Transitioning the NEMS/NMMB from NCEP operations to a broader community: application to an Alaska regional domain

DTC Visitor Project

Don Morton[1,2] & Dèlia Arnold[3]

[1]Arctic Region Supercomputing Center, University of Alaska Fairbanks

[2]Boreal Scientific Computing LLC, Fairbanks

[3]Central Institute for Meteorology and Geodynamics of Austria, Vienna

Introduction

This document describes the efforts of Don Morton and Dèlia Arnold in transitioning the NEMS/NMMB from the NCEP operational environment to a more general user environment with emphases on regional applications and portable implementations. In our original proposal we had planned an ambitious effort to deploy the NEMS/NMMB over both of our home regions. This pioneering effort would provide us with the experience to prepare a user guide for the typical Joe/Jane Sixpack community user. An overarching goal was to use this experience as a stepping-stone to begin realising the full vision of NEMS, which is to serve as a superstructure and infrastructure to couple numerous model components into a single operational system. We had hoped by engaging in this project, we would be well-positioned to start introducing coupled-model operational systems to community users.

Funding constraints led us to narrow our scope to deploying a single operational NEMS/NMMB system over the Alaska Region, perform some simple case studies, and report on our findings. Morton was funded for 6 weeks (2 weeks at NCEP in June 2012, 2 weeks at DTC in October 2012, and another 2 weeks at DTC in June 2013). Arnold was funded for 2 weeks, coinciding with Morton's June 2013 DTC visit.

Project objectives were met and, in some cases exceeded, and, as is often the case, areas of emphasis changed as the project progressed. Primary "distractions" were the efforts required to move the NEMS/NMMB into a true community environment supported by GNU compilers, and in the unexpected additional effort needed for postprocessing activities.

In this document we outline the general course of the project, while providing many of the tedious details at our publicly-available web presence,

Community NEMS (https://sites.google.com/site/communitynems/)

Progress - timeline

The project was initiated with a two-week trip by Morton to NCEP/EMC in Camp Springs, MD (just a week or two before the building was vacated by NOAA/NWS) in June 2012. Morton's primary goal for the two-week stay was to have the NEMS/NMMB running on the NOAA/NWS machine, *zeus*, on a regional Alaska domain. This would insure enough familiarity with the system to take it home and begin moving it to other computer platforms.

NCEP/EMC staff were very enthusiastic about the project and were quick to offer assistance when asked. On the first day, after a meeting with all interested personnel, Morton was briefed personally for an entire afternoon on the structure of the NEMS/NMMB, an explanation of how the run time system was currently embedded in a sophisticated regression testing framework, and what components would be of most interest. Morton spent time insuring that he could run this regression testing framework on *zeus*, and that he understood what was happening.

Since we were coming at this from a community user perspective, and we have substantial WRF experience, we decided we would try to look at the NEMS/NMMB from this angle, rather than this regression testing framework. Our aim was to understand how to flexibly run this model on different machines applied to various regional domains. A first step was to extract the basic components from the regression testing system and try to run the same cases outside of this environment, and understand the minimum requirements of a basic runtime system.

Once we felt confident in our understanding of the basic NEMS/NMMB runtime system, we proceeded to explore its deployment to a non-NCEP, Alaska domain, and this required the creation of our own domain input files. At this point, we got bogged down in the NEMS (or is it NMMB?) Preprocessing System (NPS). It's similar to WRF's WPS in many ways, but has its differences. Like the NEMS/NMMB, we were provided with an operational system which embedded the NPS components, and it was up to us to extract what we needed from a community perspective. At this time there was not a "friendly-user" NPS distribution, and we were unable to compile these tools ourselves on *zeus*, so resorted to using the binaries that had already been produced. It took a long time to get this all running correctly, and we were slowed down significantly by the lack of utilities for analysing the intermediate NPS outputs at various stages. Perhaps something was missed, but, because of the added complexity of the rotated lat-lon projection, and the absence of tools to accurately view the projection at an early stage, it was necessary to run through the NPS process, run NMMB, and then get an output file in order to use GrADS to view the actual domain we had defined.

