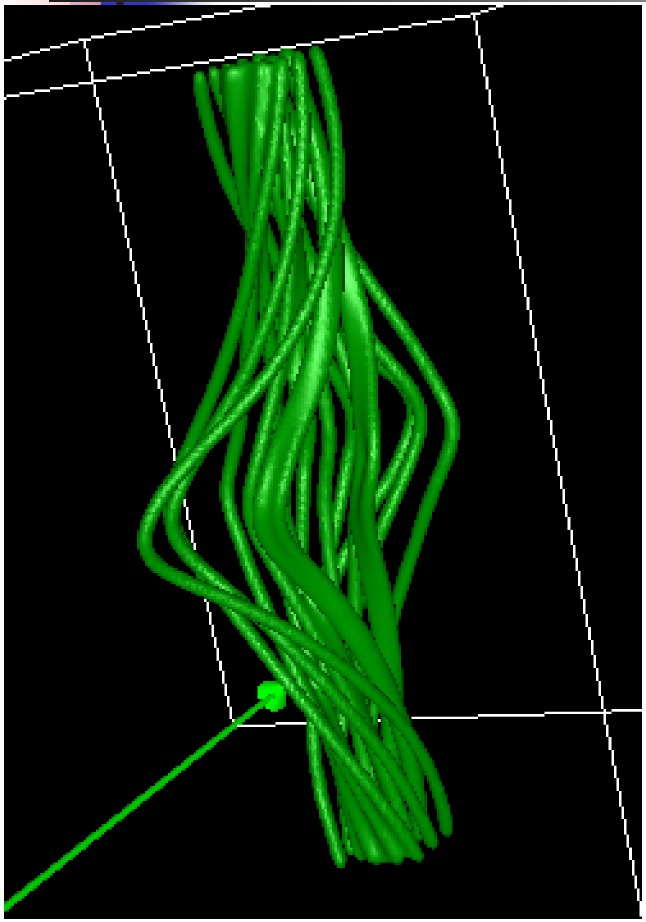
Animation Of Current Sheet Formation



E.Yu. Echkina, I.N.Inovenkov, A.V.Leonenko

Moscow State University

Faculty of Computational Mathematics and Cybernetics

Department of Research Automation

International Conference Graphicon 1999,
Moscow, Russia, http://www.graphicon.ru/

Approaches To Scientific Visualization

- Data exploring (visualization is used to explore computational data)
  - Real-time
  - Simple navigation
- Illustration (visualization is used to represent a research results to others)
  - High quality
  - Unusual representation

What Do We Study?

- We study plasma. One can think of it as a ion gas. It is a usual state of matter at high temperature.
- From mathematician point of view it is an object described by a system of nonlinear MHD equations.
- In general case it is impossible to solve those equations. That’s why the only way to study plasma is to use computers.
- Since the objects of interest are magnetic and velocity vector fields, scalar temperature field and their relations it is absolutely necessary to visualize the results of computation.

What Do We Need During Our Plasma Research?

- Visualize unsteady 3D vector fields
- Have an ability to include those images in scientific articles
- Real-time is not required since computation is very slow and we have all time we need to explore the data
- High quality is desired

so …

... to create a sequential frames of animation we use :

A “Streamball” Method

- Use a standard streamlines to represent a topology of the 3D vector field.
- Build an additional scalar field to represent a streamlines as isosurfaces.
- Produce a high quality images with reasonable effort.
- Have a great flexibility to produce streamsurfaces, timesurfaces, etc.

How It Works?

- We trace a streamlines with standard methods (4-th order Runge-Kutta + Adams-Moulton + trilinear in-cell interpolation in our work)
- At each point along a chosen streamline we define an influence function in form of :

$$I\left(\overline{x}, \overline{y}\right) = e^{-f\left(x_0 - y_0, x_1 - y_1, x_2 - y_2\right)}$$

- Finally we add all influence functions along all streamlines to create the scalar field in form of :

$$H(S, \overline{x}) = \sum_{\overline{y} \in S} w\left(\overline{y}\right) I\left(\overline{x}, \overline{y}\right)$$

- If we define an influence functions with sufficient density then almost any “good” isosurface of above field will represent a streamlines as a long tubes

What’s the Problems?

