Flux calculations for LOI: what was in the sausage

Chris Tunnell
Nikhef

A little history

- LOI = letter of intent (arXiv:1206.0294)
- LOI flux computed in early 2012
    - Accelerator design in flux (energy, circumference)
    - Still learning how best to do oscillation and rates analyses
    - Warning: everything is called VLENF for very-low-energy neutrino factory, which means nuSTORM
- The code was scattered across launchpad, which is a website we were using for the MICE experiment
    - Globes extensions: https://code.launchpad.net/~c-tunnell1/+junk/vlenf_source
    - Plotting and analysis scripts (revision 329 for ND plots): https://code.launchpad.net/~c-tunnell1/+junk/vlenf_scripts
    - For every plot in the LOI, look at the filename to see which revision and files were used.
    - However, the code got quite scary since it changed so much
- To simplify things, I made msrflux; please use this instead of LOI code since it is tested, better organized, and just better :
    - To install: pip install msrflux
    - Code: https://bitbucket.org/gnomon/msrflux
    - This was to share **algorithms**, but the default values were somewhat guesses since (from my perspective) we hadn't settled on parameters like the ND size or baseline. **You** must put in the parameters.

Regenerating ND rates

- I can regenerate ND rates (a fraction of % difference since this is MC integration and N was small)
- Agree to 1% with LOI code, big differences with msrflux?
- 1.5 m radius 100 t detector @ 50 m for 3.8 +/- 0.38 GeV beam.
  - No divergence, but 150 m straight
  - 1.7e18 muons: I had forgotten about this, but it was not always 1.8e18 since we rounded down at first to be conservative
- Something is wrong between our two parameterized codes. Not positive I'm simulating same things thought…

	LOI paper	LOI code (just regenerated; see appendix)	Difference (%)	msrflux	Difference (%)
numu bar CC	2145632	2160318	-0,68	3125588	-45,67
numu bar NC	844793	850553	-0,68		
nue NC	1387698	1398390	-0,77		
nue CC	3960421	3991121	-0,78	3716797	6,15

Testing algorithms: how I convinced myself the flux computation was right in msrflux

- What **IS** tested: Look in msrflux tests folder:
  - Compare flux/rates using GLOBES assuming point source:
    - 300 GeV NF
    - 50 GeV NF
    - 3 GeV NF
  - Against „Golden measurements at NF" hep-ph/0002108 for 50 GeV NF at:
    - 732 km
    - 3500 km
    - 7332 km
  - Take home message: believe far detector rates
- What **IS NOT** tested:
  - Averaging over detector straight, or anything for near detector; my thesis was mainly interested in far detector rates
    - note: You can see my simple integration routine in msrflux code
  - Notes: The event rates get softer when looking at ND rates, which one would expect since higher angles are sampled
  - I expect that the rates are correct to 50% (even before actually checking it for this talk), which still results in O(million) events

Mea culpa: poor knowledge transfer

- I don't remember where the defaults for msrflux came from, but I expected people to change things like detector size
  - Instead of ballpark defaults (the 5.1 diameter ND), I should have picked LOI numbers
  - Trying to give a new code that people could use easier seems to have confused things
- I'll work with David to track down why msrflux (new code) and the LOI numbers differ for near detector.
  - Step 1: get David to be able to run the exact code that was used at LOI
  - Step 2: understand msrflux and LOI difference. Should be identical…
  - Step 3: understand difference between Ao sim and parameterized models. Should be some differences since different assumptions, but not more than 20%…
  - Worst case: the new code is simple, so at least people can see what I did…

If you don’t want msrflux, here is LOI code instructions assuming Mac

sudo port install py27-matplotlib py27-scipy py27-numpy bzr

```
# environment
virtualenv-3.3 --system-site-packages -p python2.7 nustorm_flux
```
cd nustorm_flux
source bin/activate
mkdir source
cd source

globes
wget http://www.mpi-hd.mpg.de/personalhomes/globes/download/globes-3.1.11.tar.gz
tar xvfz globes-3.1.11.tar.gz
cd globes-3.1.11
./configure --prefix=$VIRTUAL_ENV
make install
cd ..
export DYLD_LIBRARY_PATH=$VIRTUAL_ENV/lib
add export DYLD_LIBRARY_PATH=$VIRTUAL_ENV/lib to $VIRTUAL_ENV/bin/activate too

nustorm extension to globes
pip install scons
sudo port install argp-standalone
bzr branch lp:~c-tunnell1/+junk/vlenf_source
cd vlenf_source
scons # this should install too
check that globes-vlenf command exists
cd ..

our analysis code

bzr branch lp:~c-tunnell1/+junk/vlenf_scripts
cd vlenf_scripts
bzr revert -r364
export PYTHONPATH=$VIRTUAL_ENV/source/vlenf_scripts/common_py
export VLENF_PATH=$VIRTUAL_ENV/source/vlenf_scripts
export GLB_PATH=$VIRTUAL_ENV/source/vlenf_scripts/glb
cd vlenf_accelerator

```
python channel_rate_acc.py vlenf_near plus # divide by ten since 1kt
python channel_rate_acc.py vlenf_near minus # divide by ten since 1kt
```