MATFOR 3

The advanced visualization toolkit built on Fortran and C++ specifically designed for programmers in scientific computing field.

- Advanced 2D/3D visualization: a set of high quality 2D/3D visualization plot
- Instant visualization: no need for Windows programming and graphics initialization
- Movie-like presentation: a dynamic way of thinking and interpreting of your simulation
- Real-time animation: real-time data monitoring as programs are executing
- Matlab-like syntax: Matlab-like syntax provided in Fortran and C++ environment
- Numerical Library: library of high accuracy and high performance (based on Intel® MKL)

MATFOR® contains a set of numerical and visualization libraries developed specially for scientists and engineers. The functions in the libraries enhance your Fortran and C++ programs with dynamic visualization capabilities, shortens your numerical codes and, speeds up your development process.

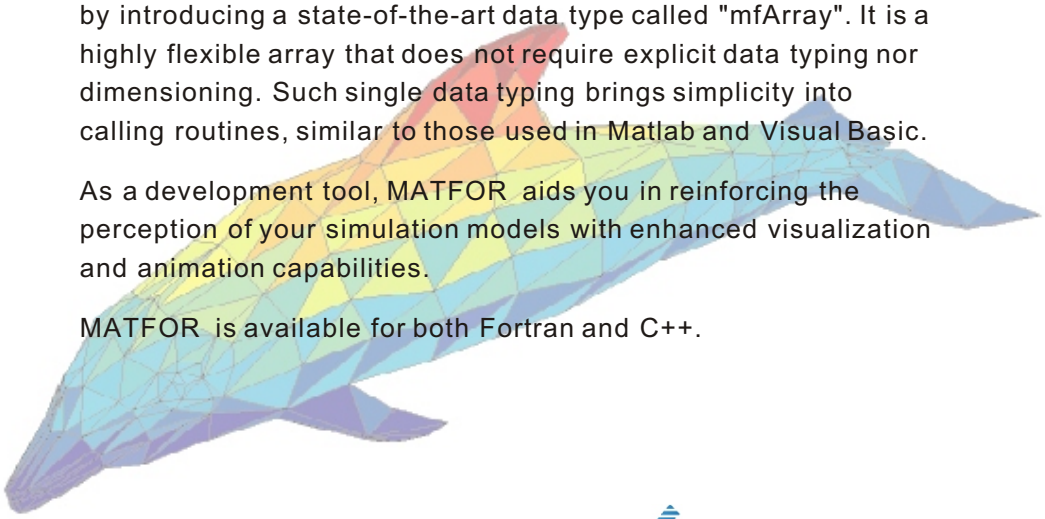
By adding a few lines of MATFOR function calls to your Fortran and C++ programs, you can easily visualize your computing results, perform run-time animations, or even produce a movie presentation file as you execute your program.

Debugging is facilitated with the debugging facilities provided by MATFOR Graphics Viewer. You can pause an animation, view the current data using MATFOR Data Viewer, and examine any aberrations.

MATFOR adopts the simple calling concept that is used in Matlab by introducing a state-of-the-art data type called "mfArray". It is a highly flexible array that does not require explicit data typing nor dimensioning. Such single data typing brings simplicity into calling routines, similar to those used in Matlab and Visual Basic.

As a development tool, MATFOR aids you in reinforcing the perception of your simulation models with enhanced visualization and animation capabilities.

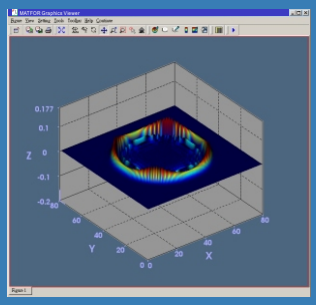
MATFOR is available for both Fortran and C++.



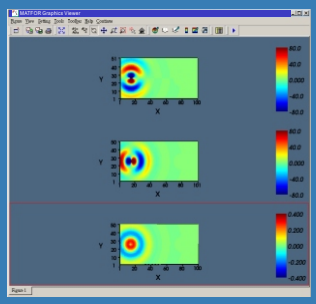
Dolphin figure. 3-D surface plot of a dolphin model.

AnCAD
Simply Faster

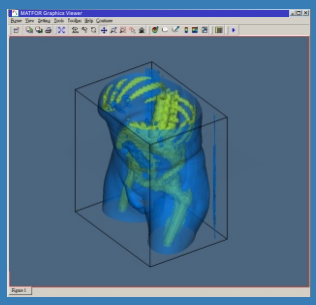
Key Features



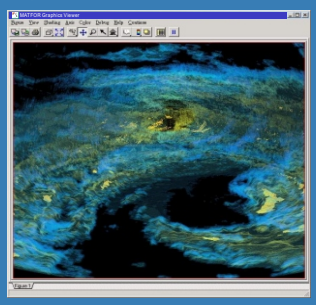
EM wave simulation. A 3-D dynamic simulation of the transmission of electromagnetic wave.



Wave analysis. An analysis of wave transmission.
Data courtesy of Precision Instrument Development Center / Taiwan.