1. Big amount of memory is required to store the scalar field
2. Slow field generation
3. Slow isosuface extraction
4. Lots of triangles in final mesh

Possible Solutions

- Use “marching cubes” algorithm to extract isosurface approximation
- Store only two slices of a grid in memory to minimize its utilization
- Use polynomial approximation to compute the influence function
- Use “spreading cubes” to extract isosurface approximation
- Use local maximum graph to locate the seed cells
- Use influence functions which guarantee local maximum conservation
- Do not store or compute any unused field values

Spreading Cubes Isosurface Extraction

- If
  - We can easily find cell neighbors,
  - We have a seed cell for each connected component of isosurface,

  Then we can construct an isosurface approximation by propagating the surface from the seed cells to its neighbors then to neighbors of neighbors and so on
- At each cell we can apply a standard marching cubes approach to construct an isosurface triangulation

Spreading Cubes Isusurface Extraction

- In case of regular grid we can easily find the neighboring cells. In other cases it could be a problem
- Its been proven that the set of seed sells could be found by generating of a set of cells which contains arcs of any connected graph of field extremas. In general case its not simple to find the extremas and this set may contain a large number of cells, but in our case this set can be made small so it will worth the efforts

Maximum Graphs

- Since we know the scalar field structure we can try to simplify the process of seed cells generation in following way:
  - Without loss of generality we can demand that the required isosurface do not have an intersections with border cells (we can always stop before a border or propagate behind it)
  - We can choose the influence function in the way that the additional scalar field will have a local maximums in the center of the influences
- In above case we can generate the connected graph of local maximums and one border cell very easily. Actually we already do the most part of its generation during streamline tracking

Maximum Graphs

- We *state* that this graph will have an intersection with all connected components of an any isosurface with possible exception of those components which lies completely inside the volume bounded by another connected components. This means what if we do not plan to draw a semitransparent surface we can use this graph to generate the seed cell list. It is possible to loose some part of isosurface, but it is an invisible part anyway.

Maximum Graphs

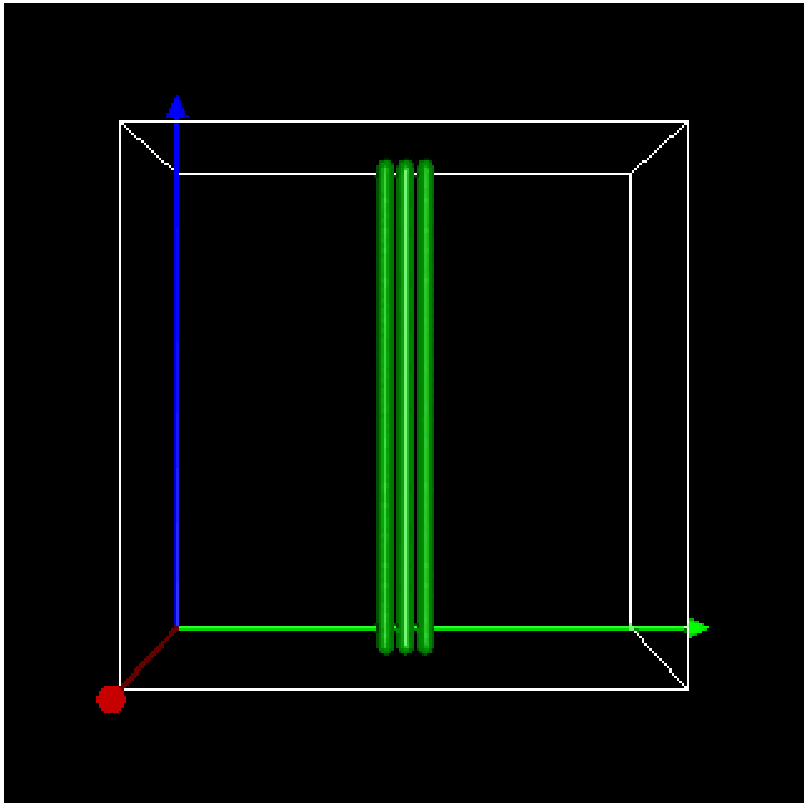
- Finally we proceed as follows:
  - Generate streamlines with standard methods. During this process, create the queue of cells visited by lines
  - Compute cells visited by a line which connects starting points of streamlines and some border cell and add them to the queue
  - For each cell in the queue we test it on intersection with isosufrace and if successful then add the corresponding neighbors to the queue and triangles to mesh
  - Proceed until the queue is empty

