INTERPOLATION ALGORITHMS

Why:

- Remeshing
- Multi-Disciplinary Code Coupling
- Boundary Conditions from Discrete Data
- Visualization

BASIC IDEA: SHAPE-FUNCTION VALUES OF COORDINATES

Given:

- Element *el* With Shape-Functions/Nodes: N^i, $\mathbf{x}_i$
- Point With Coordinates $\mathbf{x}_p$

Then:

$$\mathbf{x}_p = \sum_i N^i \mathbf{x}_i$$

Point In Element Iff:

$$min(N^i, 1 - N^i) > 0 \quad , \quad \forall i$$

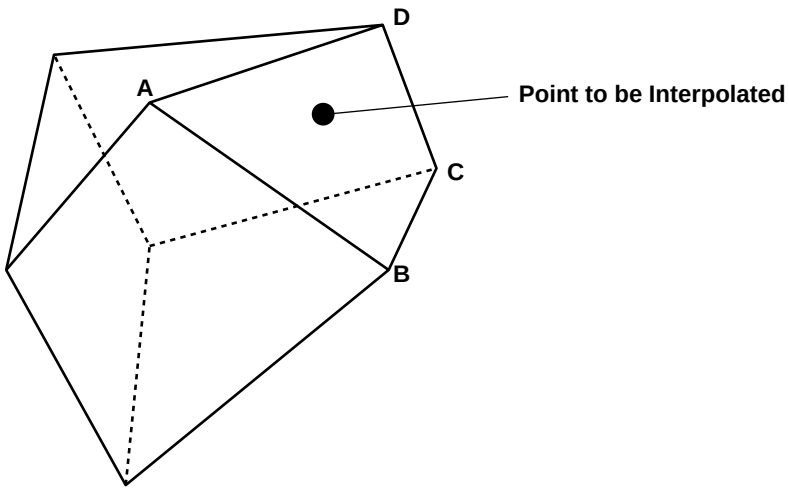


Figure 12.1 Interpolation for a Deformed Brick Element

BRUTE FORCE

Given:

- Elements and Points of Known Grid
- Point of Unknown Grid
- Other Info
    - None

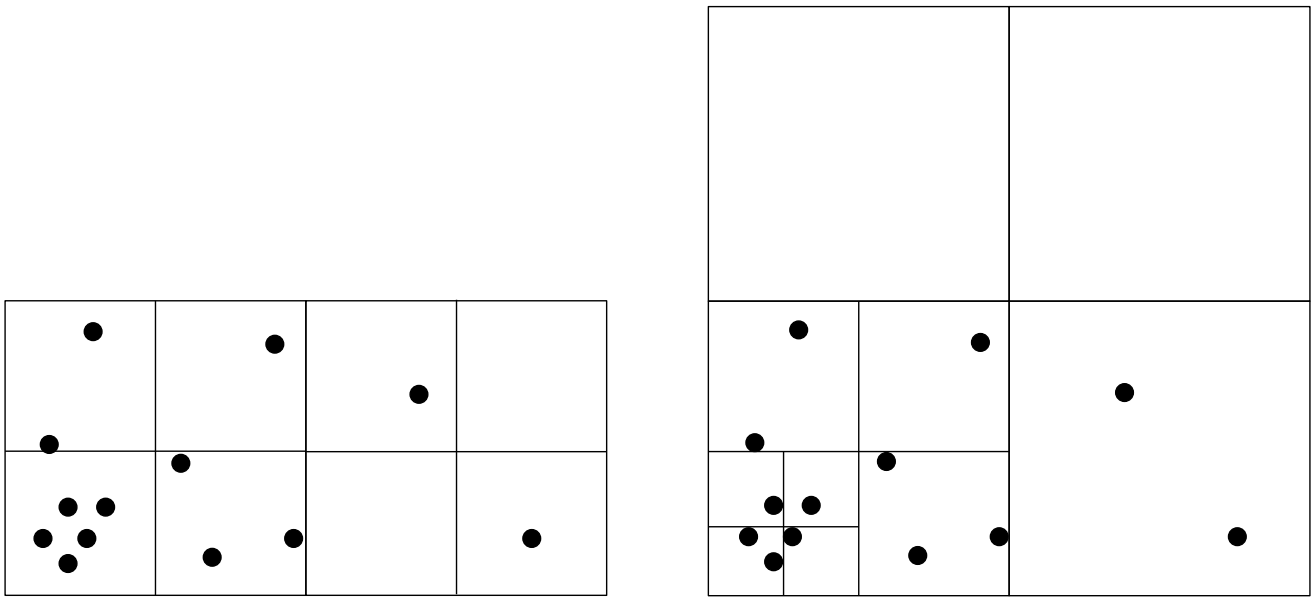
Then:

