CSCS

Swiss National Supercomputing Centre

Particle Data Management from disk archives to 3D graphics

1st SPHERIC Workshop

Rome, 10-12 May 2006

J. M. Favre, J. Biddiscombe, T. Foggia

Outline

- **Introduction**
- Particle Data Archiving
  - Ensight format
  - HDF5 to H5Part
- 3D graphics
  - Towards more realistic rendering
  - Mesh-less data to grid data
- sparticles
  - A point sprite rendering application
  - Examples of visualization
- Conclusion

CSCS *is ...*

CSCS is the **Swiss National Supercomputing Centre**, providing, developing and promoting technical and scientific services for the Swiss research community on the fields of high-performance and high-throughput computing. It is a centre of competence that pioneers new information technologies; collaborates with domestic and foreign researchers, and carries out its own research and development in computational sciences and scientific computing.

www.cscs.ch

Outline

- Introduction
- **Particle Data Archiving**
  - Ensight format
  - HDF5 to H5Part
- 3D graphics
  - Towards more realistic rendering
  - Mesh-less data to grid data
- sparticles
  - A point sprite rendering application
  - Examples of visualization
- Conclusion

Data Archiving

Needs

- Random access, seek and retrieve
- Time dependent storage
- Data exchange
- Access to data is a bottleneck, using optimal storage is crucial when working with big datasets – especially when fetching multiple timesteps is required

Many options for archiving formats

- CEI's **EnSight** Binary Gold http://www.ensight.com/
- NCSA's **HDF5** http://hdf.ncsa.uiuc.edu/HDF5/
- **H5Part** http://vis.lbl.gov/Research/AcceleratorSAPP/

Example of data readers

- EnSight
- **VTK/ParaView** (http://www.vtk.org, http://www.paraview.org)

EnSight file format

General features

- Geometry and variables in separate files
- An ASCII case file encodes the time-dependent file naming conventions
- There is a way to store only particles, without indices (Gold format)
- # files = Geometry + Time_case + N_timesteps*N_variables

⇒Many many files

Data can be written without any external libraries

Can be read by EnSight and by VTK/ParaView

HDF5 encoding

General features

- HDF5 is a general purpose library and file format for storing scientific data
- Efficient storage and I/O, allows parallel access
- Large and varied user community
- Can be read by VTK/ParaView, sparticles

H5Part, a particle data storage convention based on HDF5

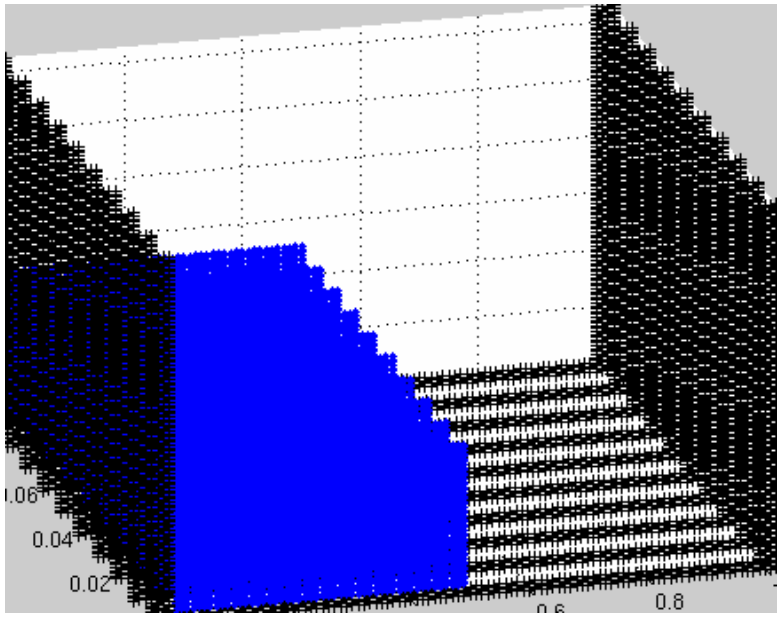
- H5Part is an API layer on top of HDF5 which simplifies reading and writing arrays of particles and their scalar/vector attributes from the file

Example of an HDF5 dataset

Look at the data structure with **hdfviewer**

Read HDF5 format with Matlab

```
particles=hdf5info(particles.x44.h5);
for i=1:size(particles.GroupHierarchy.Groups,2)
   coords=hdf5read(particles.GroupHierarchy.Filename,...
            particles.GroupHierarchy.Groups(i).Datasets.Name);
   end;
```

