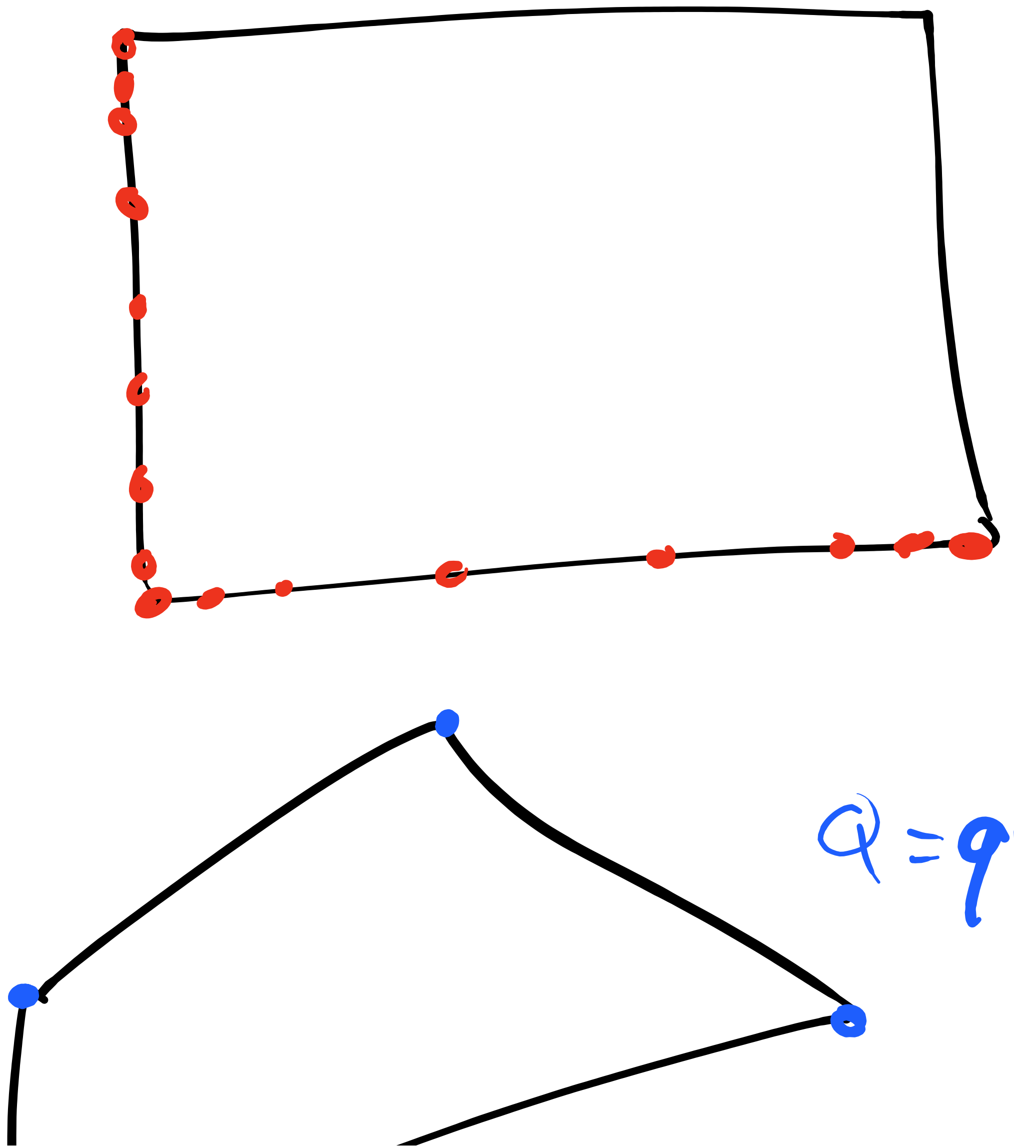
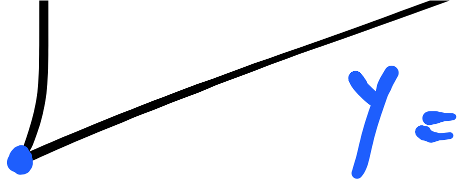
# Jonna Multishape

