LROSE AMS 2020 Mini Workshop

January 12, 2020

*The GoToMeeting for this workshop was recorded and is available **here***

SESSION I

Installation Help

- Release links:
  - LROSE-core releases
  - LROSE-cyclone releases
- Encountering errors installing with apt-get through Ubuntu (Linux subsystem) on the Windows machines
- lrose-core.rb ~30 minute install on Mac OS

Introduction

- Lidar-Radar Open Software Environment (LROSE)
- CSU team: Michael M. Bell, Jen DeHart, Ting-Yu Cha, new additions: Ali Cole, Brianna Lund
- NCAR team: Wen-Chau Lee, Mike Dixon, Brenda Javornik
- Project to develop common software for LIDAR, RADAR, and Profiler community
- Plans for no-cost extension to year 5
- Documentation @ wiki.lrose.net
  - Community wiki
  - Includes basic instructions for main work flows
  - Goal: more of detailed documentation
- forum.lrose.net
  - User forum--great place for questions
  - Goal: tutorials, open/interactive conversations
- Builds
  - Blaze--done, moving into legacy status; has a DOI and is citable
  - Cyclone--no new development, stable soon (pending bug fixes); has a DOI and is citable
  - Elle (2020)
  - Jade (2021)--last formal release under SI2 funding support
  - Topaz (2022)--community garden--will need outside help
- 2nd LROSE Users Workshop
  - NCAR
  - 26 to 30 October
- LROSE Tools
  - `RadxPrint -h`
    - Command line options

        - Good tool to check if LROSE installation is working properly
    - `RadxPrint -print_params > RadxPrint.params`
        - Only vortex is not operational under this command yet
        - On the wiki, web version of default parameters
        - Defaults work great for a variety of things, but lots of customization options exist
    - `RadxPrint -params RadxPrint.params`
        - Invoke parameter file
    - Parameter file: preferred method of doing things, has most of the information about the files
    - `RadxConvert`
        - Get data into CfRadial format (all Radx tools designed to work with this format)
        - Encourage everybody to read raw data in any format but convert to CfRadial
    - `RadxBUFR`
        - Separate package
        - Specific BUFR format for radar data
        - Not a great format, but what GSI data assimilation uses
    - LROSE Grid
        - `Radx2Grid` (3D Cartesian grid, Cartesian PPIs, regular polar grid)
            - Designed for ground-based radars
            - In airborne- or spaceborne-radar, recommend FRACTL or SAMURAI (assumptions for `Radx2Grid` break down in these situations)
        - Question: can you bypass `RadxConvert` step and go straight to `Radx2Grid` step?
            - Yes, but it is not a recommended workflow.
        - What are the pros and cons of using bilinear interpolation over Barnes/Cressman for gridding radar data?
            - For ground-based data, bilinear is better because it avoids extrapolation.
            - Barnes/Cressman will extrapolate data away from its location.
            - Bilinear uses one beam to left/right/above/below, safest option
            - Barnes/Cressman can be better for data close together; only option for airborne
            - Effect of filtering on the whole process: RadxConvert doesn't do anything, just converts format; other processing techniques involve some degree of filtering, bilinear interpolation is one of the least invasive forms of filtering; Barnes/Cressman can wipe out certain spatial scales
    - Parallelization
        - SAMURAI is parallelized

        - FRACTL is not parallelized yet, but it is well-suited for that
        - `Radx2Grid` and `RadxPid` are parallelized
        - Reading/writing of netCDF files not parallelized (writing out/compressing is slow, often a roadblock to higher speed)
        - Good idea to have multi-processors
- Getting involved/help
    - Wiki & forum
    - lrose.net
    - lrose-help@lists.colostate.edu
    - mmbell@colostate.edu

LROSE Cyclone Starter Kit Julia Notebook

- Available online via Jupyter notebooks here
    - Julia notebook is recent
    - Python notebook is slightly outdated
- Well-documented with an example
    - Illustrates basic workflow
- Question: batch processing of data, for example for all 3 hrs of radar?
    - You could use the following command to process all data between the times
    - `Radx2Grid -params ?? -start "2019 08 20 00 00 00" -end "2019 08 20 04 00 00" -debug`

Hawkeye

- Slides available **here**
- Question: does it handle different data formats?
    - Some exceptions, but for the most part, it only handles CfRadial files (`RadxConvert` is an important **first** step, puts everything into the format and also gets everything into the right date/time format, creates proper directory structure, etc.)
- Question: does Python easily read CfRadial files?
    - Package options: pyart, netCDF4
- Color scales
    - Can be specified in Hawkeye (some common one it knows about and it will try to assign those)
    - Can also be specified in params file
    - Question: can you include matplotlib colormaps?
        - Two different supported colormap formats (unsure if matplotlib colormap format is supported, but if you have rgb values you can make it work)
- Question: can you comment on the major differences between CIDD and Hawkeye?
    - Hawkeye works off a single radar, engineering type display
        - Works in radial space
        - Single-radar specific
        - Replacement for SOLO

