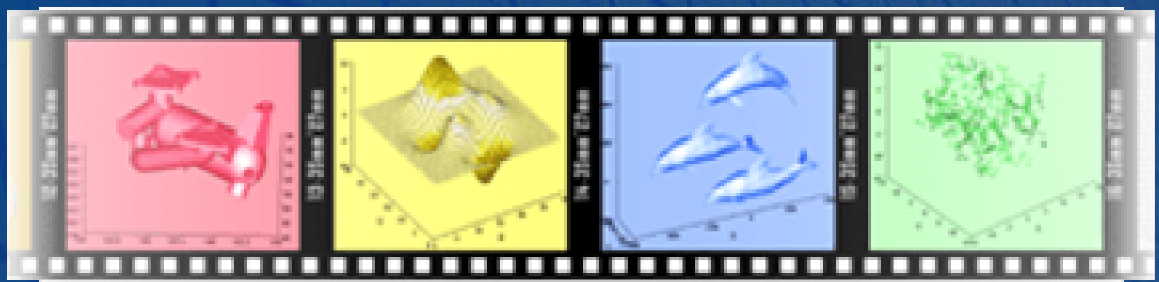
Visualizing the Physical Phenomena for Computational Science

with MATFOR® 4



Liger Chen
AnCAD **Inc.**

Outline

Introduction to **MATFOR®**

MATFOR® Functions

Cases Using **MATFOR®**

Visualization of Physical Problem by **MATFOR®**

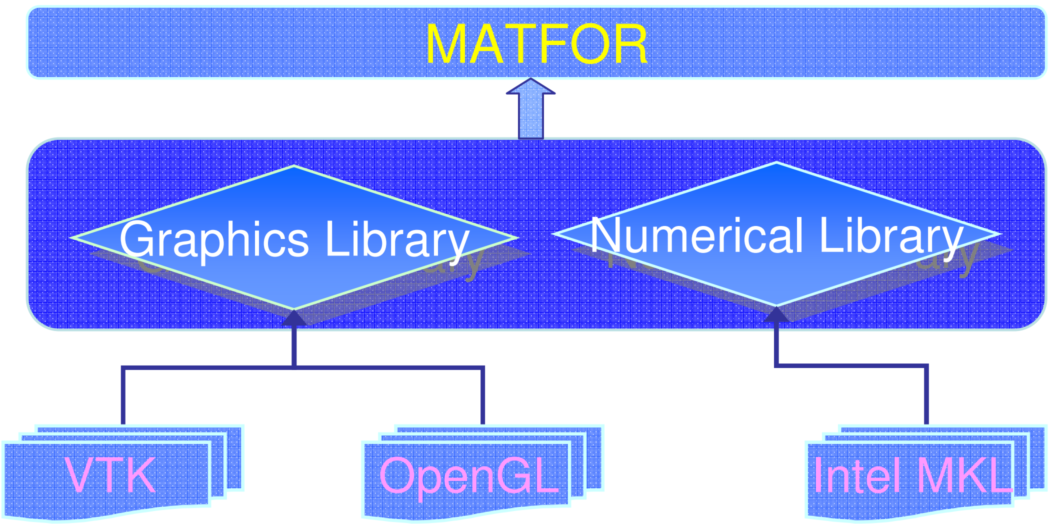
Introduction to MATFOR®

What's MATFOR® ?

- MATFOR® is a set of numerical and visualization libraries especially designed for programmers in scientific computing field.
- MATFOR® is a new generation graphics library fully exploiting the modules and the array features of Fortran 90/95 and C++ languages.
- MATFOR® is a collection of high-level graphical procedures developed with the mission of reducing time spent on the program development.

MATFOR® Structure

MATFOR®, a set of numerical and visualization libraries, is developed to enhance programming in C++ and Fortran environments. Especially designed for scientists and engineers, MATFOR® fulfills the needs of speed and advanced visualization capabilities simultaneously.



AnCAD

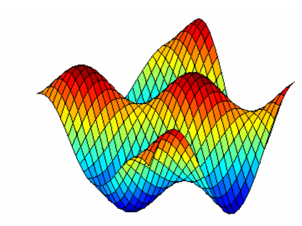
Numerical Library

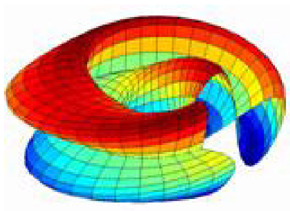
Based on Intel® MKL, the numerical library contains over 200 easy-to-use numerical functions subject to assist users with computational problem-solving.

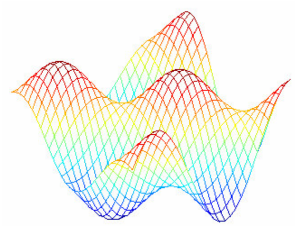
- Data Manipulation Functions:
  mfSort, mfMin, mfMax, …
- Elementary Math Functions:
  mfSin, mfCos, mfASin, mfExp, mfAbs, …
- Elementary Matrix-Manipulating Functions:
  mfEye, mfDiag, mfRand, mfZeros, …
- Matrix Analysis:
  mfEig, mfInv, mfSvd, mfQz, mfLu, mfDet, mfNorm, …
- Sparse Array:
  msSpAdd, msSpSet, mfSpNNZ,mfSpLDiv,…
- File IO:
  mfSave, mfSaveAscii, mfLoad, mfLoadAscii, …

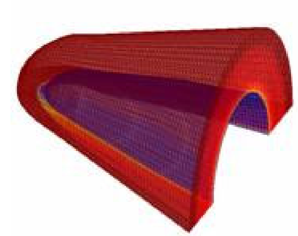
Graphics Library

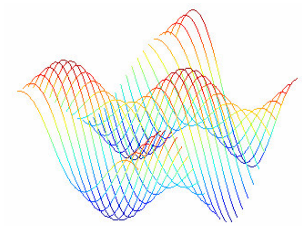
Visualization Modules I

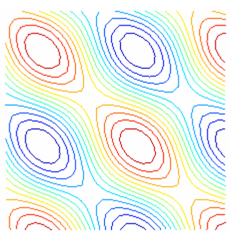

surf

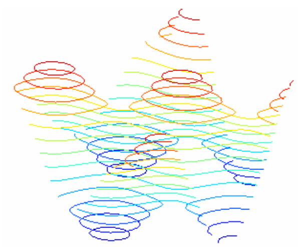

surf

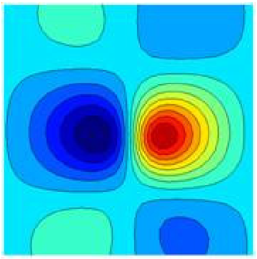

mesh

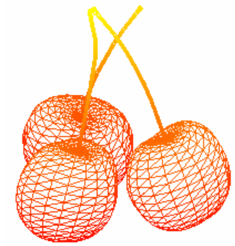

mesh

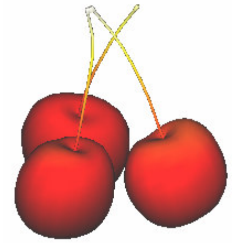

plot3

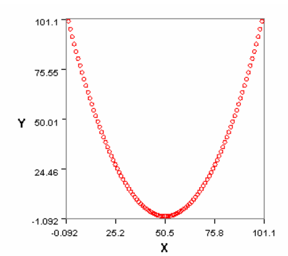

contour


contour3

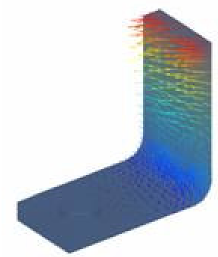

solidcontour

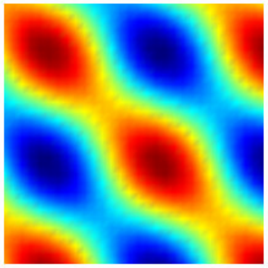

trimesh

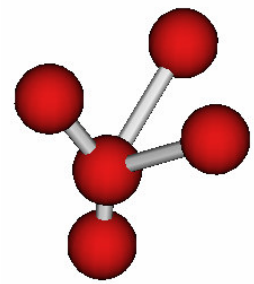

trisurf

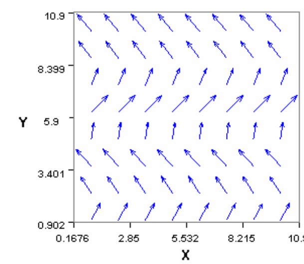


plot


quiver3


pcolor

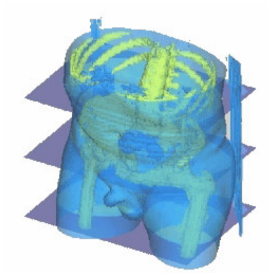

molecules

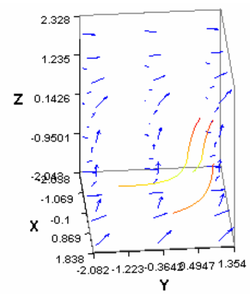


quiver

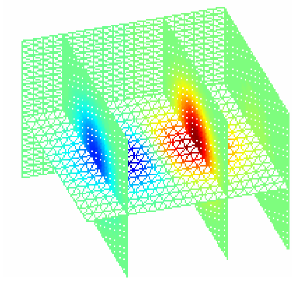
Graphics Library

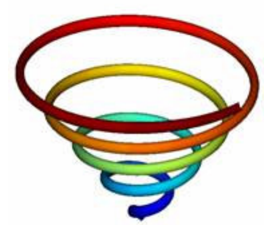
Visualization Modules II

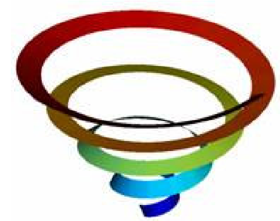

isosurface

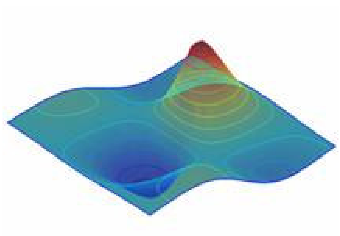


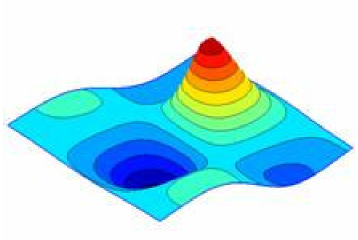
streamline

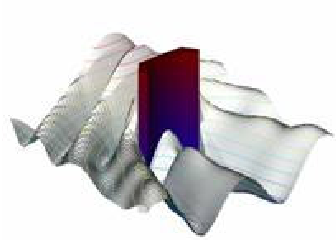

slices

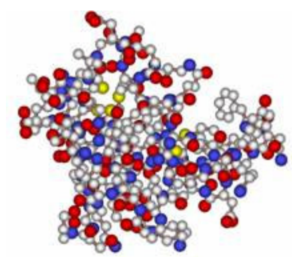

tube


ribbon

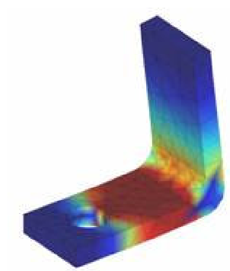

contour3

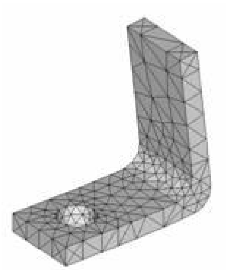

solidconour3

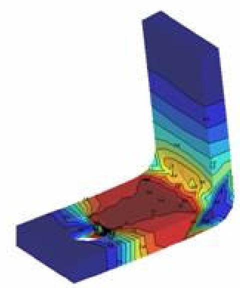

tricontour

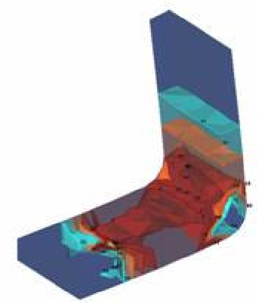

molecule

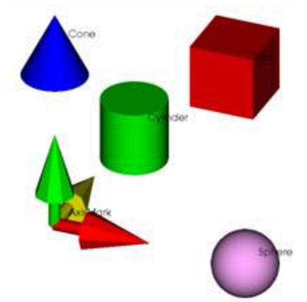

image


tetsurf


tetmesh


tetcontour

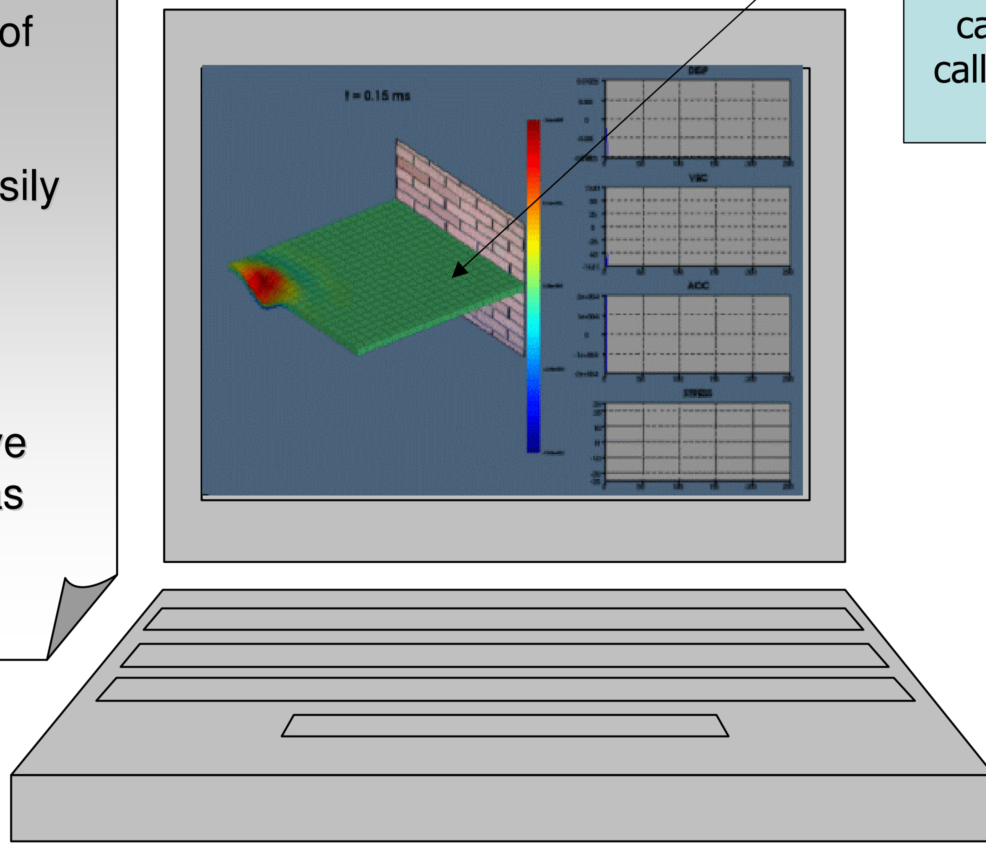

tetisosurface


cube, sphere, …

What Does MATFOR® Do?

By adding few lines of MATFOR® codes to your Fortran/C/C++ program, you can easily visualize your computing results, perform run-time animations, or even produce an interactive movie presentation as you execute your program.

```
…
call msSurf(x)
call msDrawNow
```

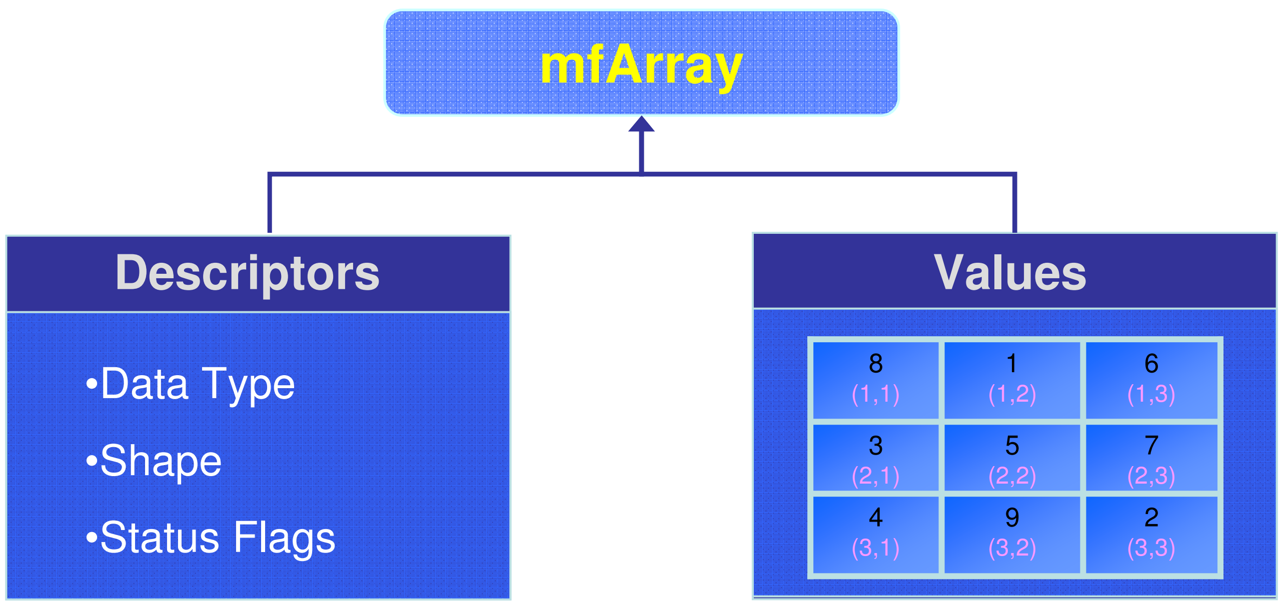


MATFOR® Functions

MATFOR® Dynamic Array I

mfArray Overview

mfArray is an advanced dynamic array defined by MATFOR® using modern features of C++ and Fortran 90/95. The mfArray data type consists of descriptors and values.



MATFOR® Dynamic Array II

mfArray Feature

A key to integrate MATFOR® toolkit into high-level programming environments

- Automatic data typing and dimensioning
- Dynamic memory allocation
- Simple calling routines with Matlab-like syntax

Construct and initialize the mfArray

C/C++	Fortran

mfArray x,y;
x = mfMagic(5);
y = mfInv(x);
	type(mfArray)::x,y
	x = mfMagic(5)
	y = mfInv(x)

Simple Using Mathmatical functions

- Besides its simple and powerful visual functions, MATFOR® enables scientists and engineers to code in Matlab fashion using the simple calling concept in Fortran and C++ environments.

MATLAB	a = inv(x)	e = eig(x)
MATFOR	a = mfInv(x)	e = mfEig(x)

- Calling msSVD in MATFOR® to perform singular value decomposition gives the same result as calling DGESVX in LAPACK. However, MATFOR® function only takes 3 pre-initialized parameters while LAPACK function takes 22 pre-initialized parameters.

```
Call DGESVX( FACT, TRANS, N, NRHS, A, LDA, AF, LDAF, IPIV, EQUED, R, C,
B, LDB, X, LDX, RCOND, FERR, BERR, WORK, IWORK, INFO )
```

LAPACK

```
Call msSVD( mfOut(A, B), C )
```

MATFOR

AnCAD

$$a(x)\frac{d^2y}{dx^2}+b(x)\frac{dy}{dx}+(c(x)-\lambda_m)y=0$$

- Solving the symmetric tridiagonal matrix problem :

$$\begin{pmatrix} C_{1,1} & C_{1,2} & 0 & \cdots & 0 \\ C_{2,1} & C_{2,2} & C_{2,3} & 0 & \vdots \\ 0 & C_{3,2} & \ddots & \ddots & 0 \\ \vdots & 0 & \ddots & \ddots & C_{n-1,n} \\ 0 & \cdots & 0 & C_{n,n-1} & C_{n,n} \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{pmatrix}=\lambda\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{pmatrix}$$

$$C_{p+1,p}=C_{p,p+1}$$

(this eigsystem problem maybe be the reduction of a 2nd order differential equation

subroutine tridib(n,eps1,d,e,e2,lb,ub,m11,m,w,ind,ierr,rv4,rv5)
lamda= w

LAPACK

eps1, lb, ub , m11 , lb , ind , ierr , rv4 , and rv5 = ????