Friday, 14 August 2020 10:47



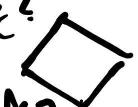
q = quad

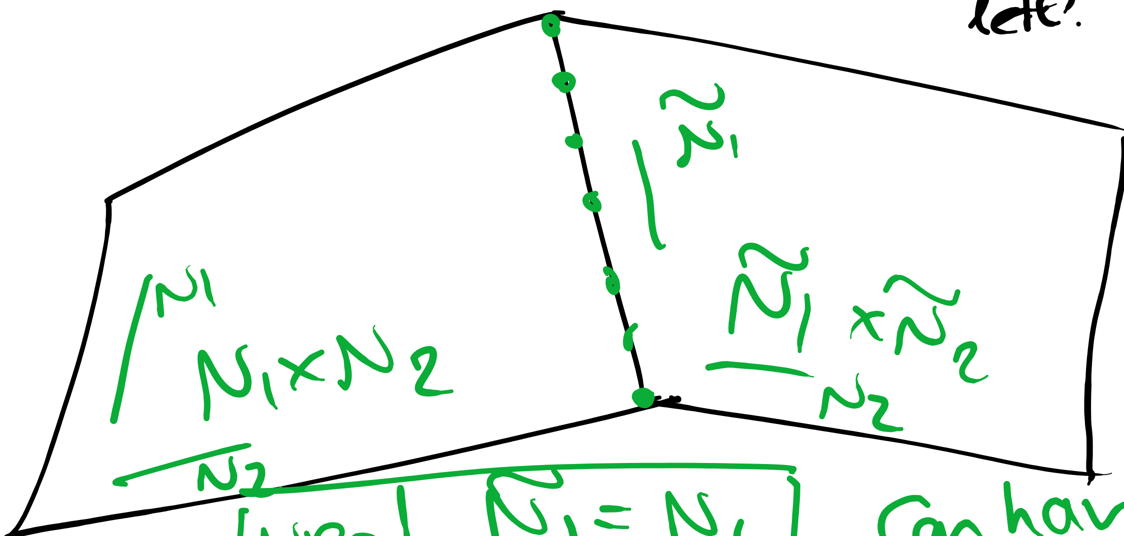


$Y$ = list of corner pts

$$Y = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

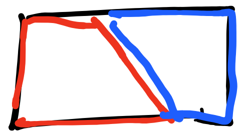
claim:
you can give
$(x_i, y_i)$ in any
order around $\phi$

