Finite Element Modelling for Ultrasound Applications

Paul Reynolds - reynolds@ca.wai.com

John Mould - mould@ca.wai.com

Course Objectives

To educate attendees in the basics of Finite Element Analysis and its application to the most common ultrasound issues, so that they can make more informed choices when selecting a numerical method for solving their problem

Why?

- Finite Element Analysis becoming essential
  - Speeds design/analysis, lowers costs
- A complex field of study in itself
  - Most users require it as a tool
- Textbooks and papers not ‘beginner friendly’
  - No ‘Dummies Guide’
- Alternative Course Title
  - “What I Wish I’d Known When I Started Modelling”

Course Layout

- Four Sections

I. Finite Element Basics
II. Wave Propagation
III. Ultrasound and Thermal Effects
IV. Efficient Use of FEA

Finite Element Analysis (FEA)

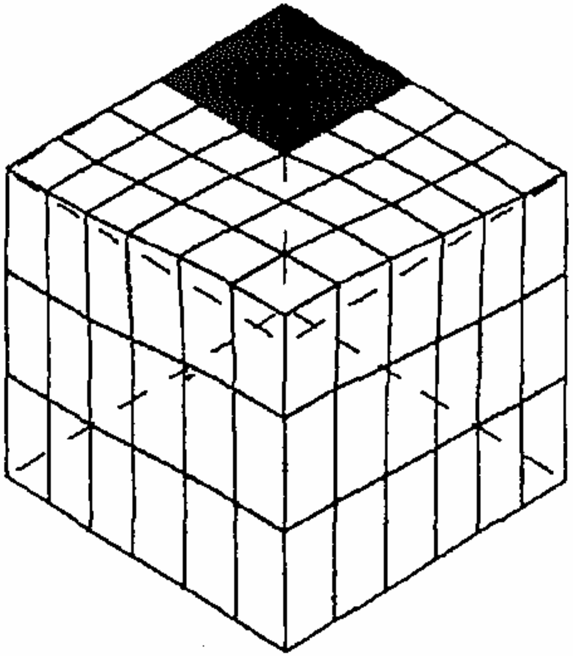
- Late 1980's saw first extensive FEA use in ultrasound
  - Combination of computing power, software availability, transducers requiring 3D analysis
- Early use mostly concentrated on piezoelectric transducer components
  - Resonant frequencies, mode shapes, impedance
- Complexity dramatically increased since 1980s
  - Codes now >100,000's lines long

FEA and Ultrasound

- Allik and Hughes, Kagawa and Gladwell,1970
- Ostergaard and Pawlak, 1986
- Lerch, Jeng 1987
- Hladky-Hennion and Decarpigny, 1993
- Wojcik, Vaughan, Mould, 1993

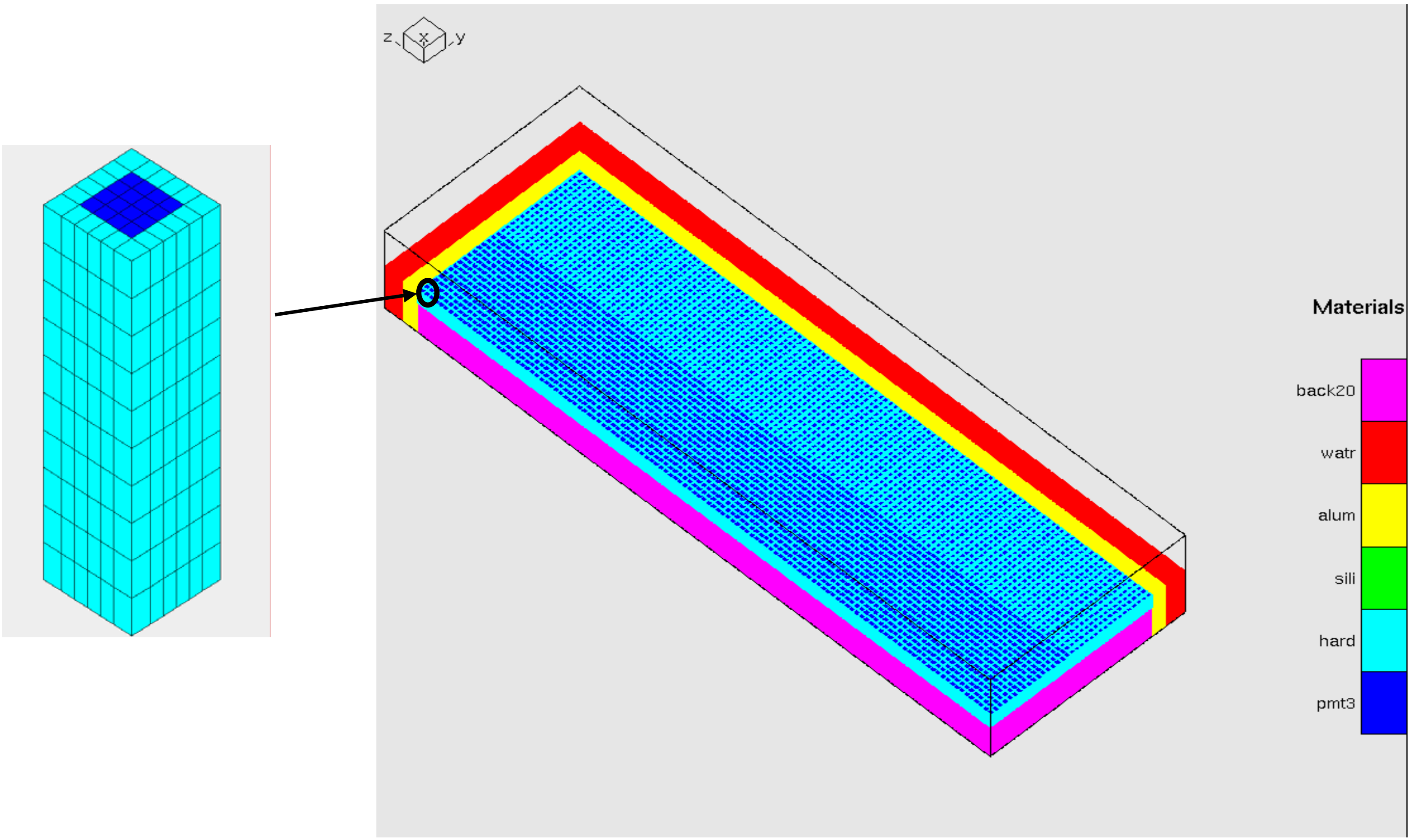
- Some commercial piezoelectric FEA packages
  - PZFlex, CAPA, ANSYS, ATILA, DYNA

Piezocomposite Device Modelling 1991



- FEA Model of Piezocomposite Cell
- “Finite Element Analysis of 1-3 Composite Transducers”, Hossack and Hayward, IEEE UFFC, V. 38 No 6, Nov 1991
- 1/8th symmetry
- 6 by 6 by 3 elements
- Solution time – 7.5 mins
  - Four frequencies

