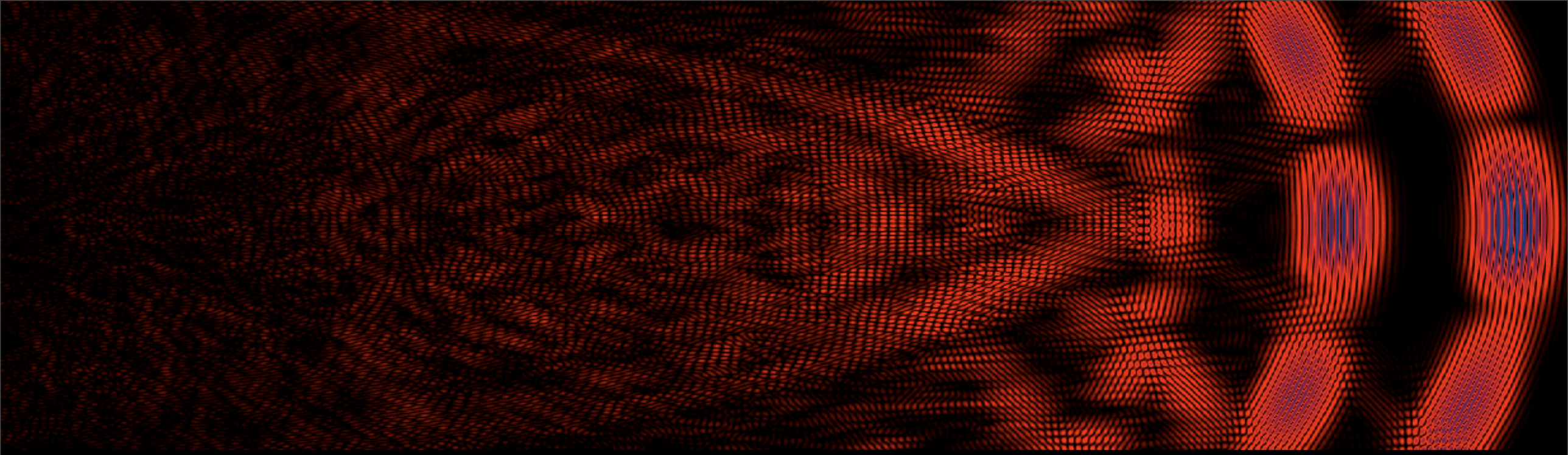


GPULib with IDL 8.0

Michael Galloy, Tech-X Corporation
michaelgalloy.com

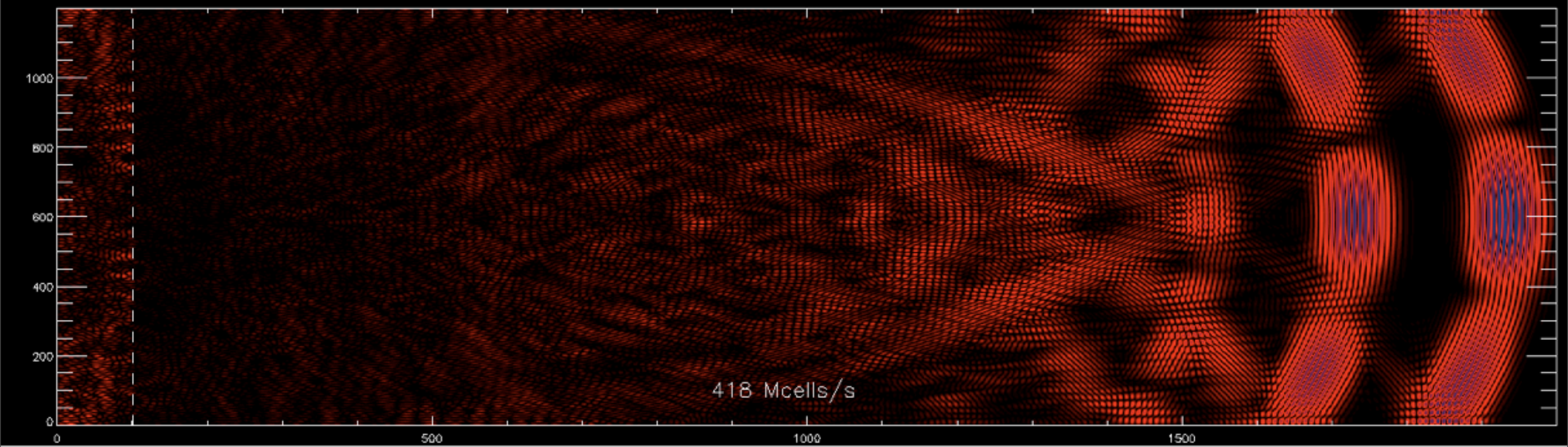
Work supported by NASA SBIR Phase II Grants #NNG06CA13C and #NNX09CA72C