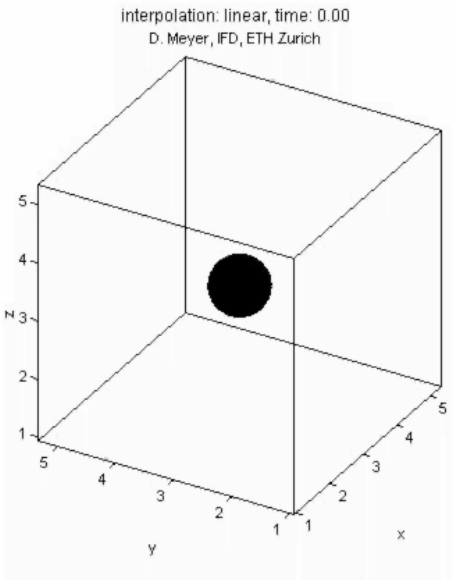


Data courtesy D.Violeau, EDF

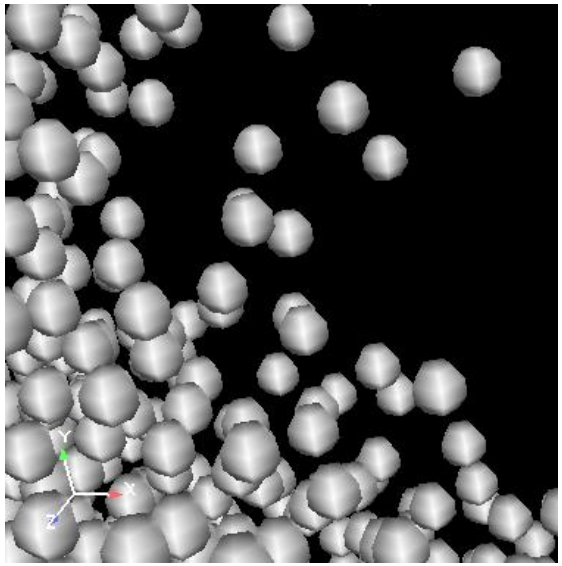
Outline

- Introduction
- Particle Data Archiving
  - Ensight format
  - HDF5 to H5Part
- **3D graphics**
  - **Towards more realistic rendering**
  - Mesh-less data to grid data
- sparticles
  - A point sprite rendering application
  - Examples of visualization
- Conclusion