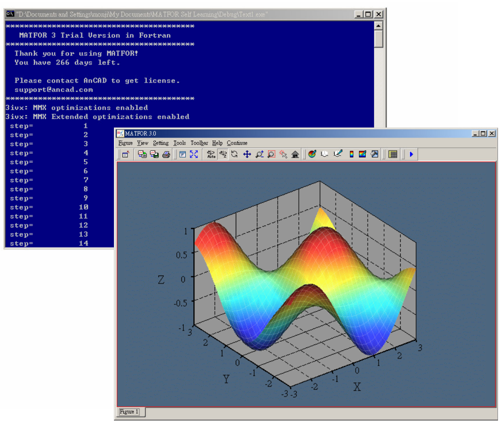
lamda = mfEig(x)

MATFOR

Real-Time Animation

MATFOR® features real-time program-monitoring mechanism to reduce time and effort spent on post-processing and debugging.

The simulation can be presented as an animation while the calculation is proceeded and shown in the console window.



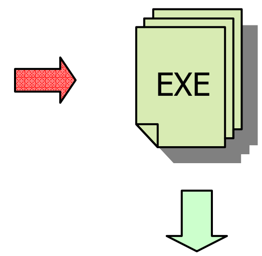
• MATFOR presenting simulation while calculating the data.

MATFOR® Sample Code

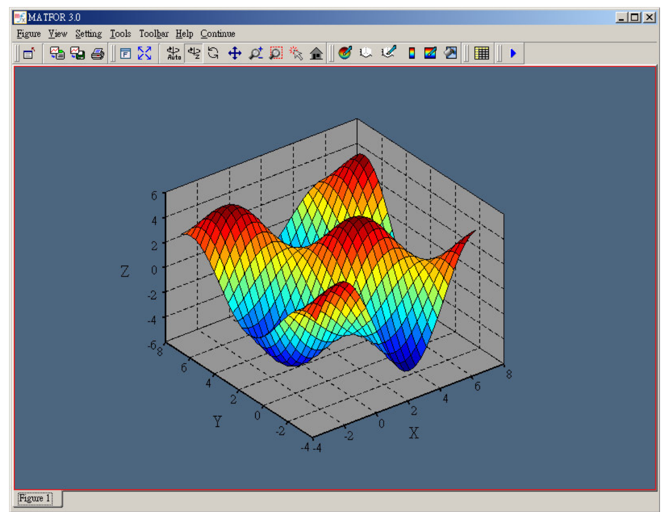
3D Presentation

```
program main

use fml
use fgl
! Declare mfArray variables
type(mfArray) :: x, y, z

! Construct grid matrices for 3D plot
call msMeshgrid(mfOut(x, y),
mfLinspace(-3,3,30), mfLinspace(-3,3,30))
! Calculate Z value
z = mfSin(x) * mfCos(y)
! Plot surf
call msSurf(x,y,z)
! Display the graph
call msViewPause
end program
```

demo codes



- MATFOR Standalone Executables



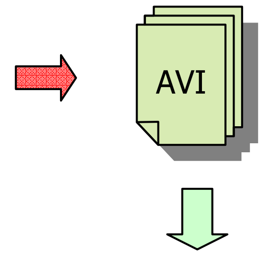
- MATFOR presents standalone executables.

MATFOR® Sample Code

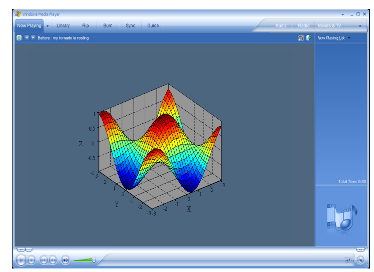
Movie Presentation

```
program main

use fml
use fgl

type(mfArray) :: x, y, z, h                     ! Declare mfArray variables
integer(4) :: I                                 ! Declare an integer variable
                                                ! Construct grid matrices for 3D plot
call msMeshgrid(mfOut(x, y), mfLinspace(-3,3,30), mfLinspace(-3,3,30))
call msRecordStart('demo.avi')                  ! Begin AVI recording
       do i=1,50                                ! Do loop
          write (*,*) 'step=', i
          z = mfSin(x+i/8.0d0) * mfCos(y)       ! Calculate Z value
          if (i==1) then                        ! Initialize handle
             h = mfSurf(x, y, z)
             call msDrawNow
          else
            call msGSet(h, 'zdata', z)          ! Update Z value
            call msDrawNow
          end if
       end do
call msRecordEnd                                ! End recording
call msViewPause                                ! Pause to display the graph

end program
```

demo codes



- Movie Files

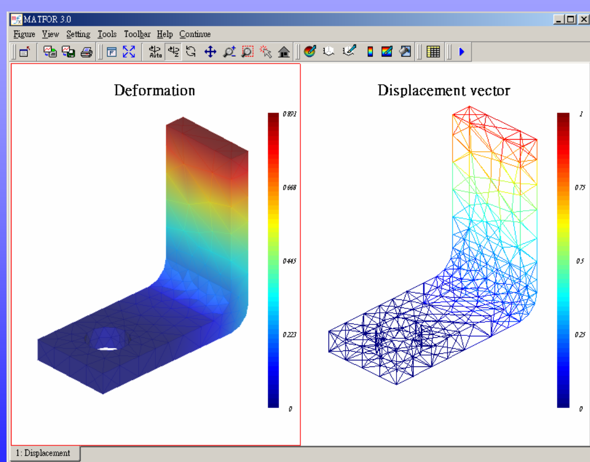


- Viewing the recorded AVI file with media player.

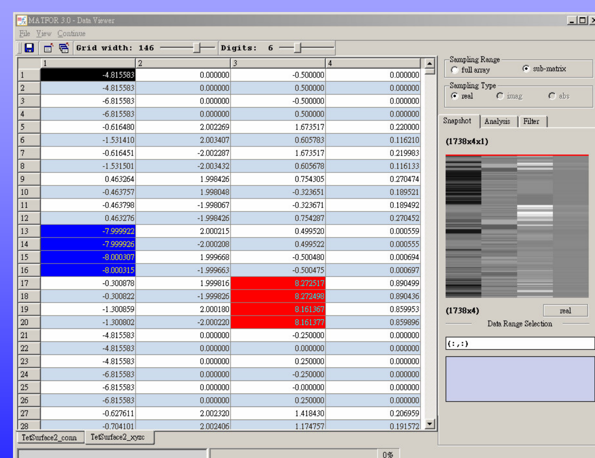
AnCAD

Runtime Data Manipulation

MATFOR® allows manipulation of the data displayed during execution; data can be examined with higher precision and customization at runtime.



- Graphics Viewer

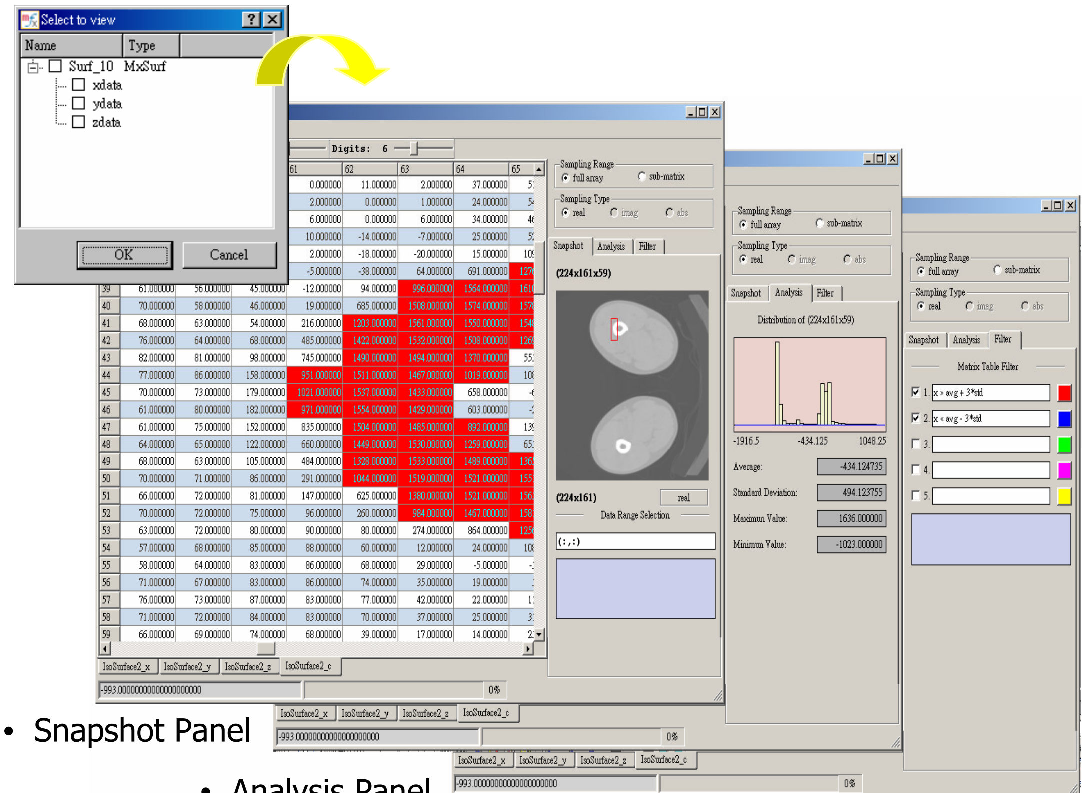


- Data Viewer

Data Viewer

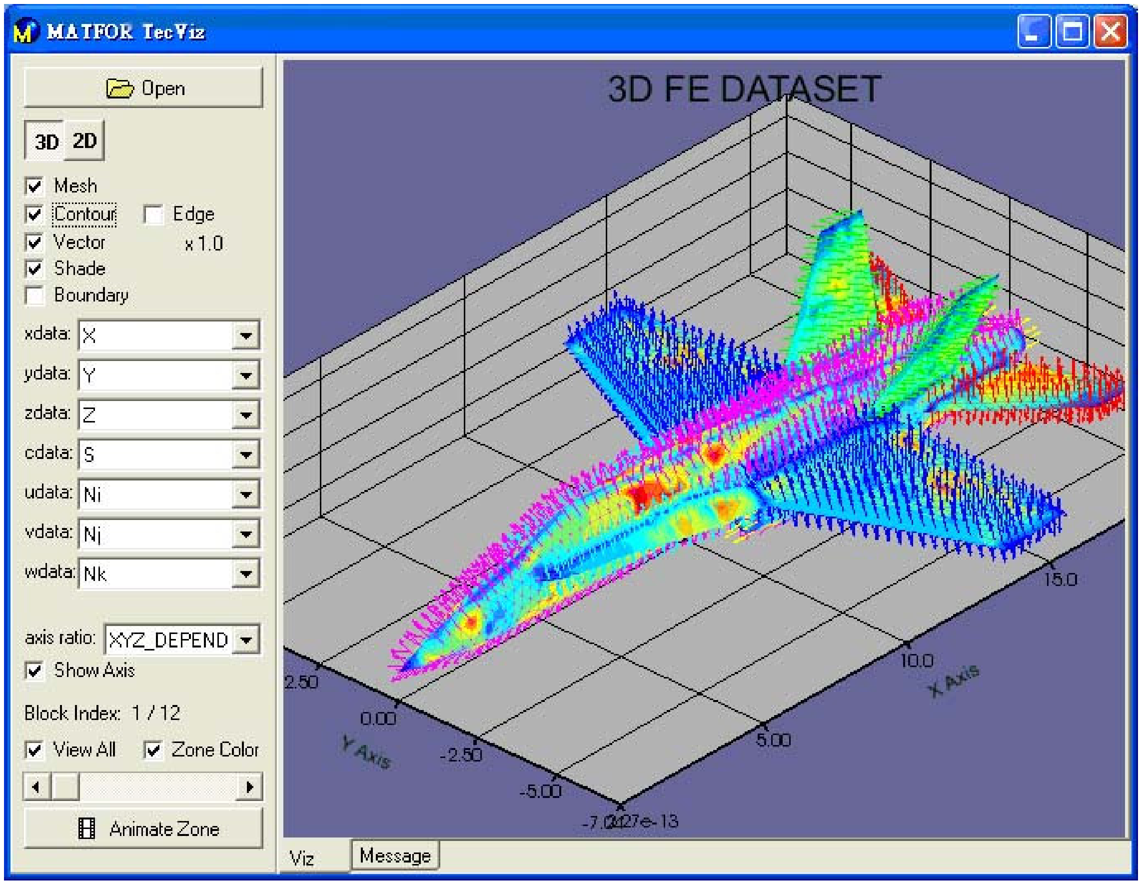
MATFOR® Data Viewer is a powerful tool that displays simulating data in a spreadsheet format.

- **Snapshot Panel**
  captures the snapshot of the distribution and size of the two dimensional data.
- **Analysis Panel**
  shows the distribution of the data including its average, standard deviation and min/max values.
- **Filter Panel**
  defines a range using conditions of inequalities.



- Snapshot Panel
- Analysis Panel
- Filter Panel

Supported Data File Formats

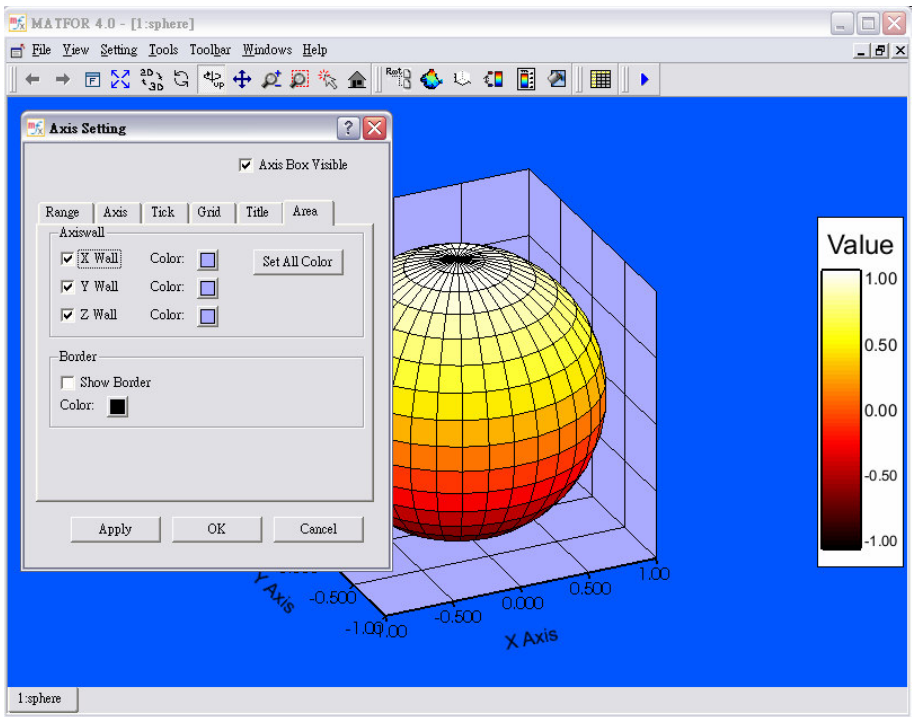


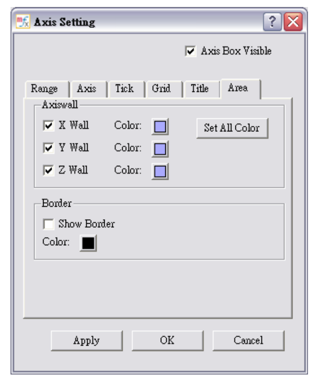
MATFOR 4 supports common software data formats and 3-D object formats to enhance the reusability and interchangeability of data.

- Matlab
- Tecplot
- 3DS, OBJ, and STL

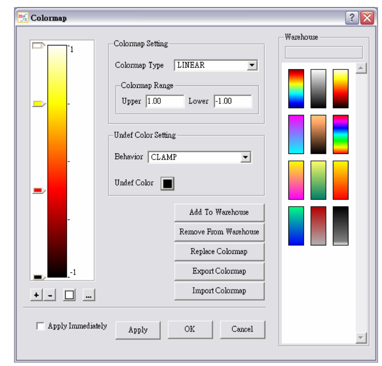
Graphics Viewer

MATFOR® Graphics Viewer provides a full range of graphical editing procedures which can be manipulated directly using the menu and the toolbar.

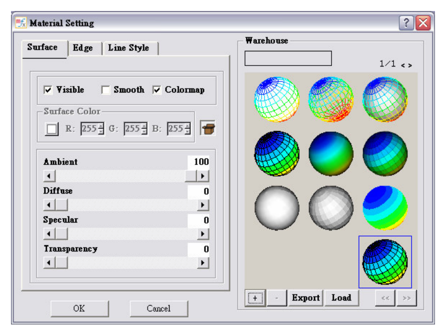




- Axis Setting Editor



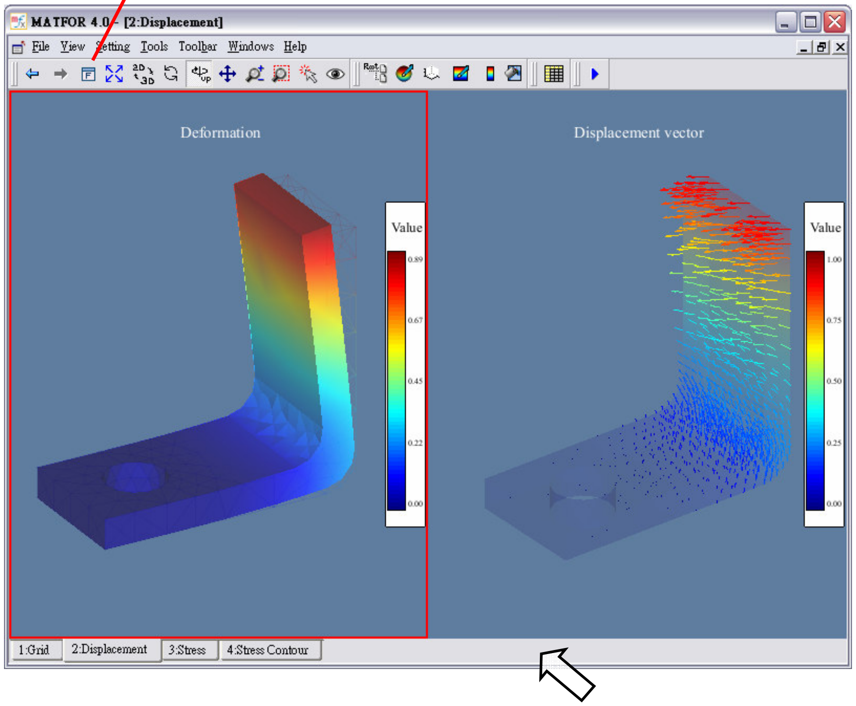
- Colormap Setting Editor



- Material Setting Editor

Enhanced Graphics Viewer
(Full-Screen Function)

Full Screen Mode



The full-screen function allows users to view and/or present data at full screen. This function also serves to eliminate the context for on-screen data capturing and printing.

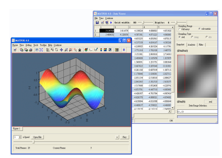
- Use the button indicated to show the graphs in full screen mode

MATFOR® mfPlayer

An exclusive Visual Tool

mfPlayer is an exclusive visual tool by which the previously saved numerical data is read and displayed as an interactive movie presentation. As MATFOR® saves the simulated data into a MATFOR®-defined MFA file, **mfPlayer** is one approach to present the recorded animation. The complete video clip can then be viewed from different angles.

- resize
- rotate
- zoom
- pause
- forward
- reward
- view data
- change colormaps

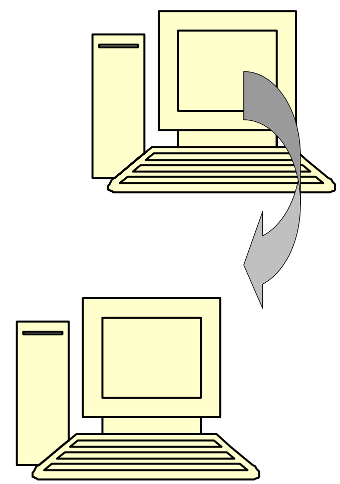


- Data Viewer
- mfPlayer

mfPlayer Standalone

- Currently, the proprietary format dominates in most visualization tools generates files that can only be executed on one specific application.
- MATFOR® possesses the ability to convert visualization files into standalone executables.
- Through the conversion, data sharing and publishing become much easier.

