INPUT OF DESIRED ELEMENT SIZE AND SHAPE

Why:

- Try to Minimize CPU/Memory Requirements

Required:

- Maximum Flexibility for Minimum Input
- Automation Where Possible

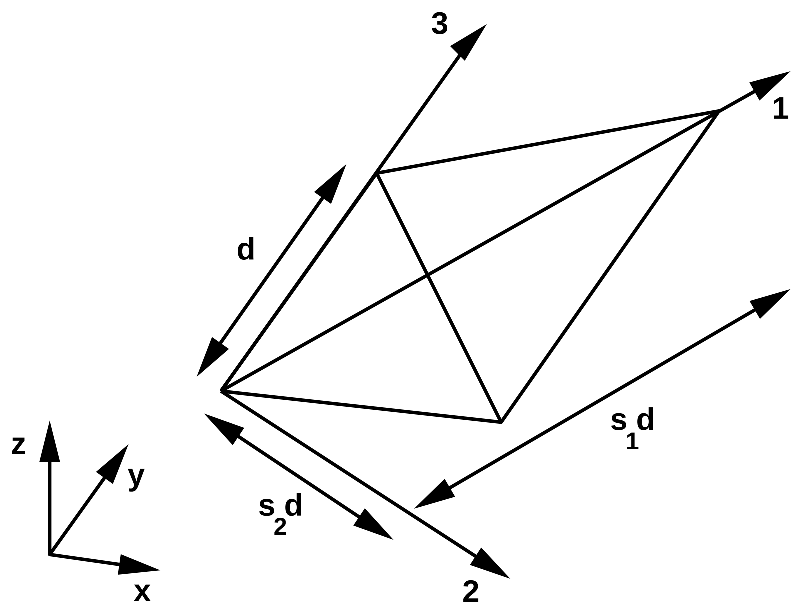
Idea: Combine

- Background Grid
- Sources
- Element Size Attached to CAD Data

BACKGROUND GRID (1)

Key Ideas:

- Define a Coarse Grid that Encompasses the Object to be Gridded
- At the Point: Define Element Size and Shape
- During Generation: Interpolate Linearly from Background Grid



Definition of Element Size and Shape

BACKGROUND GRID (2)

Pros:

- Easy
- Fast Search Possible
- Essential for Adaptive Remeshing

Cons:

- High Manual Input for Complex Geometries

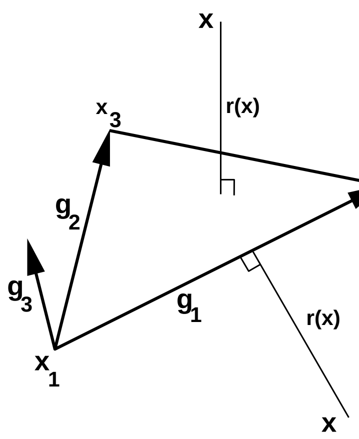
Best For:

- Adaptive Remeshing

SURFACE SOURCES (1)

Why:

- Better Than Many Line Sources
- Collapsable to Line/Point
- Use Linear Triangle for Speed



General Function:

$$\delta(\mathbf{x}) = \delta(\xi(\mathbf{x})) \quad ; \quad \xi = max(0, \frac{r(\mathbf{x}) - r_0}{r_s}) \qquad (1)$$

$r(\mathbf{x})$: Shortest Distance from Source to $\mathbf{x}$

r_0: Zone of Constant Element Size Close to Source

r_s: Scaling Length

- For Speed and Generality:

