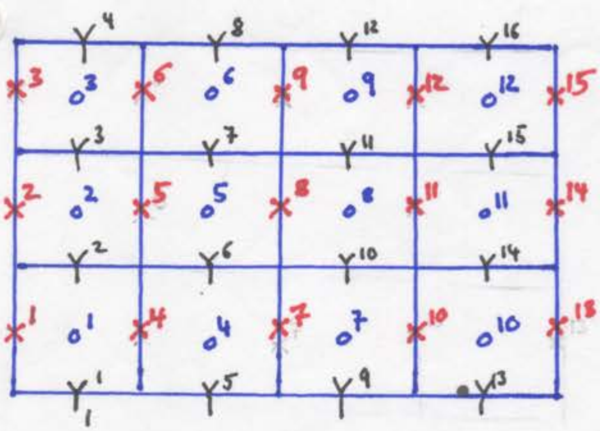
Discrete operators in 2D

Staggered grid:



$N_x = 4 \quad N_y = 3 \quad N = N_x \cdot N_y = 12$

faces in x-dir: $N_{fx} = (N_x+1)\, N_y = 5 \cdot 3 = 15$

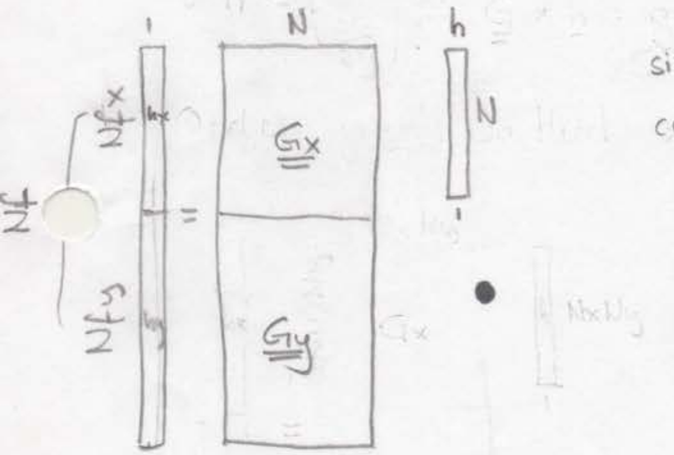
faces in y-dir: $N_{fy} = N_x\,(N_y+1) = 4 \cdot 4 = 16$

total # faces: $N_f = N_{fx} + N_{fy}$
$= (N_x+1)\,N_y + N_x\,(N_y+1)$

Gradient in 2D:

$$\nabla h = \begin{pmatrix} \frac{\partial h}{\partial x} \\ \frac{\partial h}{\partial y} \end{pmatrix} \approx \underline{\underline{G}} * \underline{h} = \begin{bmatrix} \underline{\underline{G_x}} \\ \underline{\underline{G_y}} \end{bmatrix} \underline{h}$$

$\Rightarrow$ order $\underline{\underline{G}}$ so that we first compute all x derivatives then all y derivatives.



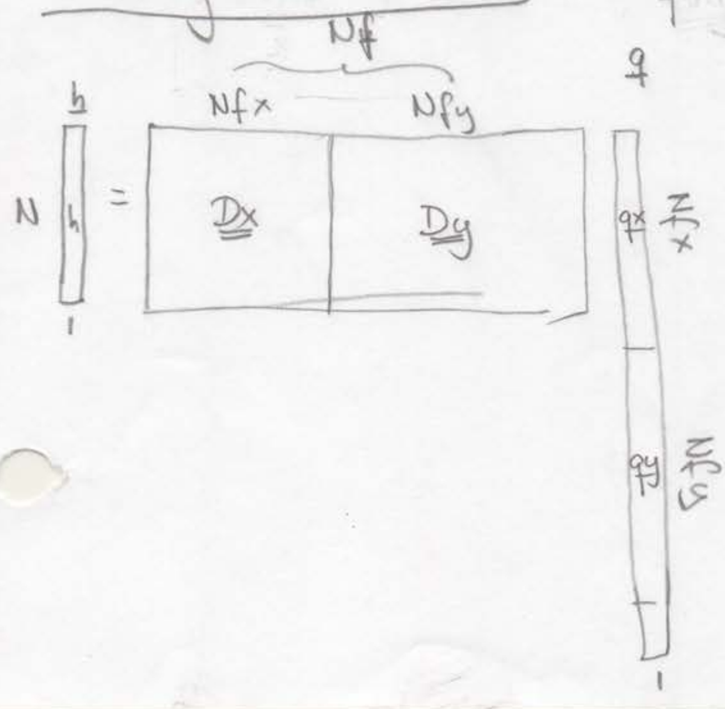
size of $\underline{\underline{G}}$ is N_f by N and it is composed of two submatrices $\underline{\underline{G_x}}$ & $\underline{\underline{G_y}}$