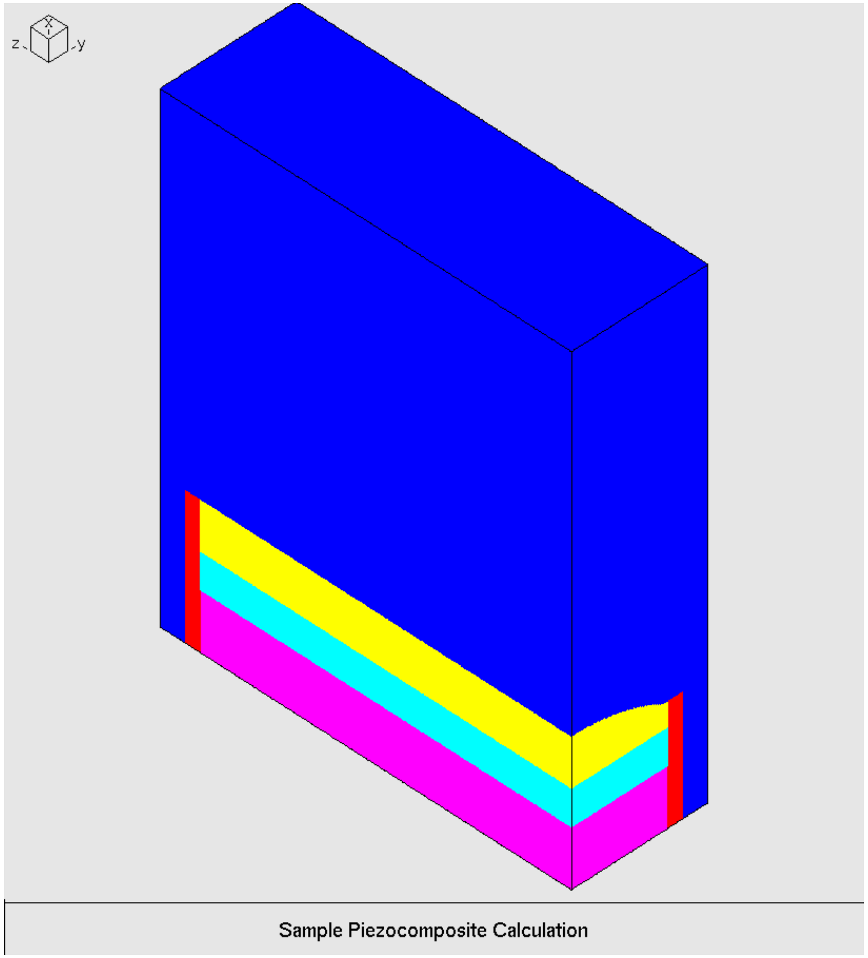
Piezocomposite Device Modelling 2005

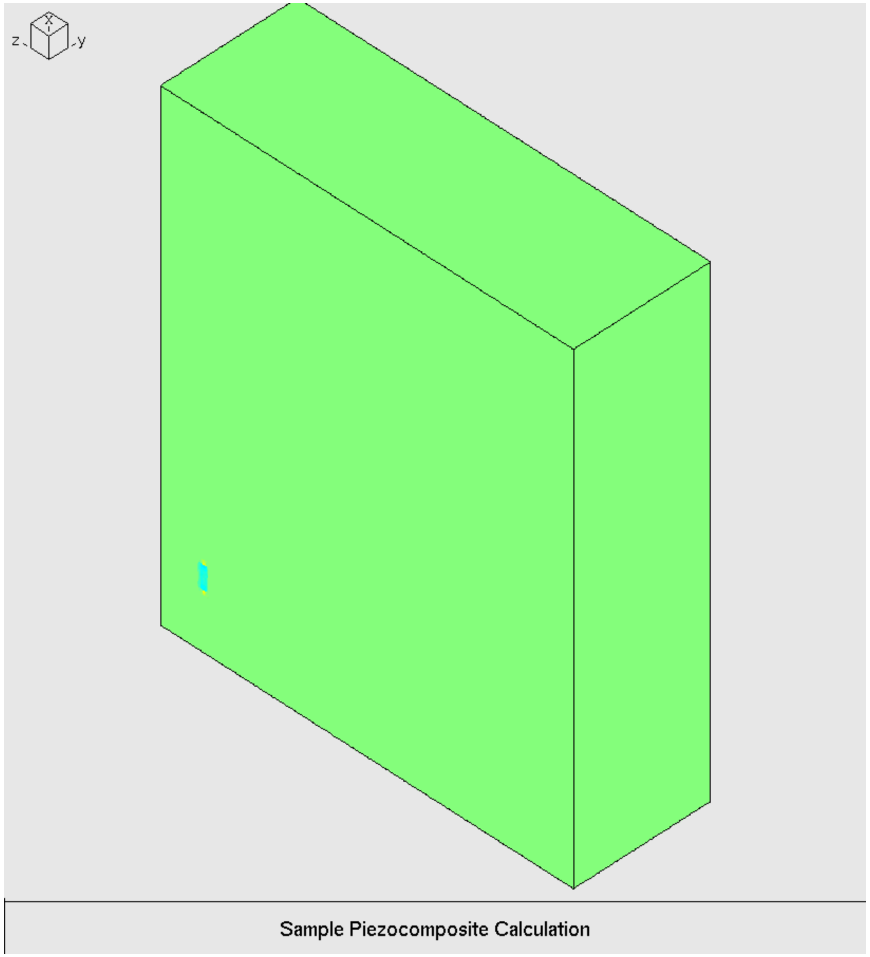


Piezocomposite Device Modelling 2005

- 23 million elements in ~3GB
- ~60 by 250 PZT pillars
- Backing, casing, lens, nonlinear fluid load
- Broadband impedance, beam profile, mode shapes, TVR
- Solves in ~6 hours on PC

Piezocomposite Device Modelling 2005





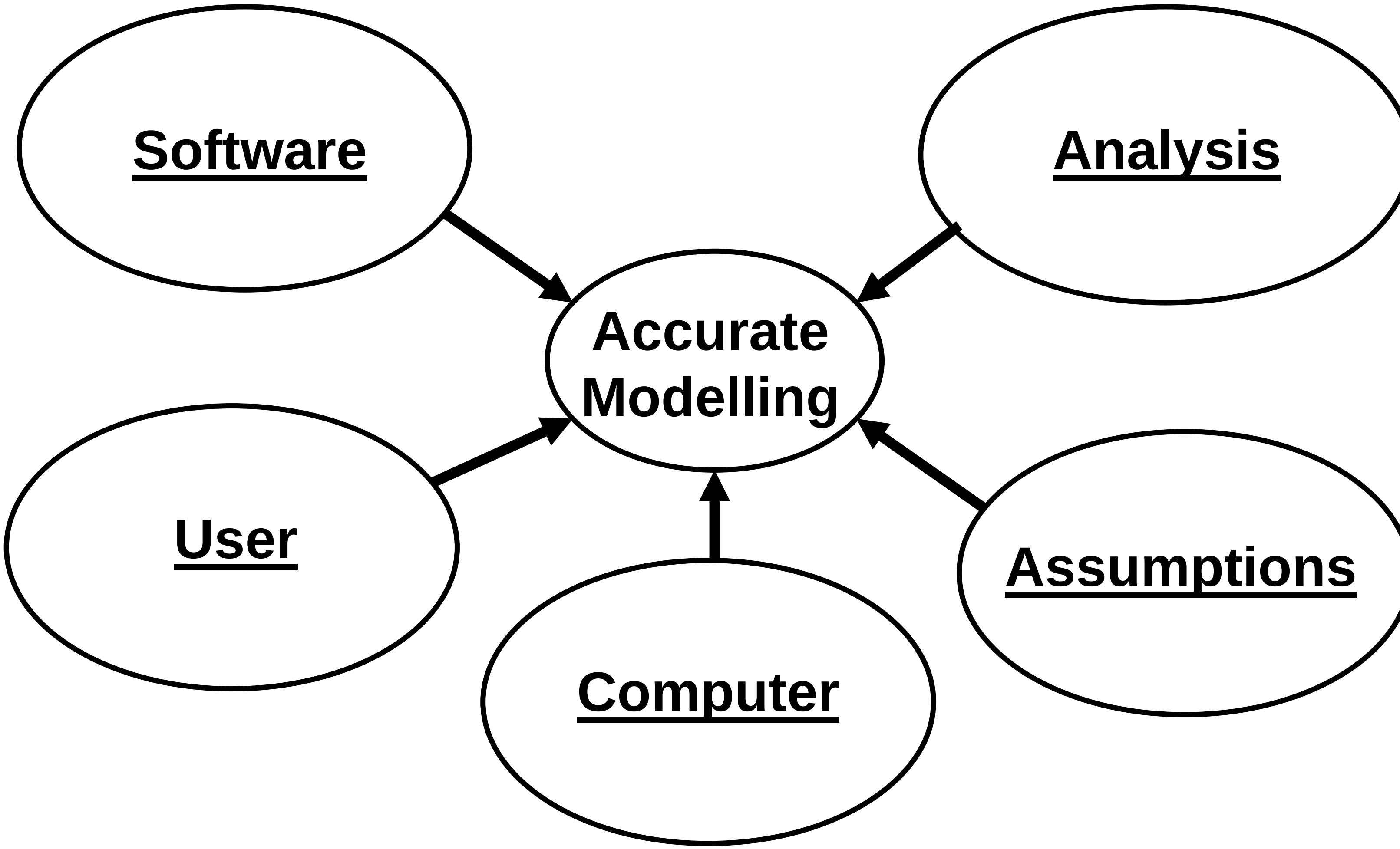
Finite Element Analysis (FEA)

- More recent use expands range of interests
  - Electrostatics
  - Thermal
  - Ultrasound imaging
    - Nonlinearity (Second Harmonic Imaging)
  - Thermal-Mechanical
  - Focused ultrasound surgery
  - Acoustic radiation force
- FEA now ‘devices’ <u>and</u> ‘applications’

Finite Element Analysis (FEA)

- Many effects tightly coupled
  - Electrical/mechanical in piezoelectrics
- Disparate size, size, and frequency scales
  - From sub-micron to 100's of meters
  - Picoseconds to minutes
  - Hz to GHz for piezoelectrics
- Software becomes complex rapidly
  - Most commercial codes have 10's to 100's of man-years development

An FEA System



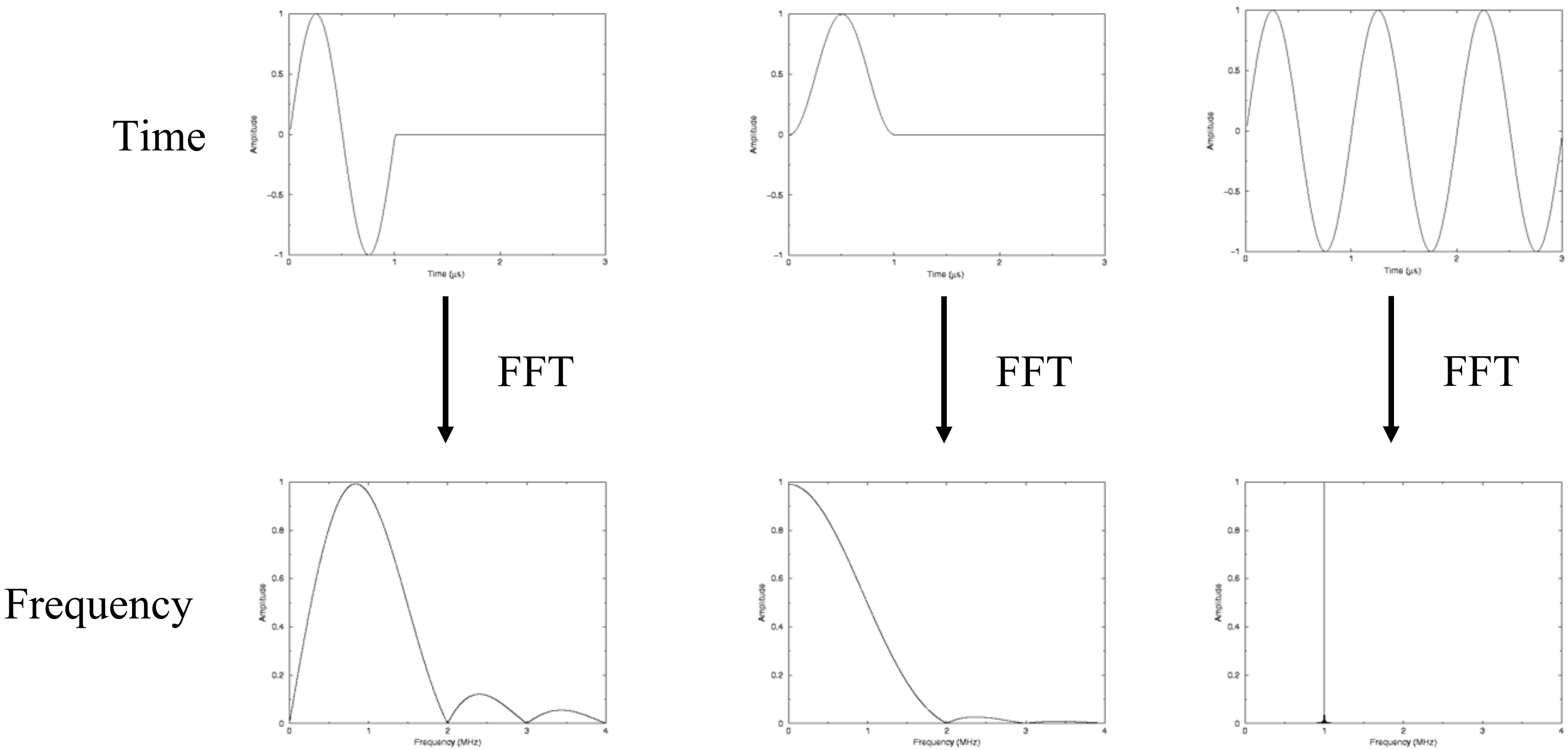
Time and Frequency Domains

- Alternative descriptions of same phenomena
  - Translate via Fourier transform / inverse transform
- Pick most efficient solver for application, then convert results if desired e.g.
  - Narrowband – Frequency
  - Wideband – Time
- More codes moving to time domain
  - Versatility and efficiency