- Vector-Loop Over the Elements of Known Grid, Checking

Remarks:

- Fastest 1-Time Start Algorithm

FASTEST N-TIME START ALGORITHM: OCTREE SEARCH



a) Bin **b) Quadtree**

Algorithms to Determine Spatial Proximity

FASTEST N-TIME START ALGORITHM: OCTREE SEARCH

Given:

- Elements and Points of Known Grid
- Point of Unknown Grid
- Other Info:
  - Octree of Points of Known Grid
  - List of Elements Surr. Points of Known Grid

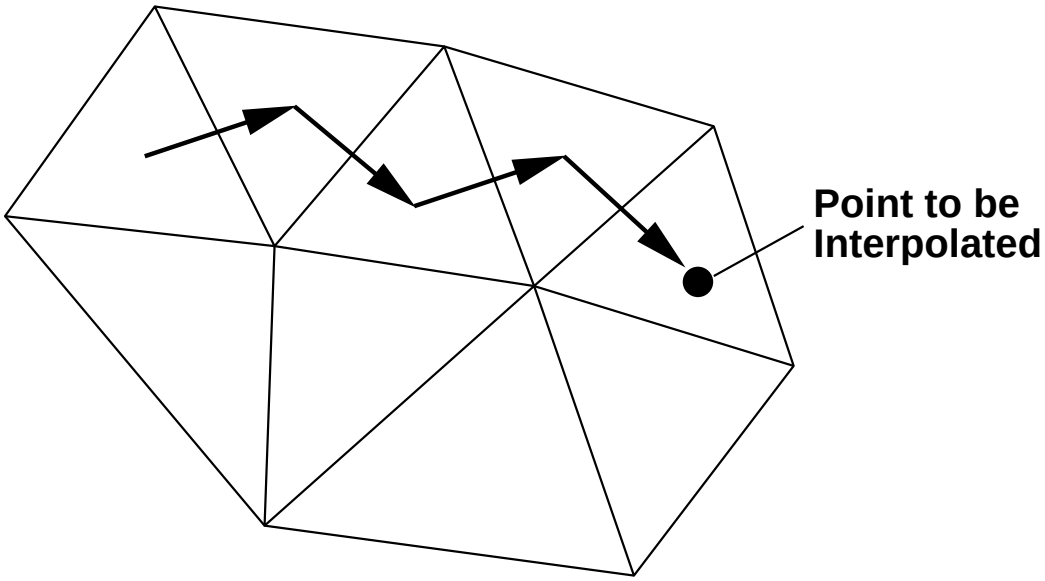
Then:

- Get Close Points of Known Grid From Octree
- Check Elements Surr. Close Points

Remarks:

- Fastest N-Time Start Algorithm
- $\Rightarrow$ Fastest Algorithm for Points

NEIGHBOUR-TO-NEIGHBOUR



Neighbour-to-Neighbour Search

NEIGHBOUR-TO-NEIGHBOUR

Given:

- Elements and Points of Known Grid
- Point of Unknown Grid
- Other Info:
  - List of Neighbour Elements of Known Grid
  - Start-Element in Neighbourhood `IESTA`

Then:

N.1 Compute Shape-Functions for `IESTA`

N.2 If Not In `IESTA`:
- Set `IESTA` To Neighbour Amsociated With $min(N^i)$;
- Goto N.1

