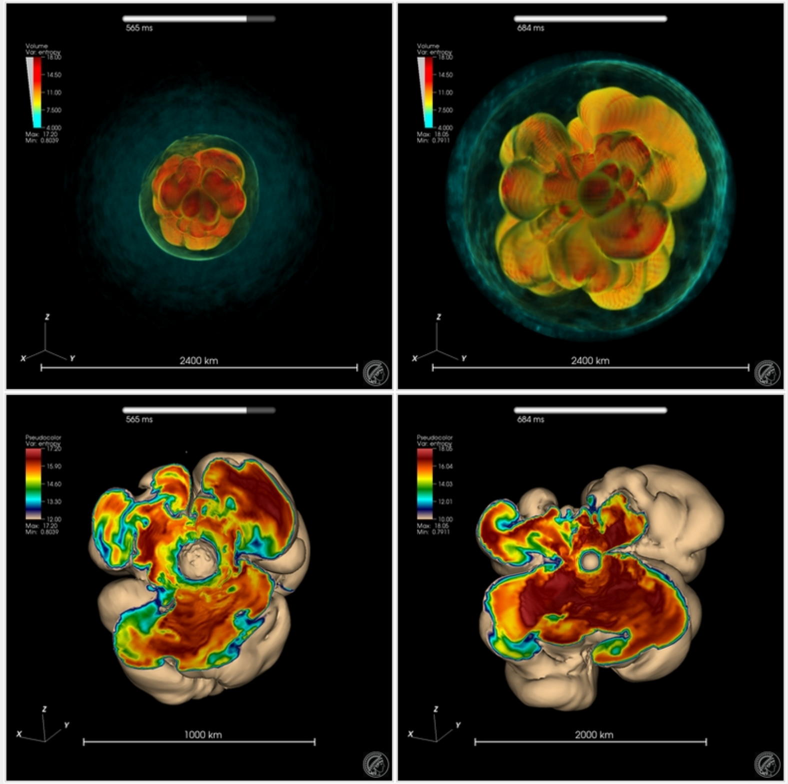
Quantitative analysis

- ▷ plots: pseudocolor
- ▷ operators: box, transform ("spherical to cartesian"), isosurface, slice (2D projection)

Application: Core-collapse Supernova

Simulations by F. Hanke & H.-Th. Janka (MPA)

- ▷ neutrino-driven explosions of massive stars
- ▷ simulation code: VERTEX (3D, time-dependent radiation hydrodynamics with detailed microphysics)
- ▷ towards first-principles modelling of core-collapse supernovae
- ▷ code writes HDF5 output
- ▷ XMDF metadata generated from HDF5 data (post-processing)