$\underline{\underline{G_x}}$ is N_{fx} by N

$\underline{\underline{G_y}}$ is N_{fy} by N

Divergence in 2D:

$$\nabla \cdot q = \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} \approx \underline{\underline{D}} * q = \underline{\underline{D_x}} * \underline{q_x} + \underline{\underline{D_y}} * \underline{q_y}$$



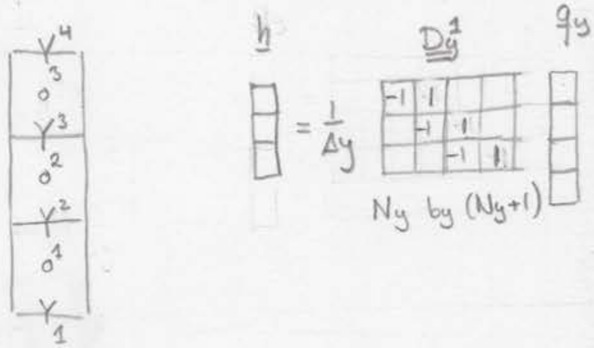
size of $\underline{\underline{D}}$ is N by N_f and it is composed of two submatrices

$\underline{\underline{D_x}}$ is N by N_{fx}

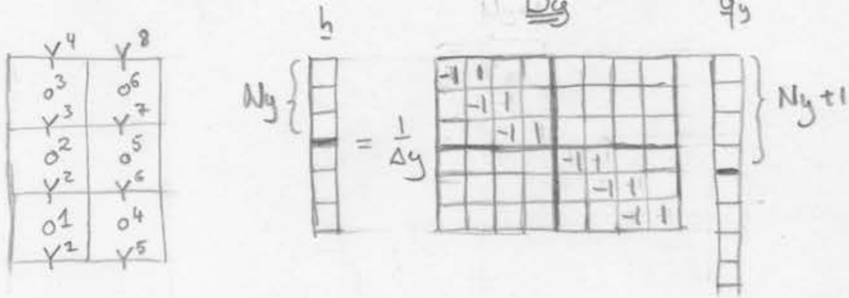
$\underline{\underline{D_y}}$ is N by N_{fy}

Divergence matrices

Start with $\underline{Dy}$ in 1D:



Suppose we add a second column of cells: $Nx=2$



$$\Rightarrow \quad \underline{\underline{Dy}}^2 = \begin{bmatrix} \underline{\underline{Dy}}^1 & 0 \\ 0 & \underline{\underline{Dy}}^1 \end{bmatrix}$$

$\underline{\underline{Dy}}^2$ is a block matrix with 2 by 2 blocks of size 3 by 4. Diagonal blocks are $\underline{\underline{Dy}}^1$!

Suppose we add a third column of cells: $Nx=3$

$$\underline{\underline{Dy}}^2 = \begin{bmatrix} \underline{\underline{Dy}}^1 & 0 & 0 \\ 0 & \underline{\underline{Dy}}^1 & 0 \\ 0 & 0 & \underline{\underline{Dy}}^1 \end{bmatrix}$$

In general:

$\underline{\underline{Dy}}^2$ is a block matrix with Nx by Nx blocks of size Ny by $(Ny+1)$. Diagonal blocks are $\underline{\underline{Dy}}^1$ and others are zero.

Tensor product construction of $\underline{\underline{D_y^2}}$

The discrete 2D operators can easily and efficiently be assembled using tensor/Kronecker products.

Definition:

If $\underline{\underline{A}}$ is an $m \times n$ matrix and $\underline{\underline{B}}$ is a $p \times q$ matrix, then the Kronecker product $\underline{\underline{A}} \otimes \underline{\underline{B}}$ is the $mp \times nq$ block matrix:

$$\underline{\underline{A}} \otimes \underline{\underline{B}} = \begin{bmatrix} a_{11}\underline{\underline{B}} & \cdots & a_{1n}\underline{\underline{B}} \\ \vdots & \ddots & \vdots \\ a_{m1}\underline{\underline{B}} & \cdots & a_{mn}\underline{\underline{B}} \end{bmatrix}$$

In tensor product notation:

$$\underline{\underline{D_y^2}} = \underline{\underline{I_x}} \otimes \underline{\underline{D_y^1}} = \begin{bmatrix} \underline{\underline{D_y^1}} & & & \\ & \underline{\underline{D_y^1}} & & \\ & & \underline{\underline{D_y^1}} & \\ & & & \ddots \;\; \underline{\underline{D_y^1}} \end{bmatrix}$$

where $\underline{\underline{I_x}}$ is the N_x by N_x identity

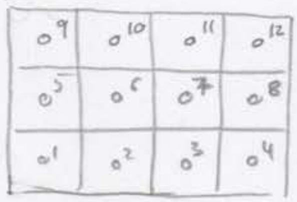
In Matlab the tensor product is obtained as

$$\underset{\substack{\uparrow \\ \text{2D operator}}}{\underline{\underline{D_y}}} = \text{kron}(\underline{\underline{I_x}}, \underset{\substack{\uparrow \\ \text{1D operator}}}{\underline{\underline{D_y}}});$$

How do we build $\underline{\underline{D_x}}$?