Towards more realistic rendering



Basic pixel rendering

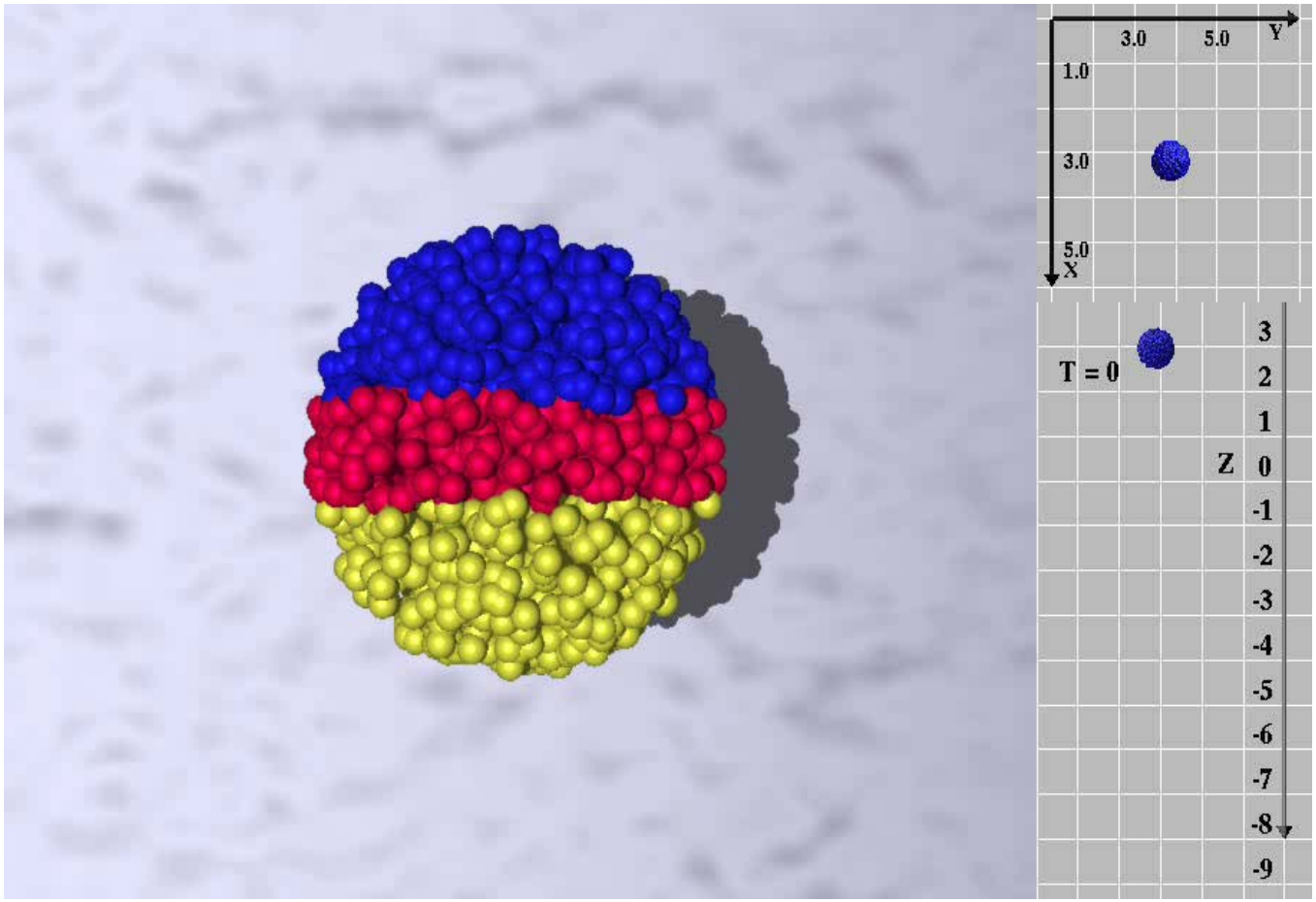


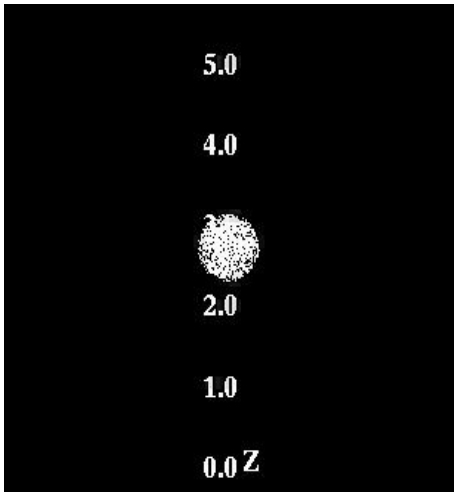
Basic sphere rendering

Simple effects will change our perception of the data

- Internal particle motion is highlighted by colored clusters
- Adding shadows help appreciate the volume occupied by the particles
- The non uniform wallpaper highlights the camera zooming in/out

Towards more realistic rendering





Video Award of the 2004 American Physics Society

Outline

- Introduction
- Particle Data Archiving
  - Ensight format
  - HDF5 to H5Part
- **3D graphics**
  - Towards more realistic rendering
  - **Mesh-less data to grid data**
- sparticles
  - A point sprite rendering application
  - Examples of visualization
- Conclusion

From mesh-less data to grid data

Goals

- Use all well-known volumetric viz tools: contouring, slicing, probing
- Free surface reconstruction

One solution: 3D volume reconstruction

- Shepard Method (uniform re-sampling with inverse weighted function)
- Delaunay 3D with non-zero alpha shapes
- Volume rendering

Structured grid volume rendering

Gaussian splatting reconstructs a structured mesh sampling the particles density at the center of the hexahedra.

Volume rendering of the density field can be real-time thanks to 2D or 3D texture mapping.

The volume is transparent everywhere except in high-gradient regions.