By the time Morton left NCEP, he had finally been successful in building (and crudely documenting) a workflow starting at domain definition and culminating in successful model run and visualisation of 2m Temperature using GrADS. This was all done on *zeus*, using the Intel compiler.

After the NCEP visit, during Summer 2012, substantial time was invested in beginning the migration of the simple Alaska regional case to supercomputers at the Arctic Region Supercomputing Center. One of the biggest concerns was that NEMS/NMMB had only been deployed in an Intel compiler environment, and deploying Intel compilers on our ultimate target machines would be difficult, if not impossible. Although we were skeptical about a Gnu port, we decided that we should try, and we saw enough incremental success to keep us going. Because the NPS source code wasn't yet available, the initial efforts were focused on the NEMS/NMMB runtime system, using the input files produced with NPS binaries on *zeus*. The porting proceeded as follows

- Utilising Morton's home Centos 6.0 machine, *alaskawx*, with 16 cores, installed Intel compilers and moved the simple Alaska regional test case from *zeus*. Compiled the NMMB runtime code, copied input files from *zeus*, and successfully ran and visualised the Alaska case
- On *alaskawx*, ported the NEMS/NMMB code and associated libraries from an Intel compiler to a Gnu compiler environment. This took up a lot of time and is described more fully below, and documented completely on our web page. By October 2012, the Gnu port had been successful, and it was possible to compile and run an Alaska NEMS/NMMB case on this system (with preprocessing still occurring on *zeus*).
- Now that the NEMS/NMMB was running in a Gnu environment, ported to the Penguin Computer Cluster machine, *pacman*, at ARSC, in its Gnu environment.

In October 2012, Morton spent two weeks at DTC in Boulder, participating in a NEMS/NMMB teleconference with DTC and NCEP personnel, completing the NEMS/NMMB porting from *zeus* Intel to Gnu on *alaskawx* and *pacman*, with simple visualisation via GrADS

The period from Autumn 2012 to June 2013 was spent continuing the porting activities. An Intel-compatible source distribution of NPS that had been compiled and executed on *zeus* was provided, so the first steps were to test it on *zeus*, then get it working on *alaskawx* with the Intel compiler, and then try to port it to Gnu on *alaskawx*. The porting to Gnu is described below, and in more detail on the Community NEMS website.

With the porting of the entire NPS and NEMS/NMMB to Gnu on *alaskawx*, it was possible to start experimenting with larger Alaska domains, and then consider the port to ARSC's Cray XK6m, *fish*, the intended platform for operational runs. *fish* is a Cray Linux system, with Gnu compilers wrapped by Cray utilities, making the port fairly straightforward. With this accomplished, the goal of transitioning the NCEP NEMS/NMMB and NPS to a Cray in Alaska, operating on an Alaska domain, had been met.

With the insight gained from extracting the NPS and NEMS/NMMB components from the NCEP operational framework and relating this to the WRF world, it was possible to build an operational NEMS/NMMB system using much of the software infrastructure from the current WRF-based High Resolution Rapid Refresh for Alaska (HRRRAK). This was implemented at 0.1 degree (approximately 10km, 300x300x51 grid points) resolution so as to be small enough to facilitate

rapid 48-hour runs, four times per day.

Finally, Morton and Arnold spent two weeks at DTC in June 2013 to build and execute some performance tests of the NEMS/NMMB operational system at 0.03 and 0.1 degree horizontal resolutions and to compare outputs of these with the 3km HRRR-AK forecasts. The NEMS/NMMB system was ported to *yellowstone* with some difficulty, an ambitious set of Python/GrADS postprocessing scripts was created to operationally visualise key NEMS/NMMB output, and to compare 0.03 and 0.1 degree cases with HRRR-AK cases, and to perform some simple comparisons of computational performance. Finally, the use of NAM, as opposed to GFS, for initialisation was explored, and documentation for the project was organised.

Porting activities

As described above, porting of the code constituted - by far - the majority of time spent in this project. The code to date had been deployed with the Intel compilers, and we felt that In order to assure a true community deployment, we would not be able to assume Intel environments and, in fact, needed to try to make it all Gnu compatible. Although we were not very confident we could accomplish such a port with such a huge system, we decided to take the risk. If, by the end of Summer 2012, a Gnu port had been elusive, we would fall back on the Intel environment and give up hopes of moving it to other machines.