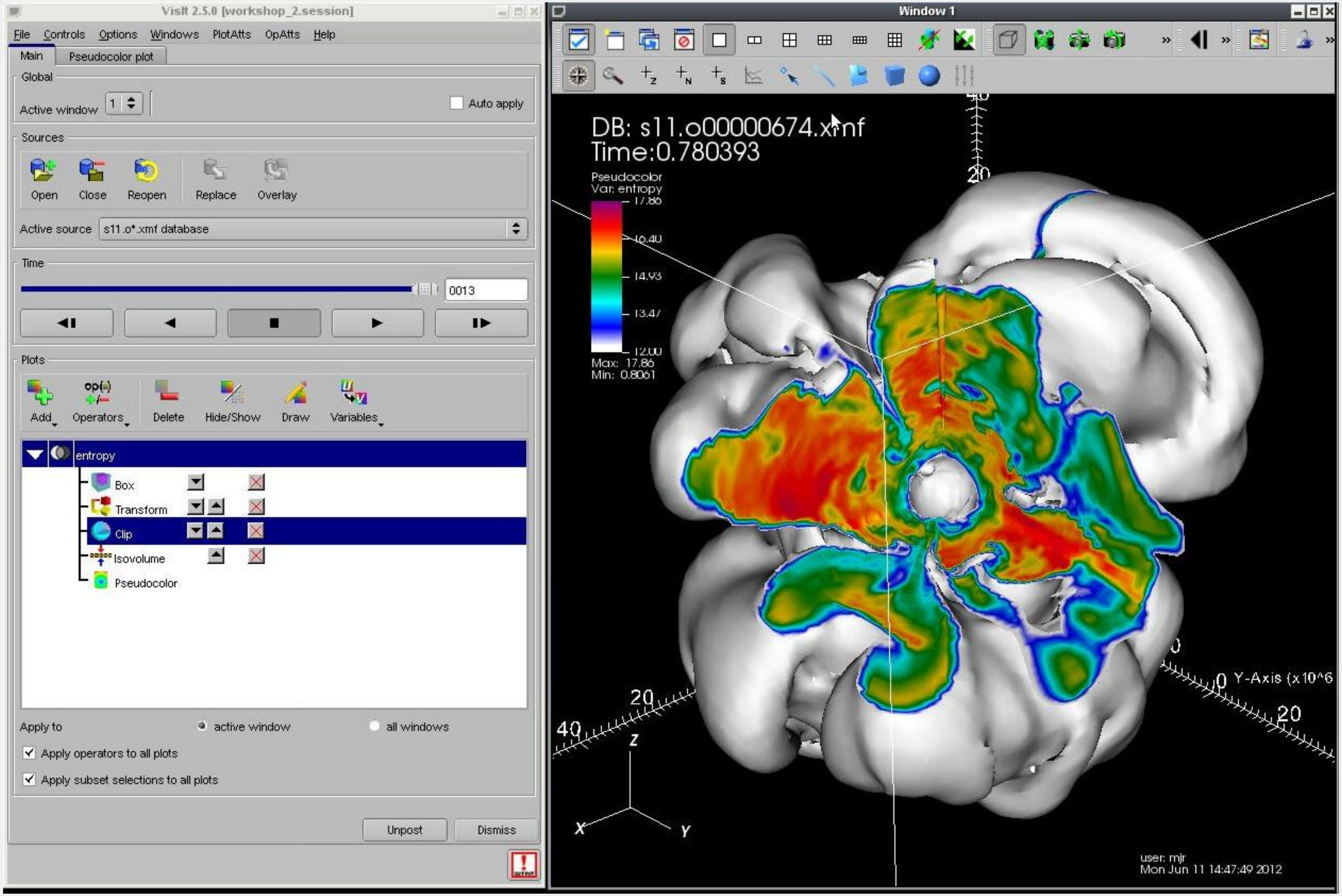


Visualisation approach (E. Erastova, M. Rampp)

- ▷ interactive data analysis: volume rendering, isovolume/pseudocolor plots, ...
- ▷ presentation: movie generation
- ▷ data: $400 \times 60 \times 120$ zones on non-uniform, polar grid, ≈ 1000 output files a 1 GB
- ▷ HDF5 output data: "decorated" with XDMF metadata description (simple perl script)

↷ no data conversion/duplication necessary

Application: Core-collapse Supernova



Application: Core-collapse Supernova

Methods

Pseudocolor plot: operators

1. Box: select radial subvolume $(\mathbf{0} \leq \boldsymbol{r} \leq \boldsymbol{r_0})$
2. Transform ("spherical to cartesian" coordinates)
3. Clip: normals $(\mathbf{1}, \mathbf{0}, \mathbf{0})$, $(\mathbf{0}, \mathbf{1}, \mathbf{0})$ $(\mathbf{0}, \mathbf{0}, \mathbf{1})$
4. Isovolume: $\boldsymbol{f_0} \leq \boldsymbol{f(x, y, z)} \leq \boldsymbol{f_{\mathrm{max}}}$

XDMF file example

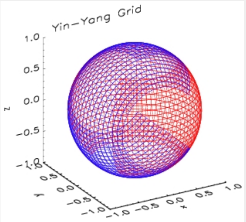
metadata for hdf5 binary file s11.o00000725:

```
<?xml version="1.0" ?>
<!DOCTYPE Xdmf SYSTEM "Xdmf.dtd" []>
<Xdmf Version="2.0">
 <Domain>
  <Grid Name="mesh" GridType="Uniform">

  <Topology TopologyType="3DRectMesh" Dimensions="120␣60␣400"/>

  <Geometry GeometryType="VXVYVZ">
    <DataItem Dimensions="400" Name="xzn" NumberType="Float" Precision="4" Format="HDF">
     s11.o00000725:/Step#000000079105/xzn
    </DataItem>
    <DataItem Dimensions="60" Name="yzn" NumberType="Float" Precision="4" Format="HDF">
     s11.o00000725:/Step#000000079105/yzn
    </DataItem>
    <DataItem Dimensions="120" Name="zzn" NumberType="Float" Precision="4" Format="HDF">
     s11.o00000725:/Step#000000079105/zzn
    </DataItem>
  </Geometry>

  <Time Value="0.830875" />

  <Attribute Name="density" AttributeType="Scalar" Center="Node">
   <DataItem Dimensions="120␣60␣400" NumberType="Float" Precision="4" Format="HDF">
    s11.o00000725:/Step#000000079105/den
   </DataItem>
  </Attribute>

  <Attribute Name="temperature" AttributeType="Scalar" Center="Node">
   <DataItem Dimensions="120␣60␣400" NumberType="Float" Precision="4" Format="HDF">
    s11.o00000725:/Step#000000079105/tem
   </DataItem>
  </Attribute>

  </Grid>
 </Domain>
</Xdmf>
```