Compiling and linking the programs with MATFOR library.

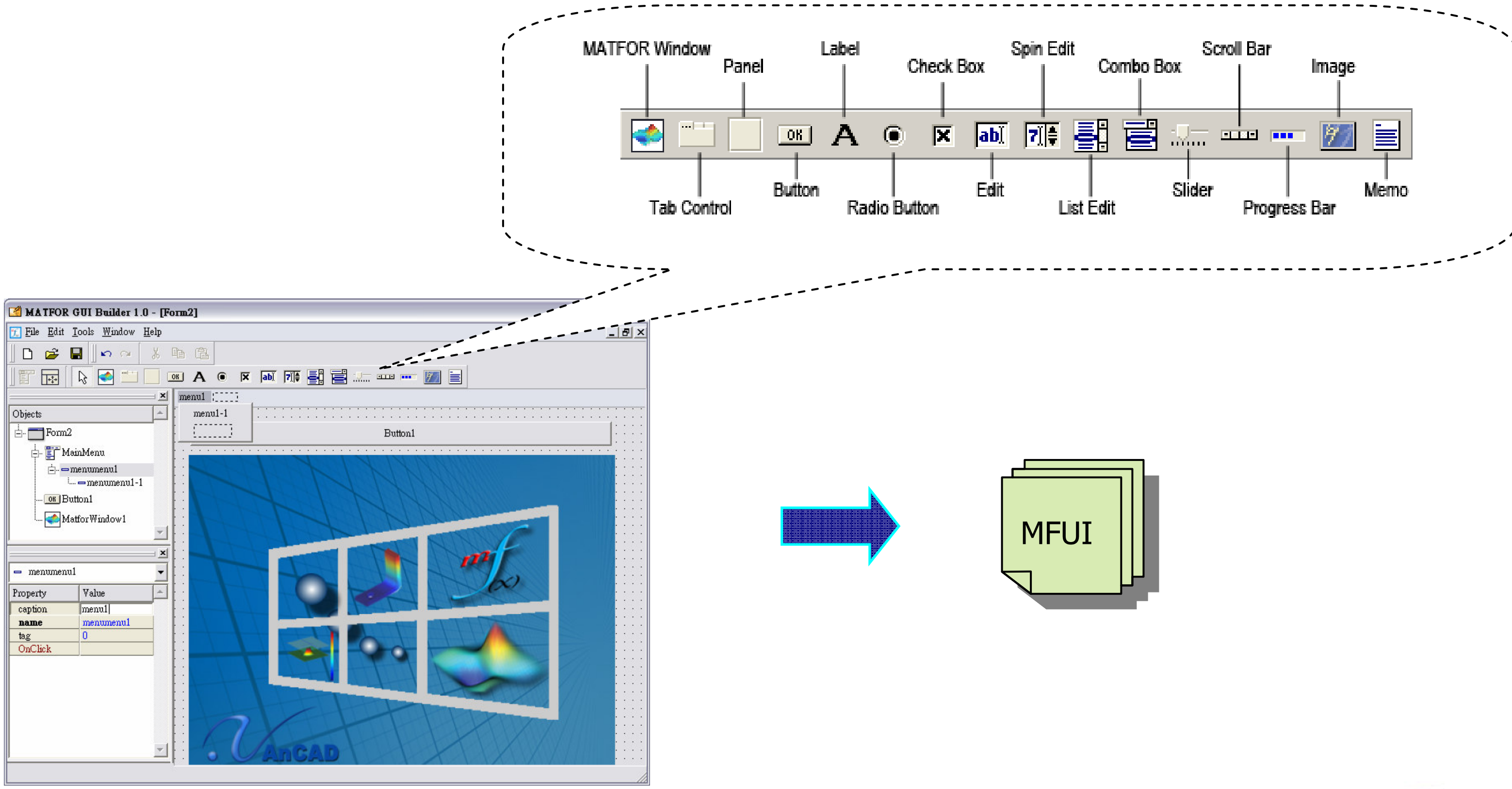


Running compiled executables without installing MATFOR.

AnCAD

MATFOR® GUI Builder

MATFOR® GUI Builder allows users to customize an interface of their own; the interface can be saved into a MFUI file (based on XML format).

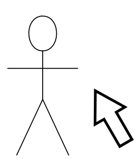


- MATFOR GUI Builder

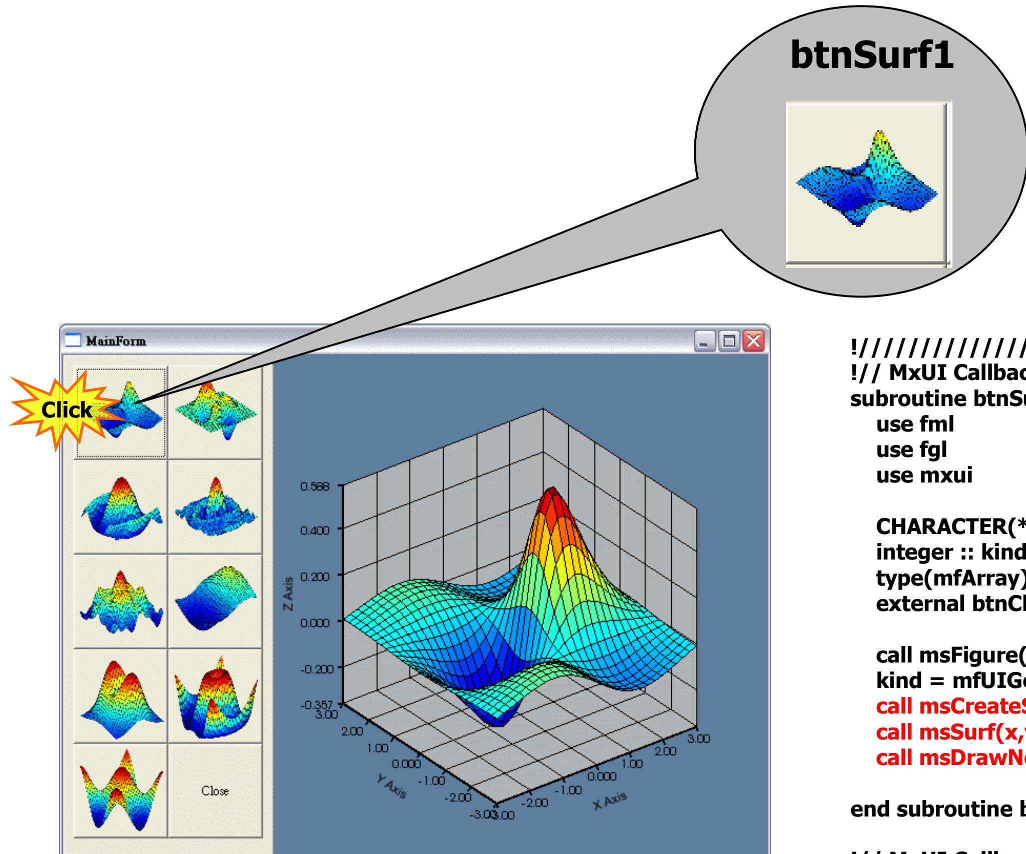
MATFOR® GUI System

(An Use Case)

- How does the communication work between the user and the application?

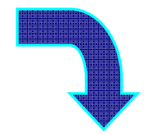


Actor



Application

UI Engine

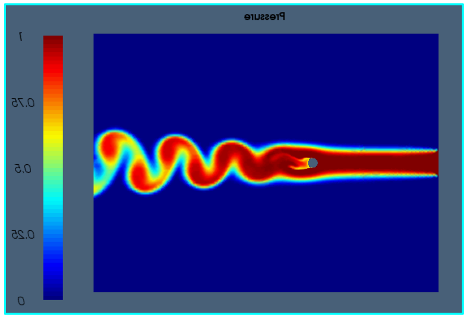


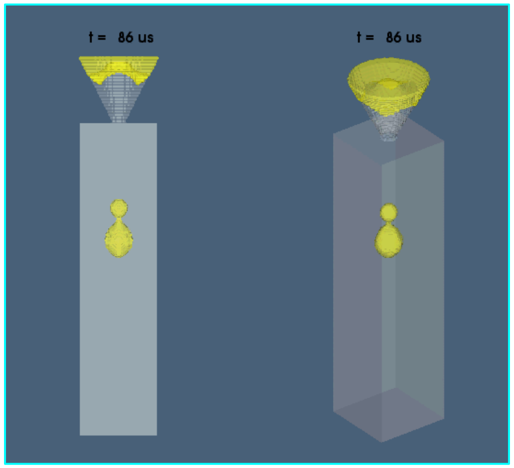
```
!//////////////////////////////////////////////////////////////////
!// MxUI Callback BEGIN
subroutine btnSurf_Click(sender)
   use fml
   use fgl
   use mxui

   CHARACTER(*) :: sender
   integer :: kind
   type(mfArray) :: x, y, z
   external btnClose_Click

   call msFigure()
   kind = mfUIGetPropertyInteger( sender, 'tag' )
   call msCreateSurfData( mfOut(x,y,z), mf(kind), mf(30), mf(30) )
   call msSurf(x,y,z)
   call msDrawNow

end subroutine btnSurf_Click

!// MxUI Callback END
!//////////////////////////////////////////////////////////////////
```

Cases Using MATFOR®

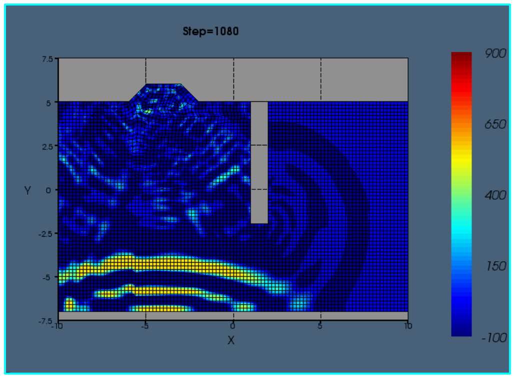
Applied Fields

- Solid Mechanics
- Fluid Dynamics
- Astrodynamics

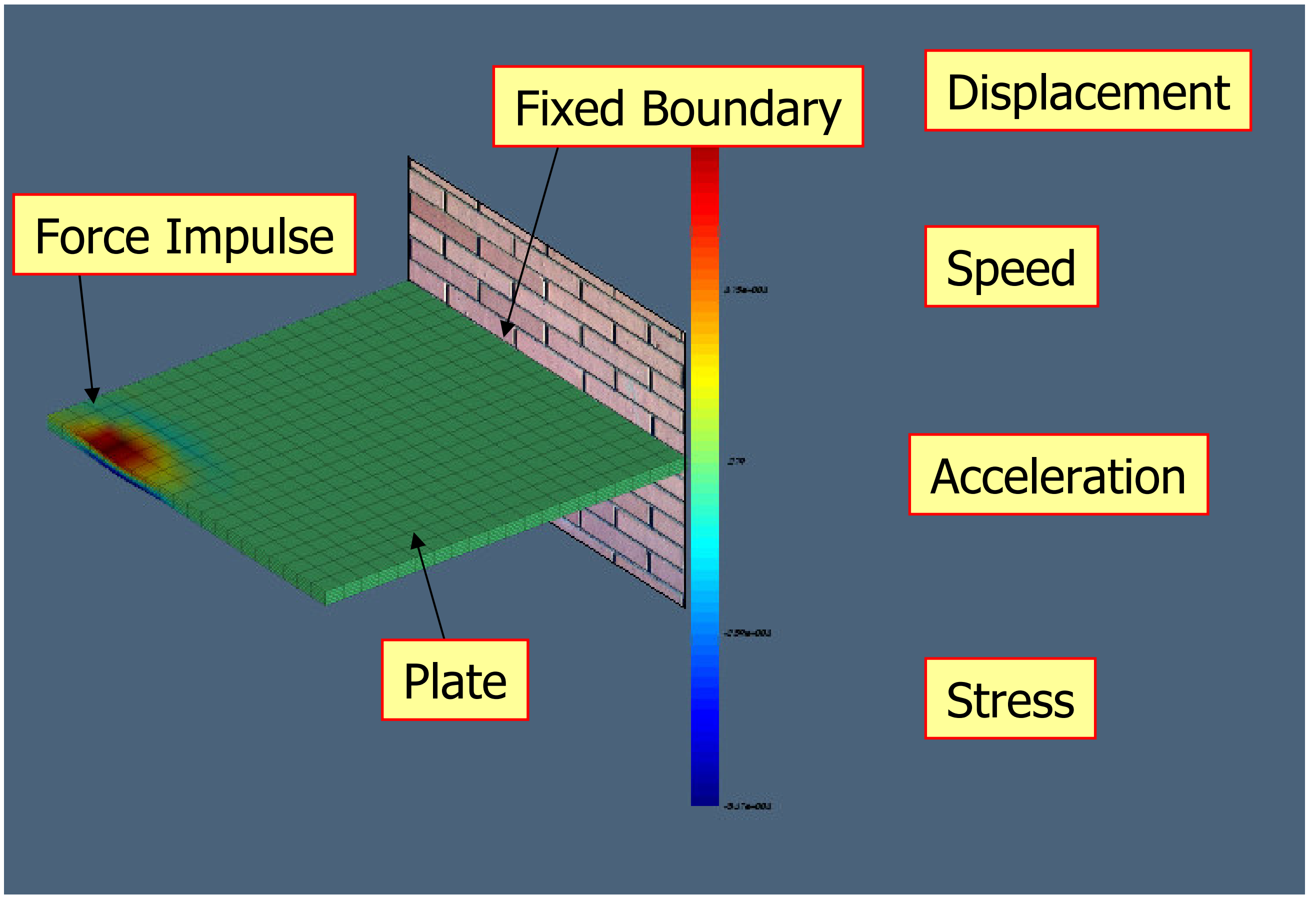




- Electromagnetic Analysis
- Heat Transfer and Geology Analysis
- Optical propagation
- Molecular Dynamics



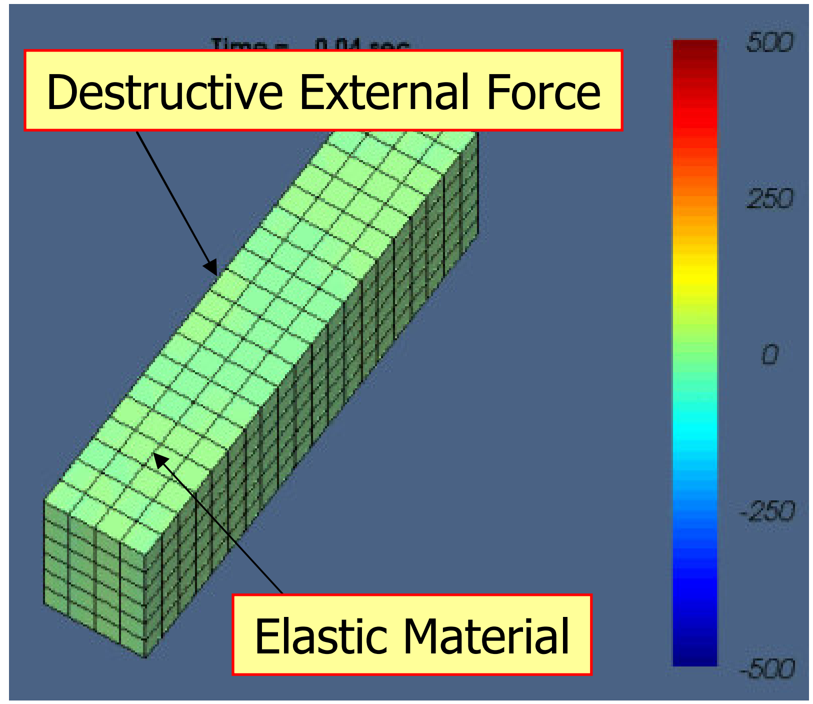
Thin-plate Vibration Analysis



(Data courtesy of Yuan-Sen Yang, National Center for Research on Earthquake Engineering)

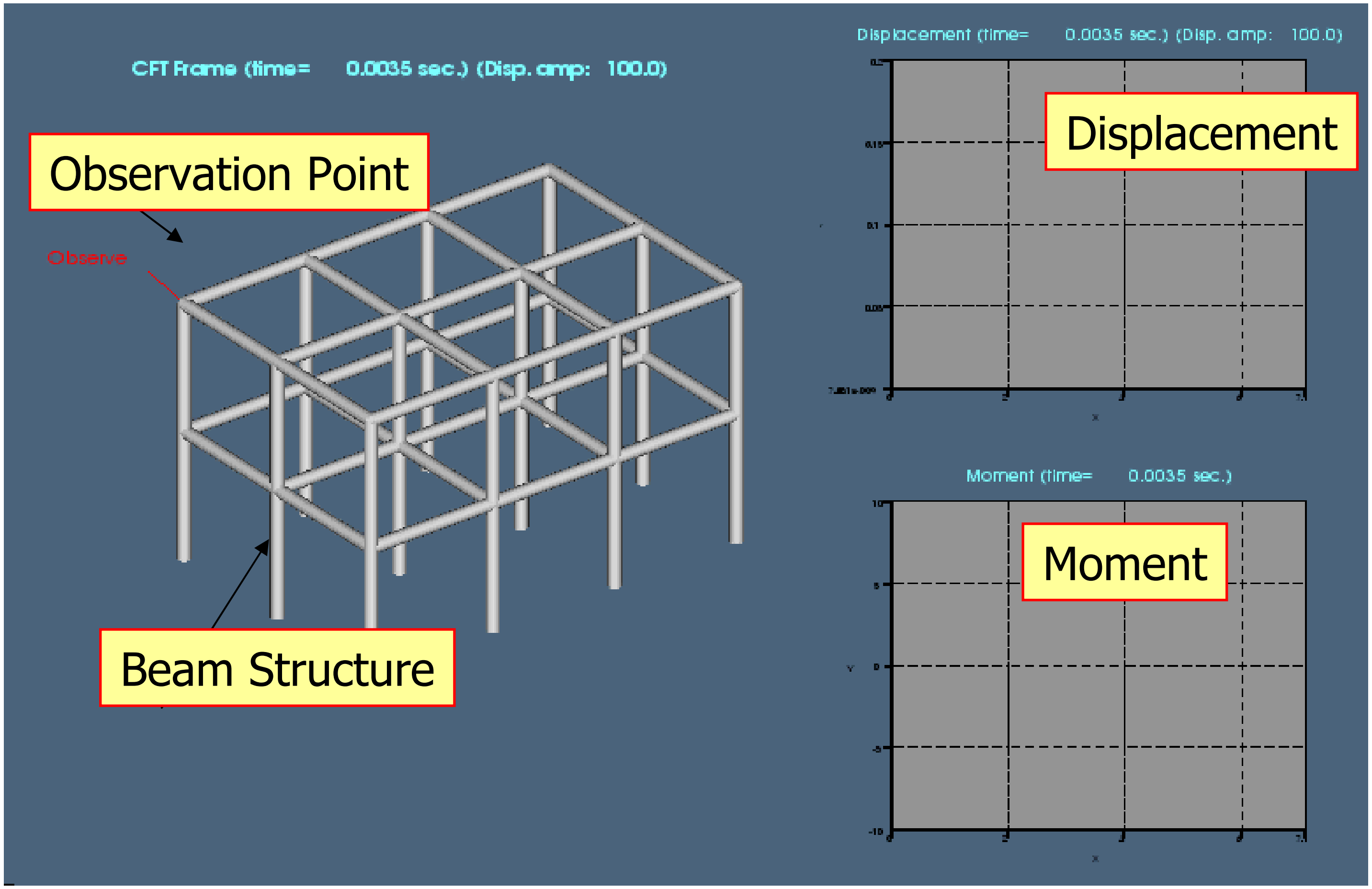
3D Co-rotational Explicit Finite Element Analysis

Simulation of the nonlinear, nonconsecutive destructive phenomenon of the dam structure based on the finite element method.