Scalar Field Evaluation

- As you can see there is to need to compute the scalar field at every grid points. It is only required to compute four values per each new cell in queue.
- This greatly reduces the memory requirements and even can speed up the field evaluation if we'll be able to solve a new problem of fast field value computation. In our work we used octree structure to speed up the search of neighboring influence points location.

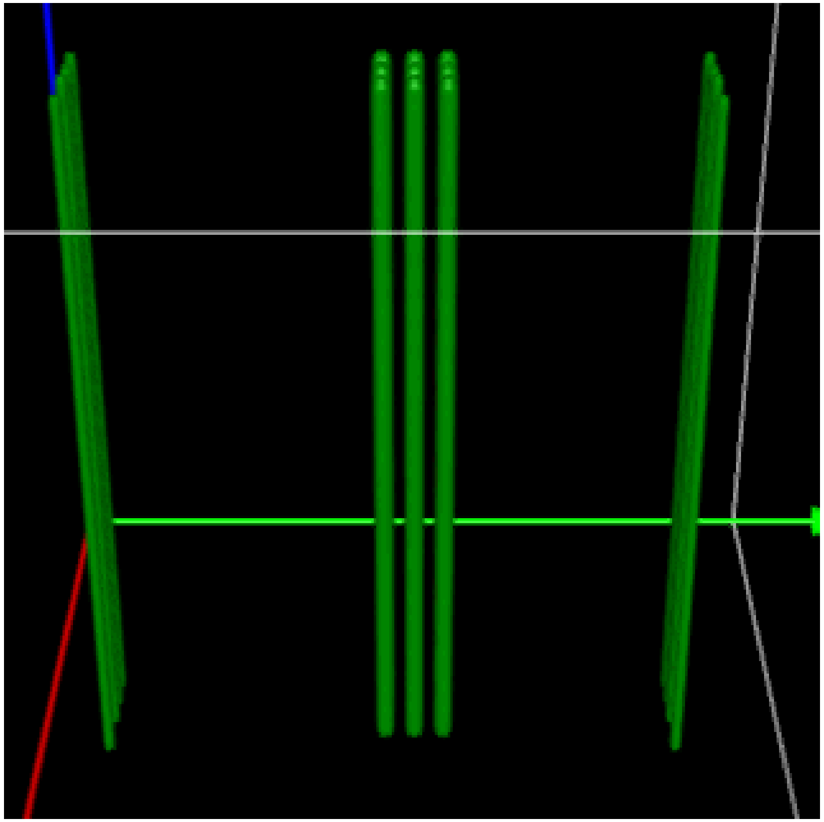
Some Results

- We present two movies made by sequentially applying of described method
- On this movie we see an evolution of magnetic field under border perturbations. We rotate the Z-border planes in opposite direction and indirectly change a velocity field. Magnetic field is changed due to "freezing-in" property of plasma.



Some Results

- On this movie we see a different behavior of magnetic field. Unidirectional currents repel each other and organize two different magnetic wisps from one.



Conclusions

- We describe a method used to create an animation of plasma processes
- It is general enough to be used in different applications which requires the streamline visualization
- This algorithm is relatively slow but it produce a high quality images and it is worth it's cost
- It is possible to merge this technique with colormapping and isoconturing to create even more informative visualization