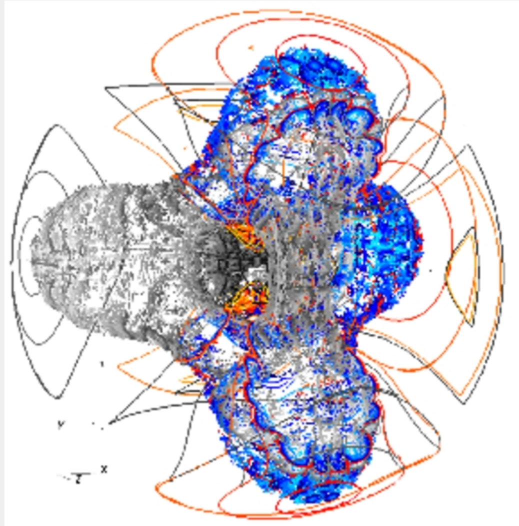
RECHENZENTRUM GARCHING rzg

Simulations by A. Wongwathanarat et al. (MPA)

- axis-free two-patch overset grid ("Yin-Yang") in spherical polar coordinates (for simulating 3D self-gravitating flows)



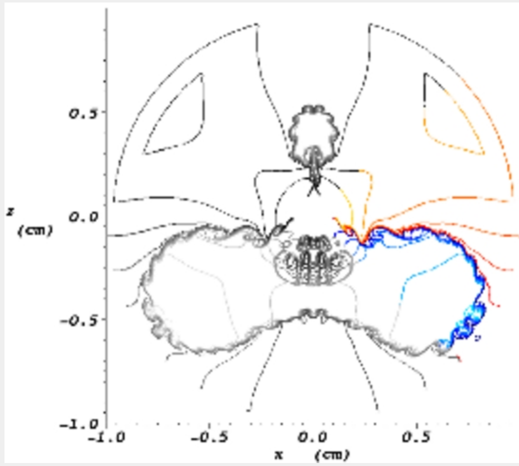
- data analysis during code development (debugging, symmetry constraints, . . .)



Visualisation approach (A. Wongwathanarat, MPA)

- dataset: $400 \times 292 \times 272 \times 2$ zones
- surfaces of constant density in 3D (top) and 2D (bottom; meridional cut) resulting from the simulation of the Rayleigh-Taylor instability.
  Yin grid: blue-yellow, *Yang grid:* white-black colors

  figures and text taken from: Wongwathanarat et al., A&A 514 (2010) A48



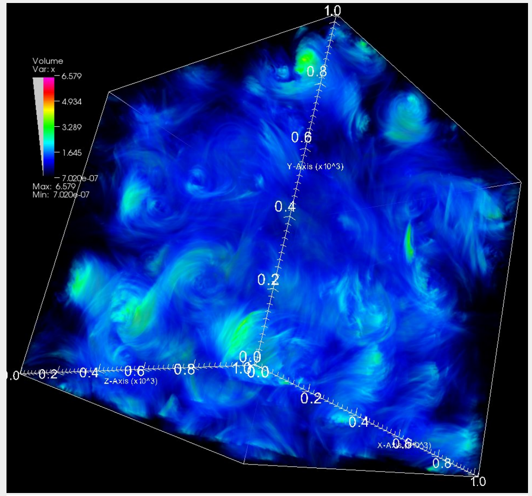
- only basic user support provided

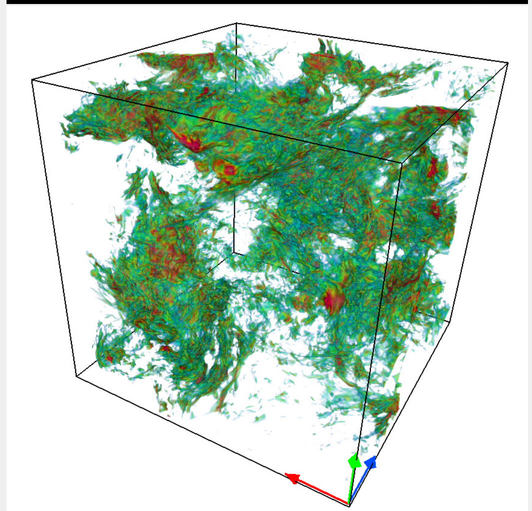
Simulations by W.C. Müller group (IPP)

- ▷ fundamental research in plasma physics for fusion devices (ASDEX UG, Wendelstein 7-X, ITER)
- ▷ grid-based MHD simulation code
- ▷ dataset: $\mathbf{1024^3} \ldots \mathbf{2048^3}$ rectilinear (cartesian) grid
- ▷ remote visualisation and interactive data exploration: volume rendering, isosurfaces, streamlines, ...

