Two Functional MDD's for the Price of One - Part 2

TODO add list of authors

November 6, 2019

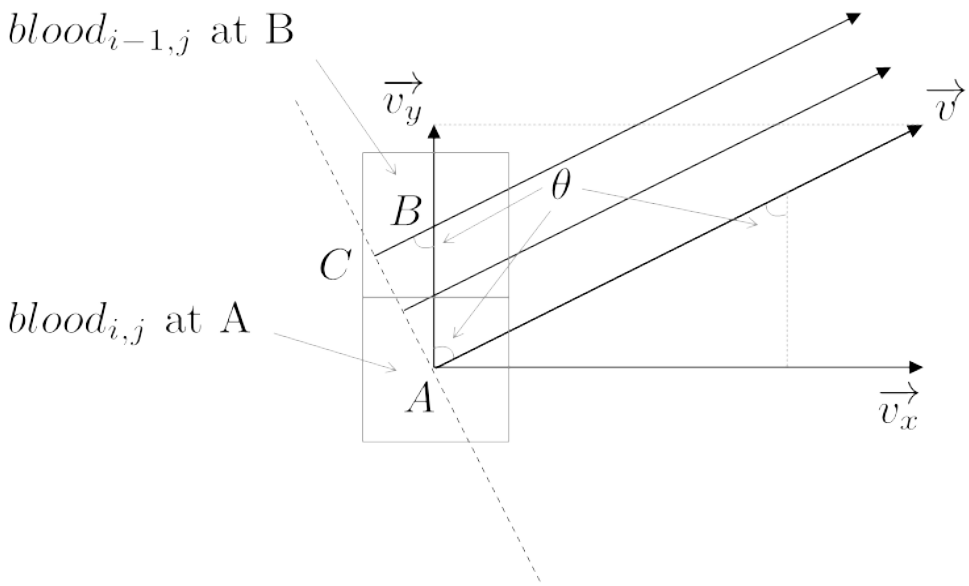
Outline

Symphony - Modeling Language for Non-Linear Optimization

- Models linear and non-linear optimization problems
- Simple declarative language
- Support for bounded parameters and constraint programming
- Generates performance oriented c code
- Solver Agnostic (plug into your solver of choice)

SYNTAX EXAMPLES

Sample Problem 1 - Velocity Problem



- MRI imaging problem dealing with blood flow
- Given vector field of blood flow: can we find how long each blood cell has been there?
- Do this by minimizing the flow over time (hence an optimization problem!)

Velocity Problem - Model Derivation

$$\Rightarrow t_{i-1,j} - t_{i,j} = \frac{CB}{|\overrightarrow{v}|} = \frac{AB\cos\theta}{\sqrt{v_x^2 + v_y^2}} = \frac{1\frac{v_y}{\sqrt{v_x^2+v_y^2}}}{\sqrt{v_x^2 + v_y^2}} = \frac{v_y}{v_x^2 + v_y^2} \tag{1}$$

$$\Rightarrow (t_{i-1,j} - t_{i,j})(v_x^2 + v_y^2) = v_y \tag{2}$$

$$\Leftrightarrow \Delta t_y (v_x^2 + v_y^2) = v_y \tag{3}$$

Velocity Problem - Optimization Model

$$\min_t \sum_{\text{pixels}} (\Delta t_x (v_x^2 + v_y^2) - v_x) * v_x^2)^2$$
$$+ \sum_{\text{pixels}} (\Delta t_y ((v_x^2 + x_y^2) - x_y) * x_y^2)^2$$

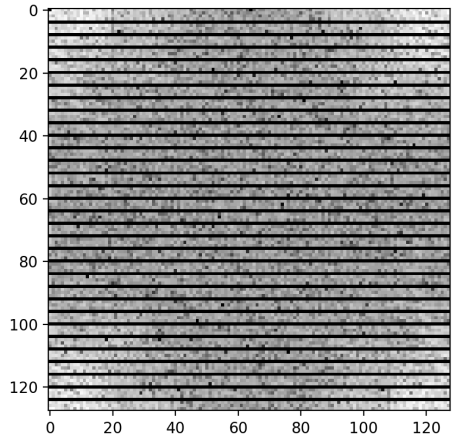
$v_{(x,y)}$ velocity in x,y direction

$t_{(x,y)}$ time in x,y direction

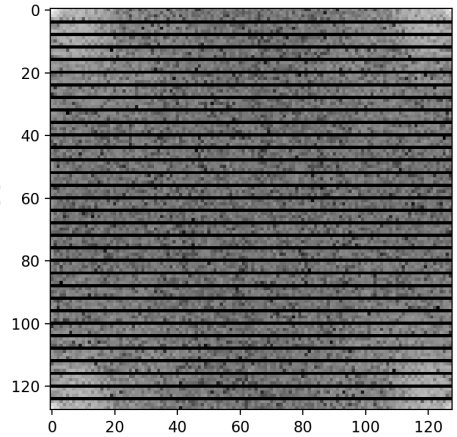
Brain Problem - 1

Data: real part (re) and imag part (im) of image's k-space received by the MRI.
Black spots are where the signal is lost.

re:

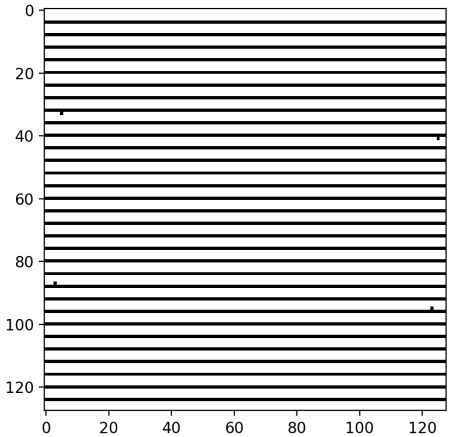


im:



Apply a threshold: signal = abs(re) $>$ 0.5 to get a matrix of where the signal is received.
signal[i][j] = 1 if there is signal in this spot, 0 otherwise

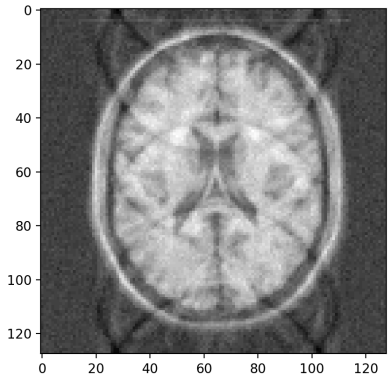
signal:



Brain Problem - 2

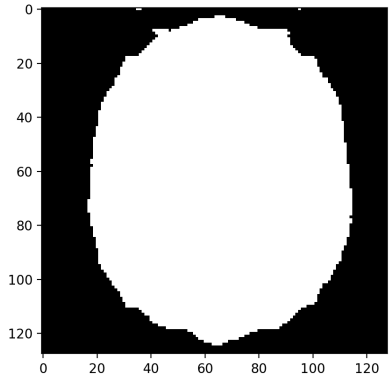
Naively reconstruct the image by taking inverse FFT, we get the naive image.

Naive image



median filter
threshold

median = normalize(ndimage.median_filter(img, 10))
head = median > 0.1



Determine the box constraint

x_lb and x_ub
- Inside head: pixel >= 0
- Outside head: -4 <= pixel <= 4

Play with generating HDF5

(Multi-Coil MRI / Constraint Programming)

Play with Scaling Factor

Play With L2-Norm / Hubar Penalty

Hashed Expression - Symphony's Backend

- Embedded Language in Haskell

References