Outline

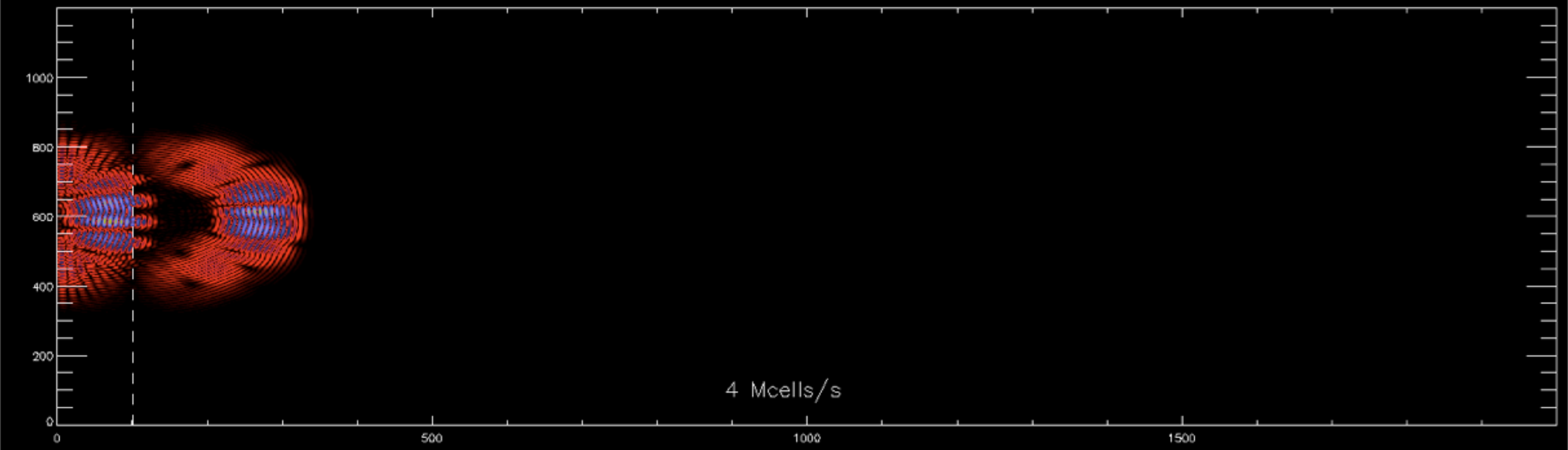
1. What is GPULib?
2. Using operator overloading with GPULib
3. Other new GPULib features
4. More operator overloading

X11 Applications Edit Window Help (1:11) May 10 2:42 pm mgalloy

Local Domain on Rank 0



Local Domain on Rank 0



```
tmp1 = gpuMake_array(…)
tmp2 = gpuMake_array(…)
rho = gpuMake_array(…)

gpuMult, x, x, tmp1
gpuMult, y, y, tmp2
gpuAdd, tmp1, tmp2, tmp1
gpuSqrt, tmp1, rho

gpuFree, tmp1
gpuFree, tmp2
```

```
rho = gpuSqrt(gpuAdd(gpuMult(x, x), $
                     gpuMult(y, y))
```

```
tmp1 = gpuMake_array(…)
tmp2 = gpuMake_array(…)
rho = gpuMake_array(…)

rho = gpuSqrt( $
        gpuAdd( $
          gpuMult(x, x, LHS=tmp1), $
          gpuMult(y, y, LHS=tmp2), $
          LHS=tmp1), $
        LHS=rho)

gpuFree, tmp1
gpuFree, tmp2
```

```
rho = gpuSqrt(x * x + y * y)
```

10000 iterations with 1000000 element arrays

CPU calculation:	144.8 secs
Procedure forms:	7.3 secs
Function forms:	15.4 secs
Function forms with LHS:	7.2 secs
Operator forms:	15.4 secs

CPU calculations performed on a 2.40GHz Core2 Duo
GPU calculations performed on a NVIDIA Tesla C1060

```
IDL> x = randomu(seed, 10)
IDL> x_gpu = gpuPutarr(x)
IDL> help, x, x_gpu
X               FLOAT     = Array[10]
X_GPU           GPUFLOAT  = Array[10]
IDL> print, x_gpu
      0.507024     0.966179    0.0294637
      0.638232     0.758752     0.102476
      0.405151     0.404657     0.151935
      0.785828
```

structures

↓

objects

```
function gpuvariable::_overloadPlus, left, right
  compile_opt strictarr

  return, gpuAdd(left, right)
end
```

Issues

- customers with various IDL versions IDL 6.4+
  - make your own `IDL_Object` class!
  - then your code will work pre- and post-IDL 8.0 (well, no operator overloading before 8.0, of course)
- . operator issues when not `self`

HDF5 classes

```
h = mg_h5(file_which('h5_test.h5'))
group = h['images']
d = h['2D int array']

e = group['eskimo']
plot, e[*, 400]
ct = group['eskimo_palette']
tvlct, transpose(ct[*])
tv, e[*], order=1
```

http://bit.ly/mg_h5_routines

RDL (Data Access Protocol in IDL)

```
url = 'http://wavelet.txcorp.com:8080/' $
      + 'opendap/data/hdf5/' $
      + 'a6_electrons_10.h5'
dap = txdap_new(url)
var = dap['group']
contour, var[*]
```

ENVI Atmospheric Radiative Transfer

- under development through NASA 2010 SBIR NNX10CB46C;
  beta version expected mid 2011
  fillmore@txcorp.com

- TxSpectralLib - correlated k distributions generated from HITRAN 2008 molecular database;
  aerosol optics - external mixtures of standard OPAC types, non-spherical dust and ice particles, option for user specified properties

- vector (polarized) radiative transfer solver; future proposal for GPU acceleration

- option for user specified anisotropic surface BRDF

- water vapor and aerosol retrieval - options for user specified scene smoothness and reflectance ratio criteria

Modern IDL

- from novice to developer
- covers new IDL 8.0 features
- hoping to publish at the same time as IDL 8.0 release
- check michaelgalloy.com

Questions?

michaelgalloy.com
mgalloy@txcorp.com

gpulib.txcorp.com
gpulib.blogspot.com