Transient and Harmonic

- Time Domain
  - Results evolve in time
    - e.g. backscatter from multiple targets in imaging
  - Inherently broadband
  - Care required during analysis
  - Consistent damping function required
- Frequency Domain
  - Direct result obtained
  - Damping specified
  - Expensive solvers

Time and Frequency Domain



Large/Small Deformation

- Large deformation
  - Account for changes in geometry
  - Geometric nonlinearities
    - Mechanical and electrical
    - e.g. cMUTS, flexible structures
- Small deformation
  - Neglect changes in geometry, angles
  - Cheaper and more efficient
  - Usually a valid assumption in ultrasound

Discrete Numerical Methods

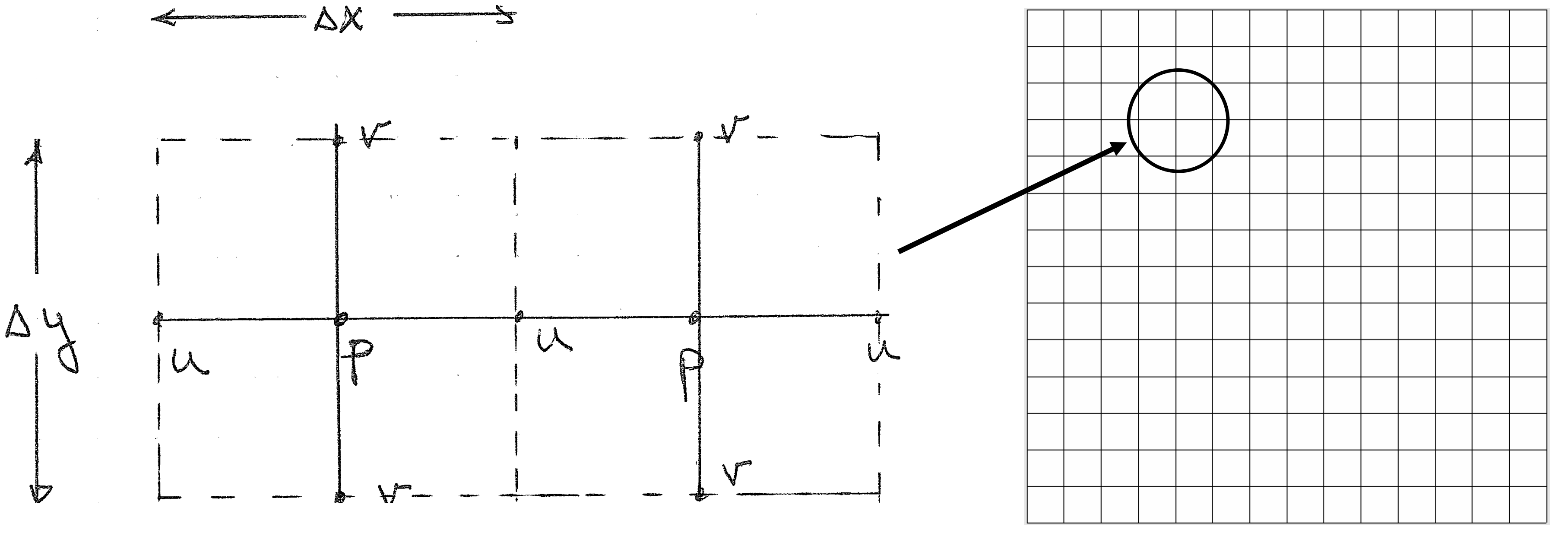
- Finite Difference
- Finite Element
- Finite Volume
- PseudoSpectral, k-space, AWP
- Boundary Element
- Element Free Methods (SPH, EFG)

- All converge at high mesh density

Finite Differences

- Typically grid is uniform and cartesian
- Approximate differentials with finite (non-infinitesimal) steps
- Typically all field values are at staggered locations
- FDTD
  - Finite Difference Time Domain

5 Point Finite Difference Stencil



pressure=stiffness * strain

$$p = K\left(\frac{\Delta u}{\Delta x} + \frac{\Delta v}{\Delta y}\right)$$

Finite Difference

- Oldest, simplest method
- Simple to code
- Fast
- Fixed, regular mesh
- Interfaces not sharp
- Boundary conditions inconvenient
- FD approximates differential equations

Other Methods

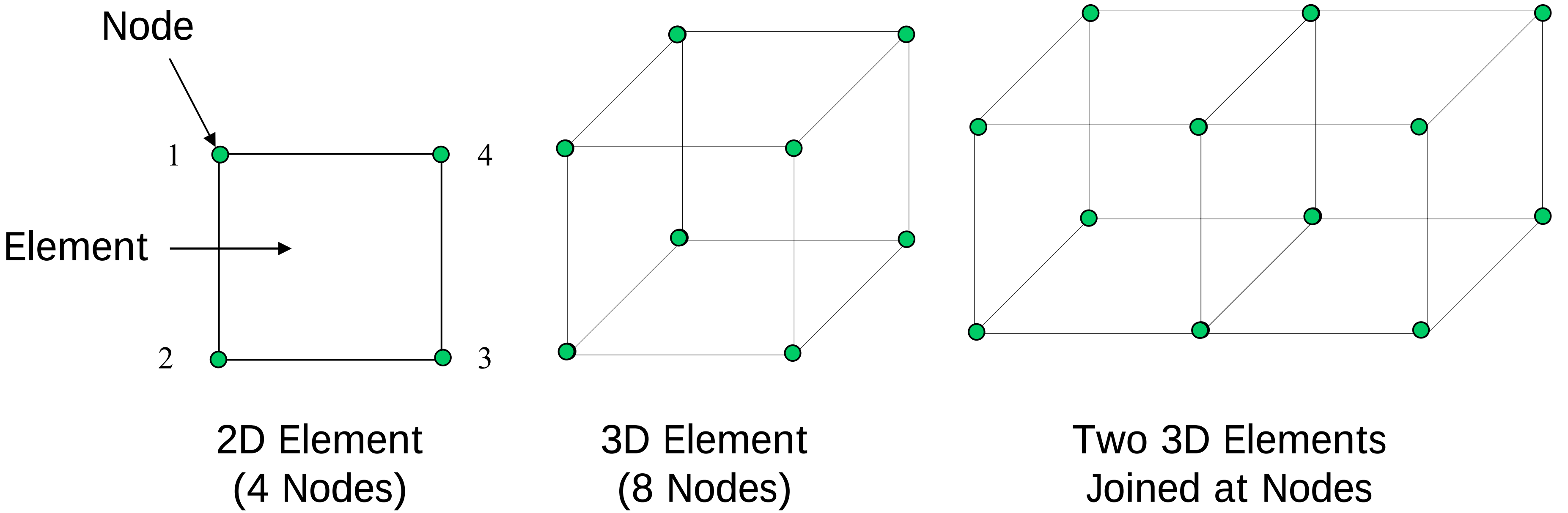
- Finite Volume
  - Typically used for Computational Fluid Dynamics (CFD)
    - Flow
- Boundary Element
  - Open domain, or limited number of interfaces
- PseudoSpectral
  - Highly accurate for long range, low contrast applications
- Meshless
  - Applications with sparse regions or changing connectivity

Finite Element

- Various mathematical formalisms
  - Galerkin, Variational principle, etc
  - Simple physical interpretation
- “Weak form” or integral technique
  - Optimal solution over a volume
- Conforming grids possible
- Handles interfaces naturally
- Boundary Conditions easier
- FE response over area/volume

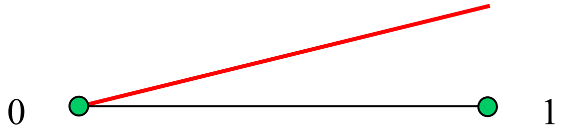
Elements & Nodes

- An element is a physical region bounded by sides defined by nodes
- "Building block" of model



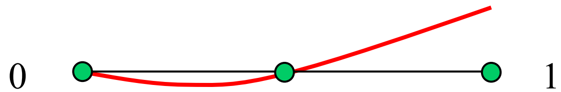
Element

- Fields (e.g. velocity, voltage) within an element are described by interpolating the nodal values of the fields
- “Shape Functions” describe interpolation method



Linear Shape Function

Two Node Element



Quadratic Shape Function

Three Node Element

Shape Function on a 1D element. Two and three dimensional models use similar shape functions in all directions

Basic Equations

- A node is subjected to an external force
  - e.g. Piezoelectric/electrostatic drive or pressure
  - Call this ‘F_t’
  - This is a time varying force
    - single value at given timestep
  - Displacement (u) of the node then depends on
    - Mass(m), Stiffness(k) and Damping(c)

Basic Equations

$$F_t = F_m + F_d + F_s$$

MASS	DAMPING	STIFFNESS
$F_m = ma$	$F_d = cv$	$F_s = ku$
Acceleration ($\ddot{u}$)	Velocity ($\dot{u}$)	Displacement (u)

- Force and velocity are vectors
  - 3 components in three dimensions
    - x, y and z directions

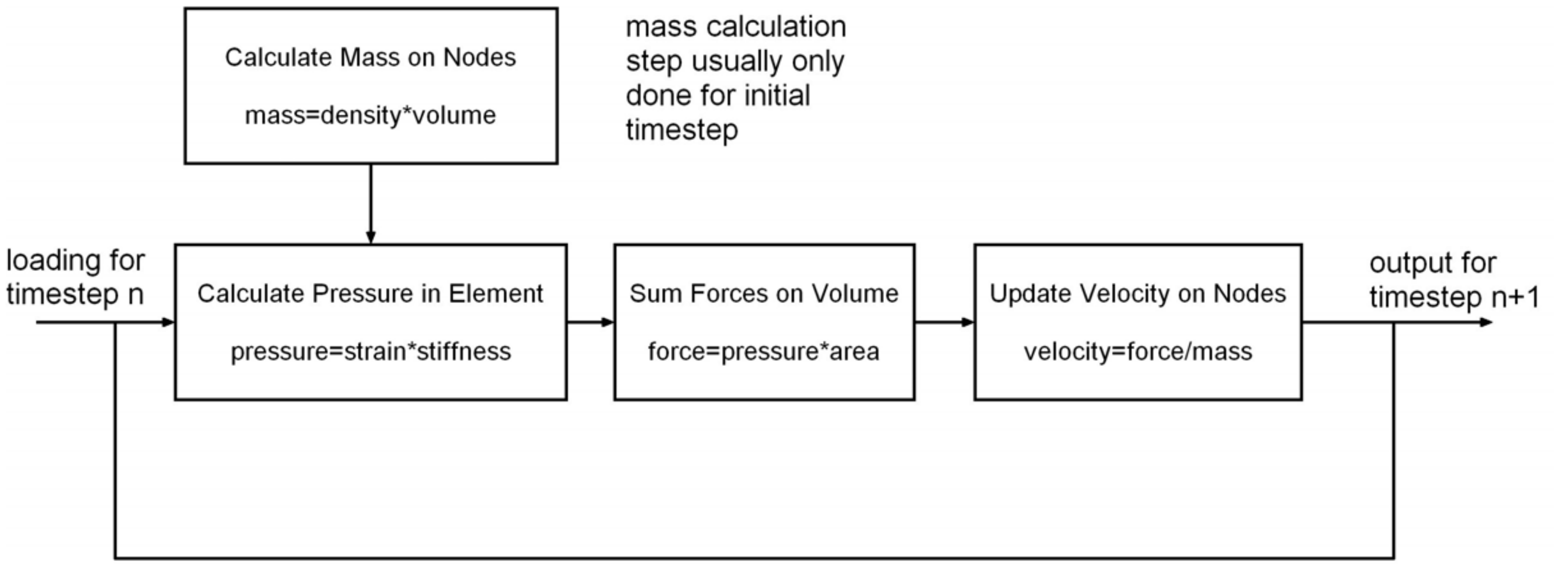
Assembly vs. EBE

- Assembly
  - Pre-compute and store each element's contribution to nodal Stiffness, Mass, Damping, Conductivity Matrices