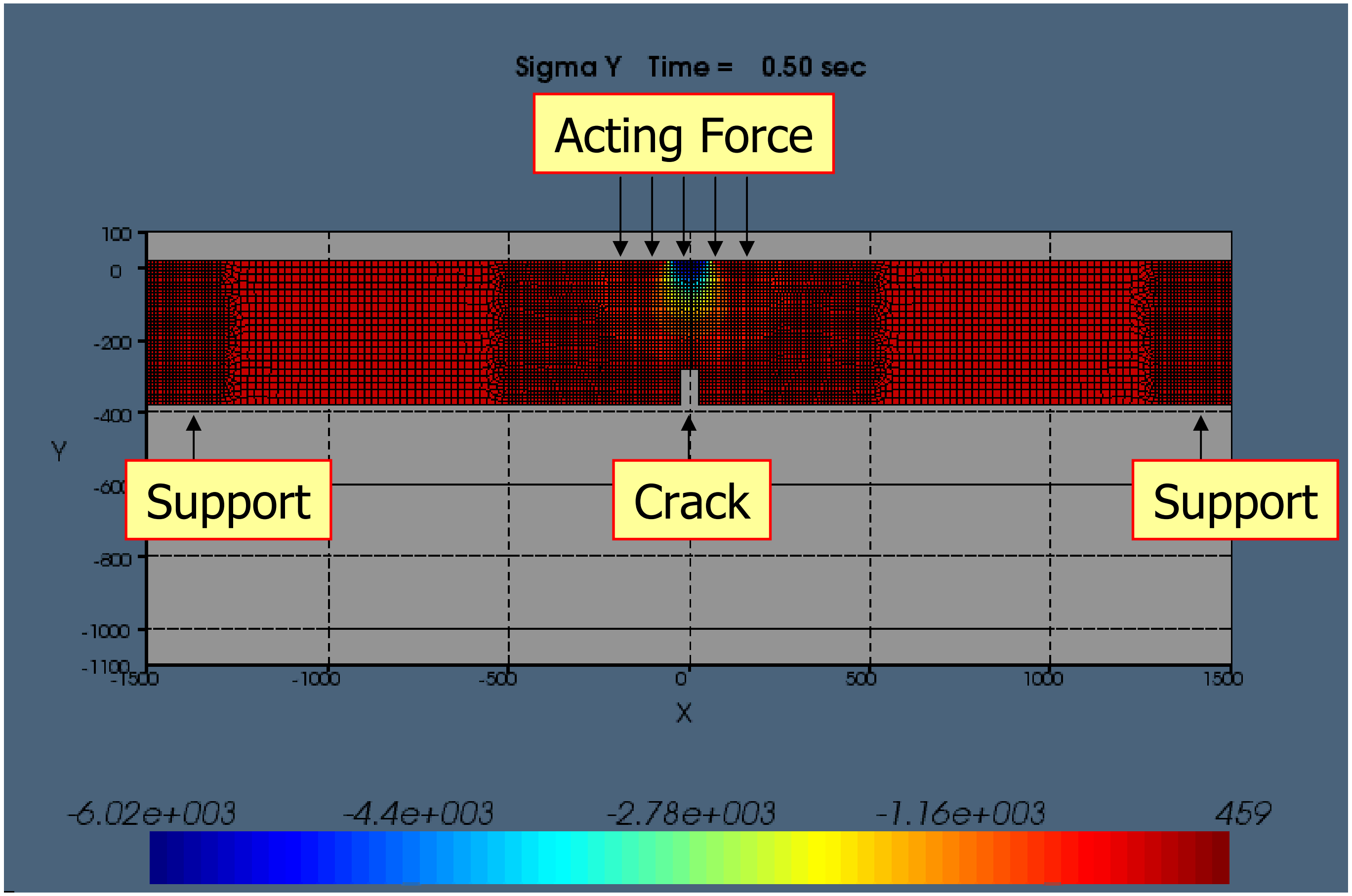
(Data courtesy of Professor Edward C. Ting/Chih-Cheng Lin, National Central University)

Structural Earthquake Response



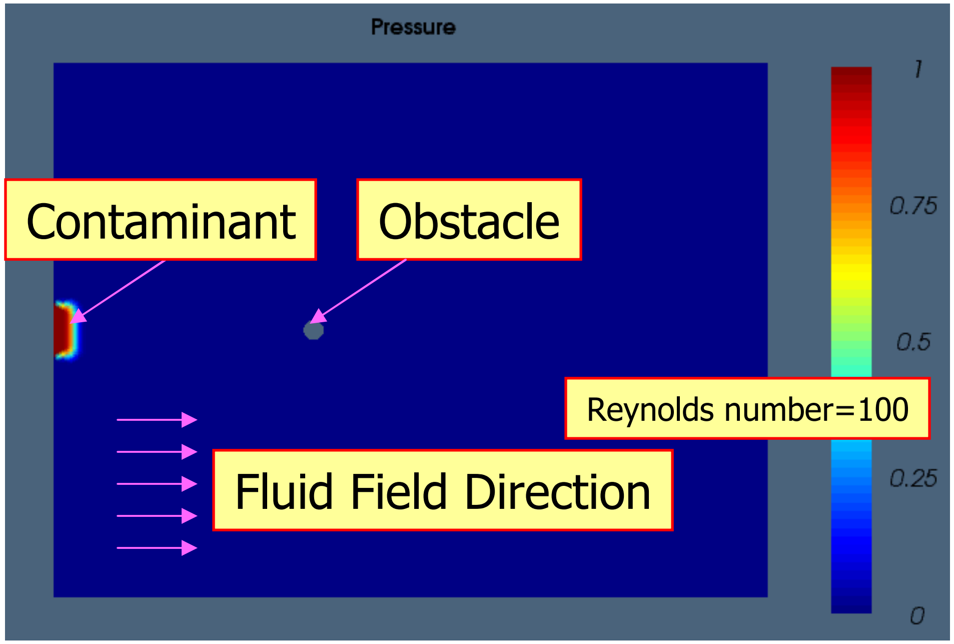
(Data courtesy of Professor Yuan-Sen Yang, National Center for Research on Earthquake Engineering)

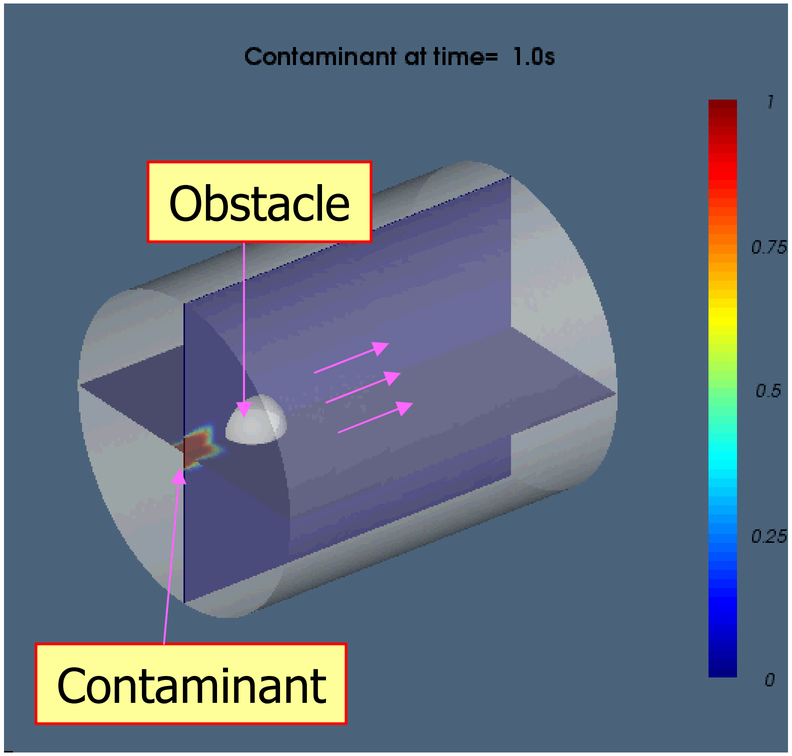
Concrete Fracture 2D Analysis



(Data courtesy of Professor Edward C. Ting/Yeh-Chan Lin/Chih-Cheng Lin, National Central University)

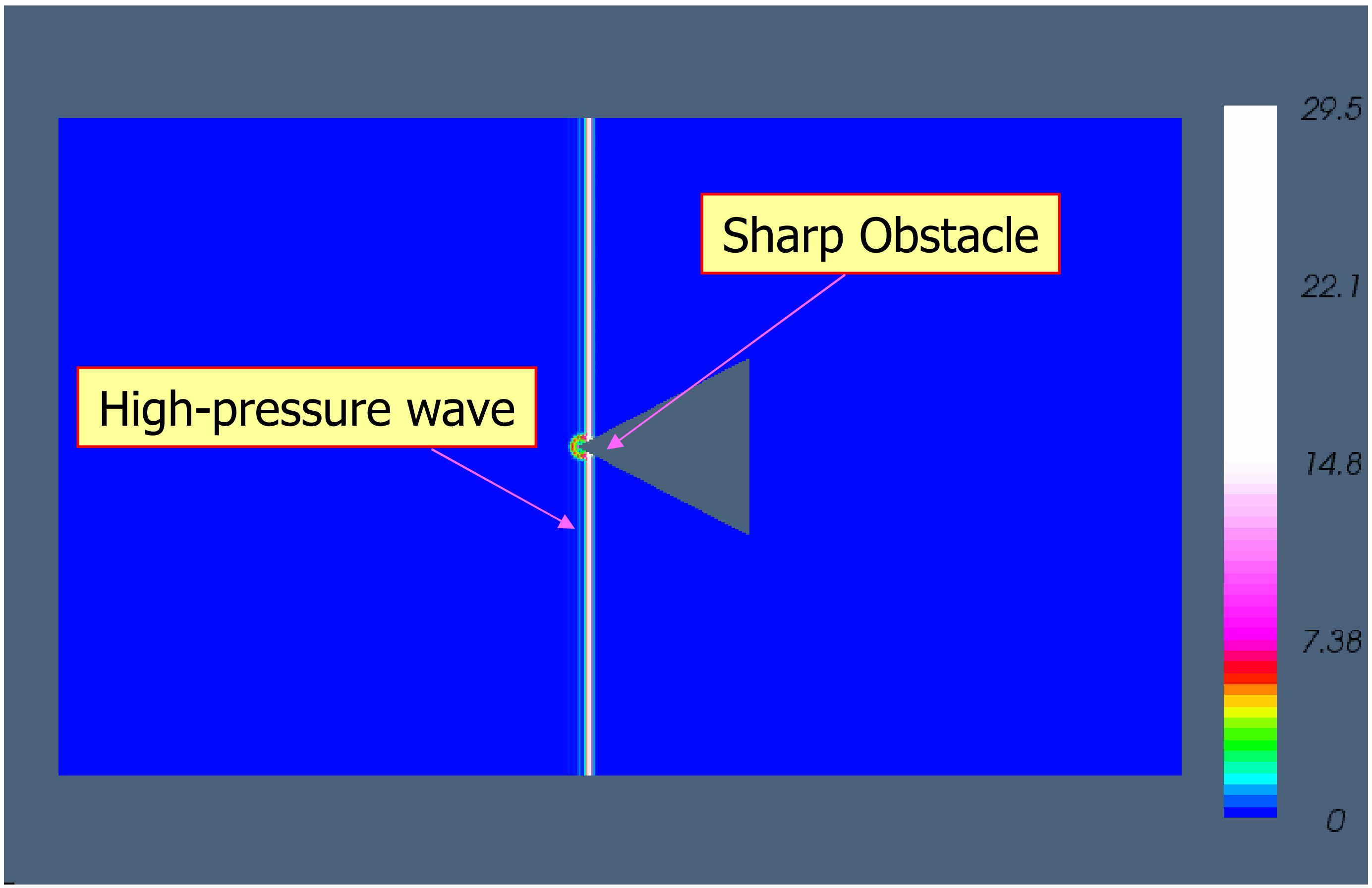
Fluid Field Turbulence I





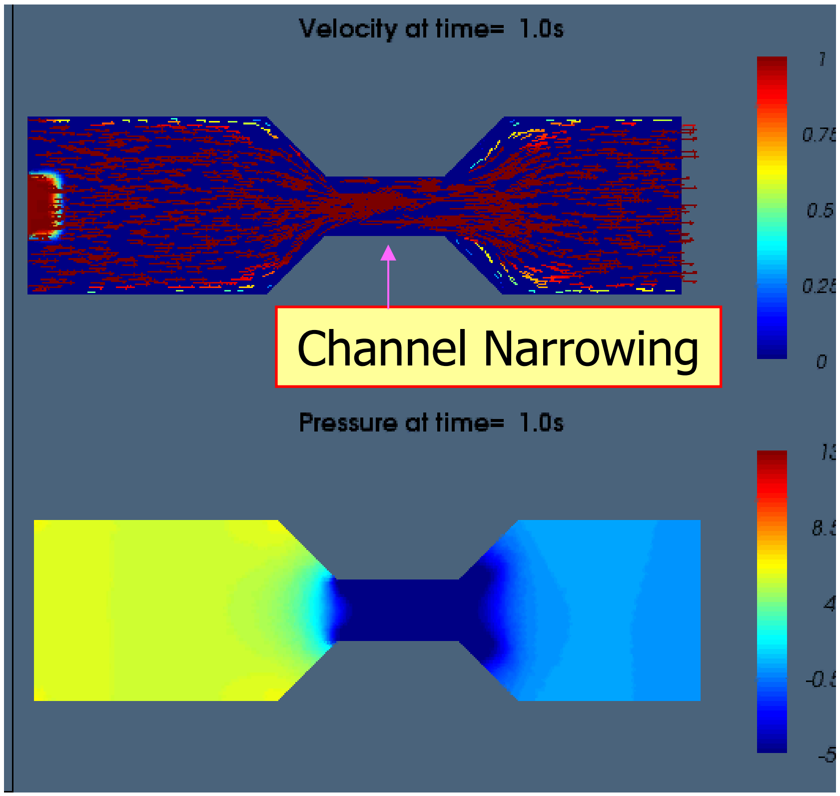
(Data courtesy of Ming-Hsin Su, AnCAD, Inc.)

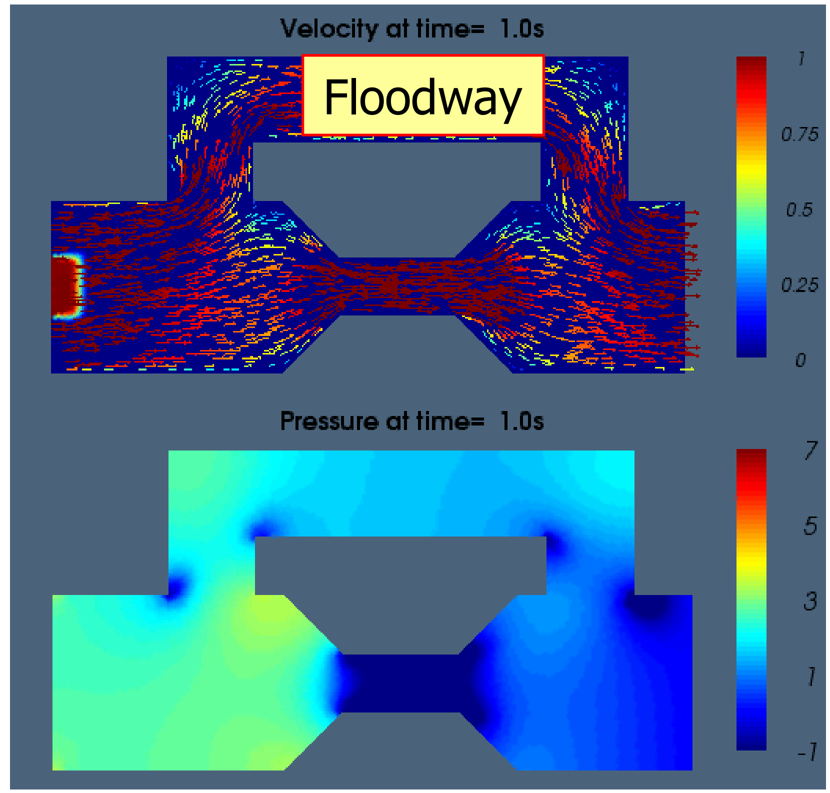
High Pressure Reflection



(Data courtesy of Dr. Lin and Dr. Huang, Institute of Science and Technology)

Fluid Field Turbulence II

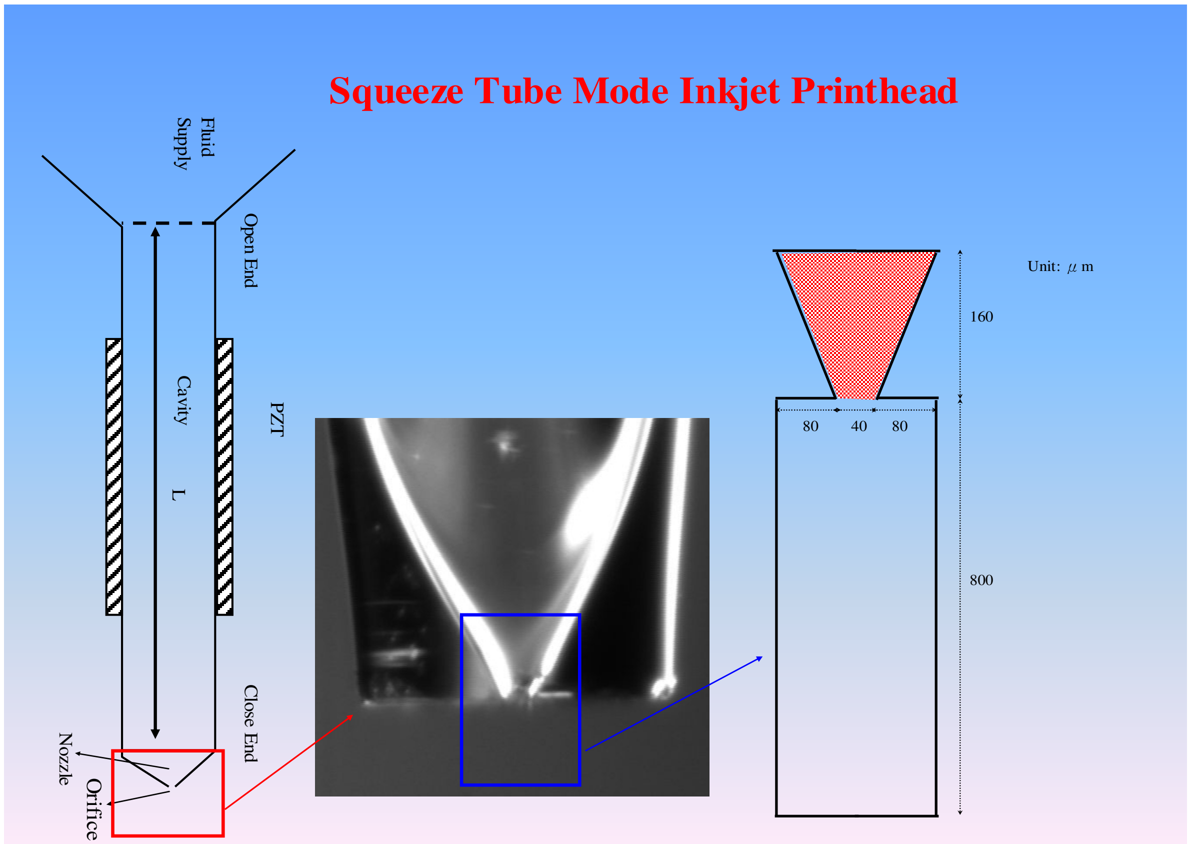




(Data courtesy of Ming-Hsin Su, AnCAD, Inc.)