So what is $\underline{\underline{Dx}}^2$?

If the grid was ordered x-first:



$$\underline{\underline{Dx}}^1 = \frac{1}{\Delta x} \overbrace{\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix}}^{Nx+1} \Big\} Nx$$

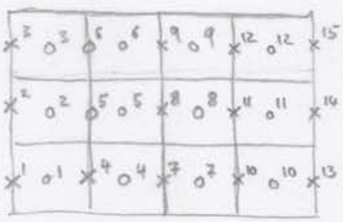
$$\underline{\underline{Dx}}^2 = \begin{bmatrix} \underline{\underline{Dx}}' & 0 & 0 \\ 0 & \underline{\underline{Dx}}' & 0 \\ 0 & 0 & \underline{\underline{Dx}}' \end{bmatrix}$$

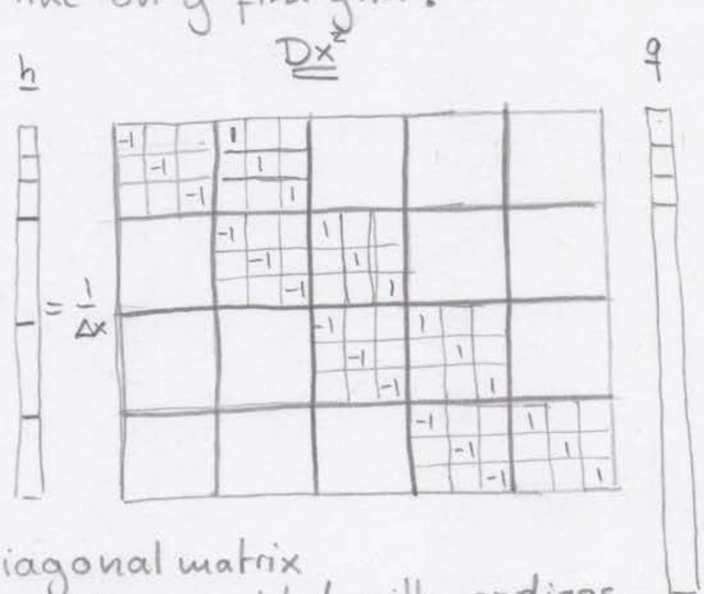
$\underline{\underline{Dx}}^2 = \underline{\underline{I}}_y \otimes \underline{\underline{Dx}}^1$

Matlab

$\underset{\substack{\uparrow \\ \text{2D op.}}}{\underline{\underline{Dx}}} = \text{kron}(I_y, \underset{\substack{\uparrow \\ \text{1D op}}}{Dx})$

What does $\underline{\underline{Dx}}^2$ look like on y-first grid?





$\Rightarrow$ $\underline{\underline{Dx}}^2$ is sparse diagonal matrix
In Matlab this could be assembled with spdiags

$\underline{\underline{Dx}}^2$ is also a block matrix built from Ny by Ny identities.

$$\underline{\underline{Dx}}^2 = \begin{bmatrix} -\underline{\underline{I}}_y & \underline{\underline{I}}_y & & & \\ & -\underline{\underline{I}}_y & \underline{\underline{I}}_y & & \\ & & -\underline{\underline{I}}_y & \underline{\underline{I}}_y & \\ & & & -\underline{\underline{I}}_y & \underline{\underline{I}}_y \end{bmatrix} = \underline{\underline{Dx}}^1 \otimes \underline{\underline{I}}_y$$

In Matlab this can be built $\underline{\underline{Dx}} = \text{krou}(\underline{Dx}, \underline{\underline{I}}_y)$

Discrete Gradient matrix

The $\underline{\underline{Gx}}$ and $\underline{\underline{Gy}}$ matices could be built using 1D matices and Kronecker products. Instead we use that fact that operators are adjoints.

$$\underline{\underline{G}} = - \underline{\underline{D}}^T$$

need to impose natural BC's. Set G on all boundary faces to zero:

dof-f-bnd = [dof-f-xmin; dof-f-xmax; dof-f-ymin; dof-f-ymax]

set corresponding rows to zero

G(dof-f-bnd, :) = 0;