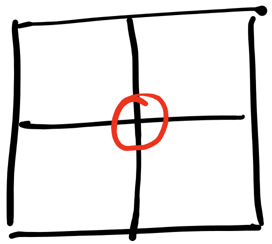


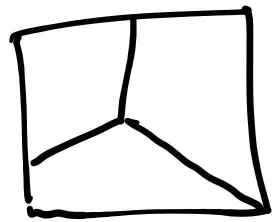



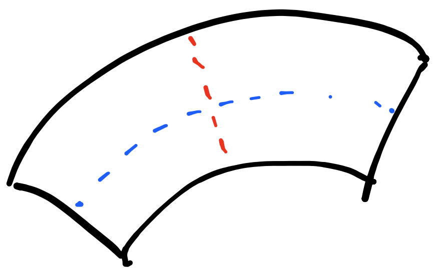

$\boxed{\text{Need } \tilde{N}_1 = N_1}$ Can have

$N_2 \neq N_2$

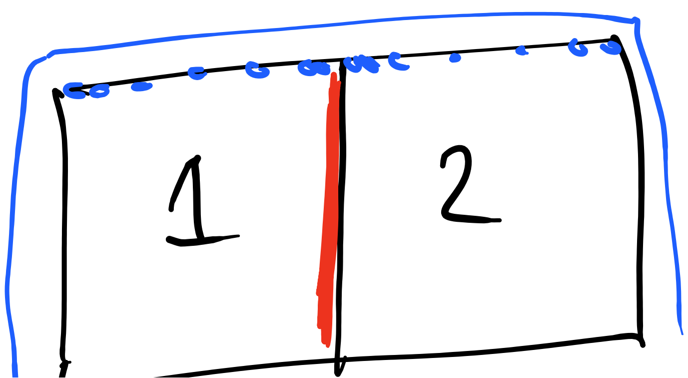





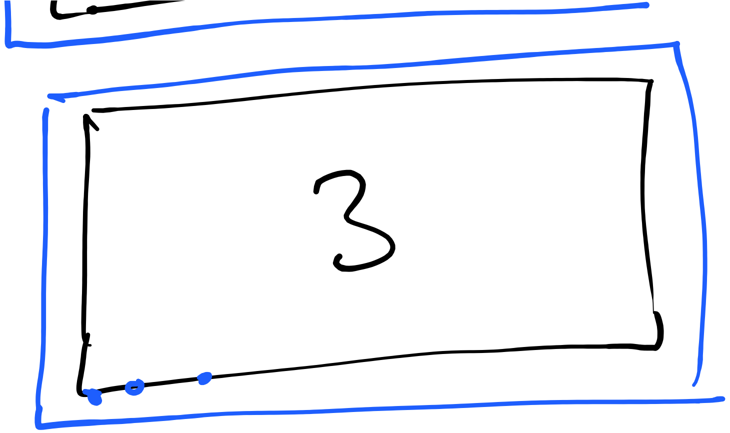




POLAR!




Pts: 2x(20x20)




(20x20)

40x20 ?

BC for MultiShape

BCs

2 options ① BCmatch

② BCnoflux (capitals?)

BCs is (nShapes x nShapes)

structure

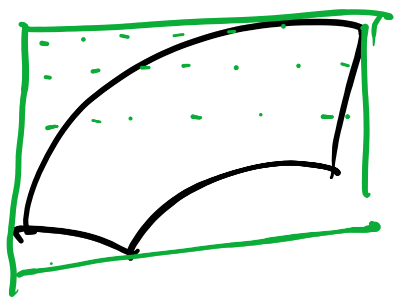
fields: function (either ① or ②)

BCs(i,j) is BC between shapes

$i$ & $j$ (only need to specify for $i<j$)

bounds: plotting/comparing on a uniform grid

(don't need for a standard computation)



MS = MultiShape(shapes)

makes a MultiShape

---

MS.Shapes(1).Shape

first quad

geom for " geom

- pts : if shape(i) has pts

(y1kv) $\rightarrow y_1^{(i)}, y_2^{(i)}$

ms pts : $\sum N_1^{(i)} \times N_2^{(i)}$ $\begin{bmatrix} y_1^{(1)} \\ \vdots \\ y_1^{(nShapes)} \end{bmatrix} \begin{bmatrix} y_2^{(i)} \\ \vdots \\ y_2^{(nShapes)} \end{bmatrix}$

MS. pts. y1_kv

- ShapeMask (PtsMask)

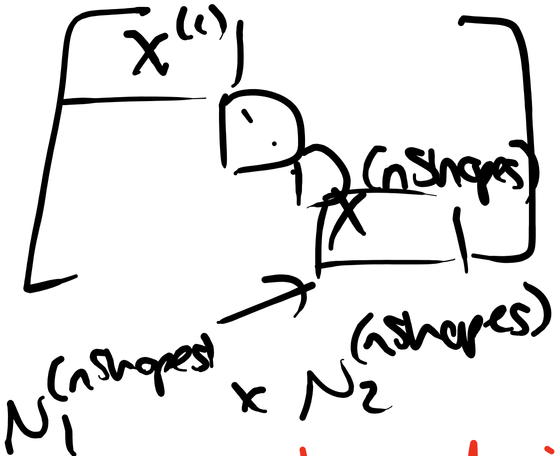
MS. pts. y1_kv (MS. Shapes (iShape)