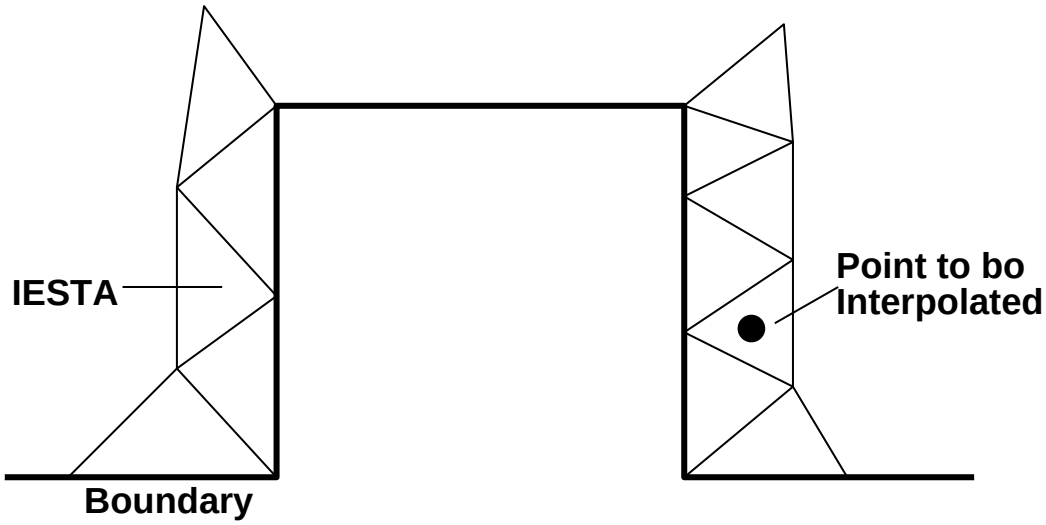
Endif

Remarks:

- Fastest Algorithm for Known Vicinity
- $\Rightarrow$ Fastest Algorithm for Grid-Grid Transfer

NEIGHBOUR-TO-NEIGHBOUR

Given:



Internal Boundaries

ADVANCING FRONT VICINITY ALGORITHM

Idea: If Complete Mesh to Mesh Interpolation Desired:

Advance Interpolation Front Using Fast Neighbour Search

Given:

- Elements and Points of Known Grid
- Elements and Points of Unknown Grid
- Other Info:
  - List of Neighbour Elements of Known Grid
  - List of Elements Surr. Points for Unknown Grid

Then:

A.1 Mark Elements and Points of Unknown Grid as Untouched

A.2 Initialize List of Front Points for Unknown Grid

A.3 Select Next Non-Interpolated Point From Front `IPUNK`

A.4 Obtain Starting Element `IESTA` in Known Grid

A.5 Attempt Nearest Neighbour Search for `NTRY` Attempts;