Though there were a number of porting issues that took a lot of time to understand, the actual changes made to the NEMS/NMMB code were not overwhelming, and in most cases made for better code. Because an NPS source distribution wasn't readily available at first, initial efforts were directed at the NEMS/NMMB. The general porting path was as follows:

- Migrate code from the *zeus* Intel environment to the *alaskawx* Intel environment. *alaskawx* is Morton's home machine, with 16 processing cores, running Linux CentOS 6.
- Port the code on *alaskawx* from an Intel to a Gnu environment. Although this was very time-consuming, the total number of changes to the original distribution was relatively small.
    - All of the Makefiles for building the various supporting libs had compilers and compiler options hard-coded in them. These needed to be changed manually. Although a scheme for automating these changes would be valuable, we felt that it was beyond the scope of our work, and feared that implementation of such a scheme might not be compatible with the NCEP way of doing things. So, we made the changes manually. Additionally, several problems in the existing Makefiles were discovered as we transitioned to the Gnu environment.
    - Because older versions of gfortran were unable to support some of the code features, we relied on gcc v4.7.1 (and have since used 4.7.2 and 4.7.3). However, we learned that the newer implementations of gfortran are quite strict, not allowing some Fortran coding practises that had become fairly common, even

if not strictly adhering to language specifications. Certain features that gfortran will catch are not addressed by other compilers such as Intel and PGI. In effect, by using later versions of gfortran we were able to find and modify non-standard constructs, hopefully making the code even more portable. The required changes to the NEMS/NMMB code distribution were in some subroutine interfaces for ESMF, and in some non-portable timing routines.

    - During this porting, we found that code "cleaning" environments were not complete, and until we recognised this, we encountered numerous problems when trying to compile new changes. In fact, the original distribution comes with some pre-made libraries and object files that need to be removed before trying to compile on another machine.

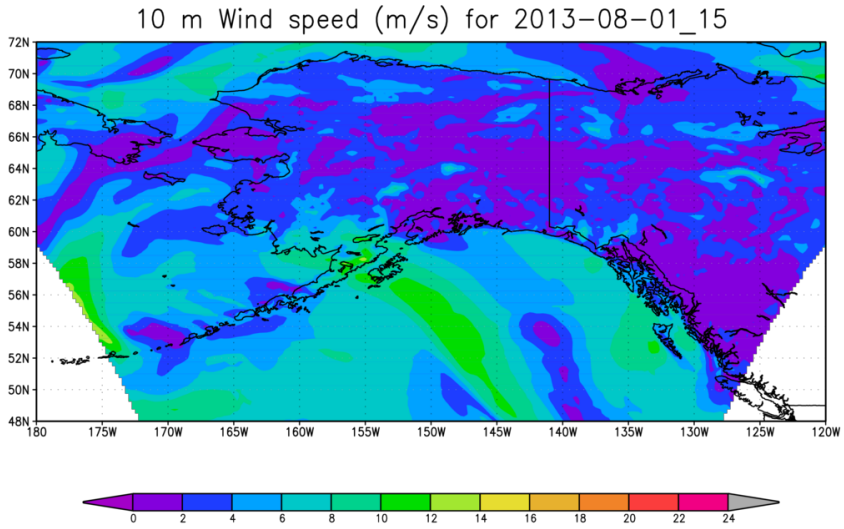
- Porting of the *alaskawx* Gnu code to the *pacman* Gnu environment. *pacman* is the Arctic Region Supercomputing Center's Penguin Linux cluster with 2816 compute cores on a RedHat Enterprise Linux operating system. After the *alaskawx* Gnu port, this was fairly straightforward
- Porting of the *pacman* Gnu code to the *fish* Gnu environment. *fish* is the Arctic Region Supercomputing Center's 1152-core Cray XK6m, Cray Linux Environment version 4 (CLE4) OS. Although *fish* has an adequate Gnu environment, Cray uses wrappers around compilers and MPI environments, so things are done a little differently. Previous experience with Crays and with WRF in this context helped with the task.
- Porting of the *alaskawx*/*pacman* Gnu code to *yellowstone*. For the most part this was straightforward, but in one case *yellowstone* doesn't support the full MPI standard for Gnu or PGI compilers, so source code needed to be changed.