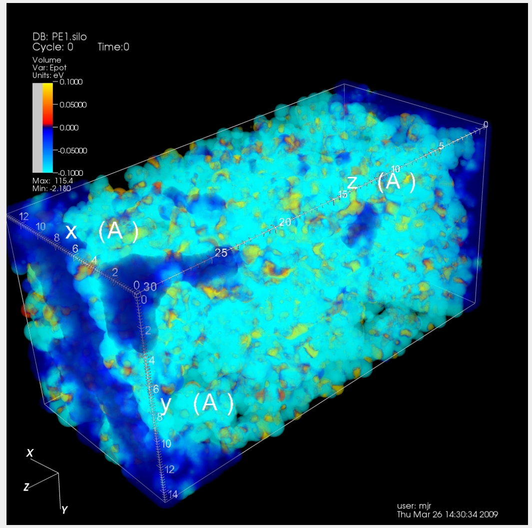
Visualisation approach (K. Reuter, RZG & J. Pratt, IPP)

- ▷ VisIt for rendering animations (offline)
- ▷ huge memory requirements
- ▷ VAPOR for interactive analysis
- ▷ rectangular grids $\curvearrowright$ VAPOR: multiresolution approach (wavelet representation: superiour interactivity)





Application: Molecular dynamics



Simulations by U. v.Toussaint (IPP)

- ▷ MD simulations of hydrogen diffusion in hydrocarbon layers
- ▷ materials research for fusion devices

Visualisation approach

- ▷ a small dataset: $28 \times 30 \times 62$ cartesian grid
- ▷ volume rendering, isosurfaces of H binding energy

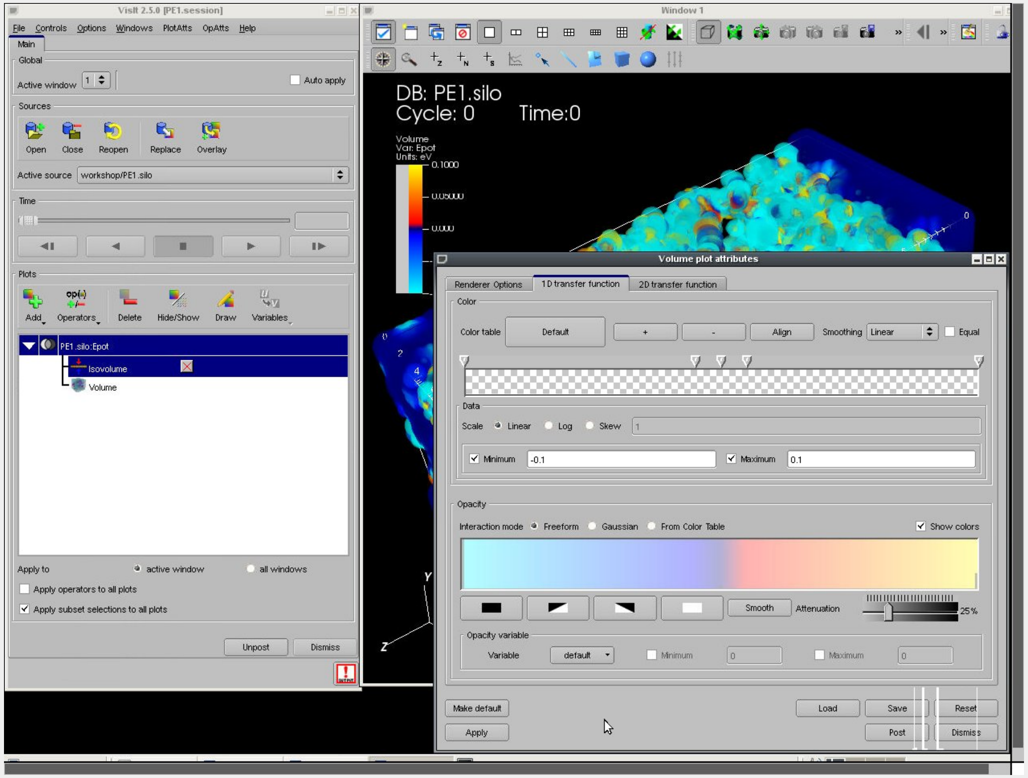
↷ a simple FORTRAN silo example (fragment) . . . remark: C-code less bloated

```
      include 'silo.inc'
      integer iopt,id,dims(3)
      data dims /28+1,30+1,62+1/

c- create an options list (returns identifier iopt)
      dbmkoptlist(5,iopt)
      dbaddiopt(iopt,DBOPT_COORDSYS,DB_CARTESIAN)
      dbaddiopt(iopt,DBOPT_NSPACE,size(dims))
      dbaddropt(iopt,DBOPT_TIME,'10.5')
      dbaddiopt(iopt,DBOPT_CYCLE,123)

c- create a silo database (returns identifier id)
      dbcreate('PE1.silo',8,DB_CLOBBER,DB_LOCAL,DB_F77NULL,0,DB_PDB,id)

c- write mesh (3 dimensional) to silo database identified by id
      dbputqm(id,'mesh',4,'x',1,'y',1,'z',1,x,y,z,dims,size(dims),DB_FLOAT,DB_COLLINEAR,iopt,RSV)

c- write scalar variable named "epot" and options to silo database identified by id resp. iopt
      dbaddcopt(iopt,DBOPT_UNITS,'eV',2)
      dbputqv1(id,'epot',4,'mesh',4,epot,dims,size(dims),DB_F77NULL,0,DB_FLOAT,DB_ZONECENT,iopt,stat)

c- close file
      ierr=DBClose(id)
```