$$\delta(\xi) = a_0 + a_1\xi + a_2\xi^2 \qquad (2)$$

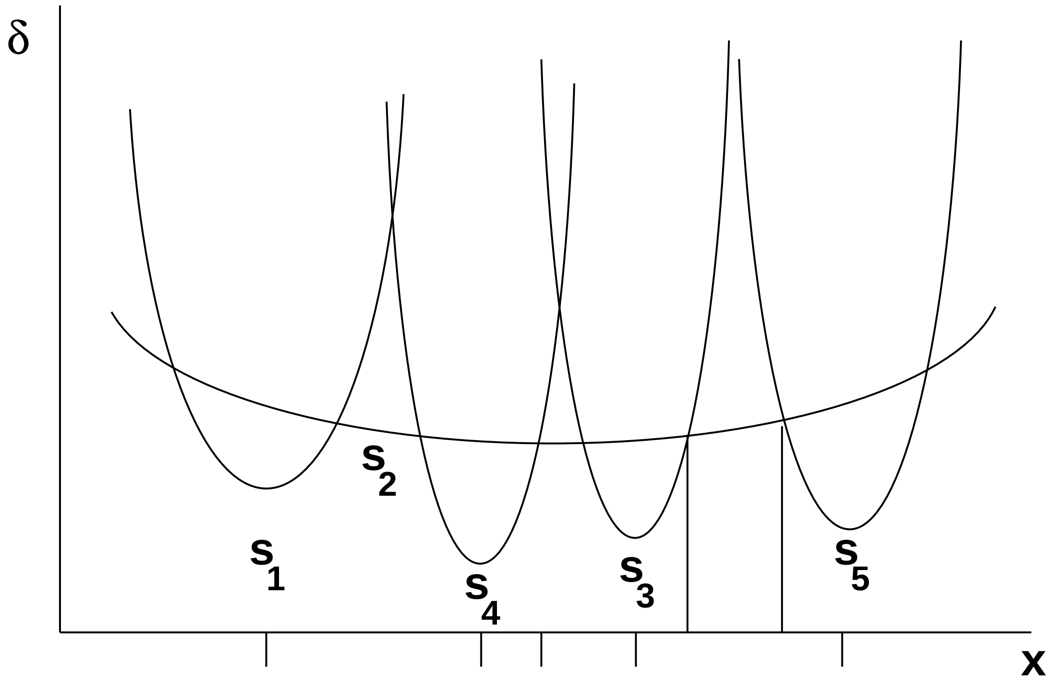
SURFACE SOURCES (2)

<u>Pros</u>:

- Fast Input
- Input Related to CAD in Most Cases
- Attachable to Bodies that Move
- Relatively Low CPU Overhead
- Vectorizable

<u>Cons</u>:

- Locality Filter Difficult
- ⇒ CPU Intensive for Many Sources



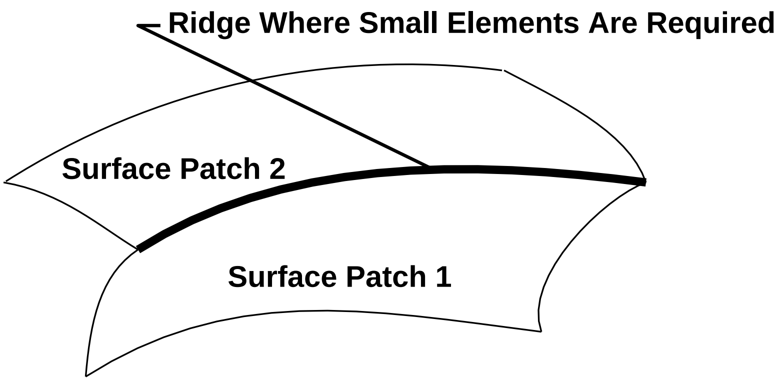
Surface With Smallest Element Size

SIZE ATTACHED TO CAD-DATA (1)

Idea: Attach Element Size to Point/Line/Surface

Pros:

- Ease of Input
- Attachable to Bodies that Move
- Flying Chunks (CSD Link)
- No CPU Overhead



SIZE ATTACHED TO CAD-DATA (2)

Implementation:

- Define Point-Size Parameter δ (`DISPO(1:NPOIN)`)
- For Any Face:

$$\delta_f = c_{in} \cdot min(\delta_1, \delta_2, \delta_3) \quad (3a)$$

$$\delta = min(\delta_{bg}, \delta_{bs}, \delta_f) \quad (3b)$$

$\Rightarrow$

- Layers of Increasing Size

- Caution: Can Create Incompatibilities !