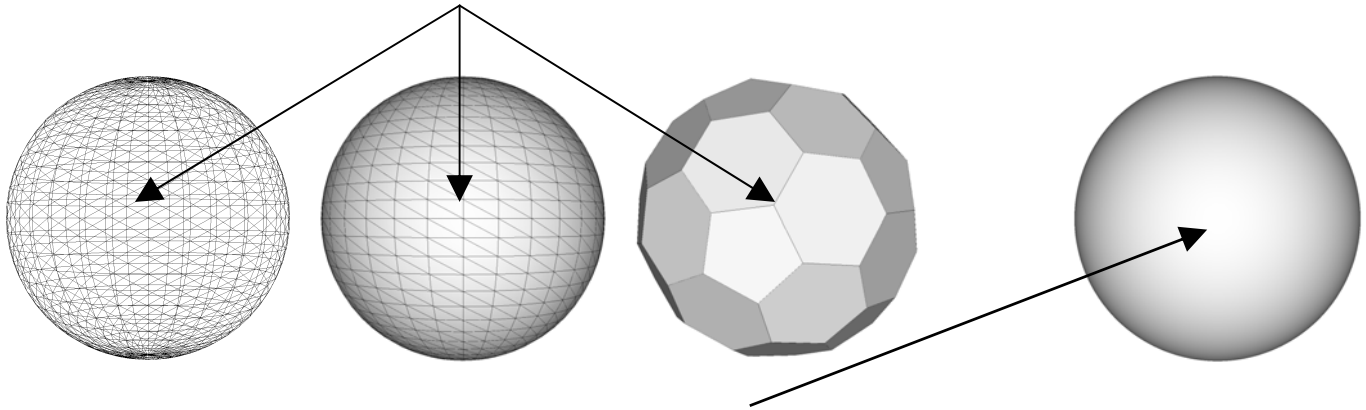
Outline

- Introduction
- Particle Data Archiving
  - Ensight format
  - HDF5 to H5Part
- 3D graphics
  - Towards more realistic rendering
  - Mesh-less data to grid data
- **sparticles**
  - **A point sprite rendering application**
  - Examples of visualization
- Conclusion

sparticles, a particle rendering application

sparticles renders using a sprite instead of polygons

- SPH type simulations are generating huge datasets
- Using Geometry based polygonal renderings is too slow



- sparticles draws a sphere “sprite” or “imposter” image at every point.
- Only 1 vertex per particle is transferred to the GPU
- Transformations of vertices from model to view space are massively reduced
- Rendering speed increases by an order of magnitude

However

- Speed can be limited by data transfer from CPU to GPU (lots of particles)
- Speed can be limited by pixel “fill rate” (complexity of GPU program)

vtkPointSpriteMapper

sparticles uses a customized vtk mapper

Renders using Vertex and Fragment Programs

Vertex program

Computes vertex transformation using standard projection

Computes particle screen size from supplied point radius

- Point radius may be independent for every point

Fragment Program

Replaces each screen pixel with evaluated function

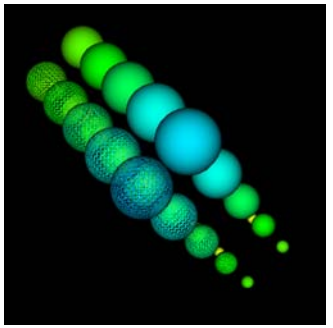
- Sphere equation implemented in current version
- Any Parametric function possible in principle

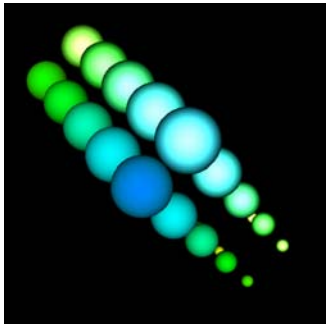
Custom fragment programs can be used for special effects

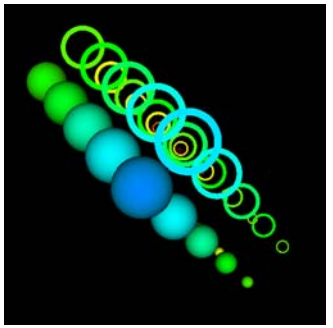
- Image 1 shows polygon spheres beside particle ones
- Image 2 shows extra lighting on the particles
- Image 3 shows particles as disks (to improve visibility whilst maintaining particle size)

The more complex the program, the slower the rendering

Particles and Geometry can be mixed freely (e.g. image1)







Key features and points

Time dependent data reading (snapshots, animations)

Variable radius/colour: **see demo**

Caching of Data

sparticles internally uses a cache to store data on a local disk to improve animation speed when viewing a previously seen dataset

Rendering speed/quality depends on graphics hardware

OpenGL shading language used to render particles if graphics hardware supports it. Complexity of Fragment program dictates speed

Point Sprite and glPoint modes supported for older computers (e.g. laptops)