Medical image. A 3-D model of a human body constructed from Computer-Tomography (CT) data.
Data courtesy of NLM, The Visible Human Project.



Lekima typhoon meteorological radar data analysis. A 3-D dynamic animation displaying the distribution of the cloud layers within a typhoon.
Data courtesy of Weather Bureau 2001/Taiwan.

Syntax

MATFOR's mfArray provides dynamic data typing and dimensioning and it acts as the basic of MATFOR. This fundamental data structure brings simplicity into your programming as it enables you to use Matlab-like syntax in Fortran and C++ environments.

Numerical Library

MATFOR's numerical library is a collection of mathematical functions which are designed to be intuitive and simple to use.They are organized into five major categories.

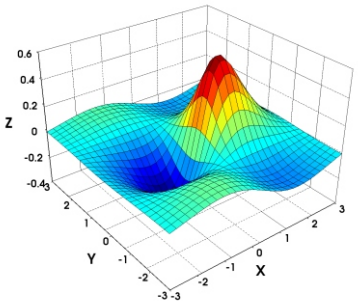
- Elementary functions (elfun)
- Elementary matrix manipulation functions (elmat)
- Matrix functions for solving numerical linear algebra problems (matfun)
- Matrix and array manipulation operators and functions (ops)
- Data manipulation functions (datafun)

Surface Rendering

Display three-dimensional surface objects using various drawing styles and color schemes. With MATFOR's Graphics Viewer, you can freely apply different representation methods to your simulation model depending on what you want to emphasize on.

- Three-dimensional iso-value surface plot from volume data.
- Three-dimensional graphs composed of colored quadrilateral surfaces. You can choose several shading options including mesh, flat, faceted, and interpolated.
- Combination of surface plot and contour plot.

A few lines of MATFOR® function calls generating a surface plot.



- Transparency and shading control API
- Colormap API

Graphics

Contain a set of high-level visualization functions for two-dimensional and three-dimensional data visualization, animation, and graphical debugging. They are easy-to-use and have a wide range of applications.

- Two-dimensional and three-dimensional linear graphs.
- Two-dimensional and three-dimensional velocity vectors.
- Two-dimensional and three-dimensional contour plots with labels. Both regular and solid contours are supported.
- Streamline from two-dimensional or three-dimensional vector data.
- 3-D Objects, including cone, cube, cylinder, sphere, ribbon, and 3-directional axis mark.
- Draw stick and ball models of molecules.
- Display orthogonal slice-planes through volumetric data along arbitrary directions and matrix indices.
- Object Eular angle(orientation, RTS) adjustment.
- Unstructured mesh.

- Unstructured grids.
- Delaunay triangulation.

Viewing

Display single or multiple figures in one or more windows using MATFOR Graphics Viewer. It also contains various editors and tools for you to customize the displaying mode and navigate through the graphics objects more easily.

- Customize window size and position
- Zooming area selection
- Transparency and shading control editor
- Axis setting editor
- Colormap editor
- Clipboard and print function
- Smart axis adjustment
- Camera manipulation

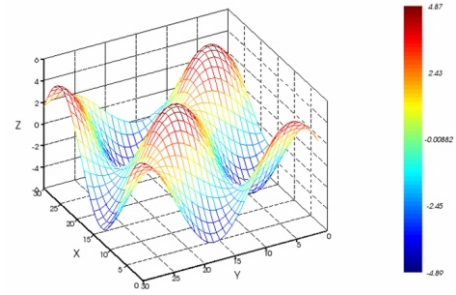
Data Viewer

Display multi-dimensional array data in a spreadsheet-like data viewer. There are some manipulating operations that can be performed on the data, including filtering, data range selection, and statistic analysis.

- Support user-defined filtering conditions. It enables you to focus on any portion of the data model.
- Quick data snapshot enables you to grab an overview of the data.
- Data histogram illustrates the distribution of data.
- Data range selection. Display sub-matrix in the combination of arbitrary two dimensions.
- Statistic analysis including average, standard deviation, maximum value, and minimum value.

Example Codes

Figure below is the plotting result and the codes for producing a mesh plot. It illustrates how the mesh plot procedure is embedded into Fortran and C++ program.

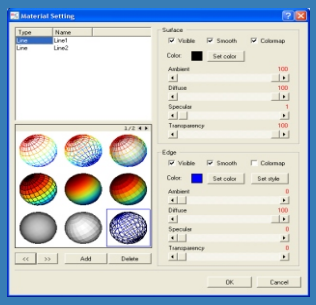


Example code in Fortran

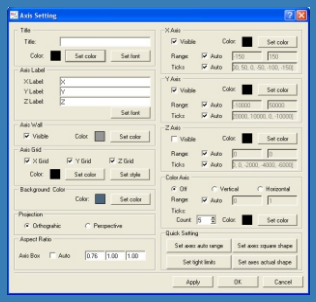
```
      Program Mesh
        use fgl
        implicit None
        integer i, j, N
        real(8) z(30,30)

C     Create data for plotting
        do 100 i = 1, 30
          do 200 j = 1, 30
            z(i,j)=3.0*sin((i+1)/
     +      4.0)*cos((j+1)/4.0)+2.0
     +      *sin((i+j)/4.0)
  200     continue
  100   continue

C     Convert Fortran variable z
C     to mfArray using mf(z), and
C     call msMesh for Mesh plot
        call msMesh(mf(z))

C     Pause to display the graph
        Call msViewPause()
      End Program Mesh
```