$$Ku + C\dot{u} + M\ddot{u} = F$$

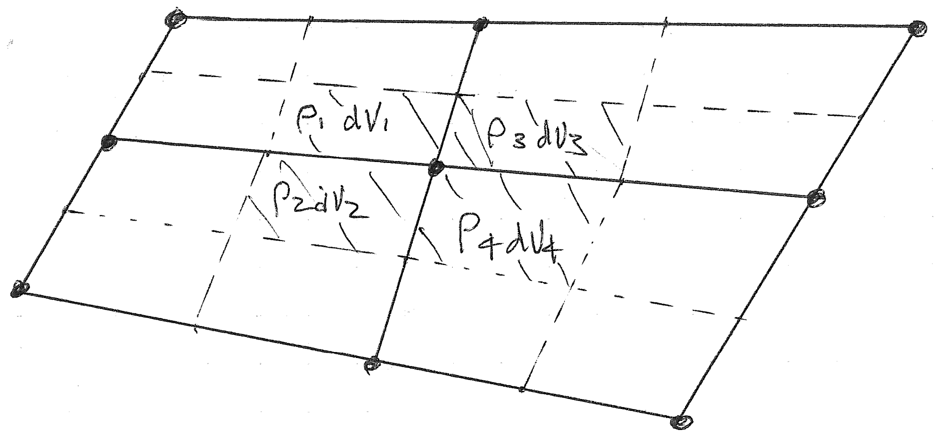
- Element-By-Element:
  - Compute Matrix-Vector product by sweeping thru elements
  - Can be used in explicit & iterative approach to implicit

Explicit Finite Element



Quantities are incremental changes per time step

Mass Lumping to Nodes



$$\mathbf{M} = \sum \rho \Delta V$$

mass = density * volume

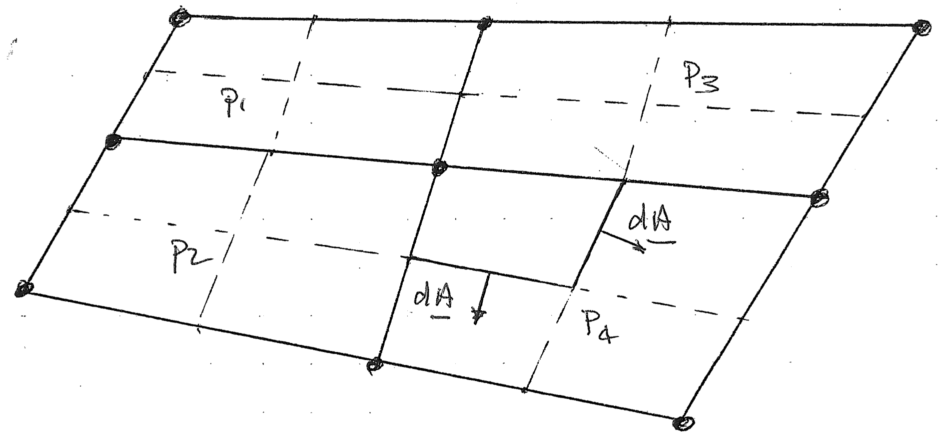
Compute Pressure in Elements

- Hooke's Law
- Update pressure in each element based on nodal displacements and Bulk Modulus
- Local to each element

$$p = K\left(\frac{\partial N_i}{\partial x} u_i + \frac{\partial N_i}{\partial y} v_i \right)$$

pressure = stiffness * strain

Sum Forces in Volume



$$\underline{F} = \sum p d\underline{A}$$

force = pressure * area

Update Nodal Velocities

- Newton's Law
  - Force = mass * acceleration
- u, v are displacements in x, y directions

$$\ddot{u} = \frac{F_x}{M} \qquad \ddot{v} = \frac{F_y}{M}$$

- Then update displacements

Calculated Responses

- Degrees of Freedom, Unknowns
  - Quantities to solve for e.g. velocity
- Primary quantities calculated
  - e.g. Force, velocity, charge, potential in PZFlex
- Other quantities derived
  - e.g. displacement, current, stress, electric field

FE Matrix Structure

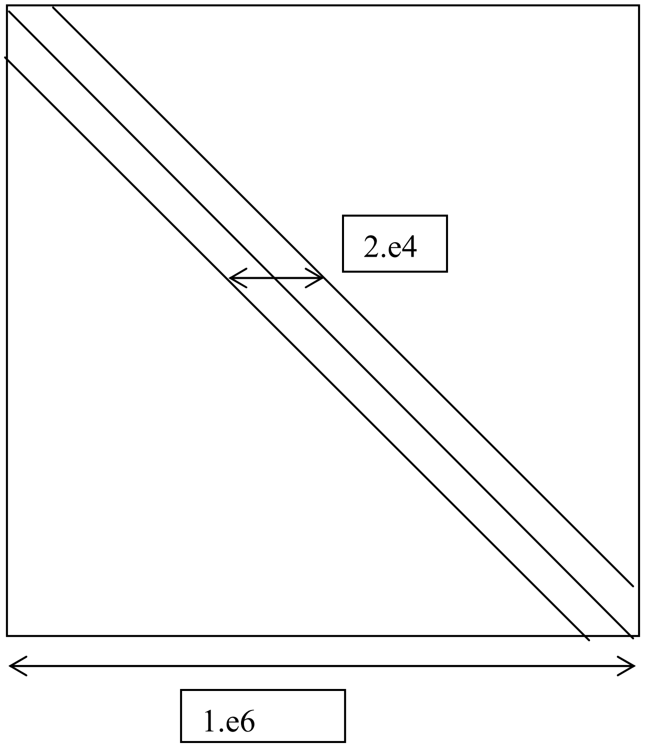
- Conventional assembled procedure
- K typically large and sparse
  - e.g. 27 entries per row (out of millions) in 3D for scalar problem
- Structured mesh
  - Banded
- Unstructured
  - Reordering to minimize fill-in

Matrices – Large & Sparse

- 100x100x100 node scalar problem
- **K** is (1,000,000 x 1,000,000) = $1*10^{12}$
  - $27*10^{6}$ nonzero entries (0.0027%)
- Structured mesh bandwidth = 20,000
  - $2*10^{10}$ nonzeros after factoring
  - Bandwidth
    - distance between first and last non-zeros per matrix row

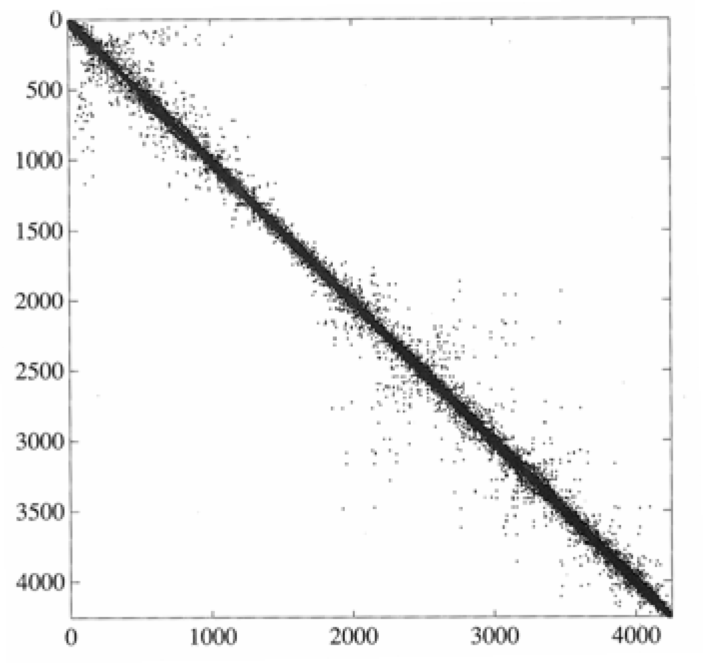
Solving Linear Systems

- Direct Methods (smaller problems)
  - Factor matrix (expensive), then backsolve (cheap) for each timestep, load case
  - Factor step is RAM & CPU intensive
- Iterative methods (larger problems)
  - Only need to store non-zero entries, or EBE to avoid storage altogether
  - More effort than a backsolve



Matrices

- K matrix for a small 2D unstructured mesh
- Re-ordering, sparse direct or iterative solvers
- Band solver inefficient