- `IF` Unsuccessful: Use Brute Force
- `IF` Unsuccessful: Stop or Skip

$\Rightarrow$ `IEEND`

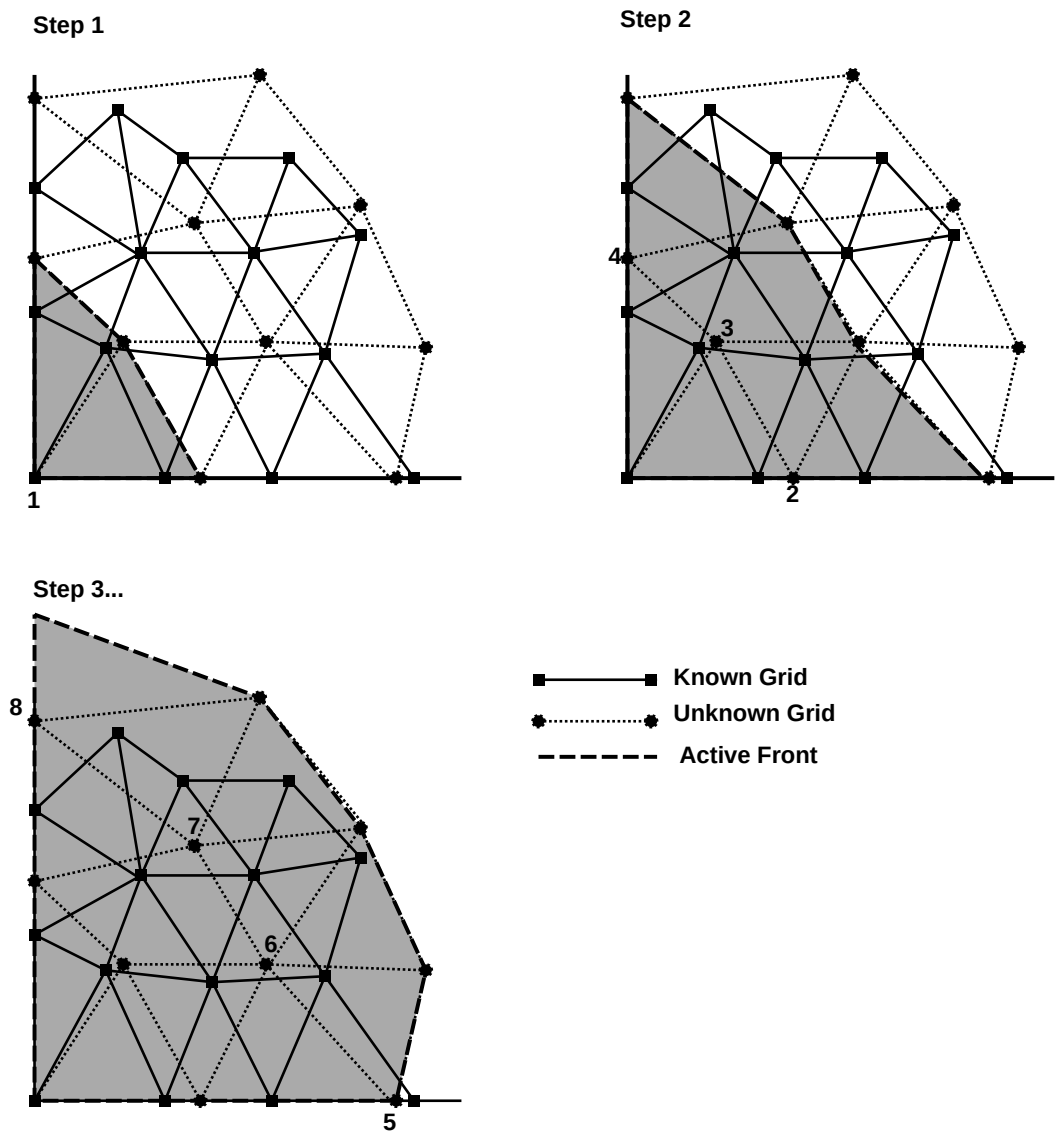
A.6 Store Shape-Functions and Host Elements

A.7 Loop Over Elements Surrounding `IPUNK`:

- `IF`: Element Has Not Been Marked:
  - Mark the Element
  - Loop Over the Points of This Element:
  - `IF`: the Point Has Not Been Marked:
    - Store `IEEND` as Starting Element For This Point;
    - Include This Point In Front;
  - `ENDIF`
- `ENDIF`

A.9 Mark Point `IPUNK` as Interpolated and `GOTO` A.3

ADVANCING FRONT VICINITY ALGORITHM



Advancing-Front Vicinity Search

IMPROVEMENTS

- Inside-Out Interpolation
    - Prefer Interior Points
- Vectorization
- Layering of Brute Force Search
    - Octree
    - Brute Force With Boundary Elements
    - Brute Force With Complete Mesh
- Concavity Measures

VECTORIZATION (1)

Key Ideas

- Interpolate Many Points At the Same Time
- Advance in Layers
- Work on 'Remaining List of Points'
- Refresh List When Depleted

VECTORIZATION (2)