30 Million particles/second

Sustained render speed on a GeForce 7800 graphics card

(1 million particles at 30fps)

Data formats supported

sparticles loads data in the following formats

- **Ensight 6** (binary/ascii)
- **Ensight Gold** (binary/ascii)
- **H5Part**

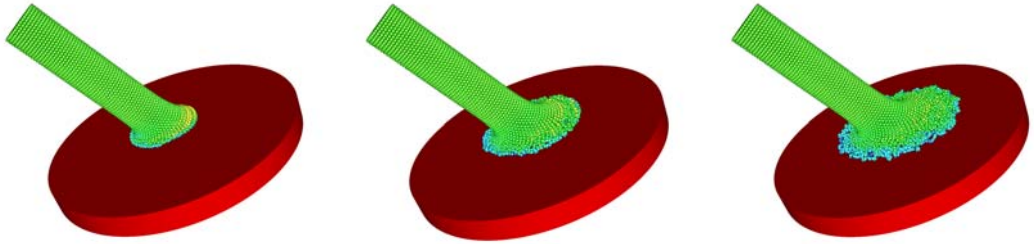
  CSCS is in the process of adapting/modifying H5Part to get the best performance possible with SPH datasets.

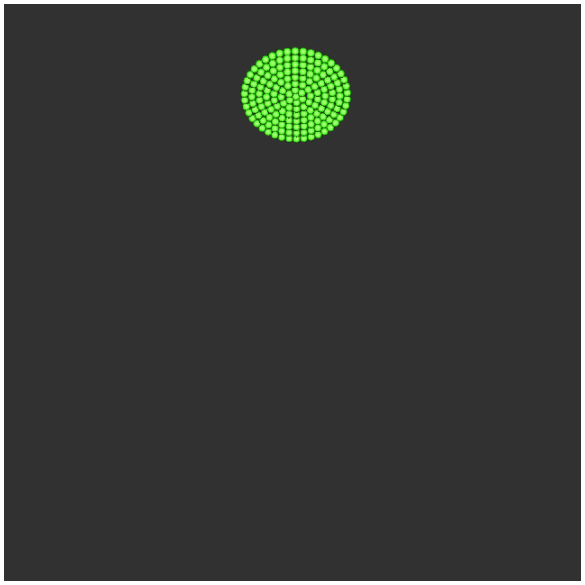
Outline

- Introduction
- Particle Data Archiving
  - Ensight format
  - HDF5 to H5Part
- 3D graphics
  - Towards more realistic rendering
  - Mesh-less data to grid data
- **sparticles**
  - A point sprite rendering application
  - **Examples of visualization**
- Conclusion

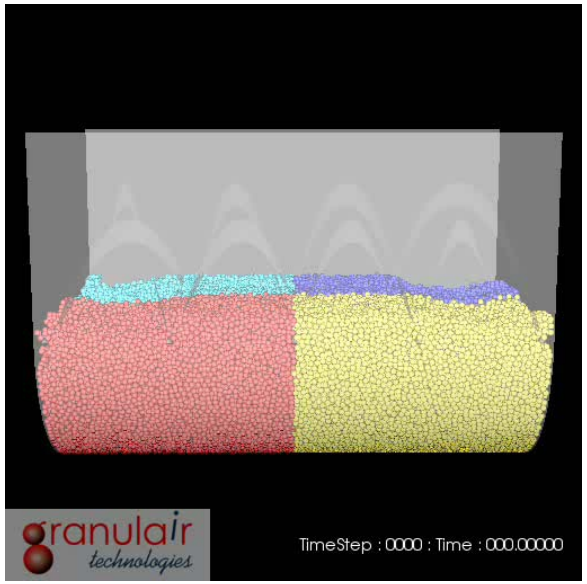
Examples of sparticles in use

Displaying SPH results

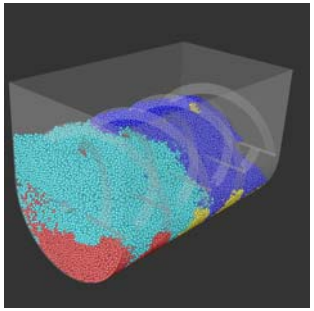


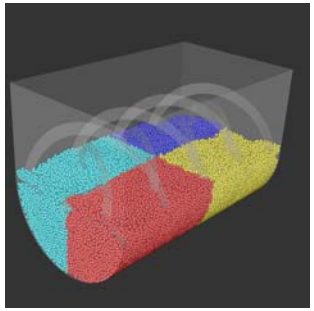


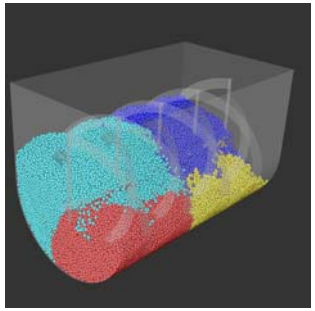
Data courtesy J. C. Marongiu, Ecole Centrale de Lyon, France; Etienne Parkinson, VA TECH HYDRO