In October 2012 the NPS source code distribution was made available. It had been working in the *zeus* Intel environment, and we went through the same general procedure to get it running in the Gnu environment of other machines

- Test in *zeus* Intel-environment
- Migrate to *alaskawx* Intel environment and test
- Port to *alaskawx* Gnu environment - as with the NEMS/NMMB, this took a fair amount of time, requiring the hard-code changes in library Makefile and dealing with incomplete "clean" mechanisms
- Subsequent ports to *pacman* Gnu environments and then *fish* Cray/Gnu environments
- On *yellowstone*, although we were able to compile NPS in the Gnu environment, the executables presented bugs that we had seen in earlier versions of WRF WPS. We were puzzled as to why this code had worked well in the Gnu environment of other machines, but failed on *yellowstone*. We never resolved this issue, and resorted to using pre-compiled NPS executables.

Operational Prototype

In May 2013, just before the final 2-week DTC visit, the NEMS/NMMB was successfully deployed on the ARSC Cray XK6m to run real-time quasi-operational forecasts. The model runs have proven to be very stable since then, running four times per day on 48-hour forecasts on 0.1 degree (approximately 10km) domain, driven by the 0.5 degree GFS. Though the routines for producing automated graphics were developed in June 2013, they weren't actually deployed until 30 July 2013.



Notes on operational deployment are available at:

https://sites.google.com/site/communitynems/home/alaska-operational-deployment

with raw binary output and graphics available at:

http://weather.arsc.edu/Operational/NEMSAK10km/Products/binaryOut/
http://weather.arsc.edu/Operational/NEMSAK10km/Products/outputGraphics/

Although our initial ambition was to run a NEMS/NMMB that looked much like the HRRR-AK, with identical outputs and verifications, various limits precluded this from happening. Besides some of the technical difficulties in working with the NMMB output, machine time at ARSC is limited. Since the Cray is relatively unused right now, we felt that with 10km resolution we would typically

get reasonable throughput on our jobs without using special scheduling mechanisms. In fact, the jobs have in general run very well on 70 compute tasks and 2 I/O tasks in less than an hour. The graphics presentation is rather bare-bones (a list of PNG files for each forecast). If potential users (with funds) are interested in something more, it may be accommodated, but given that NCEP already runs the NAM over this region, we are skeptical that these forecasts will be seen as anything more than a demonstration of Joe/Jane Sixpack users deploying the NEMS/NMMB in their own operational region of interest. Hence, further development of output products does not seem to be a priority.

Performance

Although a detailed performance assessment and comparison with the current HRRR-AK operational runs was not intended, it was decided to include a small benchmark of the case studies:

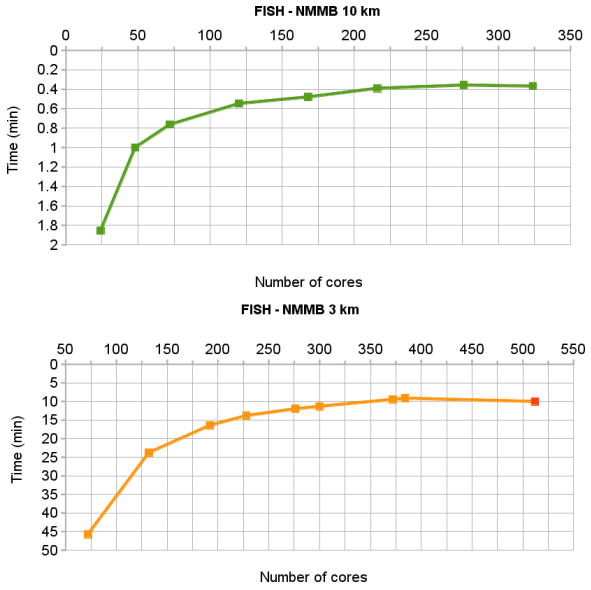
1. A one hour run with an Alaskan domain with 0.1 degree (~10 km) horizontal resolution (300x300x51 grid points)
2. A one hour run with an Alaskan domain with 0.03 degree (~3 km) horizontal resolution (1050x1050x51 grid points)

in FISH and YELLOWSTONE.