3D Inkjet System Simulation I

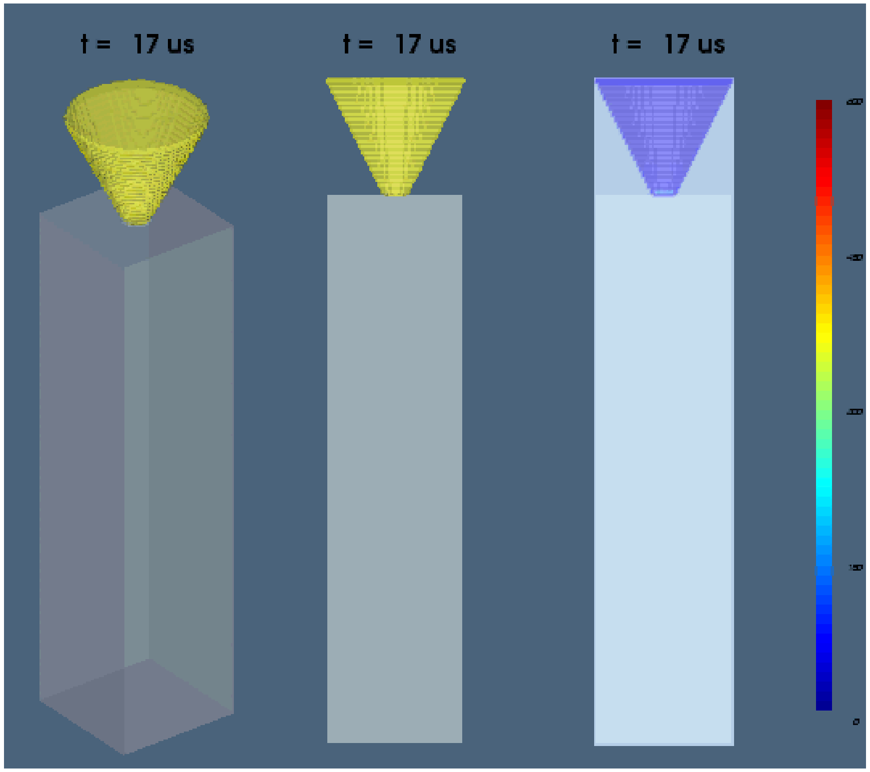
Blue Print

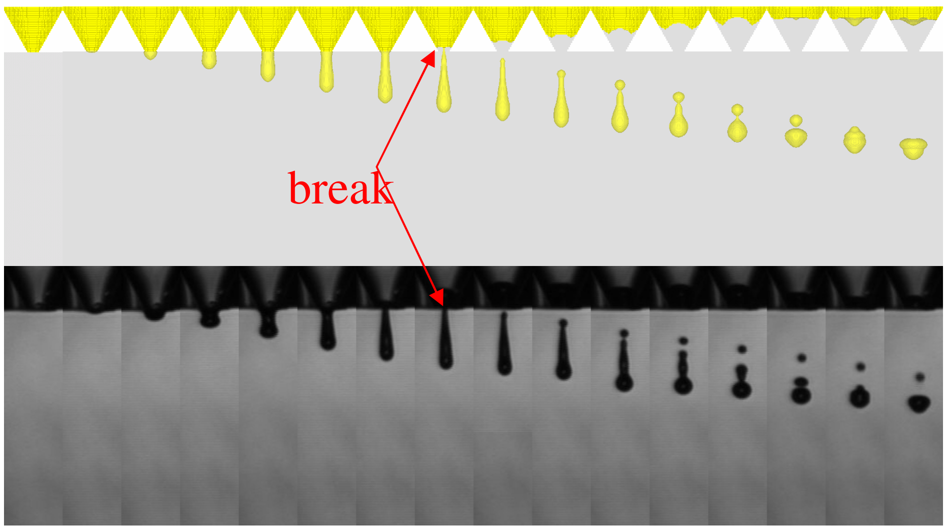


(Data courtesy of Weng-Sing Huang/Hsuan-Chung Wu, National Cheng Kung University)

3D Inkjet System Simulation II

Computer Simulation

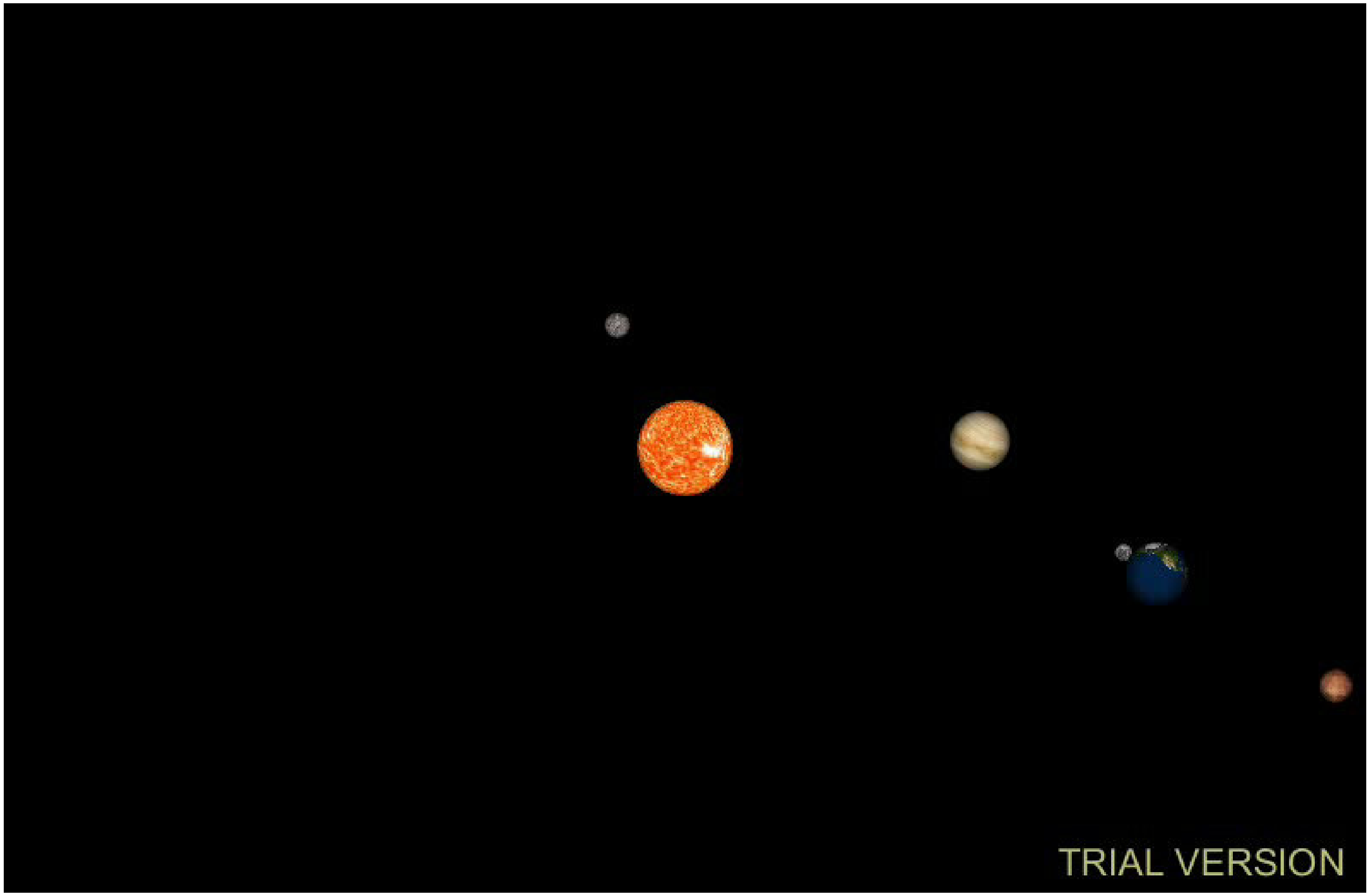




Time: 0 20 26 32 38 44 50 56 62 68 74 80 [μ s]

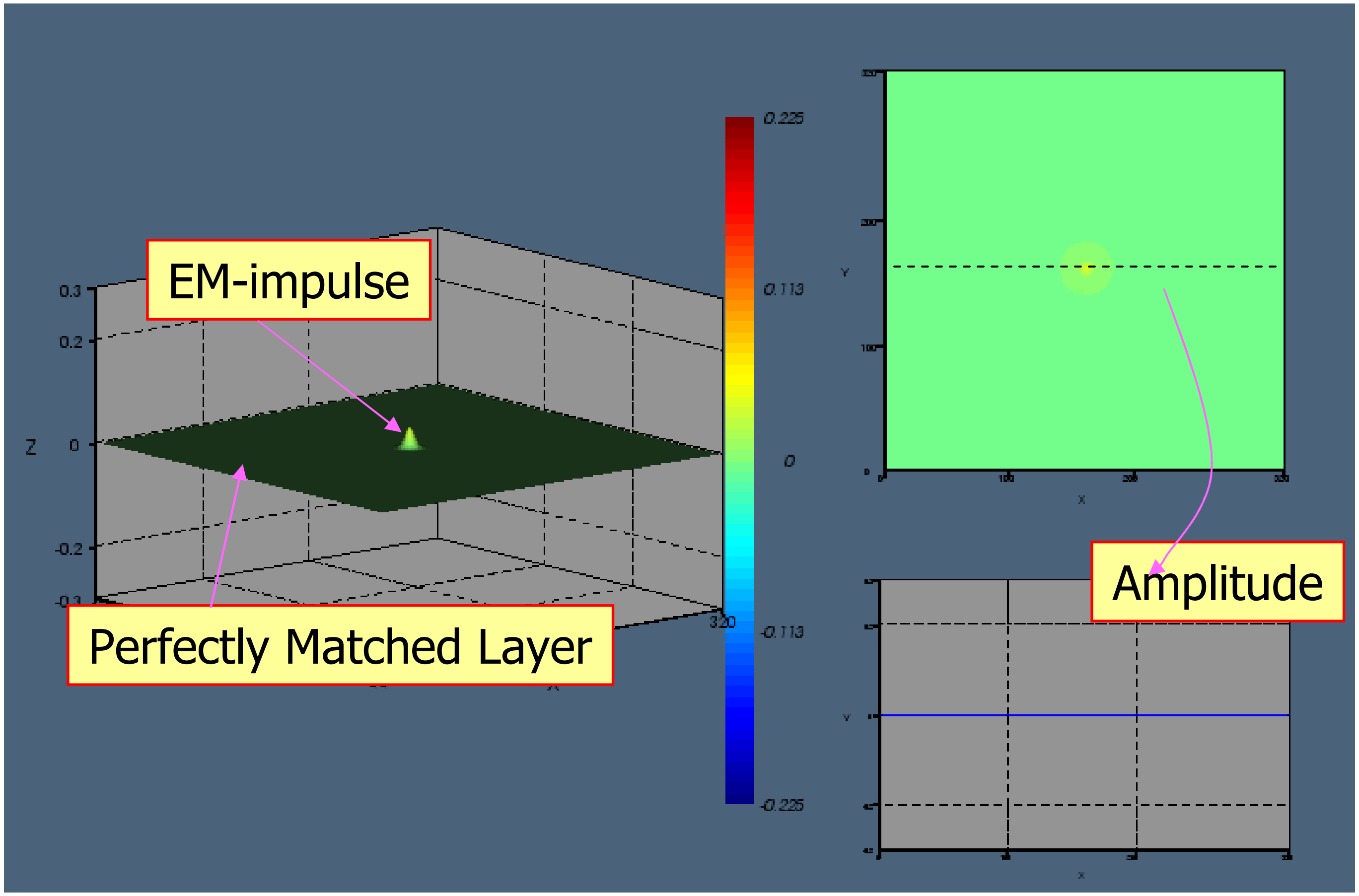
(Data courtesy of Weng-Sing Huang/Hsuan-Chung Wu, National Cheng Kung University)

3D Solar System Model



(Data courtesy of Li-Ger Chen, AnCAD, Inc.)

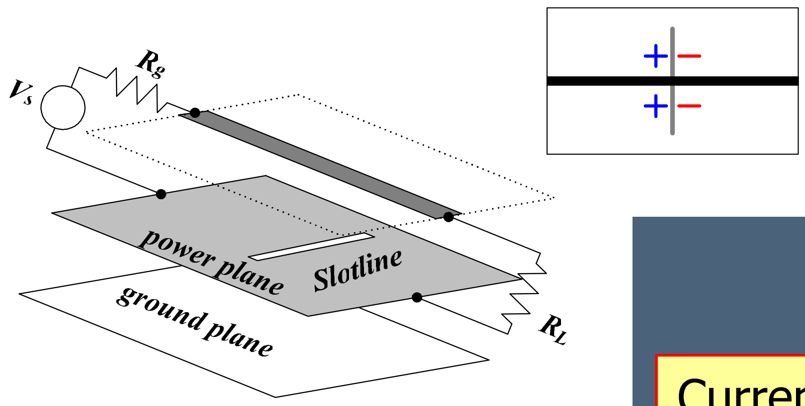
EM-wave Scattering on Perfectly Matched Layer

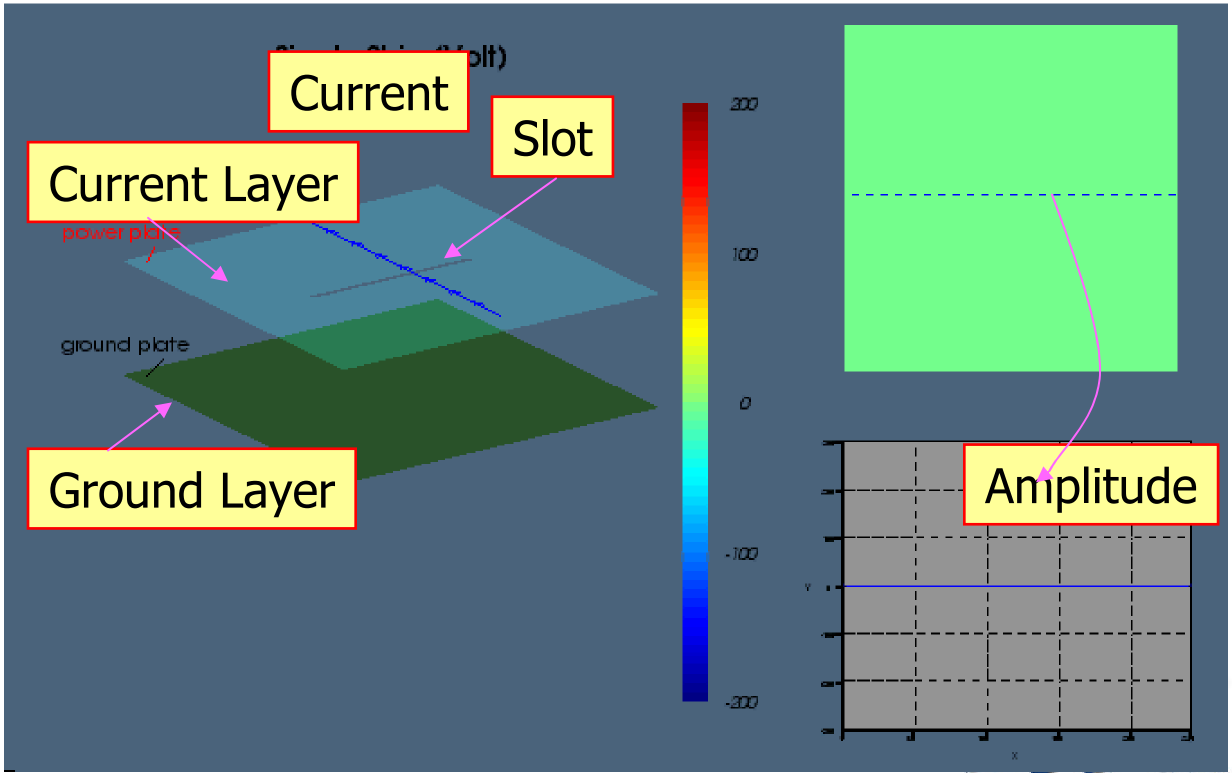


(Data courtesy of Chun-Hao Teng, AnCAD, Inc.)

Multi-layer Ground Noise in a Current Field I

Single microstrip line crossing slot

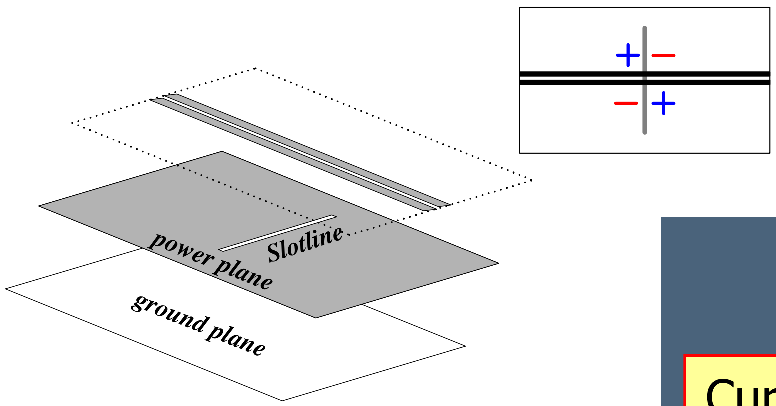


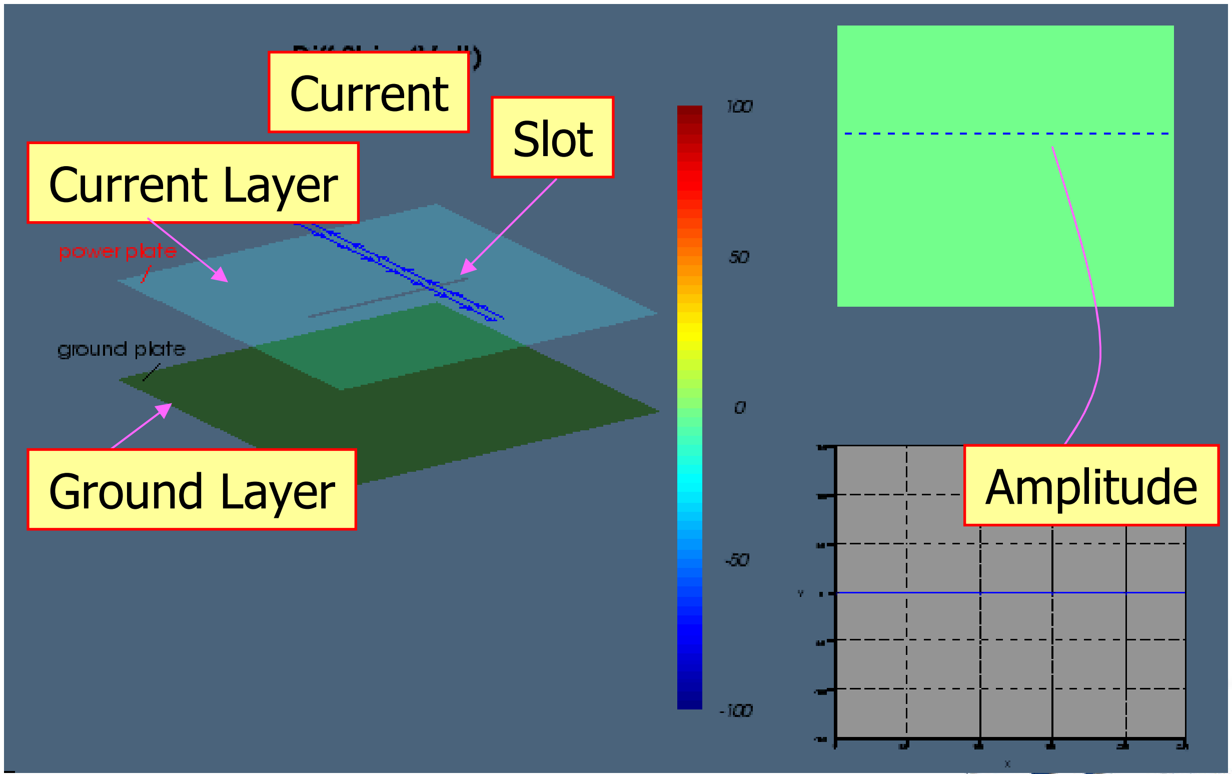


(Data courtesy of Ruey-Beei Wu/Kuang-Hua Hsueh, National Taiwan University)