Displaying DEM results





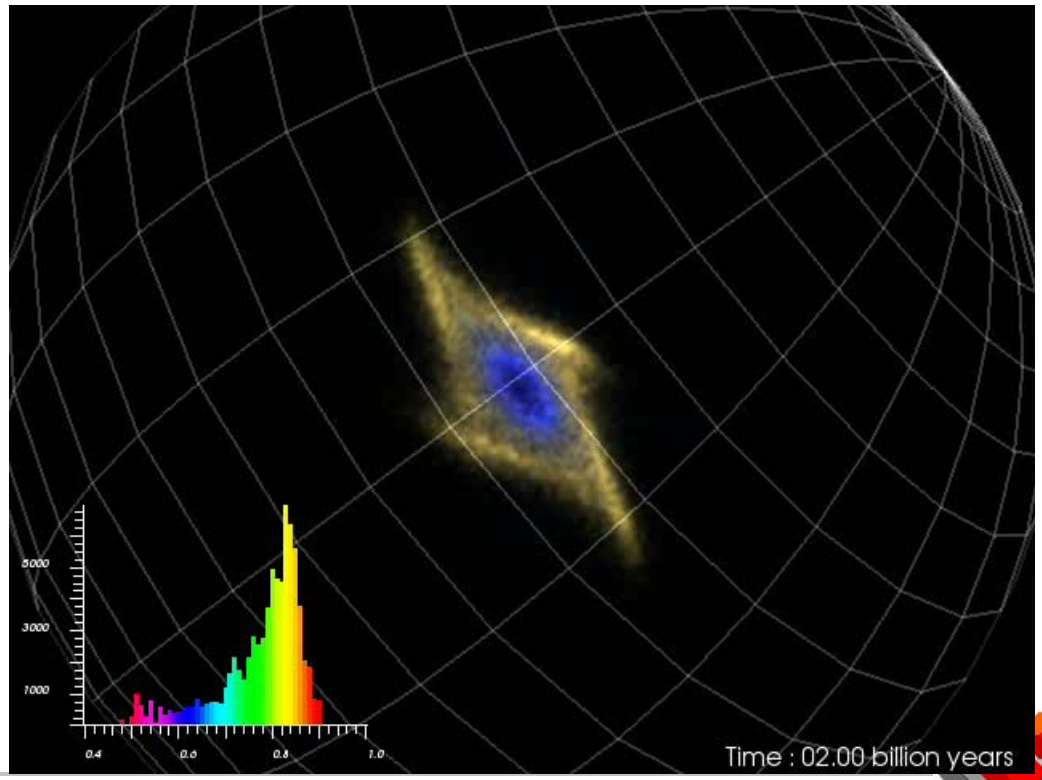


Data courtesy Mark Sawley, Granulair Technologies

Examples of usage

Displaying Astronomy data

- NB. this animation was created using a customized version of sparticles.
- By blending particles additively, the galaxy centre becomes brighter



Future work on sparticles

sparticles is being actively developed at CSCS

HDF5 - H5Part file format

- Support for geometry in H5Part file
- Faster access to coordinates and less array copying

Parallel rendering for enormous datasets (~ 50million per PC)

- Sparticles can render on multiple machines and composite results onto a single desktop

Embedding SPH and visualization into the same application

- Parallel SPH code with Parallel rendering will enable real time manipulation of simulations *(= Steering)*