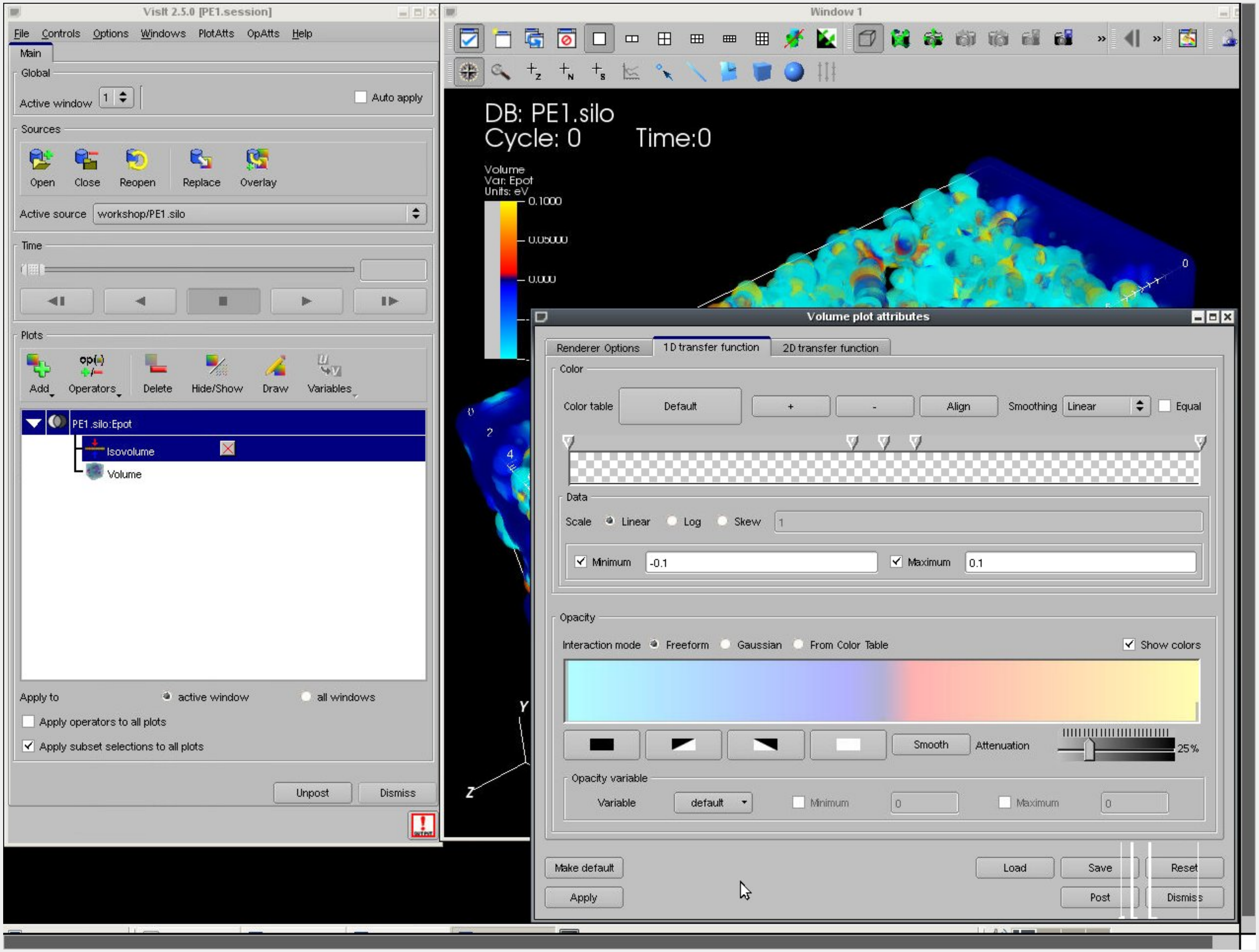
Application: Molecular dynamics

Methods

- ▷ isovolume operator selects $-\boldsymbol{E_0} \leq \boldsymbol{E(x, y, z)} \leq +\boldsymbol{E_0}$
- ▷ volume plot:
  - "hot and cold" colortable
  - explicit restriction of datarange $[-\boldsymbol{E_0}, +\boldsymbol{E_0}]$

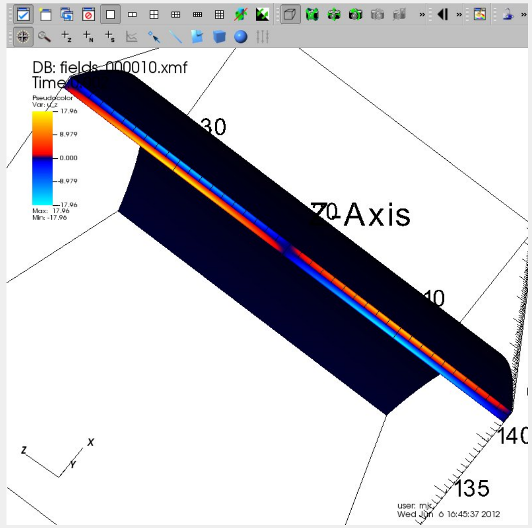


Application: Molecular dynamics



Simulations by L. Shi, M. Avila, B. Hof (MPI f. Dynamics and Self Organization)

