



IEE

Colloquium on

Visualisation of three-dimensional fields

at Savoy Place on

Wednesday, 15 November 1995

organised by Professional Group S8(Electromagnetics) and
co-sponsored by Professional Groups E2 (Electromagnetic
compatibility), E11 (Antennas and propagation) and E12
(Microwave devices and techniques)

VISUALISATION OF SCALAR AND VECTOR FIELDS WITH VISUAL PROGRAMME CONSTRUCTION

I J Curington

Field Visualisation Environment

The software package "AVS/Express" provides tools for the visualization of complex scientific data, however also allows researchers to create experimental data representations. Field data interpretation applications are constructed using a visual programming environment, and includes graphical user interface tools (GUI). AVS/Express presents a screen giving overall control of the operating paradigm. Called the "Network Editor", it contains iconic representations of programme modules. Coloured lines connecting the modules represent data and control relationships in the application. Unlike other programming environments, no distinction is made between library components and running versions of these components, and dynamic changes of the running programme is allowed.

The modules in the network editor are managed by the system provided object manager. Control, messages, and shared data areas are managed by the object manager. The objects defined in the system, as well as the visual programming system and the object manager, are based on object oriented concepts. How the system behaves with regards to a traditional C++ program development environment is summarised below:

System	AVS/Express	C++
Mode of execution	Functional	Procedural
Inheritance	Single and multiple class and prototype inheritance	Single and multiple class inheritance
Polymorphism	Polymorphism via object matching and virtual objects	Polymorphism via inheritance and virtual functions
Encapsulation	Encapsulation of underlying C/C++ implementation	Encapsulation via public, protected and private access specifiers
Build time/Run time	Seamless	Distinct
Type System	Dynamic	Static

The visualization environment defines a polymorphic data structure called a "Field". Interactive visualization of data is possible through the use of tools that automatically match to incoming data structures. Meshes presented to the system may be of a structured, irregular, unstructured, or mixed multi-block type. Generic processing algorithms are used for processing data, or interpreting data within each cell, and more specifically tuned algorithms are used where faster indexing methods are available, such as with structured meshes. C, C++ and 4GL command languages are available for creating custom interfaces to existing external data formats.

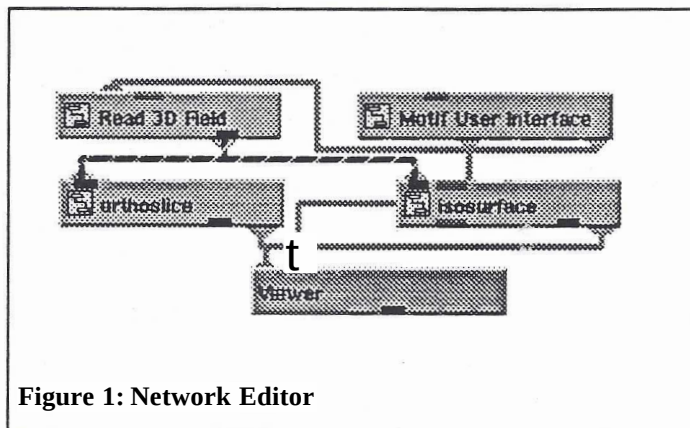


Figure 1: Network Editor

I J Curington is with AVS/UNIRAS Ltd., Chertsey, ianc@avsuk.com

Numerical field data associated with a mesh is typically a scalar quantity such as temperature or material type, a vector such as 3D flow velocity, a tensor such as stress, or a multi-component data structure. Through object oriented template matching, appropriate data types are processed by each visualization method. The