The benchmark only obtained the walltimes of the runs with increasing number of cores . The results may be found in:

https://sites.google.com/site/communitynems/home/performance-testing

In fish we have used the standard queue for all the runs except for the 512 cores and NMMB 3 km run. The latter (red marker) was run in the gpu queue. Full 12-core and 16-core nodes have been used



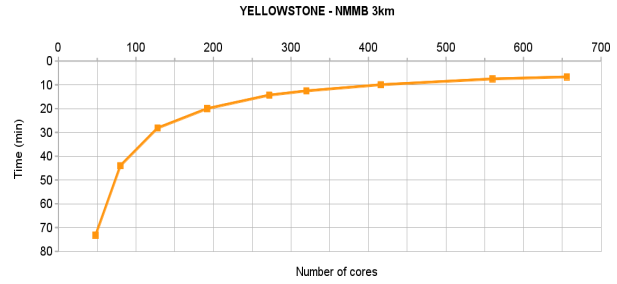


Figure 1: performance (minutes needed to make the 1 -hour run with increasing number of cores) of the 10 km (upper plot) and 3 km (lower plot) for FISH (a) and of the 3 km run for YELLOWSTONE (b)

The results are not surprising and show fairly good scalability for problems with a large amount of grid cells reaching a plateau close to the 400 processors.

Post-processing activities

One of the aspects we had to address was to produce an easy but flexible enough environment to process the output, compare it with existing models and also with observations. For examples, explanations and the source code, visit

https://sites.google.com/site/communitynems/home/postprocessing

Foreseeing a further implementation in an operational setting, a PYTHON set of wrapping routines was generated to control the calls to the plotting software, extraction of observational data and comparison of model-model output and model-obs. The manipulation of the NEMS-NMMB output in terms of plotting and obtaining variable time series was done with the GrADS (http://www.iges.org/grads/) software. GrADS was selected because of four main reasons:

1. NMMB directly produces the .ctl and binary files readable by GrADS,
2. GrADS allows for a certains degree of scripting making it easy to be used in the PYTHON structure,
3. many organisations and institutions dealing with meteorological aspects have GrADS installed and the researchers have most probably used it at some point in their careers

and thus are already familiar with it and

4. because conversion tools from WRF output to GrADS are available allowing, therefore, the comparison of the NEMS-NMMB runs in this study with the operationally running HRRR-AK WRF runs. This required to add a step to process the HRRR-AK output with ARWpost.

In this way, an environment for comparing one model against the other together with OBS data was built and allows for addressing differences between models depending on set-up aspects (some example products in figures 2-3) . Although the routines, developed basically to perform test-case studies, are hard coded in terms of the variables extracted, they are very simple to modify and adapt to the user needs. A version to be used in an operational environment has also been produced.