V.0 Set Remaining Number of Points `NPREM=NPFRT`

V.1 Perform Interpolation Checks in Vector Mode for `NPREM`

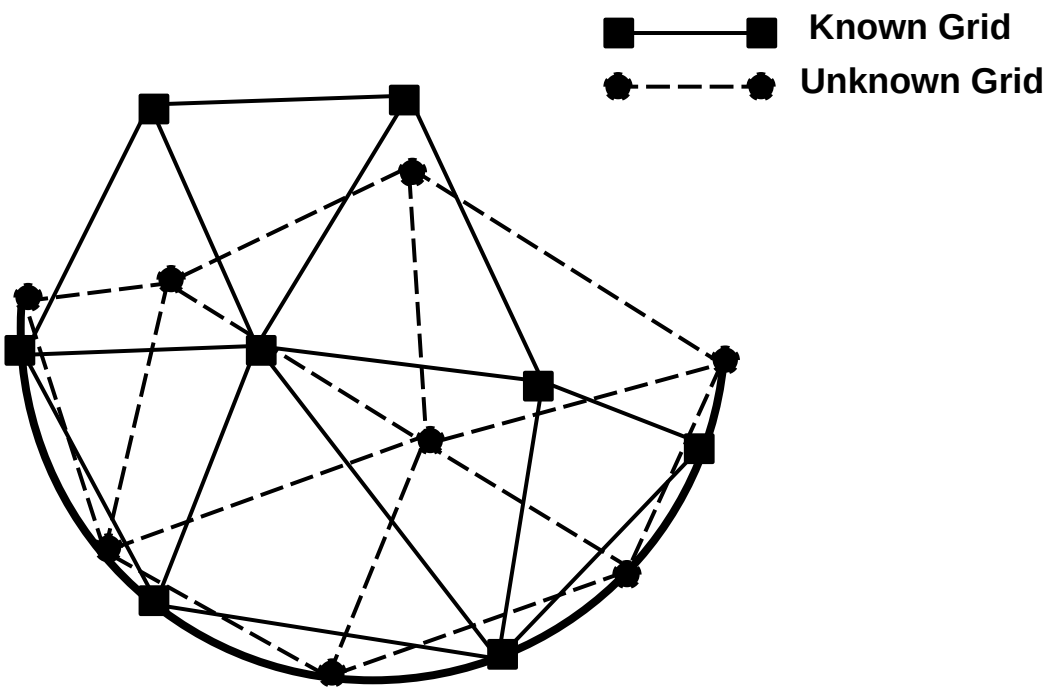
V.2 Write the `NPNXT` Points with No Host Elements into `LPCUR(1:NPNXT)`;
If `NPNXT=0`: Stop.

V.3 Write the `NPREM-NPNXT` Points with Host Elements into `LPCUR(NPNXT+1:NPREM)`

V.4 Reorder Arrays with `LPCUR`
$\Rightarrow$ Remaining Points in `1:NPNXT`

V.5 Set `NPREM=NPNXT` and go to V.1.

CONCAVE BOUNDARIES



Problems at Concave Boundaries

MEASURING CONCAVITY

Why: Concave Surfaces $\Rightarrow O(N_b^2)$-Search

Idea: Measure Concavity and Proximity

- Measure Used: Visibility of Neighbouring Faces

$$d = \alpha |min(0, \mathbf{n} \cdot (\mathbf{x}_0 - \mathbf{x}_i))| \quad , \quad 0.5 < \alpha < 1.5$$