MS.Pts.  • PtsMask )

$= $ Pts for Shape iShape

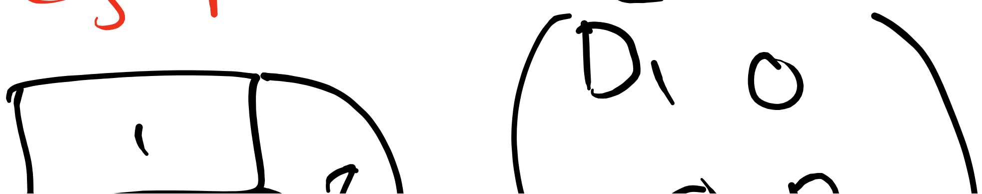
• Diff  $\swarrow D_1$  $\swarrow N_1^{(1)} \times N_1^{(2)}$

MS. Diff. X




Note: everything is stored in the coordinate system of the Shape

eg polar  $\leftarrow$ cart



$$\left( \begin{matrix} 0 & v_2 \end{matrix} \right)$$

polar

- Integration

[Int_1, ..., Int nShapes]

- Convolution (same as shape)
  (dense - global)
- Interpolation onto $\{z_i\}$ $\in \mathbb{R}^2$

challenge: need to know which

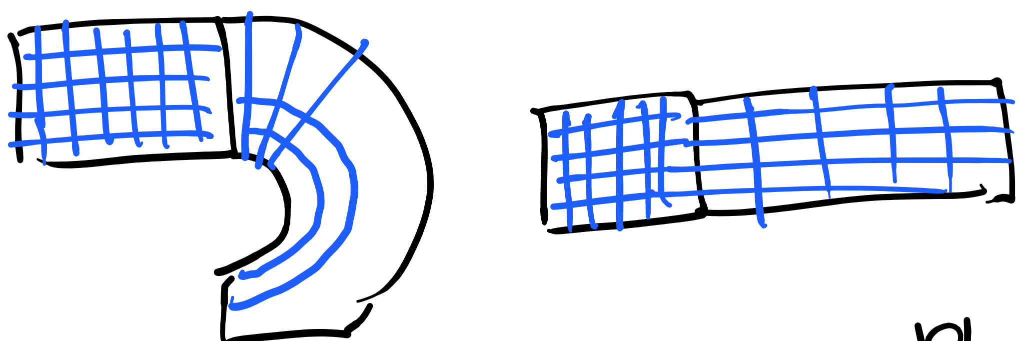
(*) shape(s) $z_i$ lies in

In shape interpolation is really done on $[-1,1] \times [-1,1]$ (ie in same space)

Compute Interpolation Matrices

- Takes a grid in [-1, 1]
- Computes interpolation onto this grid in comp-space for each shape

from native Cheb pts.

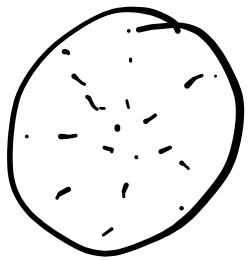


- Compute Interpolation Matrices Phys pts in physical space (y's) see (X)

1) get comp space for shape (x)

( = [-1,1] a.e.)