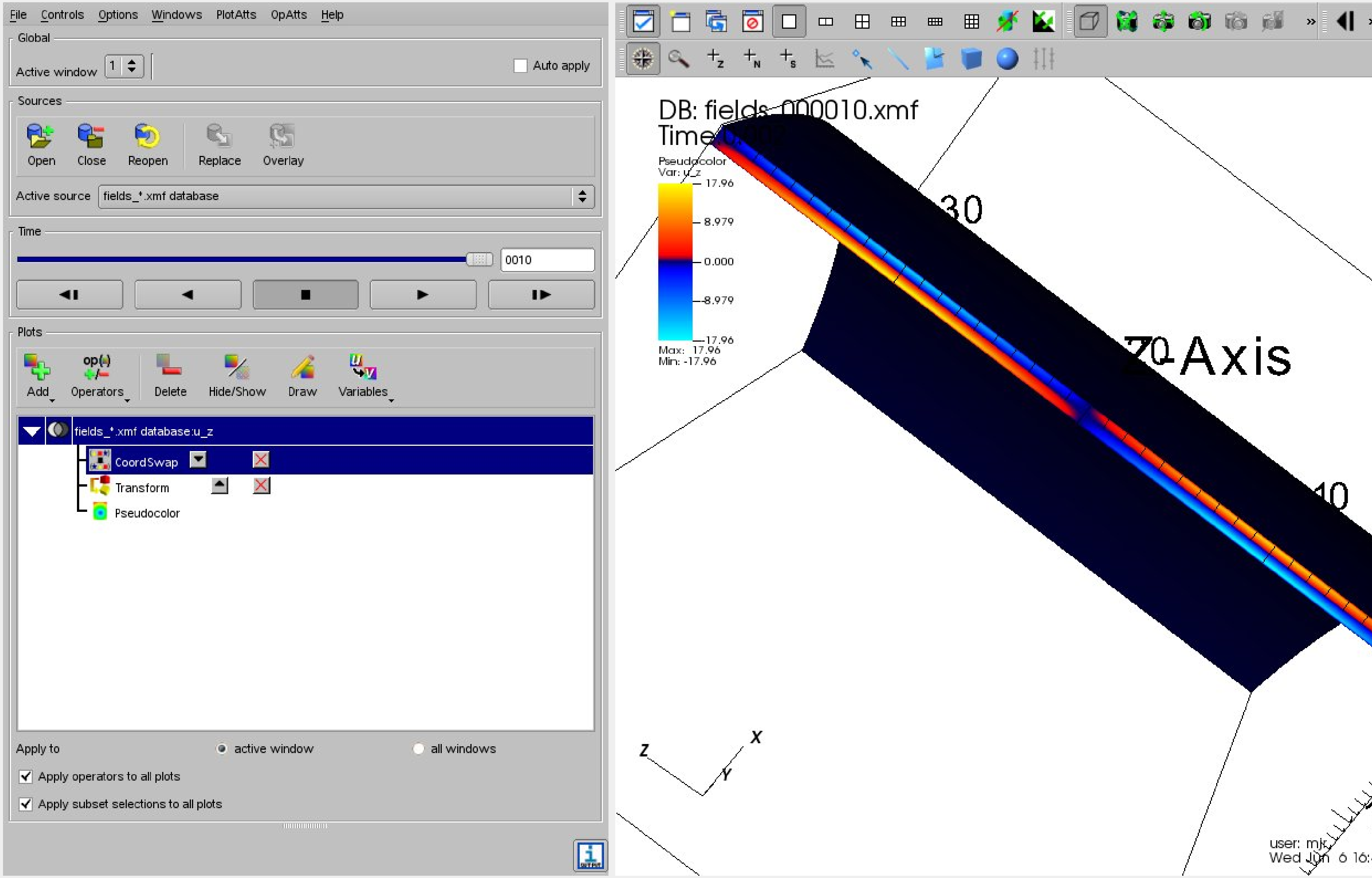
- ▷ DNS of fluids (pipe flows, Taylor-Couette flows)
- ▷ code: incompressible Navier Stokes equations, pseudospectral method
- ▷ basic research in turbulence: lab experiments, numerical simulations
  (Hof et al., Science (2010), Avila et al., Science (2011))



Visualisation approach (work in progress: M. Rampp, L. Shi)

- ▷ $100\ldots200 \times 380 \times 760$ zones on non-uniform cylindrical grid
- ▷ $\approx 100\ldots1000$ output files (timesteps)
- ▷ developed an I/O and visualisation strategy "from scratch":
  - simulation code:
    - I/O is time-critical and needs to be highly optimized
    - parallel HDF5 output of physical variables, $p(\theta, z, r)$, (u_θ, u_z, u_r)
    - generation and output of XDMF metadata in separate XML files
  - visualisation with VisIt:
    - "swap coordinates" operator handles transposed coordinates: $(\theta, z, r) \rightarrow (r, \theta, z)$
    - how to do this (straightforwardly) in ParaView?