Solution Methods

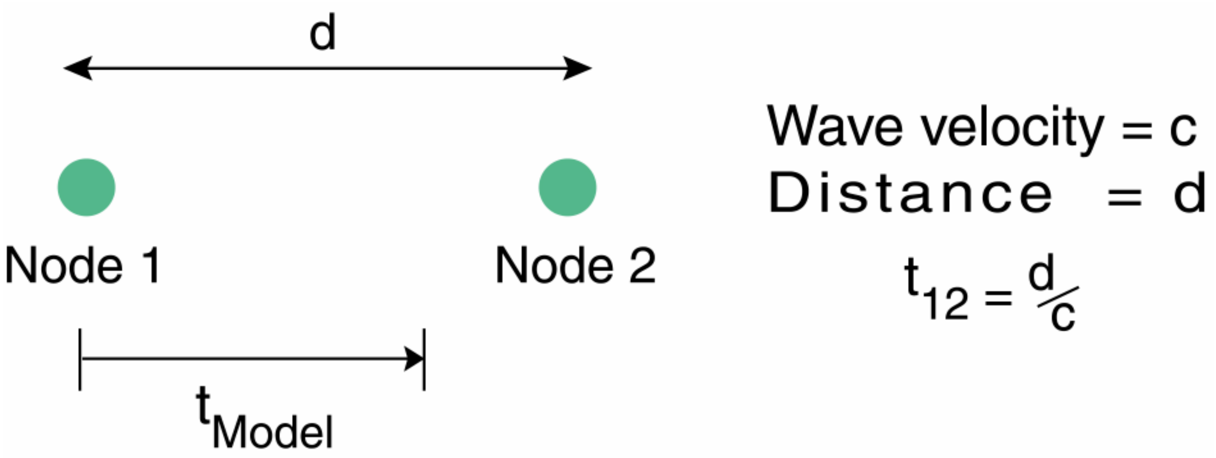
- Frequency Domain Implicit
  - Indefinite (or complex non-Hermitian)
    - Most expensive solvers (LU, GMRES, QMR…)
    - Solve 1 linear system per frequency
- Time Domain Implicit
  - Positive Definite
    - More efficient solvers (CG, Cholesky,…)
    - Solve 1 linear system per timestep
- Time Domain Explicit
  - Decoupled
    - Most efficient - in certain cases
    - Explicit recursion, no linear systems to be solved
    - 1 sweep per timestep

Implicit and Explicit Methods

- Implicit
  - Classic approach
  - Generate ‘matrix’ for entire structure, then solve
  - Considers effect of each node on all others
- Explicit
  - Sets time step to ‘decouple’ nodes from one another
  - Can be >100 times faster than implicit
    - For certain classes of problems
    - Also less memory required

Explicit Method

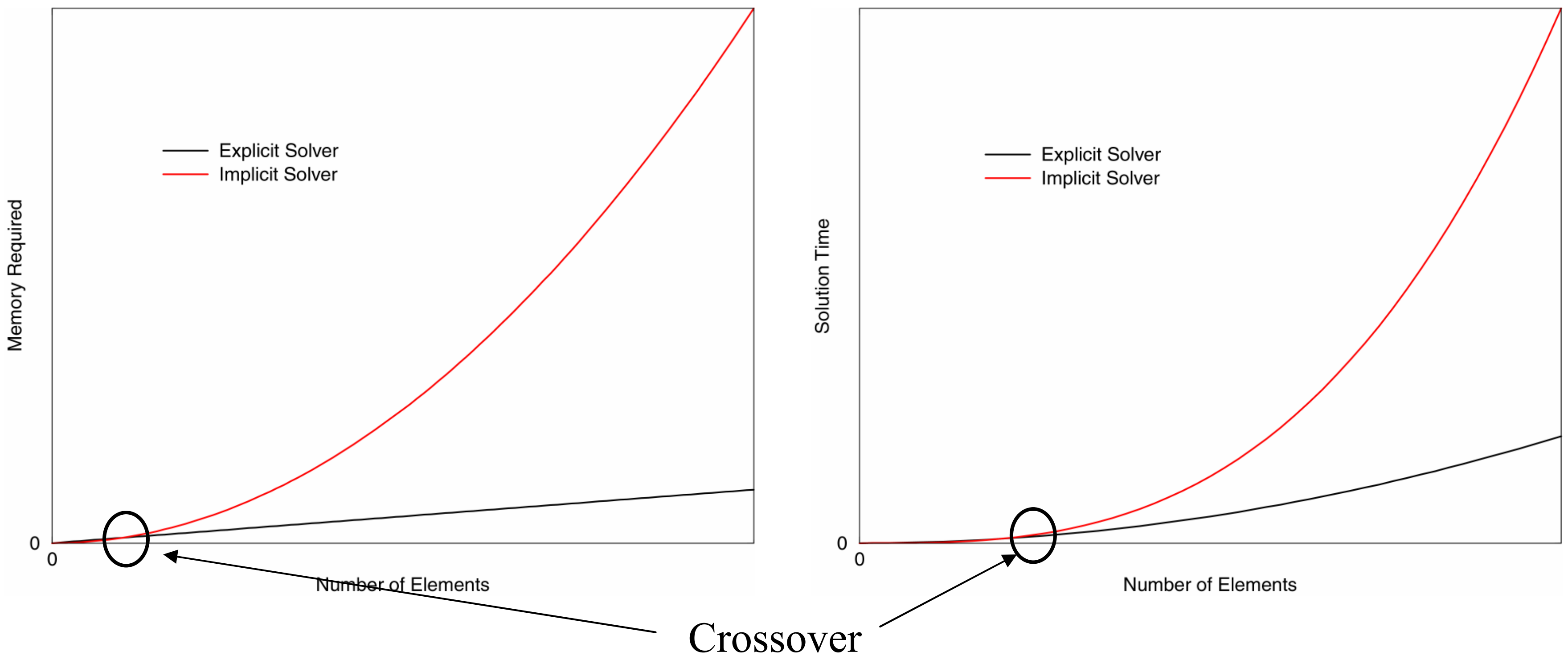
Decoupling the nodes:



t_{12} is the maximum stable timestep

If t_{model} is less than t_{12} then nodes are *decoupled*

Implicit and Explicit Methods



Small Models – Either method acceptable

Large Models – Explicit if possible

Graphs show results for 3D cube – 'best' comparison for explicit

Limitations of Explicit Method

- Effective for most mechanical wave propagation
- Limited in piezoelectric material
  - Electrical wave propagation much faster than acoustic
  - Effectively instantaneous travel between nodes
  - Consider electric field to be static at mechanical time scale
    - Electrostatic solution
  - Implicit solve needed for electric field
    - Positive Definite
- Models with vastly differing scales
  - Wavelength large compared to element size

Stability and Accuracy

- Accuracy
  - How good is the solution we are interested in?
- Stability
  - How does the algorithm treat the high frequencies?
    - Unconditional stability
      - Stable for any timestep
      - Timestep governed only by accuracy considerations
      - Implicit
    - Conditional stability
      - Stable for timesteps smaller than a threshold. E.g. $dt < dx/cp$
      - Explicit

Stability Implications

- Assume 2nd order implicit and 2nd order explicit methods produce about the same accuracy
- Explicit method is much cheaper per timestep
- If accuracy requires
  - dt ~= dx/c
  - Typical for wave propagation
  - Explicit is the obvious choice
- If accuracy allows
  - dt >> dx/c
  - Implicit may be preferred
    - low frequency structural vibration problems

Order

- Solvers have ‘Order’ for space and time
  - N^{th} Order
- Higher order are
  - Usually more accurate
  - More expensive in CPU and RAM
  - Less effective at interfaces
- At least 2^{nd} order in both space & time usual
  - Doubling number of elements reduces error by 2^N
- Match space and time order for error cancellation

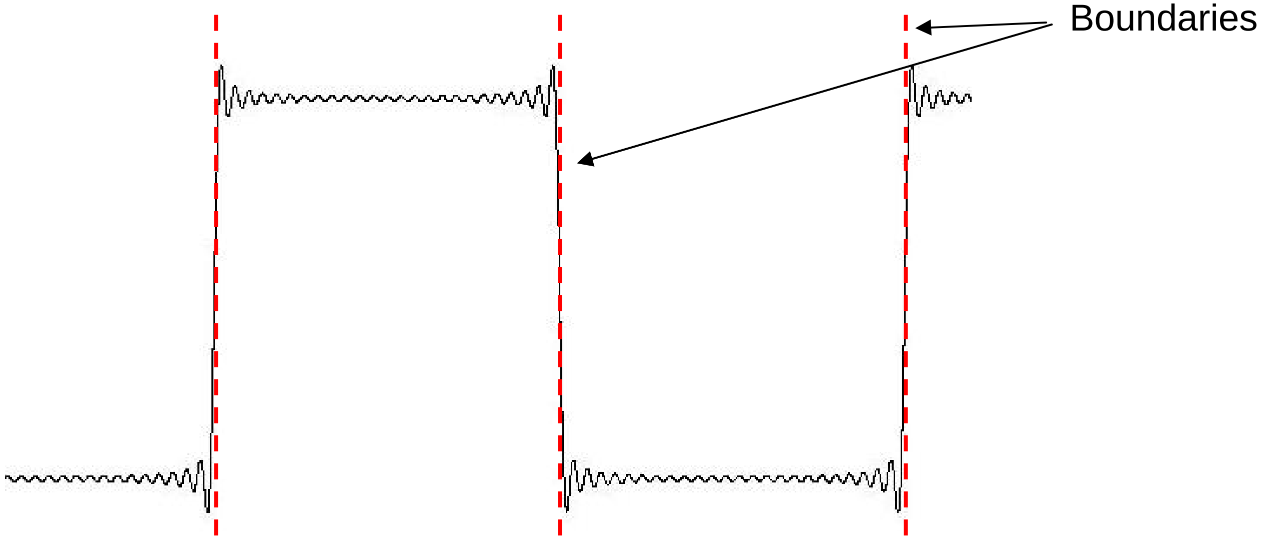
Smoothness

- Smoothness
  - Larger number of continuous derivatives
- Smooth problems
  - Benefit from higher order methods
  - Non-smooth better represented with lower order
- Limit 2 nodes per wavelength for exact solution
  - e.g. Pseudospectral, high order FD/FE
  - Nyquist limit
- Gibb’s phenomenon

Smoothness & Gibb’s Phenomena

Smoothness

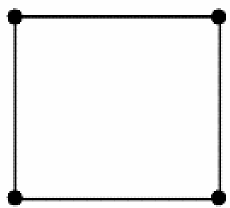
Gibb’s
Phenomena



Element Types

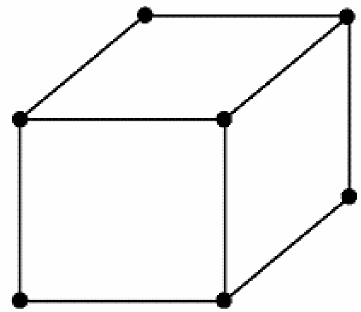
- Triangles/Tetrahedra
  - Least accurate
  - Convenient for auto grid generation
- Quadrilaterals/Hexahedra
  - Efficient and accurate
  - Auto grid generation not always possible
- Higher order
  - More nodes per element