2) if polar convert y to polar

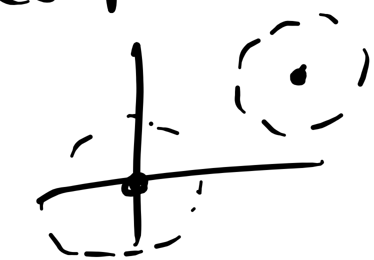


+ origin

← pt at zero

$\in [-1,1] \times (0,1]$ comp space

cart2pol $(y_1, y_2)$

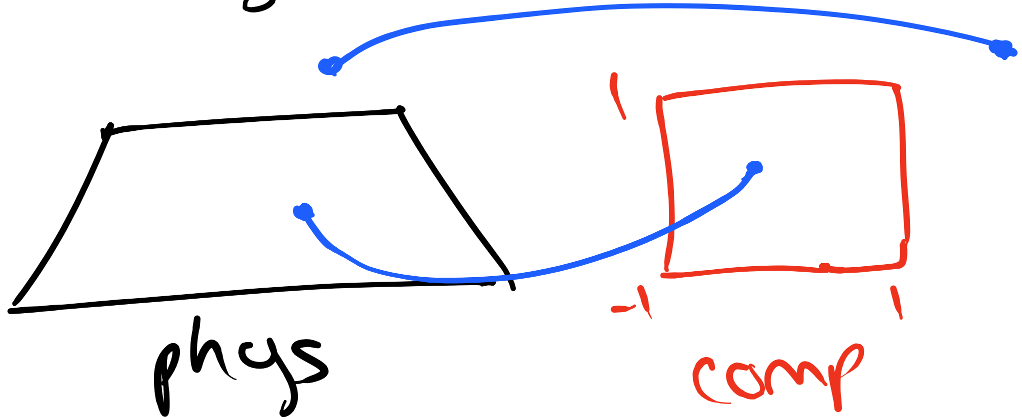


3) compspace $(y) = X$

↑ from shape

$X \in [-1,1] \times [-1,1]$ then

If $x \in (-1, 1)$ ...
$y$ lives in that shape




Vectors can be tricky

on a shape a vector is

$$\begin{array}{c} v_1 \\ \\ v_2 \end{array}\left( \begin{array}{c|c} - & - \end{array} \right) \quad 2N_1 \times N_2 \qquad \underline{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$\varphi = \begin{bmatrix} \varphi \\ \varphi \end{bmatrix}$$

MakeVector$(\varphi, \varphi)$

$\nabla V_2$

$.* (\mathrm{Grad} \;.*\; V_2)$

$= \cancel{\begin{pmatrix} \rho \\ \rho \end{pmatrix} .* DV_2}$

Make Vector
Split Vector
} use MS. y1mask, y2mask

y1mask ~ $\begin{bmatrix} \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix}$

$N_1^{(1)} \times N_2^{(1)}$

$N_1^{(1)} \times N_2^{(1)}$

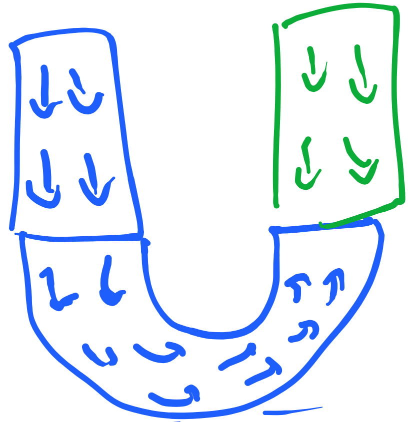
$\begin{pmatrix} V_1^{(1)} \\ V_2^{(1)} \end{pmatrix}$

$2 \sum N_1^{(i)} \times N_2^{(i)}$

grad $\begin{bmatrix} \mathrm{grad1} & & 0 \\ & \ddots & \end{bmatrix}$

[ 0 . gradnShapes ]

complete aside:



$v_1 = 0$ (vectors of right length)