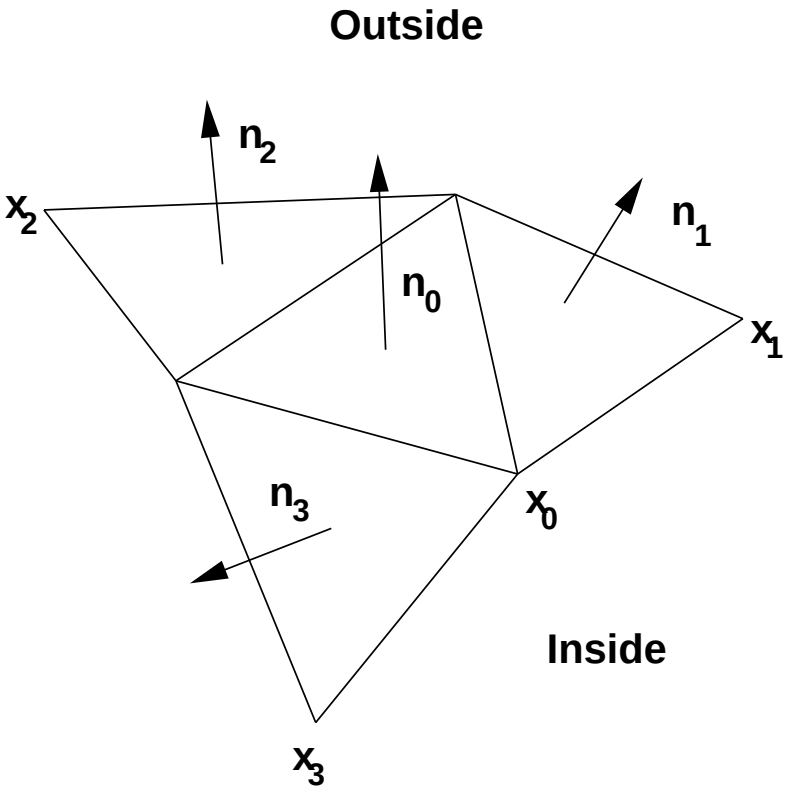
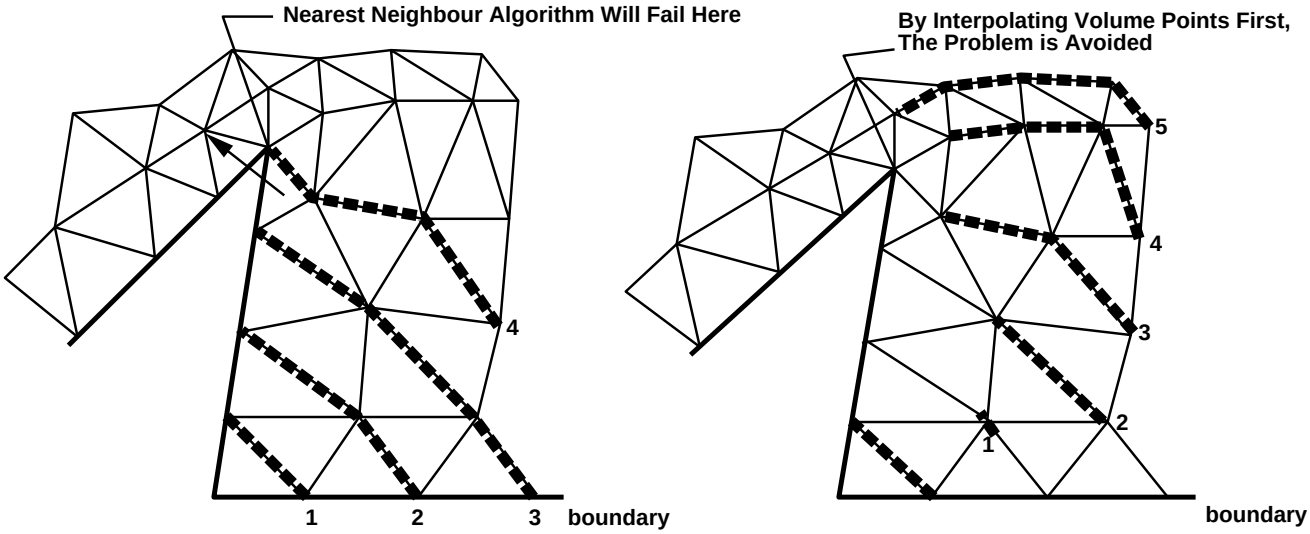


Figure 12.9 Measuring Concavity

INSIDE-OUT INTERPOLATION



Inside-Out Interpolation

Table 1 Interpolation Timings

Case	NELEM$_1$	NELEM$_2$	# BFS	CPU-scalar	CPU-vector	Sp
Cube$_1$	34,661	30,801	0	0.1399	0.0283	4.9
Cube$_2$	34,661	160,355	0	0.5360	0.1104	4.8
Train$_1$	180,670	243,068	31	1.1290	0.3405	3.3
Train$_2$	180,670	243,068	0	0.9905	0.2020	4.9