Multi-layer Ground Noise in a Current Field II

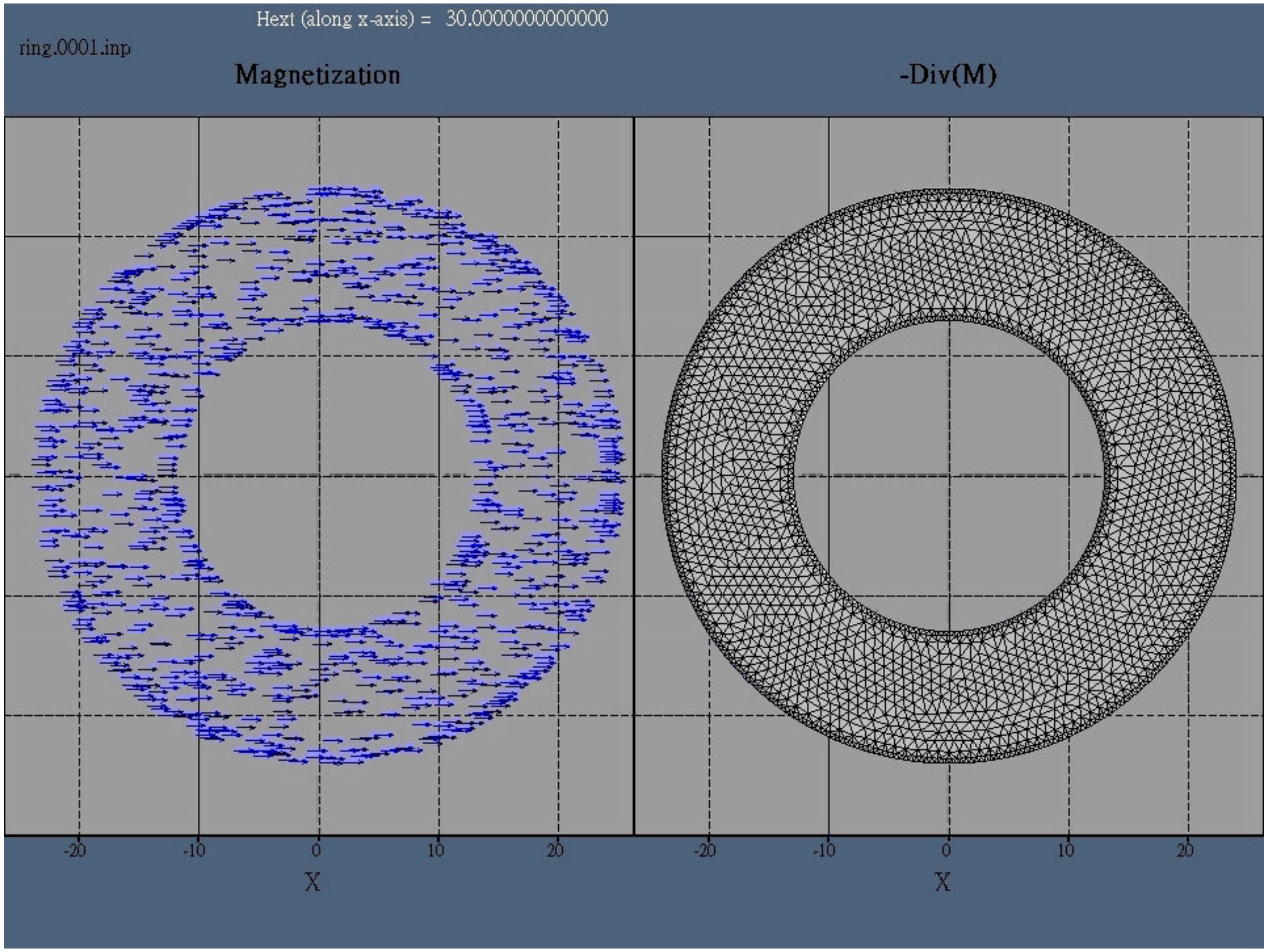
Differential microstrip line crossing slot (coupling factor = 0.305)





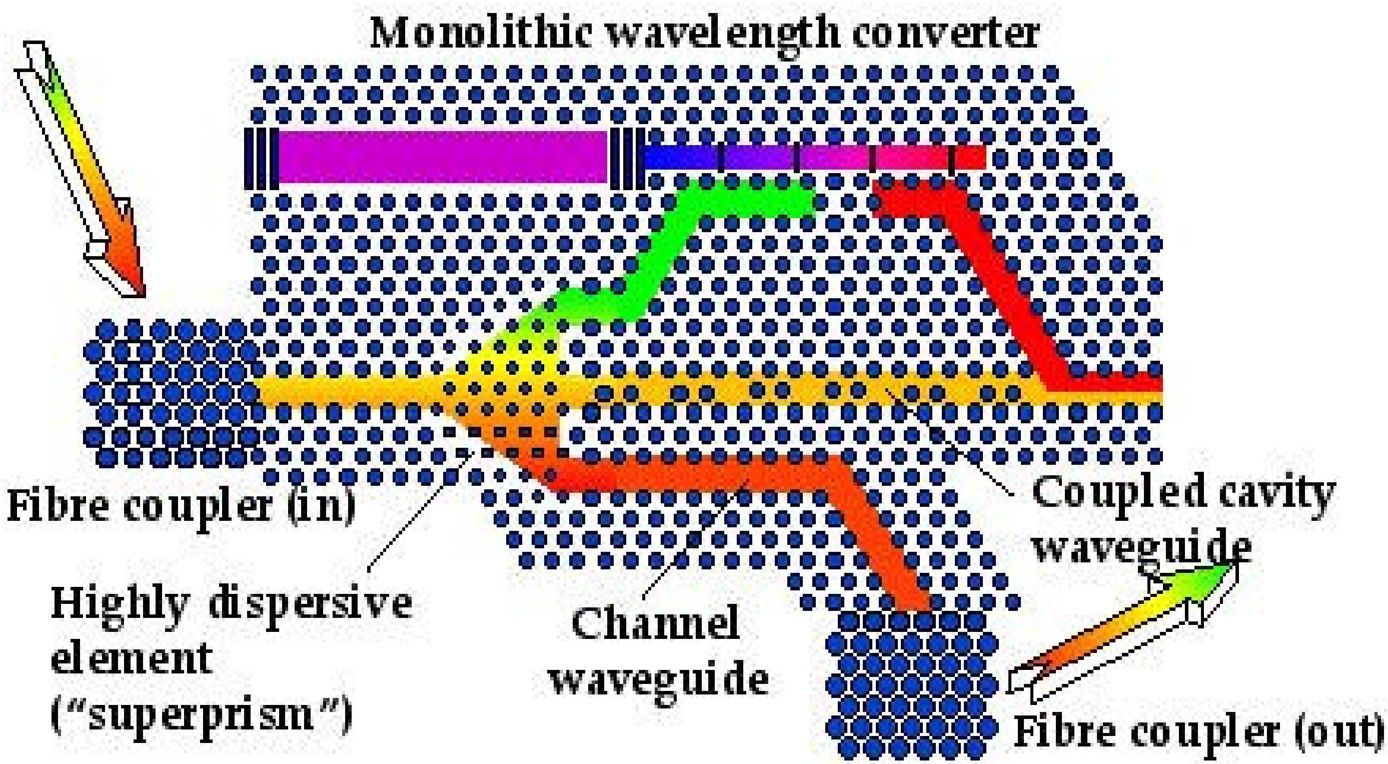
(Data courtesy of Ruey-Beei Wu/Kuang-Hua Hsueh, National Taiwan University)

Micro-magnetic Simulations



(Data courtesy of Professor Lee, Ching-Ming, Chungchou Institute of Technology.)

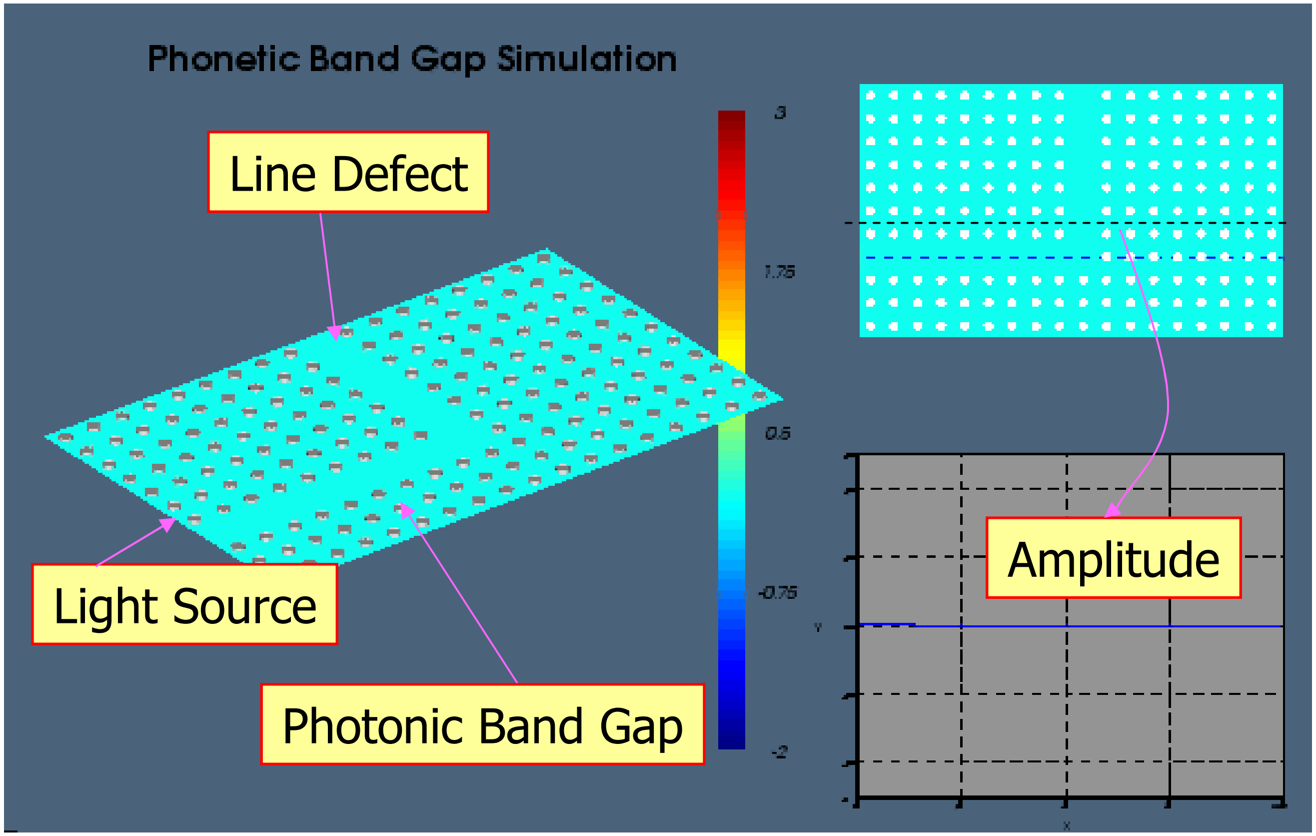
Photonic Band Gap Waveguide Transmission I



The blueprint of future integrated optical circuits.

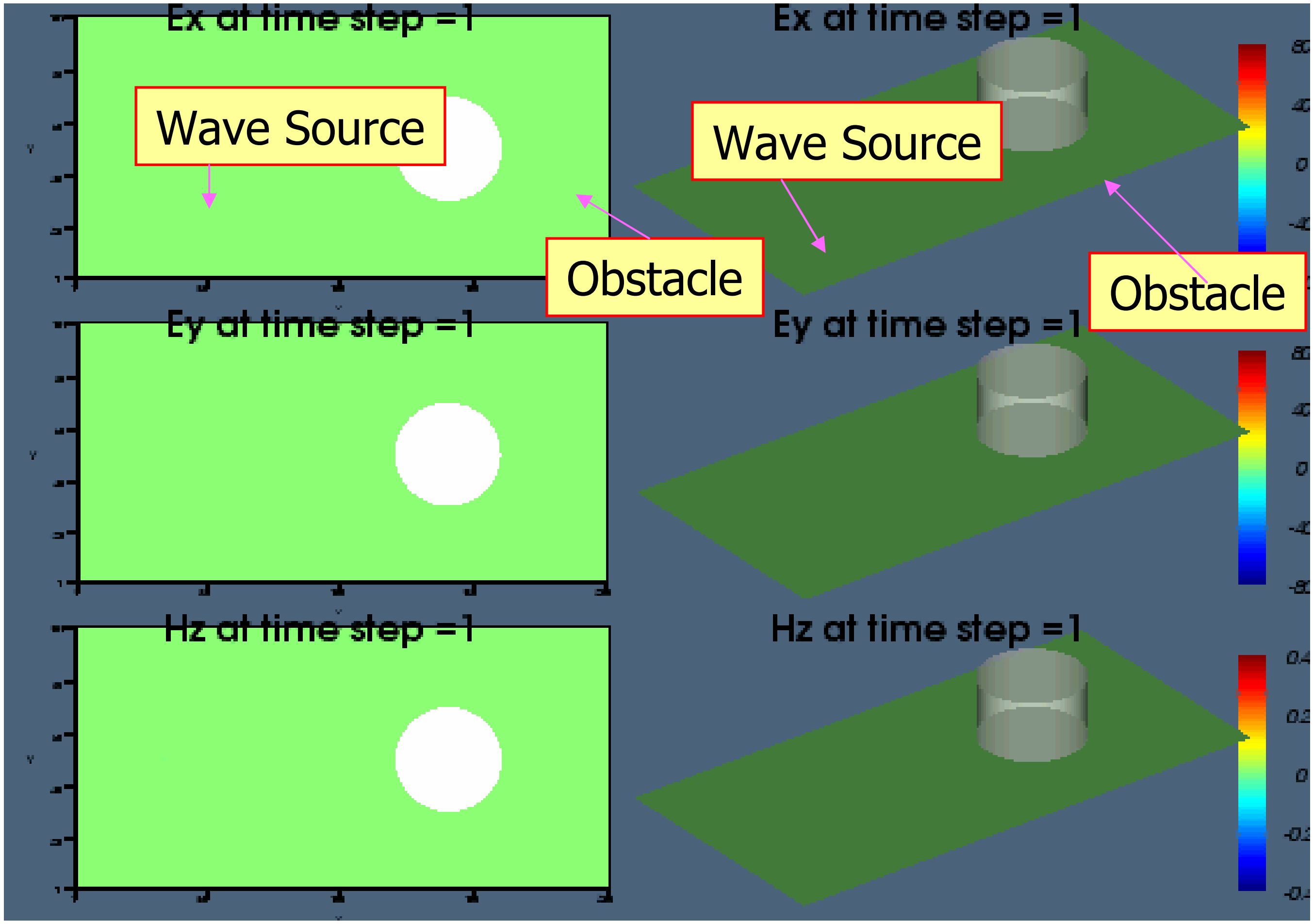
(Data courtesy of Department of Physics & Astronomy, University of St Andrews, UK)

Photonic Band Gap Waveguide Transmission II



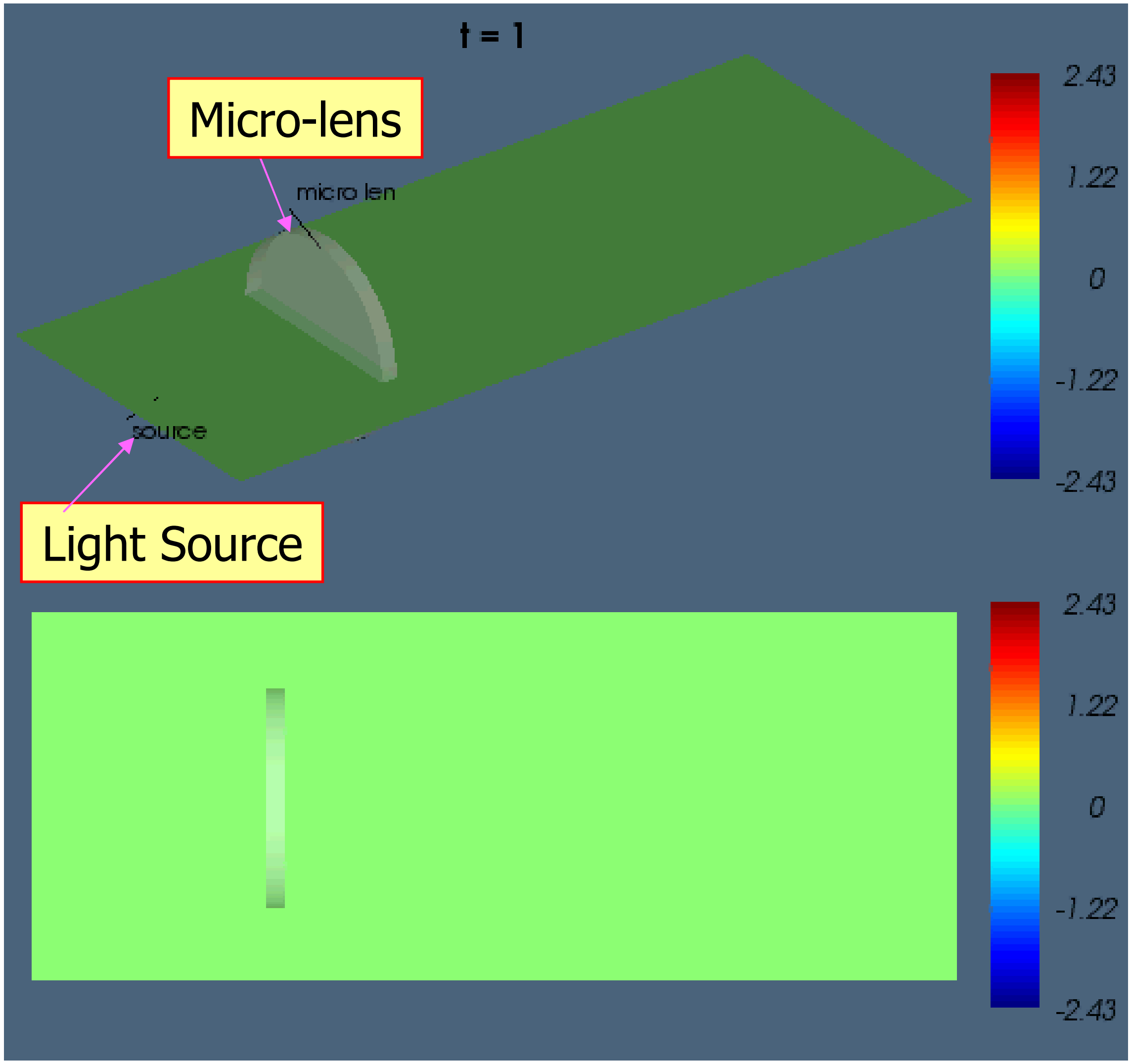
(Data courtesy of Dr. Pei-Kun Wei, Academia Sinica, Taiwan)

Finite-difference Time-domain Analysis



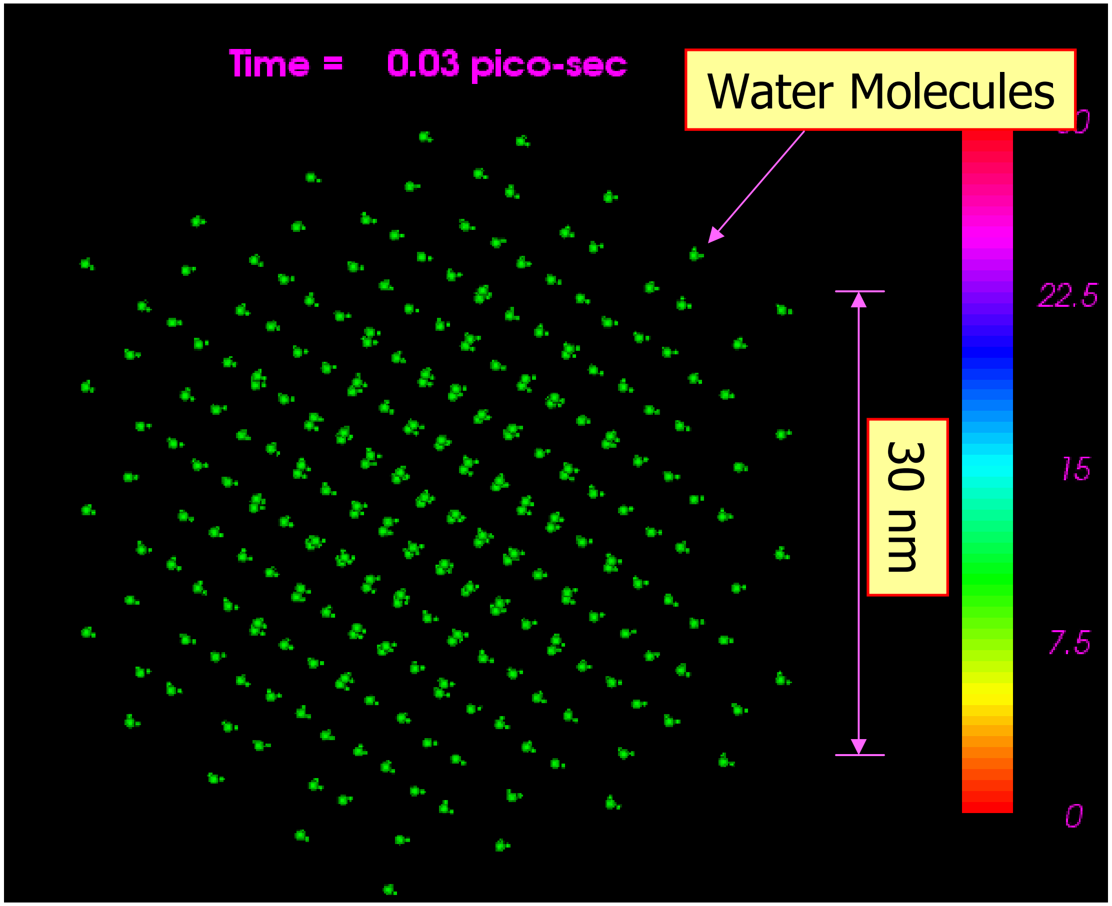
(Data courtesy of Yu-Jen Lin, Precision Instrument Development Center)