        - CIDD allows you to combine radar from multiple radars, satellite data, model data, etc
            - Puts it into a map projection (lat/lon, Mercator, Cartesian, etc)
            - Integrating display
            - Next generation of IDV
- Setup
    - Time slider at the bottom
    - WIDTH column allows you to switch between different fields
    - File-menu allows you to do things like open, save (nice for generating images for publications, etc. in a png file format), time-control, overlays, freeze, show-click (query data and look at individual values), boundary editor (more later), zoom/unzoom (drag a square and it will zoom) ((on new version of Hawkeye, zoom is on the actions))
    - Right-click (ctrl-click on Mac) brings up a menu (with options such as parameters)
- Question: can we use Hawkeye for displaying gridded data?
    - No, it's designed for CfRadial format
    - ncview works really well for a quick look
    - Actual plots need Python, Julia, etc.
- Can be installed via homebrew
- Question: is there an effort to make most important parameters the ones at the beginning of the file?
    - Yes, with `Radx2Grid` and `RadxPid`. Not true of `RadxConvert`, because that is mostly used from the command line.
    - For Hawkeye, this is not the case at the moment, so you will have to search for what you want a bit.

LROSE Polarimetric Data

- Python notebook, available **here**
- Forum link to test polarimetric workflow data here
- Question: what is the best way to document these workflows for the users? (Something to discuss)
    - At this early stage, the forum is a good option.
    - Storing/sharing radar data may prove to be challenging.
    - Agreement that uploading workflow scripts on a github repo would be helpful.

"Cyclone" Multi-Doppler Tutorial

- FRACTL (Fast Reorder and CEDRIC Technique in LROSE), SAMURAI (Spline Analysis at Mesoscale Utilizing Radar and Airborne Instrumentation)
- Fall workshop should have a detailed tutorial on this
- Inputs for the tutorial **here**--link updated later today
- Slides **here**
- Multi-dopper synthesis
    - Doppler velocity is a projection of full wind along the radar beam

    - Need ~ 30° separation
- Running similar to anything else in LROSE
- Mandatory things to change in FRACTL params
    - lat/lon origin
    - input/output
    - Variable names
    - Grid dimensions (just need to put in an increment and FRACTL will determine a range)
- Optional things to change in FRACTL params
    - Condition number cutoff
    - Leise filtering (implemented from CEDRIC)
- Mandatory things to change in SAMURAI params
    - Use background
    - input/output
    - Variables names
    - Grid dimensions (note: SAMURAI also supports cylindrical coordinates)
    - Reference time in param file
    - Reference center file (can move the grid as the storm is moving)
        - Can generate with provided Perl script
- Optional things to change in SAMURAI params
    - Gaussian and Spline filter lengths
        - These can be difficult to determine, email Michael or lrose-help for assistance
- Question: do FRACTL and SAMURAI take care of the time offset between files from two different radars? Or do they always have to be synchronised?
    - Both are 3D-var in that sense. They will take whatever data you put in as a pseudo dual-Doppler solution. Because SAMURAI has the moving ability, it will take care of a basic time correction.
    - FRACTL does not have that shifter functionality (want to add it)
- Question: can we use HRRR/NAM output as the background?
    - Yes, you just have to get it from the grib into the txt format that SAMURAI uses, but yes, anything you have can work as a background field.

SESSION II

Updates on Latest Features

- Slides available **here**
- LROSE-core has 400+ ‘apps’ with various levels of capability
- NSF Radar Workshop 2012 Survey defined radar software needs and level of need
    - Progress has been made in all of the key areas identified
- Ultimate plan is to move away from list-serv; aim to move to discourse
- Question/comment: having different workflows spread out among different repositories is a good idea; also having one code that installs multiple packages could be beneficial. When it comes to publishing, being able to accurately cite everything is valuable. Mixing and matching between disciplines can be beneficial, too (i.e. radar and aircraft)
- `RadxConvert` has large read capability; if formats are not included here, would be happy to include them
- `RadxBUFR` is a different app because of format and functionality
- Hawkeye
    - Engineering and real-time display suitable for both scanning and vertically pointing radars
    - Ideally creating an end to soloii
    - Editing support will be in Elle
    - Polygon tool
        - Now allows to click/drag the points to adjust them
        - Can add new points
        - Can be saved as boundary
    - Circle tool
        - Draw circle with radius
        - Can be saved as boundary
    - Brush tool
        - Allows you to click and draw a boundary
        - Can be saved as a boundary
    - Boundaries go to CfRadial file (if you reload, they will stay with it)
- LROSE Display
    - Jazz
        - Java-based
    - CIDD
        - C++ based
        - Must be built separately
        - Available in the docker container
    - TITAN
        - Nowcasting, storm-tracking display
        - Rview display
- LROSE QC
    - `RadxMergeFields` (cyclone)