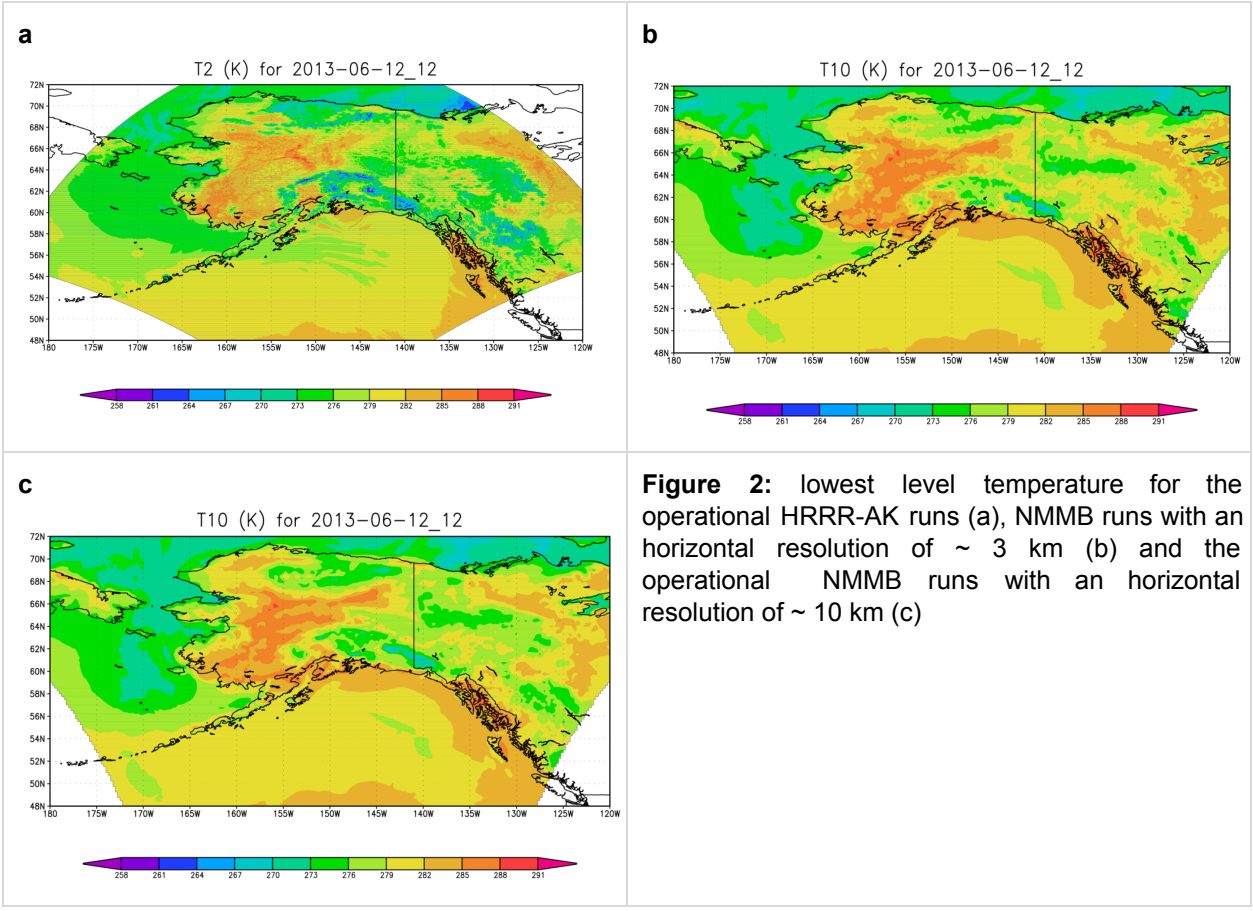


Figure 2: lowest level temperature for the operational HRRR-AK runs (a), NMMB runs with an horizontal resolution of ~ 3 km (b) and the operational NMMB runs with an horizontal resolution of ~ 10 km (c)