$v_2 = -1$

$v = \text{MakeVector}(v_1, v_2)$

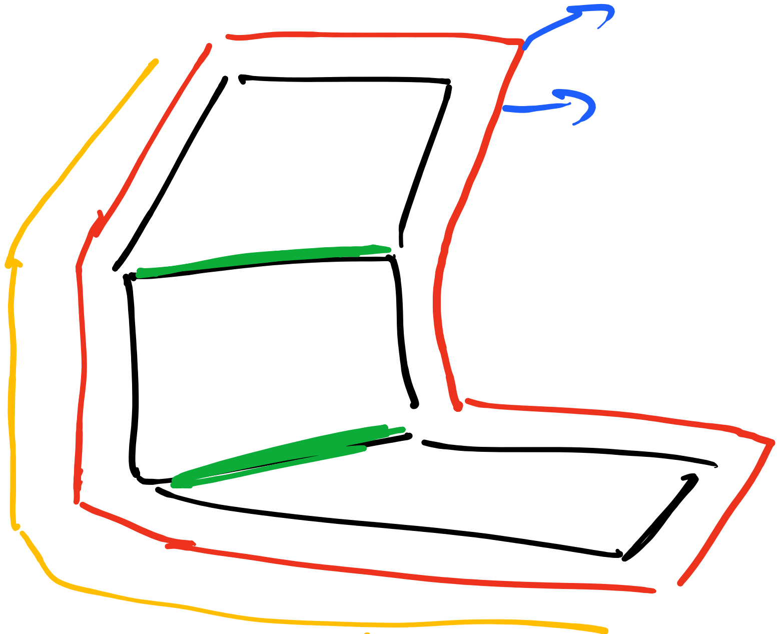
Indexing:

bound

normal

ill-defined at best

"left, right, top, bottom,
radial 1, radial 2, outer, inR"

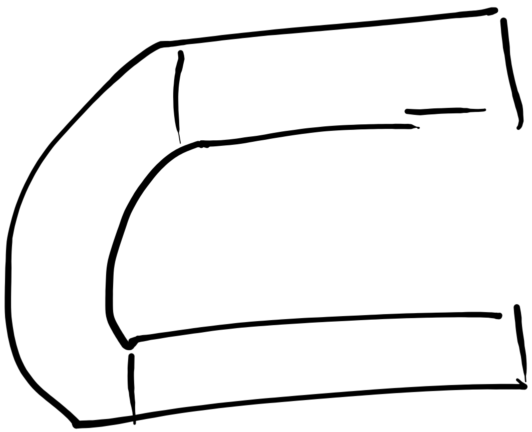


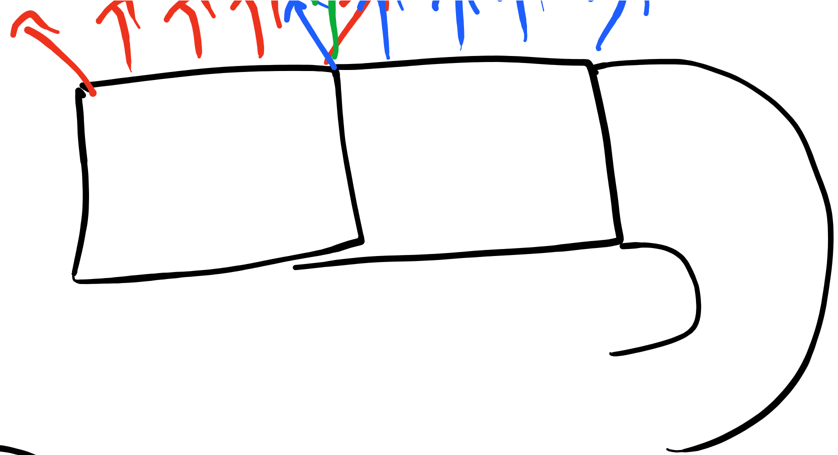
left

bound
intersections
(i, j)
normals

↑
reliable

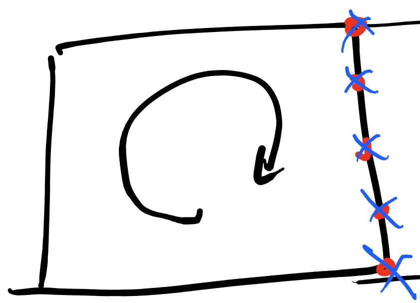
left





flip

$X = \cdot\ ?$



$[\ \ ] \stackrel{?}{=} [\ \ ]$