        - Combine fields from CfRadial files
    - `RadxPersistentClutter` (cyclone)
        - Persistent ground clutter identification and removal
    - `RadxDealias` (elle)
        - 4D velocity dealiasing (James & Houze 2011)
    - `RadxQC` (elle)
        - Remove AP & sea clutter, RF interference, chaff
    - `AirborneRadarQC` (elle)
        - Airborne navigation corrections and automated editing (Cai et al. 2017)
- LROSE Grid
    - `Radx2Grid`
        - More detail above
    - Airborne radar/any kind of moving radar need reorder-style gridding (available in FRACTL and SAMURAI)
- LROSE Echo
    - `RadxRate` (formerly `RadxPartRain`)
        - Has functionality of `RadxKdp` and `RadxPid`
        - KDP, Z & ZDR attenuation, NCAR PID, precip rate
    - `RadxQPE`
        - Accumulated quantitative precipitation estimate
    - `RadxBeamBlock`
        - Bream blockage identification
    - `ConvStrat` (elle)
        - Convective-stratiform partition
    - `Refract` (elle)
        - Refractivity calculations
- LROSE Wind
    - Multi-Doppler
        - FRACTL (Fast Reorder and cedric Technique in LROSE)
            - Highly optimized classic dual-Doppler solver
        - SAMURAI (Spline Analysis at Mesoscale Utilizing Radar and Airborne Instrumentation)
            - Spline-based 3DVAR technique
    - Single Doppler
        - `RadxEvad`
            - Enhanced VAD
        - `VORTRAC`
            - GBVTD and GVTD
- LROSE installation
    - Everything cannot necessarily be supported
    - Lots of different combinations are being tested and lots of packages implemented
    - Testing being done currently
    - lrose-help (and the forum) is a good contact here

Discussion/Comments

- Question: can you say a little bit more about philosophy/plans for how to best approach documentation?
    - Wiki is part of that, i.e. the idea of making it a community-based documentation. Also lots of documentation in the parameter file.
    - Practically difficult due to time commitments of writing documentation.
    - Goal to reflect the software piece and the usability piece (bringing in the meteorology side) in the documentation.
- Comment: three types of documentation--how you use it/how does it do what it does & what does it do/what are the features of it & how to run it/examples.
    - Three types should be separated.
    - First type should be in the code directly.
    - You learn from examples and they should be there, but these tutorials should be separate from the details (use cases).
    - Use cases should have separate documentation.
- Comment: delicate balance between power and ease of use
    - Adding more functionality can decrease ease of use
    - Needs to be accessible for easy things to do, but also have the ability to be more powerful
- Comment: the flowchart style of showing how to use things is very beneficial
    - i.e. that in Michael's slides showing the paths to time series displays, engineering displays, scientist displays, and web-enabled integrated displays
- Comment: when publishing a paper using LROSE, the code from that use case should be published so people can follow along with it
    - Extra comment: papers that do not do this should be rejected, software needs to be open (esp if software is important to it)
    - Could be thrown into an appendix in your paper, include a link and acknowledgments, etc.
- Comment: at some point radar data may be too big to download and it will need to be accessed through the cloud; this is something we will want to eventually look at as a community and work with going forward
- Comment: a docker image/container can hold all of the software used to do an analysis; limited in size but maybe some amount of data along with it; should document everything done in the analysis and the steps can be reproduced; advantage is it holds specific versions of things, could make it easier to produce
- Comment: the "L" in LROSE; how is the progress going towards working to the "L" and the silent "P"; both communities are not as coherent as the radar community, important to make sure their voices are heard
    - Response: nothing in LROSE right now that would prevent you from using LIDAR data; as long as you can get it into CfRadial, it should (keyword) work
    - Representation at the LROSE workshop for both of these communities can be beneficial and make sure that those voices are being heard

    - Response: are people in LIDAR converting their data to the CfRadial format? (answer: geosphere LIDARs and NCAR LIDARs have been)
        - Response: huge diversity of LIDAR formats and a lot of those are not in `RadxConvert` ("give us the data and we'll try to get it in there")
        - Hawkeye has a vscan capability, you can do those types of displays; if you had a scanning LIDAR you could grid it like anything else
        - Base is there, but we need more people testing and using it
- Question: are there any major things that we're not addressing right now?
    - Comment: examples in documentation would be really beneficial; a sheet that says which software is related to which case would also be helpful; hail integration could be beneficial, too
        - Response: writing up a method in MATLAB (as a prototype), for example, then sending it to be converted to LROSE would be helpful to know what the best approach is in the eyes of the community
        - Response: need to make sure that things being put in the garden are things that we know what they do and what they are; ex: chose one dealiasing method published in the literature as a starting point
    - Comment: the majority of the software in the LROSE core has been developed somewhere and there is a need in the community for it to be included in our basic suite
- Comment: Future DISPLAY's:
    - Weather Mod and Met Service Nowcasting still uses Rview/TimeHist on a daily basis
    - It would be nice to get a new App, merging Rview/TimeHist and CIDD overlay capability and SPDB plotting
    - `RadxPartRain` filtering is very good but there are small gaps