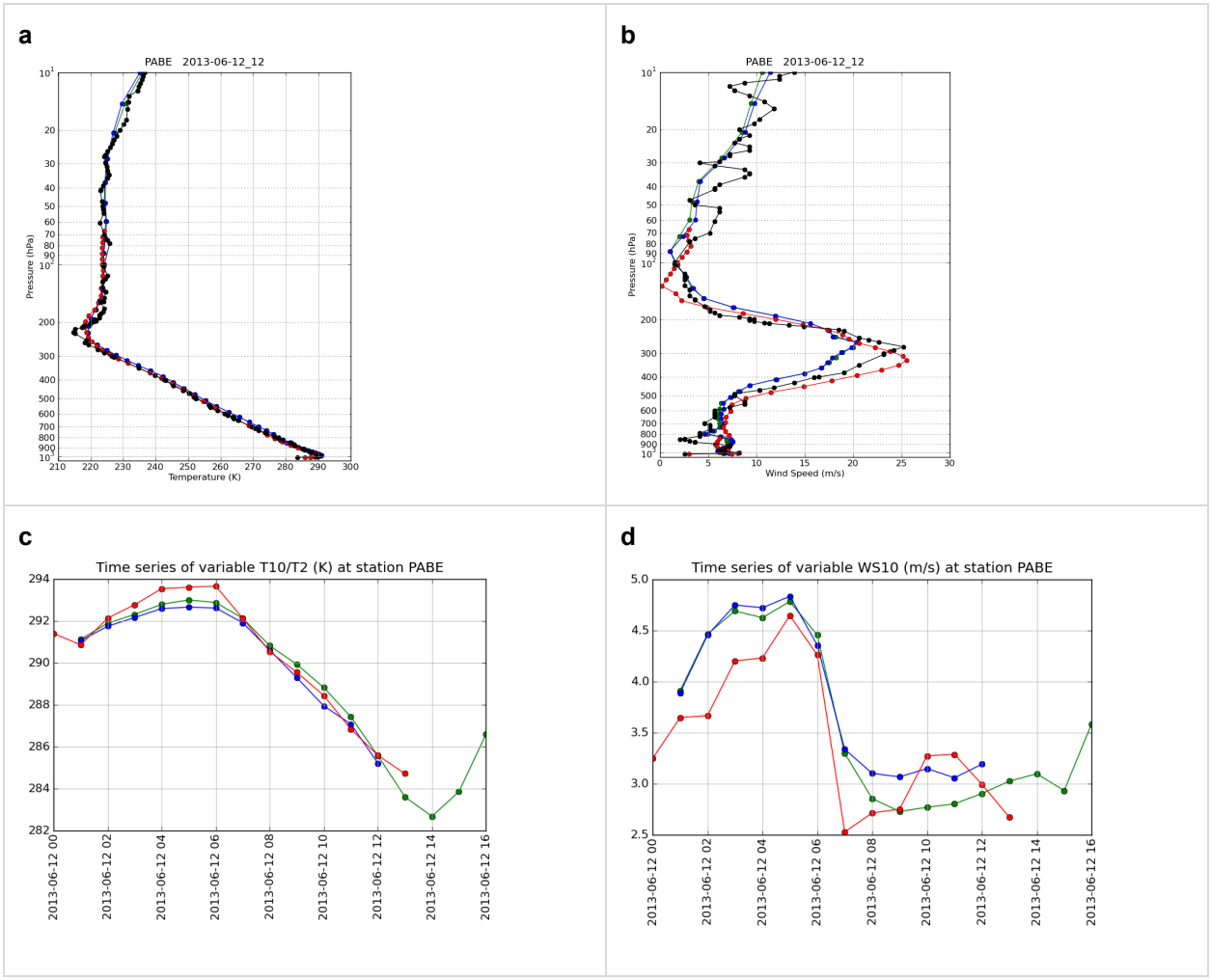


Figure 3: soundings and time series for the Bethel station in Alaska. Black are the observations, NMMB 10 km (green), 3 km (blue) and HRRR-AK (red) .

Summary and Recommendations

The original intent expressed in our proposal had been to devote a total of 4.0 person-months to deploying and documenting the NEMS/NMMB in a way that it could start being used regularly in the general community. Funding limitations led us to narrow the scope to a simple regional operational deployment and demonstrate usability of the system by performing case studies. In retrospect, even if the original proposal had been funded, it would have been difficult over the course of the year (even with 4.0 months funding) to make this ready for first community users - there was a steep learning curve in getting things started and running on Gnu-based machines that would have held up progress in other areas.

We are pleased with our accomplishments in this project, having taken - with the help of NCEP personnel - the NEMS/NMMB system from the somewhat "restrictive" environment of NCEP to an environment more familiar to the general WRF and Open Source user. Through our efforts we have demonstrated that NEMS/NMMB can now be used for research and even some operations in typical Linux environments, as well as on state of the art Cray computing systems (a semi-typical Linux environment).

Our pre-conceived notions were that the NEMS/NMMB would be WRF-like, meaning relatively portable, and that it could be inserted into an operational environment much as we have already done with various WRF deployments. However, we had a fair amount of unexpected additional work in moving this system to a Gnu-based environment, and more so in the creation of forecast products. We had naively hoped to post-process NMMB output with UPP, so that we could deliver GRIB file output to National Weather Service AWIPS and use some of our existing NCL and MET infrastructure for visualisations and verification. For visualisation, we didn't have the resources to do anything more than GrADS, and for verification we have to hope that UPP may be able to work with NMMB output so that it can be converted to GRIB. This has yet to be explored.