Element Types



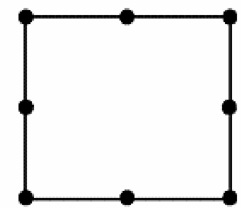
2D Quadrilateral

Low Order



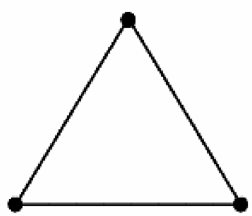
3D Hexahedron

Low Order



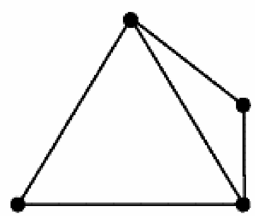
2D Quadrilateral

Higher Order



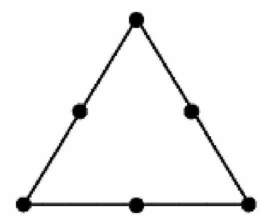
2D Triangle

Low Order



3D Tetrahedron

Low Order



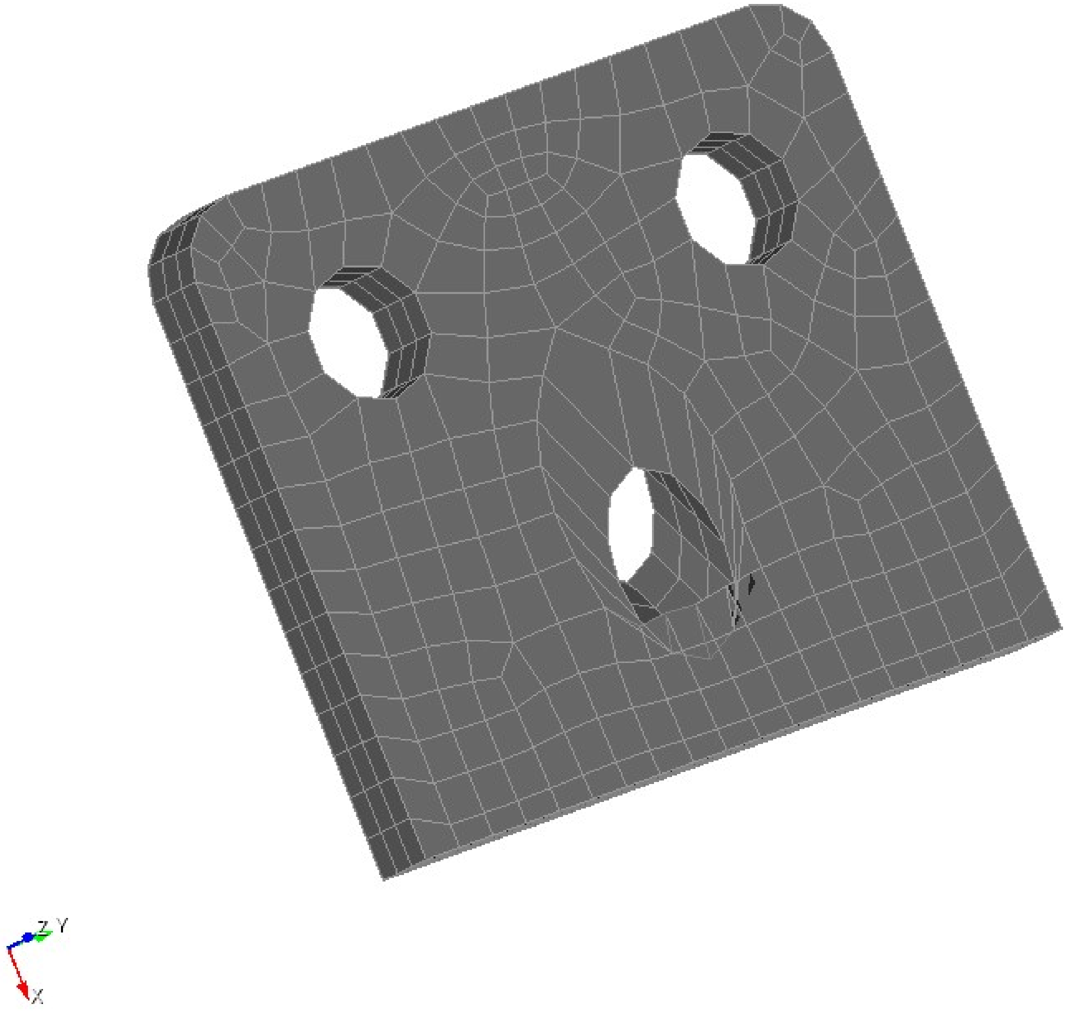
2D Triangle

Higher Order

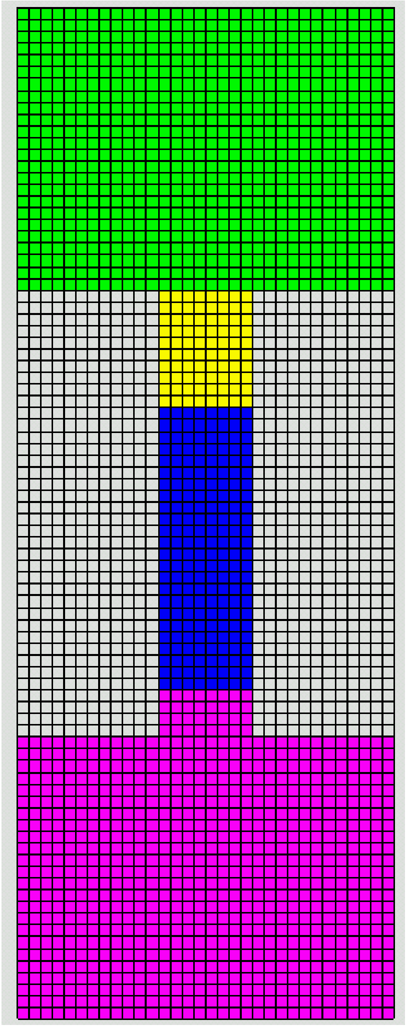
Grids

- Structured Grid (ijk)
  - Implicit connectivity
  - Not necessarily cartesian
  - Less storage
  - Faster
- Unstructured grid
  - Connectivity must be defined and stored
  - Easier to mesh complex geometries
- Block Structured Mesh
  - Structured pieces are bonded together