Application: DNS hydro simulations



RECHENZENTRUM GARCHING

Methods

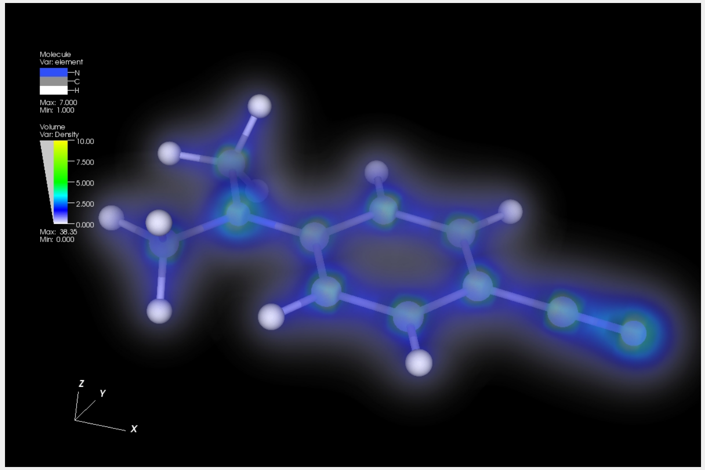
- ▷ "swap coordinates" operator: $(\theta, z, r) \rightarrow (r, \theta, z)$
- ▷ "transform coordinates" operator: $(r, \theta, z) \rightarrow (x, y, z)$
- ▷ pseudocolor, volume, vector, ... plot, expressions: $\mathbf{vorticity}(u_r, u_\theta, u_z)$, ...

```
OpenDatabase("localhost:/ptmp/mjr/nsCouette/fields_*.xmf␣database")
SetPipelineCachingMode(0)

AddPlot("Pseudocolor", "pressure", 1, 1)
AddOperator("CoordSwap", 1)
AddOperator("Transform", 1)
SetActivePlots(0)

CoordSwapAtts = CoordSwapAttributes()
CoordSwapAtts.newCoord1 = CoordSwapAtts.Coord3
CoordSwapAtts.newCoord2 = CoordSwapAtts.Coord1
CoordSwapAtts.newCoord3 = CoordSwapAtts.Coord2
SetOperatorOptions(CoordSwapAtts, 1)

TransformAtts = TransformAttributes()
TransformAtts.transformType = TransformAtts.Coordinate
TransformAtts.inputCoordSys = TransformAtts.Cylindrical
TransformAtts.outputCoordSys = TransformAtts.Cartesian
TransformAtts.vectorTransformMethod = TransformAtts.AsDirection
TransformAtts.transformVectors = 1
SetOperatorOptions(TransformAtts, 1)

DrawPlots()
for n in range(0,GetDatabaseNStates()-1):
    SetTimeSliderState(n)
    s = SaveWindowAttributes()
    s.format = s.JPEG
    s.width = 800
    s.height = 800
    s.fileName = "movie_%04d" % n
    SetSaveWindowAttributes(s)
    SaveWindow()
```

Application: "Ab initio" molecular simulations

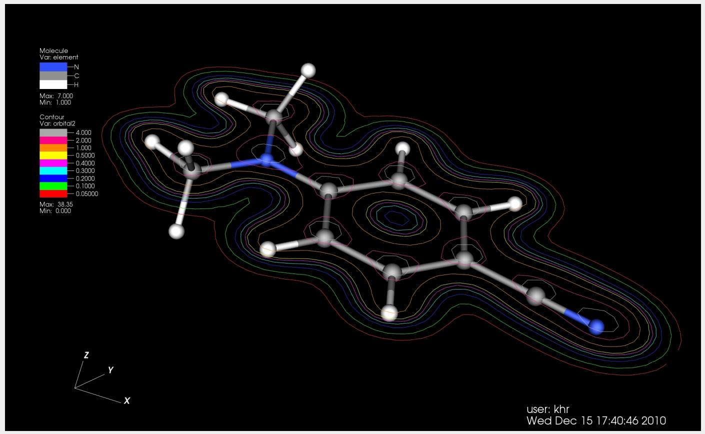
Collaboration with V. Blum, H. Appel (FHI)

- ▷ visualisation of molecular simulations
- ▷ DFT simulations with FHI-aims
- ▷ datasets: atom coordinates, Hartree potential, . . .
  (small) testcases for establishing methodology, e.g. *4,4'-dimethylaminobenzonitrile (DMABN)*
- ▷ remote visualisation and interactive data exploration: volume rendering, isocontours, . . .



Visualisation approach (A. Marek, K. Reuter, RZG)

- ▷ developed an I/O and visualisation strategy:
- ▷ postprocessing of FHI-aims output
  ↷ *cube* format for scalar fields
- ▷ VisIt for interactive analysis and rendering
  - "balls and sticks" for nuclei
  - volume rendering for scalar DFT fields (electron density, Hartree potential, . . .)