Recommendations (listed in no particular order)

- Documentation - obviously, much needed, especially on various namelist options. We have at least documented our notes for deployment, and persistent users should be able to get as far as we did with minimal extra effort. A progression of examples, similar to the type we have prepared for WRF users - borrowing heavily from the WRF examples, themselves would be useful -

  http://weather.arsc.edu/Training/WRFTutorial-June2009/BasicWRFTutorial_June2009.pdf

  http://weather.arsc.edu/Training/WRFTutorial-June2009/WRFNesting_June2009.pdf

http://weather.arsc.edu/Training/WRFTutorial-June2009/SampleCaseStudies_June2009.pdf

In addition to GFS, we minimally explored ingest of NAMS regional data. It took a long time and as is often the case, a simple namelist variable change solved our problems. Another area for documentation is more guidance on namelist setup and I/O quilting. Although we learned through trial-and-error what kinds of combinations of compute and I/O nodes worked for us, we feel that additional information as to the "why" would be useful.

- Compilation issues
    - Better, more thorough cleans
    - Set up (perhaps like the WRF model with a configure script first) to compile, without various hard-codings, on at least the Intel and Gnu Linux environments and, since we have already explored this area, also in a Cray Gnu environment.
- The NMMB message outputs (the PET files for each process) are for the most part, useless, confusing, and misleading to Joe/Jane Sixpack. Even when things are running correctly, there are numerous warning and error messages that can distract the new user.
- Provide tools for viewing intermediate outputs. Especially, a tool like the WPS plotgrids.exe that will allow for viewing model domain based solely on the contents of the namelist file. Although I tried to compile and use the plotgrids.exe that came with an initial NPS distribution, I could never get it to plot in the rotated lat-lon projection, so the only way I could truly understand how my domains were formed was to run NPS all the way through, and then run NMMB on some data, then view the output files with GrADS - clearly an efficient way to initially set up model domains.
- Output formats - difficult and non-standard to work with. I didn't have the time to explore these formats in depth, but, at the very least, I would have liked to have understood tools for interfacing with this data (I undestand some API's were available, but I didn't look into this). One user outside our project was much more forceful, suggesting that the community is based on NetCDF these days, and it simply makes no sense to put data out there in a "unique" format.

Disappointments

- Feel like we failed to achieve our goal of understanding how to work in the NEMS

framework as a whole, realising its full vision of coupling various model components in unique ways to create unique models. Perhaps it was naive to think that this might be the foundation for a "plug and play" coupling environment, but that's what we had hoped to begin understanding. Although we feel we understand the NEMS/NMMB much better, we don't feel like we've gone beyond learning how to deploy "another weather model."

- From a community perspective - this is just my opinion - I think the overall vision of NEMS as a coupled modeling system, capable of (with some work) plug-n-play model components sells. My original hope was to experiment with this and introduce it in universities in order to encourage novel modeling experiments. If the goal is simply to get the existing NEMS/NMMB out into the community, I frankly have to wonder if it's worth the heavy investment. Current DTC projects, like WRF, GSI, MET and UPP have wide-ranging applications, and a long-history of support by the community. I fear that NEMS/NMMB, if supported in isolation, without coupling in other models, will simply not have great interest in the community, and maybe the resources necessary for support are better applied elsewhere. On the other hand, a move to demonstrate the utility of this NEMS plug-n-play capability (yes, we recognise it's not THAT easy), using the NEMS/NMMB as a foundation, could be quite valuable, opening the door for a bright future of model coupling experiments and deployments.

Our future use

- We would be very interested in exploring, contractually, many of the exploratory and documentation issues mentioned above. This might consist of activities like
    - Collaborating with NCEP and DTC personnel to package and document the existing NEMS/NMMB system so that community users, coming from a WRF background, can begin to work with it somewhat easily.
    - Exploring the issues related to examining intermediate NPS outputs, much in the way that WPS tools do, to detect problems in model setup and data ingest as they happen.
    - Exploring the issues related to moving NMMB output into formats that are more compatible with existing tools such as UPP (perhaps it already works - we don't know) and NCL.
    - Explore and document the use of NEMS, with the NMMB application, in a true coupling of model components, such as hydrology, sea-ice, etc.
- As part of a private venture, I am attempting to demonstrate the utility of cloud computing environments for various models, and am contemplating the use of NEMS/NMMB (as well as WRF and other atmospheric transport models) as demo cases.