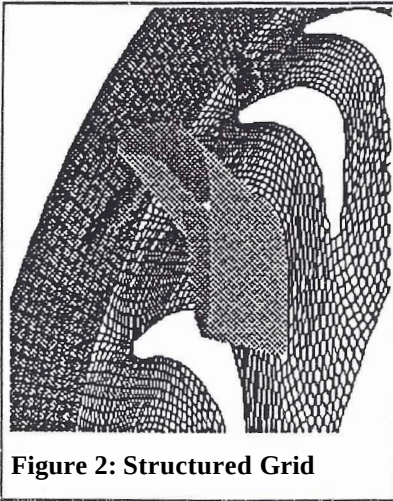


Figure 2: Structured Grid

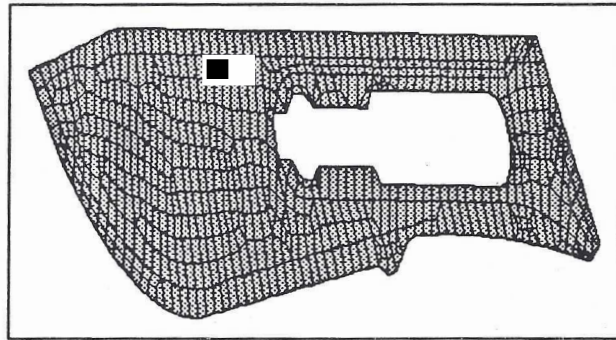


Figure 3: Unstructured Grid

network editor interface only allows connections of similar data types during application construction.

AVS/Express contains a large library of visualisation modules, used for creating 2D and 3D graphical representations directly from numerical data. Any 3D Field may be processed through tools which slice, cut, extract surfaces, create isosurfaces from constant scalar field values, crop subregions, map alternate data onto surfaces, create scatter plots, contour plots, cross section line plots, and direct 3D transparent volume projections. Each tool can be decomposed and modified, such that custom data processing algorithms are introduced to the system.

Vector fields are interpreted by sampling at nodes of the mesh, or by arbitrarily positioned sampling meshes. Once vector sample positions are established, a separate library of visual representational objects are connected to create a 3D view. For scalar data, “glyph” object such as a sphere, cube, arrow, wind-barb is replicated at each of the sample points, with a data dependent colour and size. In this way 2D or 3D scatter plots are created, showing field variations. If a sampling position system is used, this can be moved within the field under interactive mouse control, so that the researcher can search for field variations of interest.

Vector data may be displayed by the same methods as scalar data, with the added attribute of orientation and directional scaling taken from the vector components. By using an arrow “glyph” object, with a 2D plane sample system, a standard vector arrow plot is constructed. A further level of glyph indirection is added through a glyph index lookup system, where discreet data levels are assigned to different glyph objects.

Further vector field interpretation may be made with “streamlines” or “particle advection” modules. These are applied to a continuous vector field by integrating a series of positions for synthetic massless particle. The integration method used within the model cells is Runge-Kutta 3rd order, and uses adaptive step size to achieve accuracy in high dynamic range field data. Particles are animated through the field and shown in a dynamic display window.

Direct Interaction

All 2D and 3D graphic display windows allow for mouse interaction, through a “virtual trackball” interface. This interface allows direct interaction with viewing position, object position, object selection, interactive data probing, and region of interest sketching and editing. If the interactive probe is connected to the “glyph” system, the probe will change colour, size and orientation to represent the sampled data as the user moves the probe position through the field.

Simulation and analysis programs may be linked into the AVS/Express system as visible modules in the network editor. Once in the environment, they may be dynamically coupled to 3D graphics and mouse interaction objects. User interface components such as sliders and dials can be coupled directly to control parameters of the simulation, allowing users to see the simulation progress in a graphics window, and the same time “steer” the simulation by changing controls.

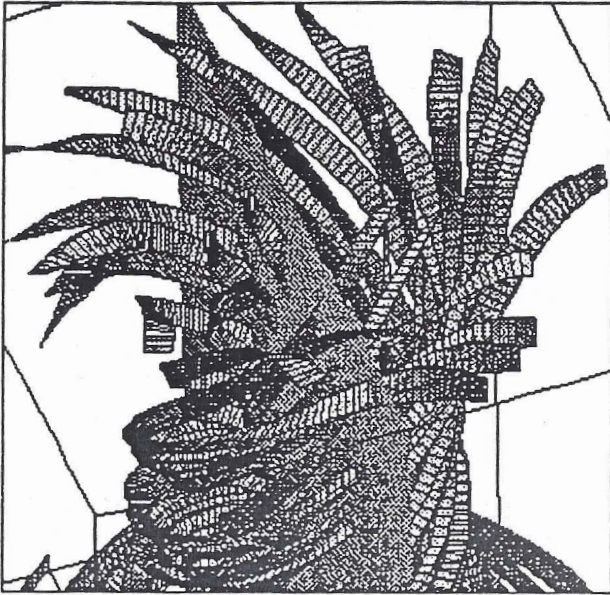


Figure 4: Vortex Stream Ribbons