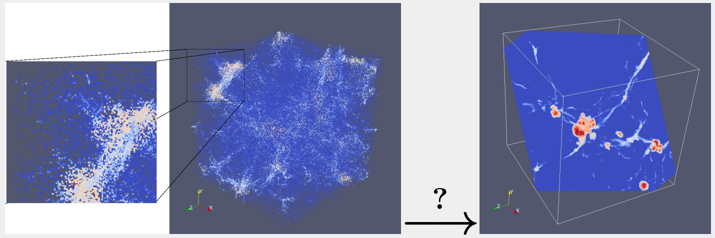
Simulations by S. Kochfahr et al. (MPE)

▷ SPH simulations produce "point clouds" with (strongly) varying particle density

▷ background: SPH "particles" sample scalar fields, particles carry size information (smoothing kernel)

▷ very limited support by standard software, special-purpose software (Splotch, Splash) does not cover full spectrum of features

▷ tailored approaches could not be employed for interactive, quantitative analysis



Visualisation approach (C. Simion, MPE & K. Reuter, RZG)

▷ mapping to unstructured(!) grids which can be handled by VisIt, Paraview

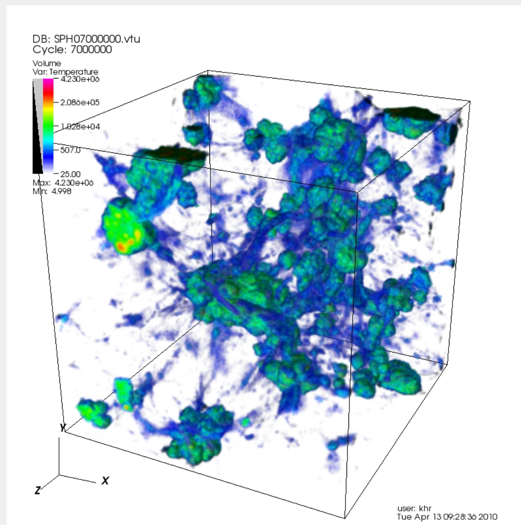
▷ approach: Delaunay triangulation

- preserves resolution, avoids interpolation

but: VTK library implementation (employed by VisIt 2.0.0 ?):

- CPU time scales as N^2, huge memory requirements

↷ Qhull, GPU implementation, . . . ?



Part 4: Conclusions

Experiences and some lessons learned

General remarks

- ▷ VisIt is a tool definitely worth considering for multi-D data visualisation & analysis
- ▷ highly competitive with ParaView, VAPOR, and also commercial tools
- ▷ VisIt has its roots clearly in massive (astrophysics) data visualisation ...
- ▷ ... but has matured to a comprehensive, general-purpose visualisation and analysis tool
- ▷ VisIT employs intuitive concepts and controls
- ▷ personal experience: quite often, the right tool (operator, plot, ...) is *just there*
- ▷ observation: increasing number of VisIt plots in scientific publications (our perception bias ?)
- ▷ parallel capabilities allow to utilize CPU, RAM of a *cluster* (multiple nodes) for a single visualisation session
- ▷ parallel version not trivial to install
- ▷ parallel efficiency: not trivial to achieve; very good support by H. Childs (UCD)
- ▷ movies: Python scripting superior (in flexibility and work effciency) to keyframing

RECHENZENTRUM GARCHING

Observed deficiencies (partly VisIt 1.x)

- a "stop rendering" button is missing
- long-running renderings: more control required (start/stop, pause)
- cleanup of remote rendering processes not reliable in case of crashes
- tedious "select $\longrightarrow$ open" dialogs (improved in VisIt 2)
- progress bar mostly useless (shows percentage of individual rendering phases)
- changed spherical coordinate system (VisIt 1.x: $\boldsymbol{r, \phi, \theta}$, Visit 2.0: $\boldsymbol{r, \theta, \phi}$)
- coordinate transform (spherical $\longrightarrow$ cartesian) for vectors broken (VisIt 1.x)
- integration of earth-mapped data ? ($\curvearrowright$ good results with ParaView in a recent project)
- quantitative analysis of massive SPH data ?

Conclusion

How to get (really) started yourself

1. read the (few pages) *VisIt Getting Started Manual*
   https://wci.llnl.gov/codes/visit/manuals.html
2. possibly consult the tutorial at
   http://www.visitusers.org/index.php?title=Short_Tutorial
3. ask your assistant (or favourite computing centre) for help on converting data
4. start rendering

Recommendation for next steps

- ▷ read "*Getting Data Into VisIt*": strategies for data handling
  https://wci.llnl.gov/codes/visit/manuals.html
- ▷ consult *VisIt User's Manual* and *VisIt Python Interface Manual*
  https://wci.llnl.gov/codes/visit/manuals.html
- ▷ consider Python scripting as soon as possible (for making your work more efficient and more easily reproducible)