Analysis of Optical Thin-film Coated Micro-lens



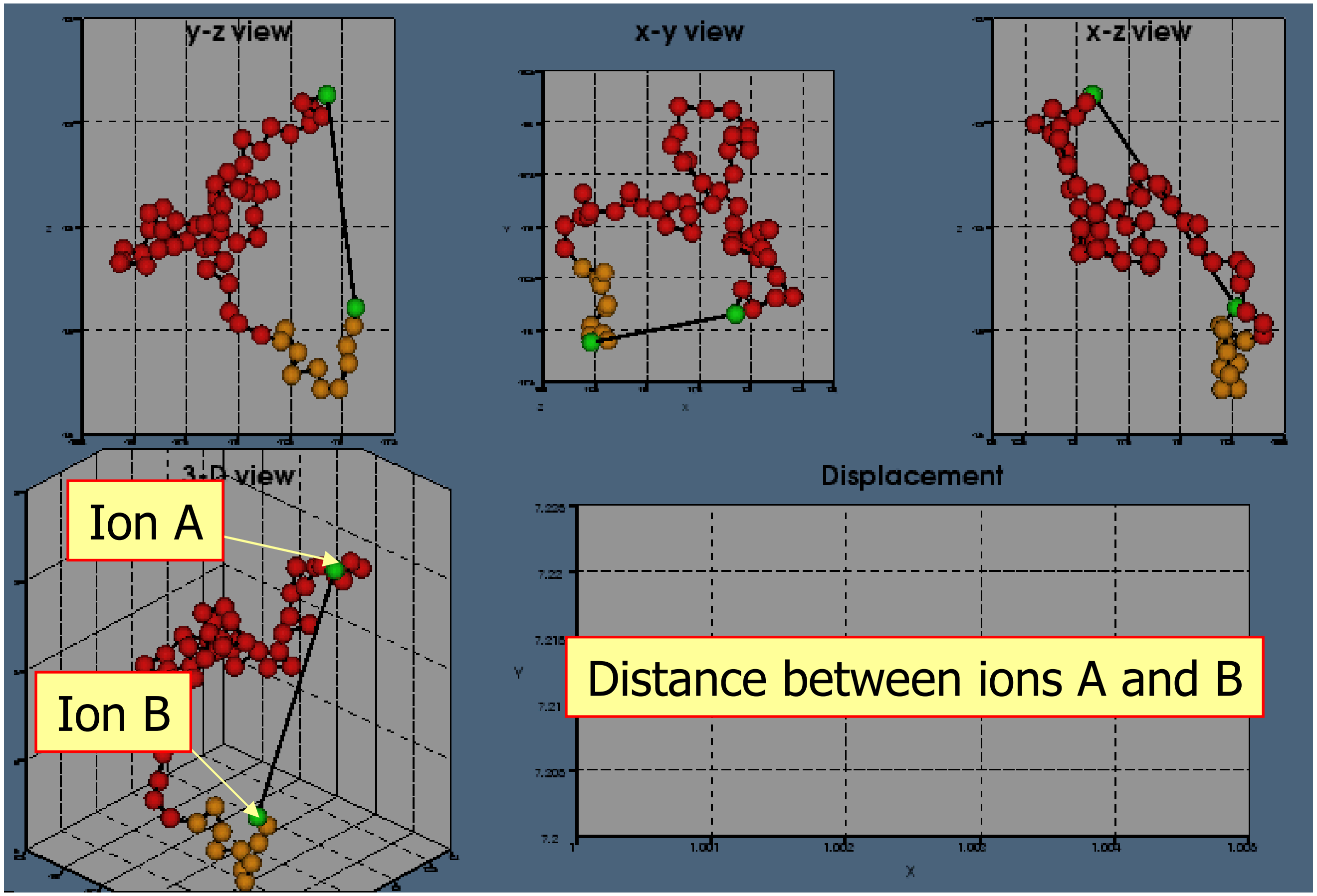
(Data courtesy of Tung-Lung Fu, Precision Instrument Development Center)

Water Molecule Phase Transition



(Data courtesy of Chin-Hsiang Cheng, Tatung University)

Ion Perturbation Analysis



(Data courtesy of Yu-Chang Sheng, National Taiwan University)

Visualization of Physical Problem by **MATFOR®**

Electromagnetic Radiation

- Dipole

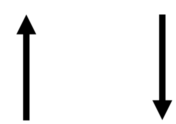
$$\left\langle \vec{S} \cdot \vec{n} \right\rangle = \frac{1}{2} \frac{p_0^2}{4\pi} \frac{\omega^4}{c^3} \sin^2(\theta) \frac{1}{r^2}$$

- Quadrupole

axial separation

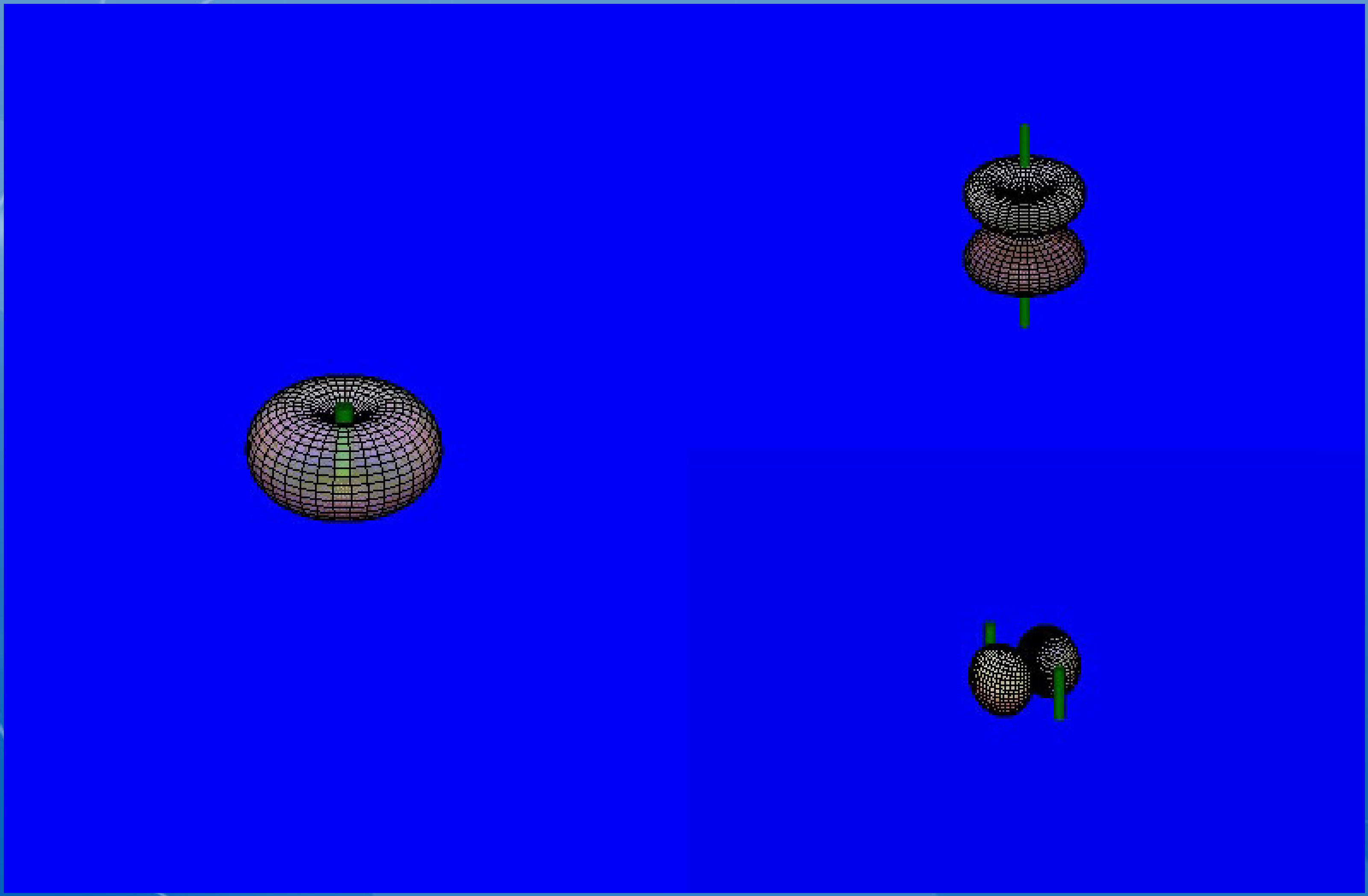
$$\left\langle \vec{S} \cdot \vec{n} \right\rangle = \frac{1}{2} \frac{Q^2}{32\pi^2} \frac{\omega^6}{c^5} \sin^2(\theta) \sin^2\left[kd\cos(\theta)\right] \frac{1}{r^2}$$

laterial separation



$$\left\langle \vec{S} \cdot \vec{n} \right\rangle = \frac{1}{2} \frac{p_0^2}{\pi} \frac{\omega^4}{c^3} \sin^2(\theta) \sin^2\left[\frac{kd}{2} \sin(\theta) \cos(\varphi)\right] \frac{1}{r^2}$$

Visualizing the Radiation Pattern



Diffraction Integral

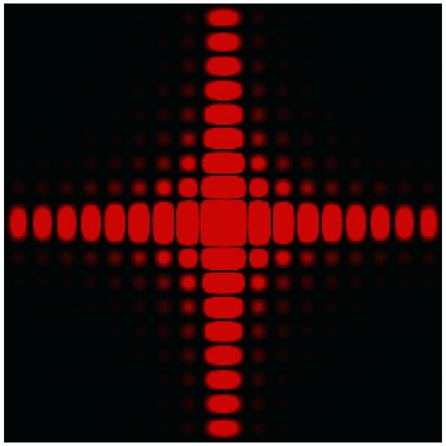
- Fresnel-Kirchhoff :

$$\psi(P) = \frac{1}{i}\frac{\psi_0}{2\lambda}\int_\sigma \frac{e^{ikr}}{r}(1+\cos(\theta))da$$

- Frauhofer :

$$\psi(\alpha,\beta) = \frac{1}{i}\frac{\psi_0 e^{ikR}}{\lambda Z}\int_\sigma e^{-ik(\alpha\xi+\beta\eta)}d\xi d\eta$$

($R >> \lambda >> D \quad \therefore r \approx R(\alpha\xi+\beta\eta)$)



Slit Diffraction

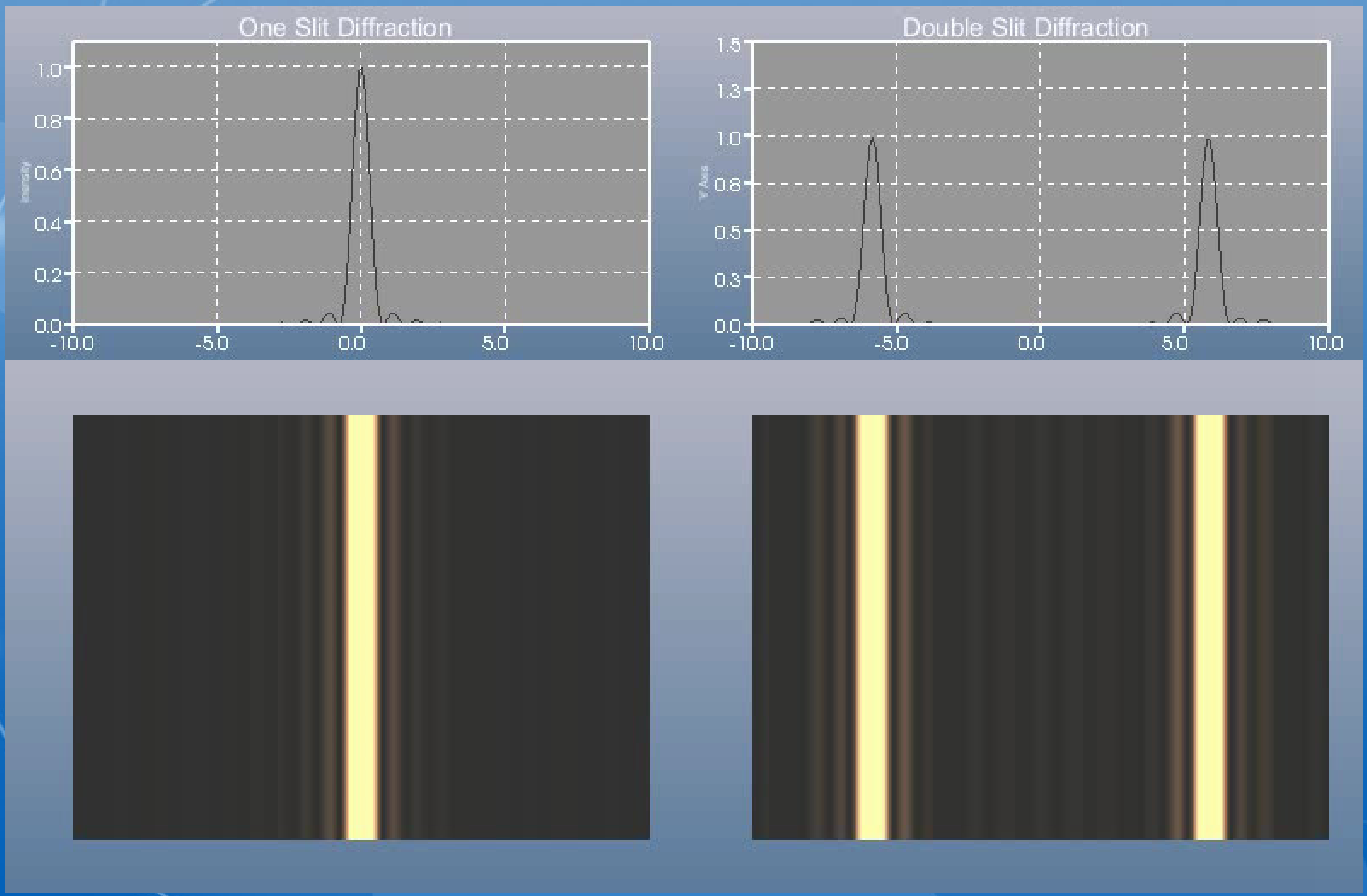
- In one dimensional ,The Fraunhofer Diffraction Integral becomes :

$$\psi(\alpha) = C\int_{\sigma} e^{-ik\alpha\xi} d\xi$$

- If the slit is of width 2a, then the integral is :

$$\psi(\theta) = C\int_{-a}^{a} e^{-ik\alpha\xi} d\xi = \underline{\underline{2Ca(\frac{\sin(u)}{u})}} \quad , u = ka\alpha = \frac{2\pi}{\lambda} a\sin(\theta)$$

Visualization of the Slit Diffraction



Circular Aperture

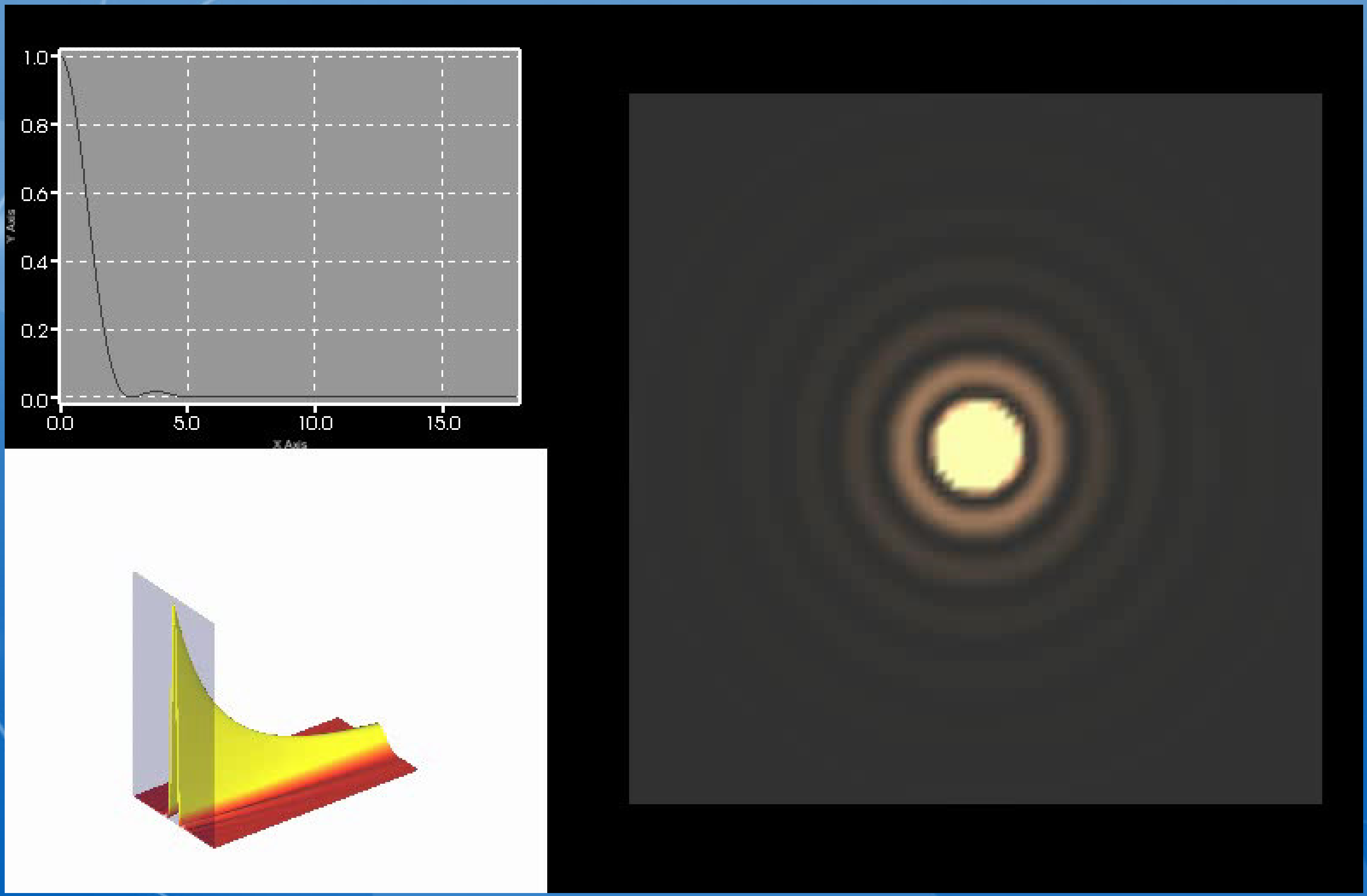
$$\psi(P) = C\int_{\sigma} e^{-ik(\alpha\xi+\beta\eta)}d\xi d\eta \quad \xi = \rho\cos(\varphi), \eta = \rho\sin(\varphi)$$

$$\alpha\xi + \beta\eta = \rho\theta\cos(\varphi - \varphi')$$

$$\therefore \psi(\theta) = C\int_0^a\int_0^{2\pi} e^{-ik\rho\theta\cos\varphi}\rho\cdot d\theta d\rho = 2\pi C\int_0^a J_0(ka\theta)\rho d\rho = 2\pi Ca^2\left(\frac{J_0(u)}{u}\right)$$

$$u = \frac{2\pi}{\lambda}a\theta, \quad \left|\pi a^2 C\right|^2 = \left(\frac{\pi a^2}{Z\lambda}\right)\left|\psi_0\right|^2$$

Visualizing the Circular-Aperture Diffraction



Optical Soliton : Kerr Effect

This refractive index variation is responsible for the nonlinear optical effects of self-focusing and self-phase modulation, and is the basis for Kerr-lens modelocking.