Fragment programs

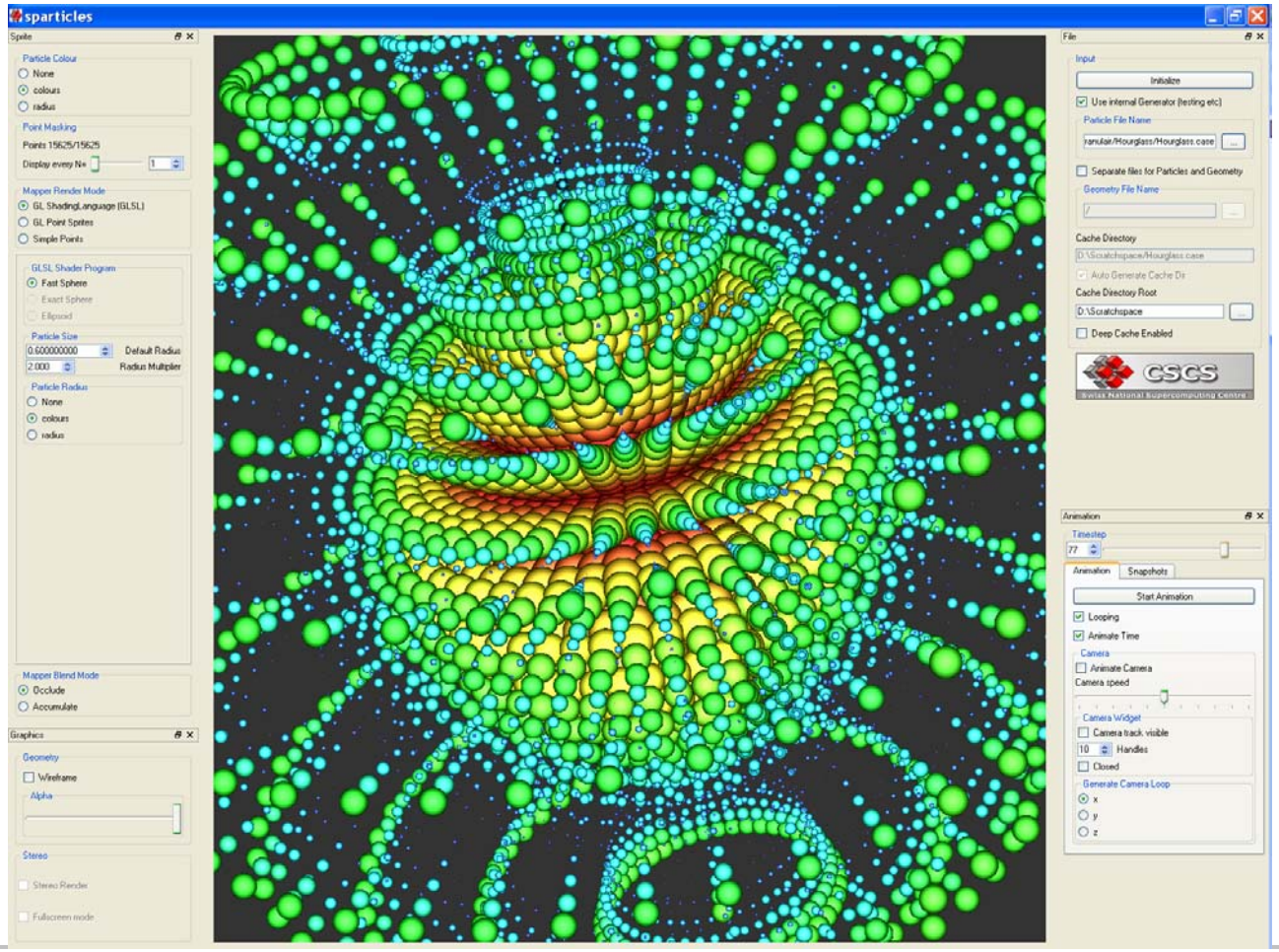
- Sparticles renders using a fast sphere approximation but some simulations require overlapping spheres and other effects such as transparency.
- Ellipsoids required for vector display

User interface and features – Lots to do

Download sparticles (beta version)

On the CSCS web site:

http://www.cscs.ch/a-display.php?id=170



Outline

- Introduction
- Particle Data Archiving
  - Ensight format
  - HDF5 to H5Part
- 3D graphics
  - Towards more realistic rendering
  - Mesh-less data to grid data
- sparticles
  - A point sprite rendering application
  - Examples of visualization
- **Conclusion**

Conclusion

The data archiving structure is a relevant issue with the development of very large particle simulations (millions of particles, thousands of time steps).

Advanced rendering techniques allow better perception of the data and significantly increase the speed of particles visualization.

The CSCS visualization tool **sparticles** makes possible the rendering of huge particle datasets.