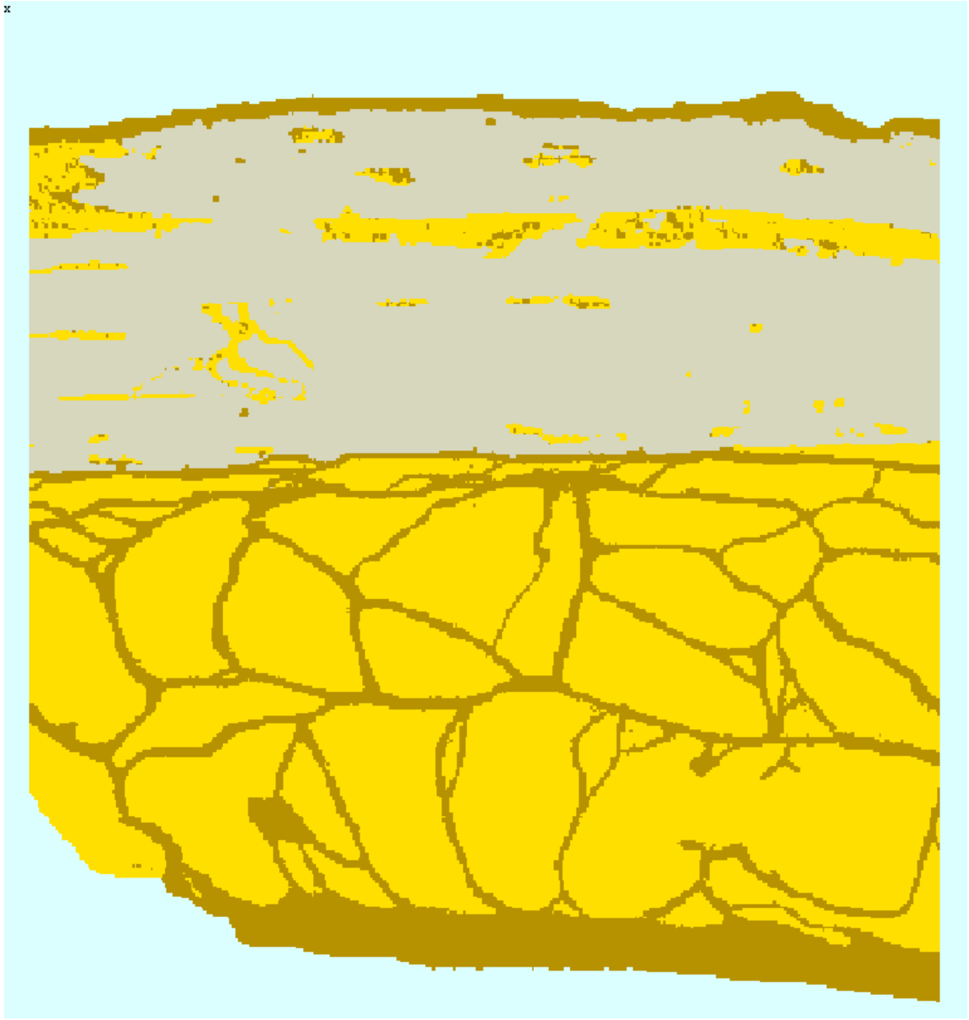
Unstructured Grid



Structured Cartesian Mesh

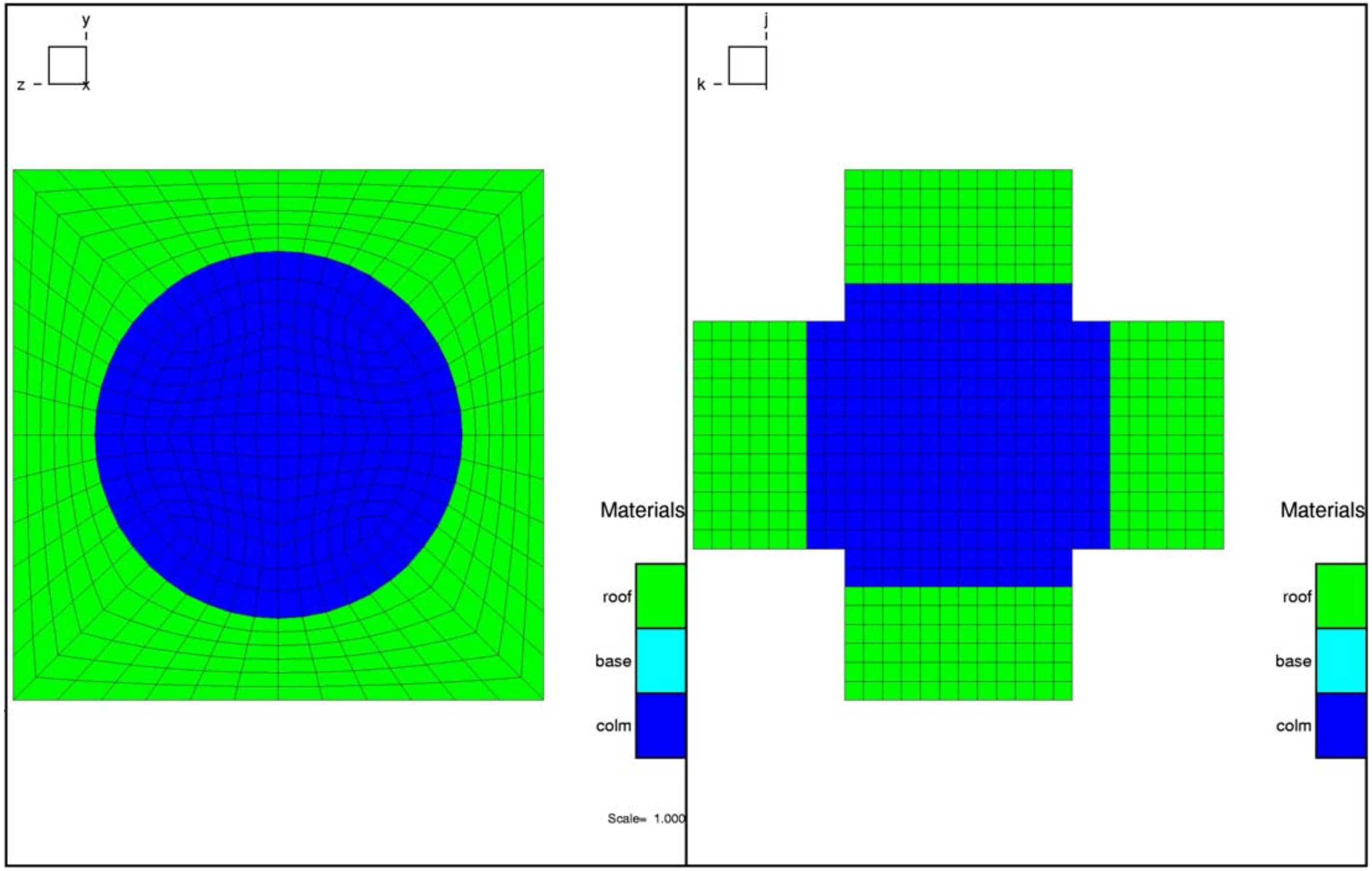


Piezoelectric Element

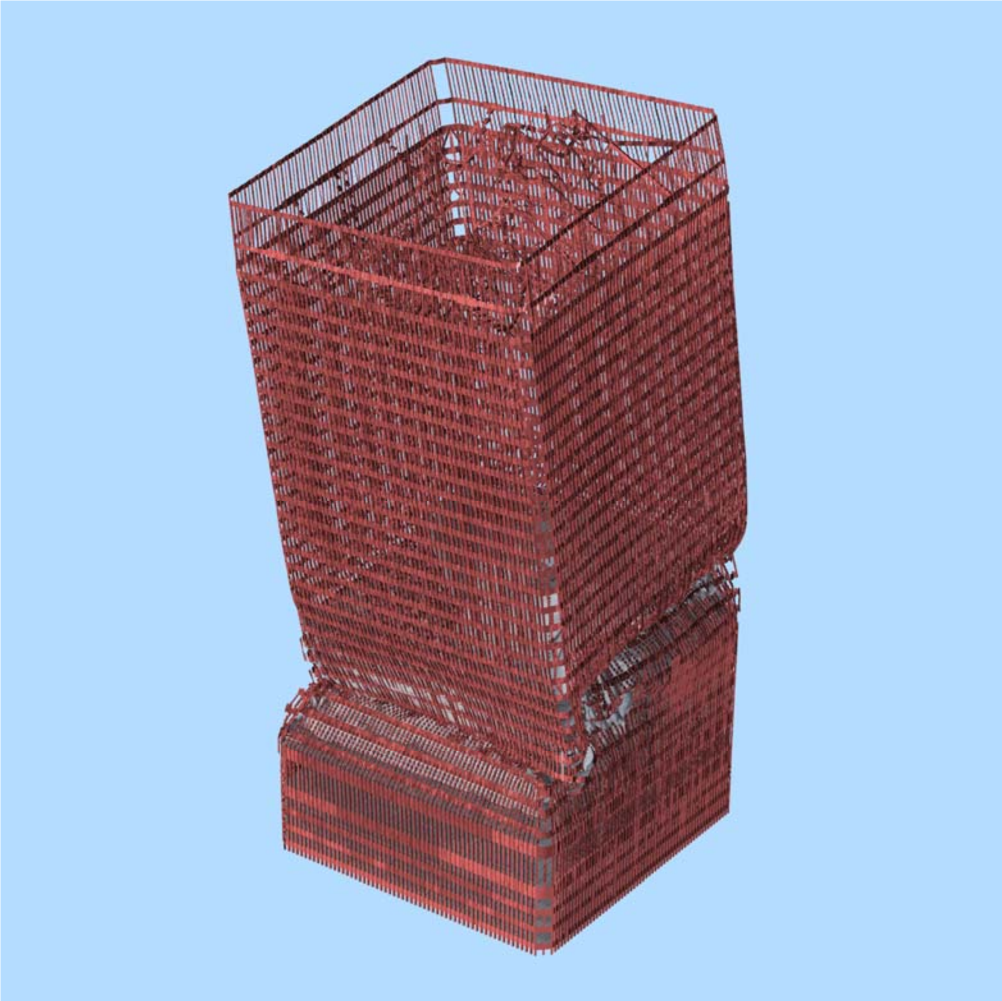


Tissue Model

Structured Non-Cartesian Mesh



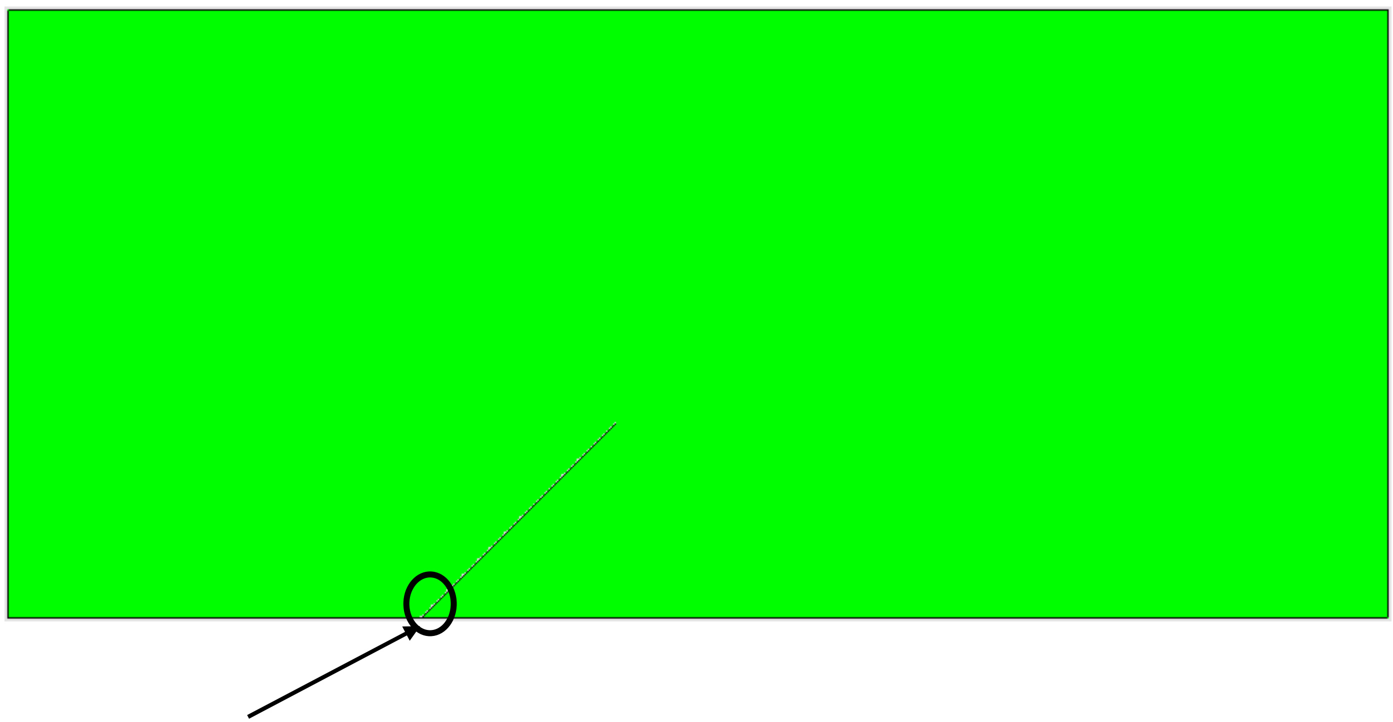
Structured Grid, Non-cartesian

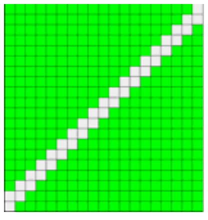


Stair Stepping vs. Conforming

- Conforming elements
  - More accurate representation of interfaces
  - Skew reduces element accuracy
  - Skew dramatically increases element cost
  - Use when geometric details are large compared to wavelength

