

FrameLab: Development Guide

September 21, 2014

1 Overall Design

FrameLab 1.0 is designed to have an object-oriented, user friendly scripting interface with compute intensive routines written in compiled languages such as C and CUDA/C. The current scripting language is Matlab, using MEX as an interface mechanism to pull in compiled libraries. In the future we plan to implement Python/iPython as an alternative to Matlab, to keep the entire code open-source.

2 Matlab OOP System

The primary goal of Framelab is to solve general linear inverse problems of the sort

$$Au = f_0 + \eta$$

FrameLab supports models of the type:

$$\min R(u) \quad \text{such that} \quad F(u) < \epsilon$$

where $R(u)$ is a generic regularization term, typically of the form

$$R(u) = \|Wu\|_1$$

3 Compute Kernels

3.1 Computed Tomography

From [?]

Compiling MEX Libraries

```
mex -L"/usr/local/cuda/lib64" -lcudart -I"." Ax_fan_mf.cpp Ax_fan_mf_cpu_siddon.cpp  
Ax_fan_mf_cpu_new.cpp Ax_fan_mf_cpu_new_fb.cpp Ax_fan_mf_gpu_siddon.cu  
Ax_fan_mf_gpu_new.cu Ax_fan_mf_gpu_new_fb.cu find_area.cpp sort_alpha.cpp
```

Possible error message about invalid conversion from int to mxComplexity: change

```
plhs[0]=mxCreateNumericMatrix(nx*ny*nt,1,mxSINGLE_CLASS,0);
```

to

```
plhs[0]=mxCreateNumericMatrix(nx*ny*nt,1,mxSINGLE_CLASS,mxREAL);
```

in any mex interface files

4 Another Section