AUTOMATIC GRIDDING (1)

Why: Very Complex Geometries

Options:

- Background Grid
  - General
  - Fast ($O(1)$)
  - Tedious
- Background Sources
  - General
  - Fast ($O(N)$)
  - Tedious for Very Complex Geometries
- Prescribed Point/Line/Surface/Domain Sizes
  - Easy
  - Geometry-Size Link
  - Possible Incompatibility

AUTOMATIC GRIDDING (2)

Problems Not Addressed:

- Geometric Adaptation
- Geometrical Stiffness
- Smooth Transition
- General Deforming Bodies

AUTOMATIC GRIDDING (3)

General Algorithm:

- Given: Background Grid
- Determine Point-Sizes from Line-Sizes
    - Possible Smoothing
- Correct/Refine Background Grid
- Generate Sides
- Correct/Refine Background Grid
- Generate Faces
- Correct/Refine Background Grid

BACKGROUND GRID ADJUSTMENT

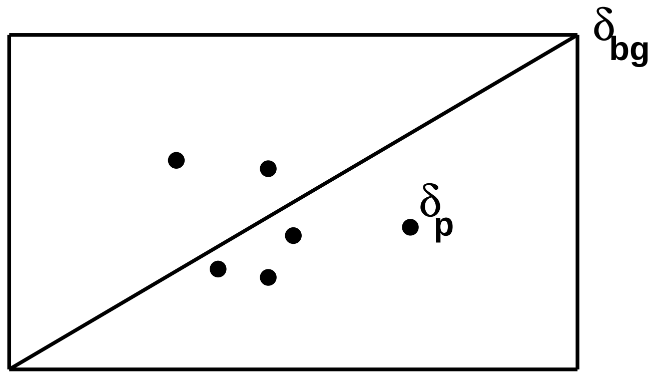
Given: For Each Point

- δ_p
- Background Grid Host Element

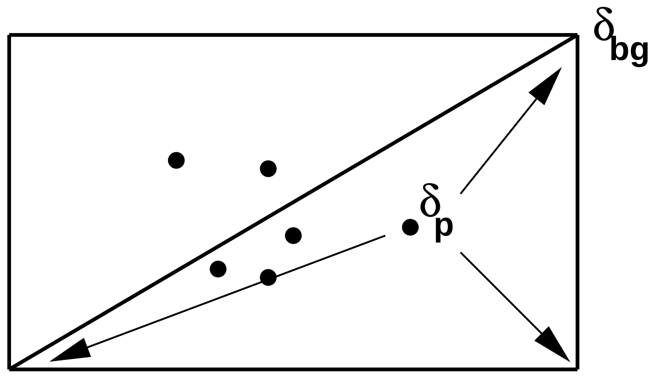
Then: For Each Point

- Get Host Element $\Rightarrow$ `IELBG`
- `DO`: Loop Over the Element Nodes
  - Get Background Grid Point and Distance-Parameter
  - $\Rightarrow \delta_{ibg}$
  - Get Distance to Point: $\rho = |\mathbf{x}_p - \mathbf{x}_{ibg}|$
  - Determine Distance-Parameter Assuming Geometric Increase
  - $\Rightarrow \delta'_{ibg}$
  - Set: $\delta_{ibg} = min(\delta_{ibg}, \delta'_{ibg})$
- `ENDDO`

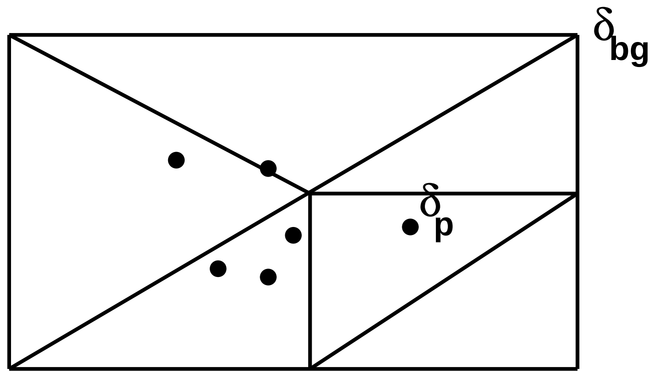
a) Generated Surface Points



b) Adjust B.G. Element Size to Surface Point Size



c) Refine B.G. As Required



BACKGROUND GRID ADAPTATION

Given: For Each Point

- δ_p
- Background Grid Host Element

Then: For Each Point

- Get $\delta_{min} = min(\delta_1, \delta_2, \delta_3, \delta_4)$
- `IF`: $\delta_{min} > c_t \delta_p \Rightarrow$ Refine