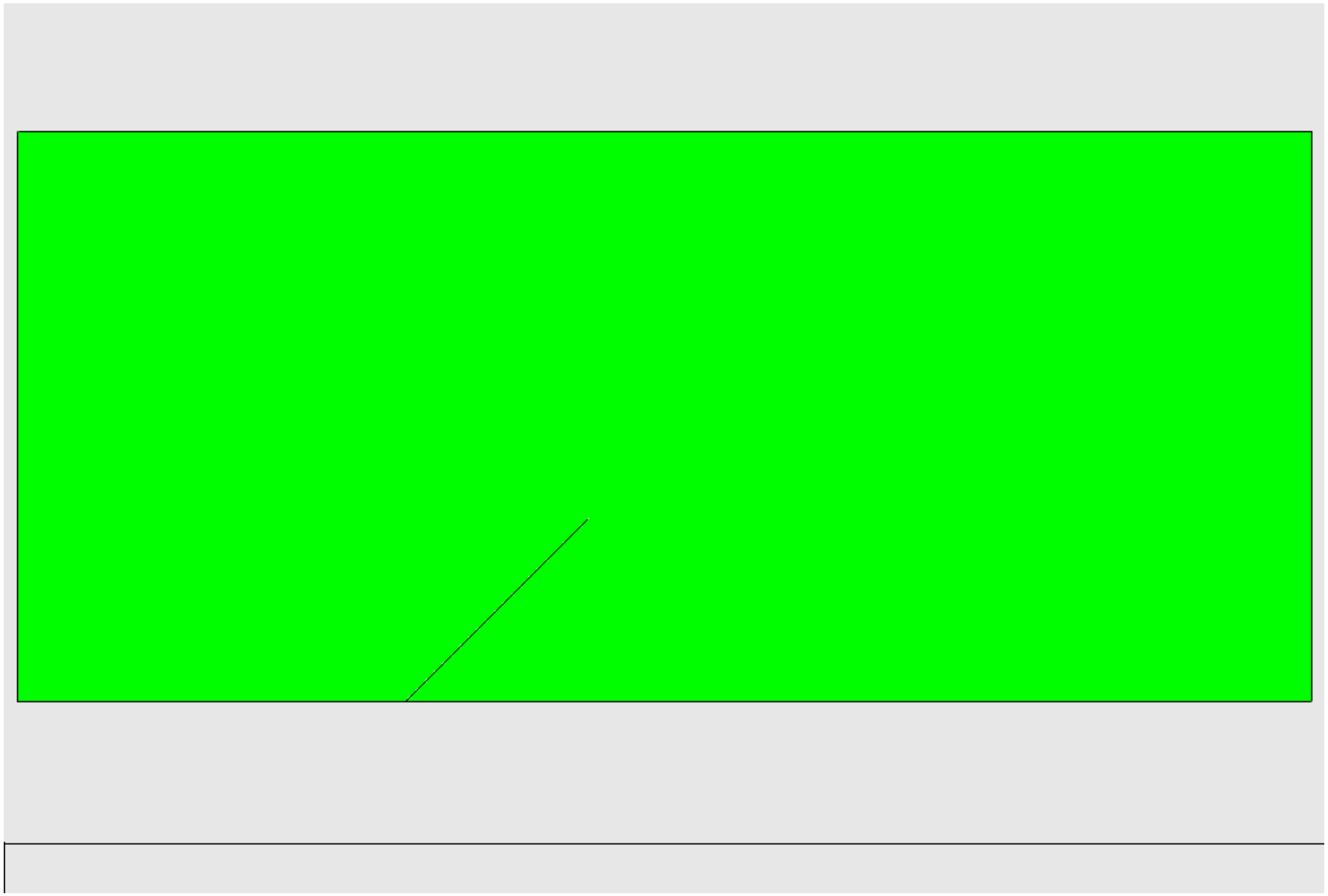
Stair Stepping vs. Conforming





- 45 degree notch in steel block represented by non-conforming elements
  - ‘Stair-stepping’

Stair Stepping vs. Conforming



Nonlinearities

- Material
  - PZT for high drive applications
  - B/A, cavitation in acoustic materials
- Geometric
  - Contact (cMUT)
  - Change of geometry
  - Large strains, electric fields
  - Membrane forces

Nonlinear Solution Algorithms

- More costly than linear
- Re-formulate, update or iterate
- Superposition & scaling not possible
  - Solution at 100 Volt is not 100x the solution at 1 Volt
- Linear often adequate
- ‘Falls out’ naturally in the time domain
- Explicit implementations are easier than implicit

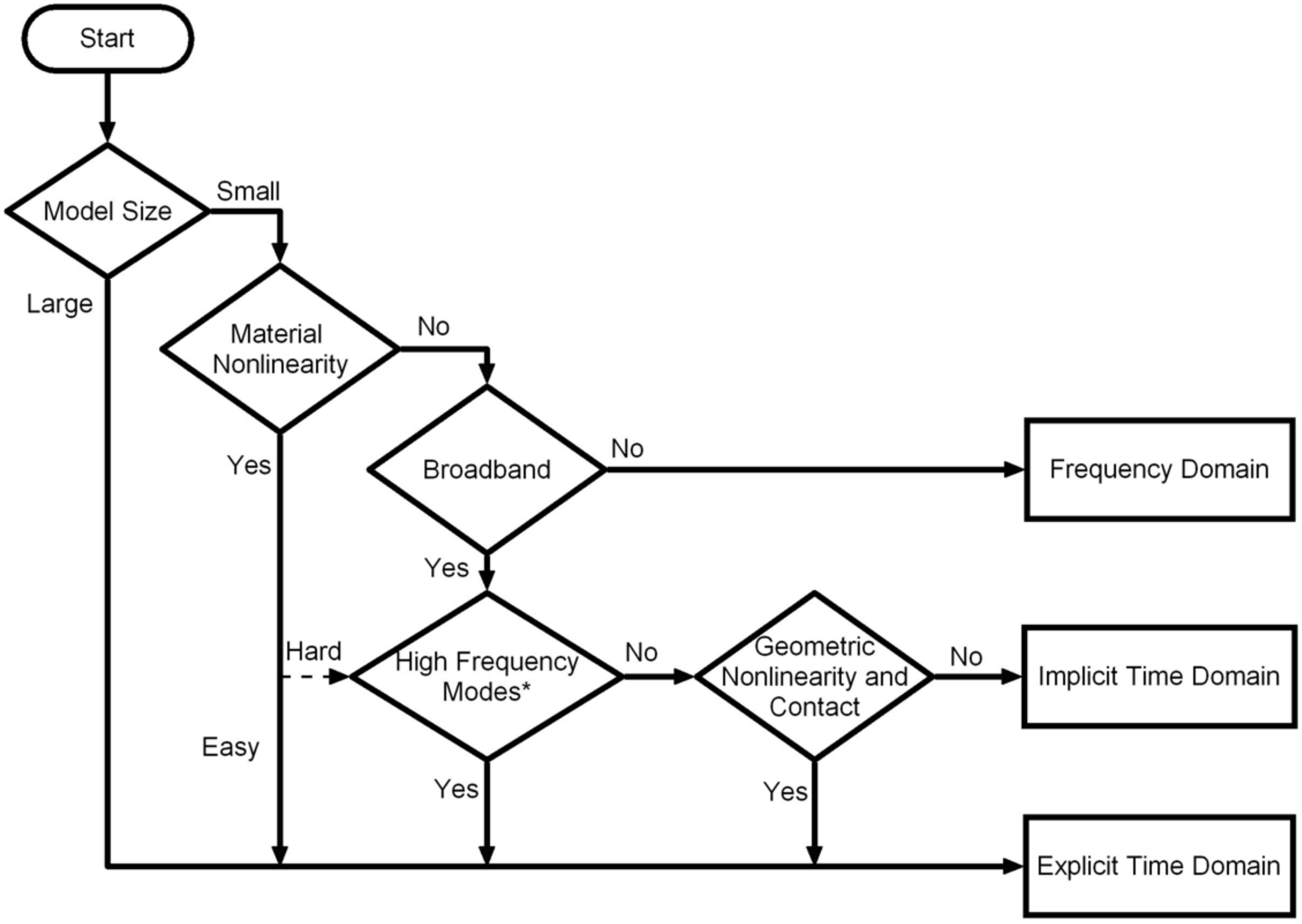
Boundary Conditions

- Model must be finite in size for efficiency
- Edges of model must simulate interaction with ‘rest of the world’
  - Symmetry
  - Periodicity
  - Free
  - Pressure
  - Velocity
  - Radiation conditions (AKA absorbing boundaries)
    - For “open domain” problems

Radiation Conditions

- Desired result
  - Waves propagate out of model, no reflection
- What happens at the edge of the grid?
  - Material interfaces
- Analytical approximations
  - Accuracy vs locality, cost
  - Linear / nonlinear
- Multiple wavetypes incident?
- PML – Perfectly Matched Layer
  - Artificial damping, balanced to be non-reflecting
- Perfect vs Cheap vs Versatile

FE Choices for Efficiency



*When timestep to resolve frequencies of interest » timestep due to element size

FE Choices: Examples

- Piezoelectric transducer
  - Requirements
    - Disparate multi-layered materials
    - Broadband
    - Time and size scales comparable
    - Compact structures
  - Solution
    - Implicit for smaller models
    - Explicit for larger
    - Frequency dependent attenuation
    - Low order for boundaries

FE Choices: Examples

- Wave propagation through tissue
  - Requirements
    - 100’s of λ propagation distance
    - Soft tissue (low shear wave speed)
    - Small variation in material properties (smooth)
    - Broadband
  - Solution
    - High order elements or Pseudospectral methods
    - Longitudinal waves only
    - Imaging from backscatter – time domain

FE Choices: Examples

- cMUT
  - Same requirements as piezoelectrics but also
    - Wide spatial and temporal ranges
    - Broadband in water, narrowband in air
    - Bending membrane
    - Highly nonlinear
  - Solution
    - Implicit for small to medium numbers of elements
    - Explicit for contact and large numbers of elements
    - Higher order elements for bending
    - Hybrid approach?

Typical Biomedical Devices

- Moderate size for most applications
  - 2D or 3D models needed, 10s of λ along each side
  - Some imaging applications 100s of wavelengths in size
- Mostly elastic and acoustic materials
  - Piezoelectric/electrostatic typically small percentage of overall volume
- Transient behaviour
  - Broadband, wide frequency ranges needed
- Potentially nonlinear effects
- Potential thermal effects

- FE code should be chosen to meet application needs
  - Each code has a ‘sweet spot’

Summary

Summary

- Finite Difference
  - Simple, fast
  - Regular grid, difficulty with BC's and interfaces
- Finite Element
  - Conforming grids, BC's and interfaces simplified

- Structured Grid
  - Low computational/memory cost, implied connectivity
  - Innefficient for some structures
- Unstructured Grid
  - Simplified construction of complex models
  - Higher memory/computational requirements

Summary

- Frequency Domain
  - Single frequency, direct result obtained
  - Multiple simulations for broadband, boundary effects issue, solver can be expensive
- Time Domain
  - Matches ‘real world’, frequency domain through use of Fourier techniques
  - Faster for large problems, versatile
  - Analysis of data requires more care, can be slower for smaller problems

Summary

- Implicit
  - Forms matrix for entire structure
  - Used for small or ‘low frequency’ models
  - Becomes very expensive for large models
  - Nonlinear more difficult/costly
- Explicit
  - Time domain only
  - Decouples nodes to reduce memory/CPU required
  - Not appropriate for all cases e.g. piezoelectric
  - Enforces time step limit for stability
  - Nonlinearity straightforward to include

Summary

- High Order Elements
  - Usually more accurate
  - More expensive, requires ‘smooth’ interfaces
- Low Order
  - Extremely inexpensive, good at interfaces
  - Can require large number for accuracy

- Linear/Nonlinear
  - Linear assumptions simplify models
  - Application in Time domain easier than Frequency
  - Nonlinear more expensive