Type Kerr effect:

$$n = n_0 + n_2 I$$

Slow Vary Approximation

$$\nabla^2 E + n^2 k_0^2 E = 0, \quad \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}$$

$$\left[\frac{\partial^2 a}{\underline{\underline{\partial z^2}}} + i(2n_0 k_0) \cdot \frac{\partial a}{\partial z} + (-1)(n_0 k_0)^2 + \frac{\partial^2 a}{\partial x^2} \right] + (nk_0)^2 a = 0$$

$$for \quad k \sim \frac{1}{\lambda} \quad , \Delta z >> \lambda \quad \rightarrow \left| k_0 \frac{\partial a}{\partial z} \right| >> \left| \frac{\partial^2 a}{\partial z^2} \right|$$

$$and \quad n^2 - n_0^2 = (n - n_0)(n + n_0) \approx (2n) n_0 I$$

$$eq. \Rightarrow \frac{\partial a}{\partial z} = i \frac{1}{2n_0 k_0} \frac{\partial^2 a}{\partial x^2} + i\left(\frac{n_0 k_0 n^2}{2} |a|^2\right) \cdot a$$

$$set \quad \frac{1}{2n_0 k_0} \equiv \alpha, \quad \frac{n_0 k_0 n^2}{2} \equiv \beta$$

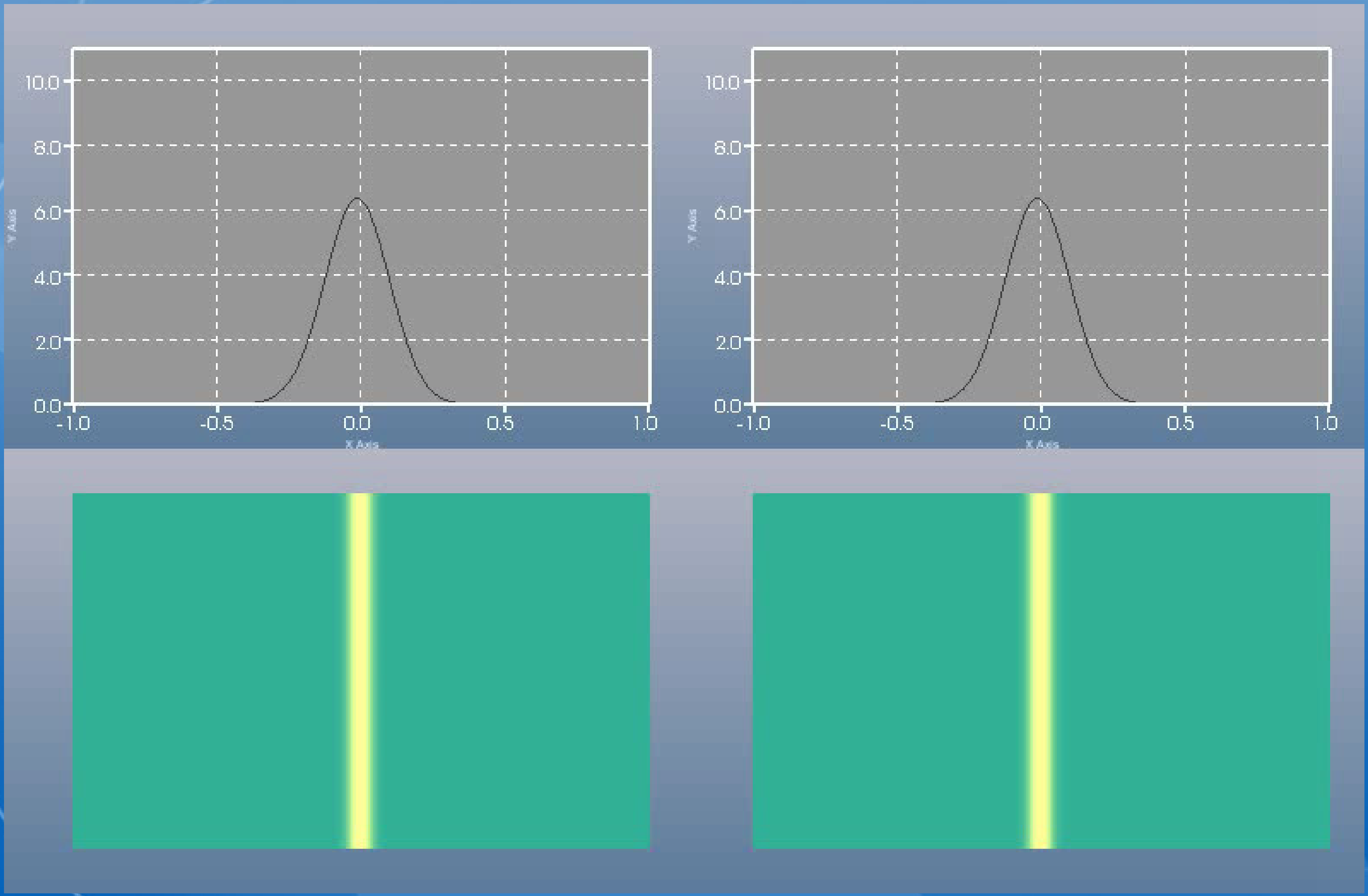
Split Step Fourier Method

$$\frac{\partial a}{\partial z} = i\alpha \frac{\partial^2 a}{\partial x^2} + i(\beta |a|^2) \cdot a$$

$$\Rightarrow \frac{\partial a}{\partial z} = \left(\hat{L} + \hat{N}\right) a \quad \Rightarrow a = e^{(\hat{L}+\hat{N})z} a_0 \approx e^{\hat{L}\cdot z} e^{\hat{N}\cdot z} a_0$$

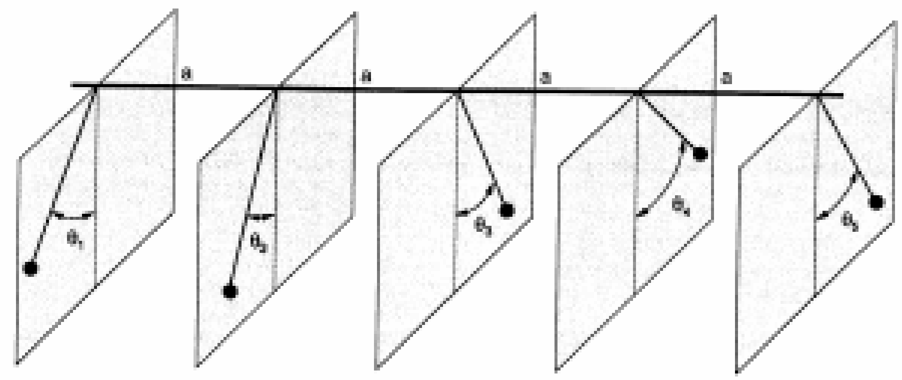
$$\Rightarrow \underline{\underline{a_{n+1} = e^{i\alpha(-k^2)\cdot \Delta z} F[e^{i\beta |a|^2} a_n]}}$$

SSF simulation result for Kerr soliton.



Sin-Gordon Equation

- Considing a 1-D chan of identiacl pendu;la connenected by a torsion bar.



- The euqation of motion for pendulum j follows from Newton's law for rotational motion in terms of torques:

$$\sum_{j \neq i} \tau_{ji} = I\alpha_j$$

$$-\kappa(\theta_j - \theta_{j-1}) - \kappa(\theta_j - \theta_{j+1}) - mg\sin(\theta) = I\,\frac{d^2\theta_j(t)}{dt^2}$$

2D Sin-Gordon Equation

- If the wavelength in a pulse are much longer than the repeat disance a, the chain could be approximated as a continuous medium.

$$\theta_{j+1} \approx \theta_j + \frac{\partial \theta}{\partial x} \cdot a$$

$$-(\theta_j - \theta_{j-1}) - (\theta_j - \theta_{j+1}) \approx \frac{\partial^2 \theta}{\partial x^2} \cdot (a)^2$$

$$\Rightarrow \lim_{a \to 0} \frac{\partial^2 \theta}{\partial t^2} - \frac{\kappa a}{I} \frac{\partial^2 \theta}{\partial x^2} = \frac{mg}{I} \sin(\theta)$$

- Then generalizing the spatial derivatives:

$$\nabla^2 \theta - \alpha \cdot \frac{\partial^2 \theta}{\partial x^2} = \beta \cdot \sin(\theta)$$

Sin-Gordon Equation: 1D solution

1-D analytic solution :

$$\frac{\partial^2 \theta}{\partial t^2} - \frac{\partial^2 \theta}{\partial x^2} = \sin(\theta) \quad , \theta(x,t)$$

$$\xi = t \pm \frac{x}{v}$$

$$\frac{d^2 \theta}{d \xi^2} = \frac{v^2}{v^2 - 1} \sin(\theta)$$

$$for \quad v \sim \pm 1$$

$$\theta(x - vt) = \begin{cases} 4 \tan^{-1}\left[\exp\left[\frac{x - vt}{(1 - v^2)^{\frac{1}{2}}}\right]\right] \\ 4 \tan^{-1}\left[\exp\left[-\frac{x - vt}{(1 - v^2)^{\frac{1}{2}}}\right]\right] + \pi \end{cases}$$

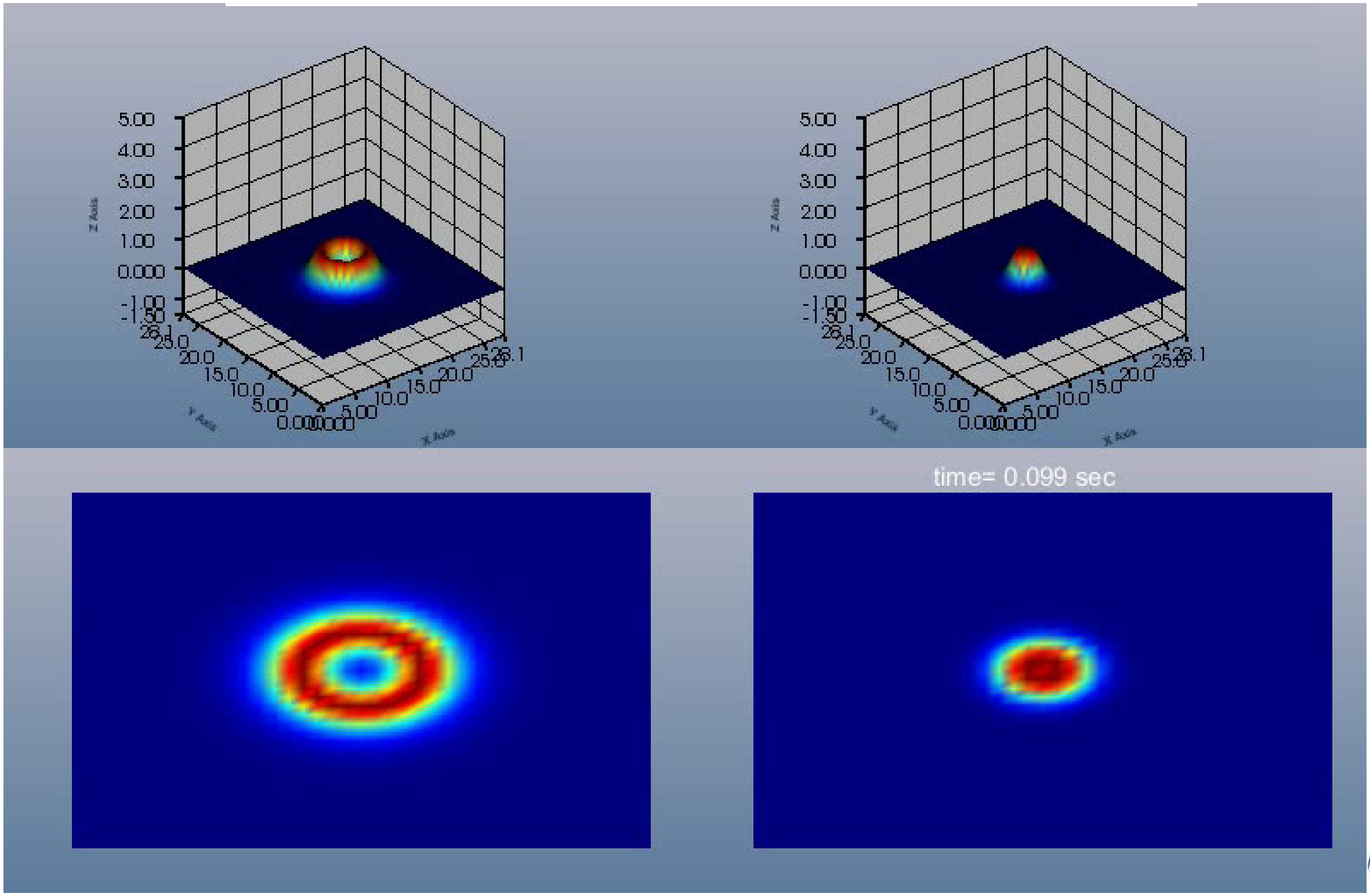
So the 2D initial gauss be :

$$\theta(x, y, t = 0) = 4 \tan^{-1}\left[\exp\left(c - \left(x^2 + y^2\right)^{\frac{1}{2}}\right)\right]$$

where c be some constant.

2D Dynamical Sin-Gordon Equation

Initial condition, left: above equation; right: Gaussian form.



Quantum bound

- In multiple-particle nature, the interaction leads the nonlocal potential :. It changes the interaction term in Schrodinger equation :

$$V(r)\psi(r) \rightarrow \int V(r,r')\psi(r')dr'$$

- Then the equation becomes :

$$H\psi = E\psi$$

$$\frac{p^2}{2\mu}\psi_n(k) + \frac{2}{\pi}\int_0^{\infty} V(p,k)\psi_n(k)p^2 dp + E_n\psi_n(k)$$

- One way to deal with the equation is by going to k-space(Bessel transformation):

$$\frac{p^2}{2\mu}\psi_n(k)+\frac{2}{\pi}\int_0^\infty V(p,k)\psi_n(k)\,p^2dp+E_n\psi_n(k)$$

the k-space potential is obtain by a double Bessel transform

$$(2\pi)^3\int_0^\infty\left(\int_0^\infty j_l(kr')V(r,r')r'dr\right)j_l(pr)rdr=(2\pi)^3\int_0^\infty j_l(kr)V(r)j_l(pr')r^2dr$$

Numerically, The Schrodinger equation be :

$$\frac{k^2}{2\mu}\psi_n(k)+\frac{2}{\pi}\sum_{i=1}^{N}k_j^2V(k,k_j)\Delta k_j\cdot\psi_n(k_j)=E_n\psi_n(k)$$

- In matrix form :

$$[H]\cdot[\psi_n] = E_n[\psi_n] \quad H_{ij} = \frac{k_i^2}{2\mu}\delta_{ij} + \frac{2}{\pi}V(k_i, k_j)k_j^2\Delta k_j$$

$$[\psi_n] = \begin{pmatrix} \psi_n(k_1) \\ \psi_n(k_2) \\ \vdots \\ \psi_n(k_N) \end{pmatrix}$$

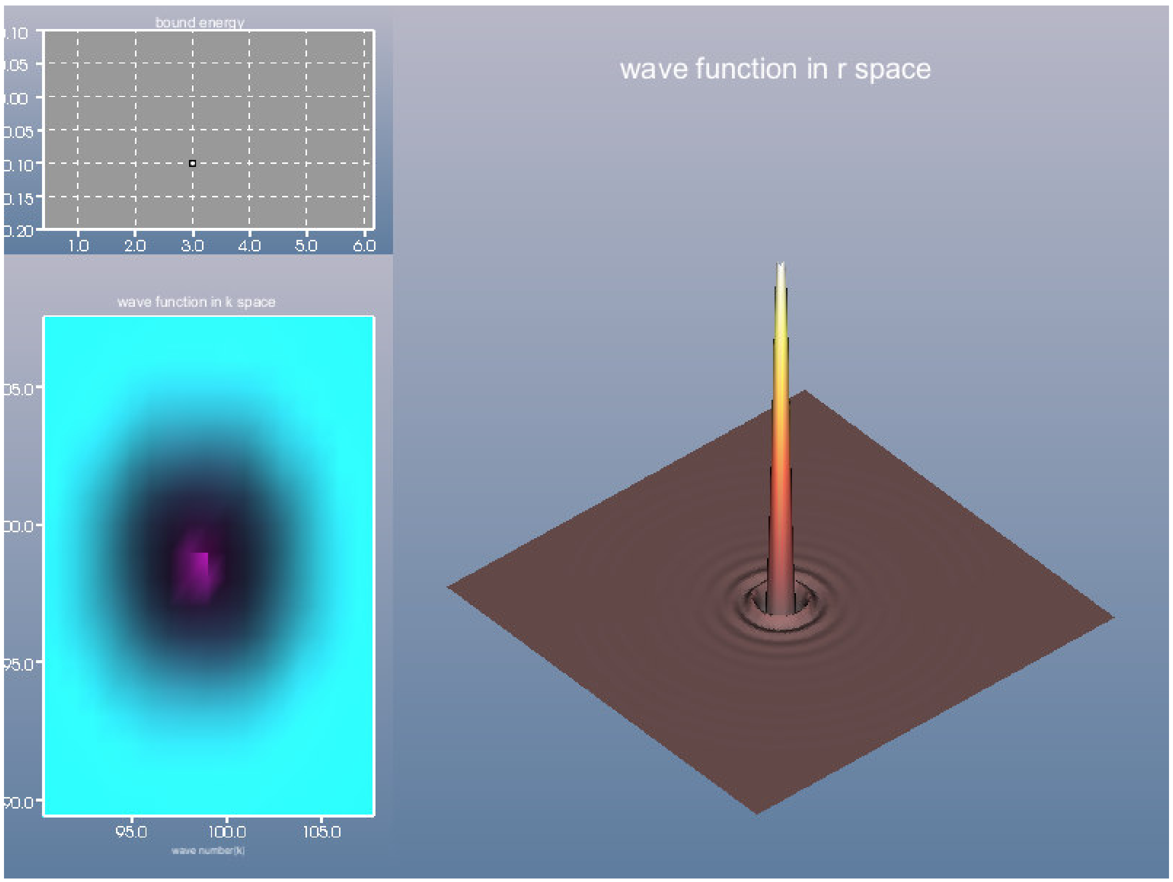
- To solve the eigensystem problem, we could receive several eigenvalues, but only one satisfys $|H - E_n I|$ (boundary condition). Once we find the $\psi(k)$, by inverse Bessel transformation we could get the wave function in r-space without any difficault : .

$$\psi(r) = \int_0^\infty \psi_n(k) j_l(kr) k^2 dk$$

$$V(r) = \frac{\lambda}{2\mu}\delta(r-b)$$

The simplest case : $V(r) = \frac{\lambda}{2\mu}\delta(r-b),\qquad V(k,p) = \int_0^\infty j_l(kr)V(r)j_l(pr)r^2dr = \frac{\lambda b^2}{2\mu}j_l(kr)j_l(pr)$

.



Summary :MATFOR Components

- **mfArray**
  - integrates the entire MATFOR toolkit into high-level programming environments such as C++ and Fortran, simplifying the syntax and facilitating object-oriented programming.
  - Basic data structure.
- **Numerical Library**
  - contains useful linear algebraic functions subject to assist users with computational problem solving.
  - Use mfArray to do numerical computation
- **Visualization Library**
  - collects well-designed graphical procedures and controls to support a variety of 2D and 3D visual functions.

MATFOR Components (Cont)

- **Data Viewer**
  - organized in spreadsheet format, is one convenient platform for data management, filter, and analysis.
- **Graphics Viewer**
  - besides its highly customized user interface, overthrows the convention of post-processing as it instantly visualizes scientific and engineering data.
- **mfPlayer**
  - captures picture frames, animates simulation results, and allows additional graphical manipulations.

- MATFOR®-

A Simple Yet Powerful Solution to Meet Your Needs!