Example code in C++

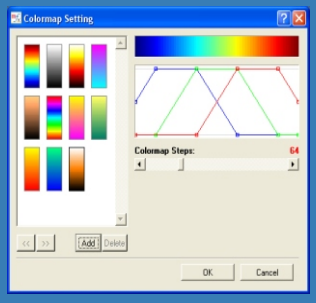
```
#include <math.h>
#include "fgl.h"

void main()
{
  int i,j;
  double z[30][30];

  // Create data for plotting
  for (i=0; i<30; i++){
    for (j=0; j<30; j++){
      z[i][j] =
      3.0*sin((i+1)/4.0)
      *cos((j+1)/4.0)+2.0
        *sin((i+j)/4.0);
    }
  }
  // Plot a mesh grid using
  // mfArray for the grid
  // intersections
  mfMesh(mfArray(&z[0][0],30,30));

  // Pause to display the graph
  mfViewPause();
}
```



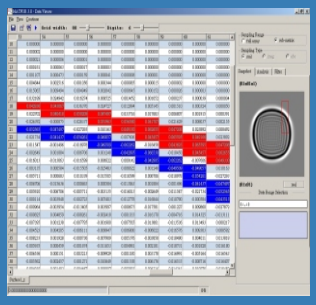
Transparency and shading control editor. Allowing you to perform material shading intuitively.



Axis setting editor. Allowing you to perform axis adjustments after the program is run.



Colormap editor. Allowing you to customize your own colormap very easily.



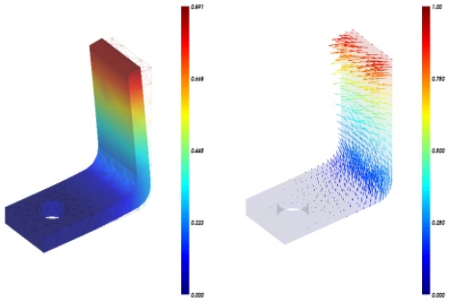
Data Viewer. Allowing you to represent your data in spread sheet format and do some data manipulation.

Presentation

Present results of your simulation models in a movie-like animation and allows you to do graphical manipulation on it. The supported animation recording techniques are:

- Frame capturing. Pictures displayed on screen can be captured and saved into picture files in format like bmp or a movie file in format like avi (Microsoft Audio Video Interleave). The avi file can be replayed using media players.

Display of the deformation and the displacement vectors of an L-shape steel board.



Platform and System Requirements

Platform	Operating System	Compiler	
		In Fortran	*In C++*
Intel Based systems 32-bit	Windows 98/NT/2000/Me/XP	• Compaq Visual Fortran 6.6 above • Intel Fortran 7.1 • Lahey Fortran 5.7 • Microsoft Fortran PowerStation 4.0 • Intel Visual Fortran 8.0 • PGI Fortran 4.0 above	• Intel C/C++ 7.1 above • Visual C++ 6.0 • Borland C++ Builder 4.0 above
Intel Based systems 32-bit	Red Hat Linux 8.0 above	• Intel Fortran 7.1 • Intel Fortran 8.0 • PGI Fortran 4.0 above	• Intel C/C++ 7.1 above • GNU C++
Intel Based systems 64-bit	Windows Server 2003	• Intel Fortran 7.1 • Intel Visual Fortran 8.0	• Intel C/C++ 7.1 above
Intel Based systems 64-bit	Red Hat Linux 8.0 above	• Intel Fortran 7.1 • Intel Fortran 8.0	• Intel C/C++ 7.1 above
SGI 64-bit	IRIX 6.5	• NAGWare Fortran 95	

*** Upcoming supports are in Red.**

(Feb. 2004)

AnCAD, Inc.

AnCAD has been established and devoted to the field of scientific computing in Taiwan since 1999, with an initial emphasis on developing software related to scientific computing, such as matrix-based operation, visualization library, mesh generator, translator and gateway.

Such software in the scientific computing industry can be applied to the evolution of the most advanced technology and scientific research, like high frequency interconnect analysis, molecular simulation, IC design simulation, electro-magnetic analysis and simulation, molding analysis, heat transfer, and Computational Fluid Dynamics (CFD).

The perspective of AnCAD is to provide the best scientific development tool set and parallel computing, cross-platform programming environment. Moreover, AnCAD elaborates applications for the field of bioinformatics, MEMS (Micro-Electro-Mechanical System), financial engineering, etc. In the coming future, AnCAD is at the expectation of being the leading corporation in worldwide integrated scientific computing environment.

AnCAD Simply Faster

AnCAD, Inc.
5F, No. 67, Sec. 1, Yung-Ho Rd.,
Yung-Ho City, Taipei, Taiwan
Tel: (886) 2 8923-5411
Fax: (886) 2 2928-9364
Email: sales@ancad.com
http://www.ancad.com