If you want to use your G4/G5 with Mac OS X for some good numerical work .. this could be useful to you. I have here, binaries, source, documentation and instructions to install **Fortran**, **MPI**, **OpenMP**, **PVM**, **Octave**, **GDL**, **Cactus**, **Globus**, **RNPL**, **GIMPS**, **GRAVSIM**, **FEYNMAN**, **GNU Java**, etc. on **Darwin**. If you are interested in harnessing the full power of a dual G4/G5, i.e. using **AltiVec** and both processors, **click here**.

While most of these packages have been recently recompiled/tested on **PANTHER OS X 10.3** and its new Developer Tools, I would expect the binaries included here to work with almost any version of OS X. I've included **Jaguar** versions of some binaries at some places as needed. Feel free to contact me with questions or problems! Note: Panther does not include **/usr/local/bin** in its default **PATH**. Therefore, assuming you're using the default **bash** shell, please type (or include in **/etc/profile**) **PATH=/usr/local/bin:$PATH** and **export PATH** before using any of these tools.

The binary install option provided here, will literally take a few seconds! Many consider this the quickest and easiest install for some of these packages available on the web. If you choose to get the source and compile, be warned that it may take several hours even when everything goes perfectly!

NEW! Most binaries updated with current CVS source. Screenshots of some of these software packages and a short blurb about **Apple's CHUD Tools** is now available at **this link**!

Computation Tools :: Fortran

g95 3.5 :

Recently compiled using source code from the GNU CVS servers. This contains gfortran (g95) version 3.5 compiler. This is an alpha version of the much awaited, free, open source, F95 compiler! Download my binaries, and **cd** to the download folder. Then **gunzip g95-bin.tar.gz** (if your browser didn't do so already) and **sudo tar -xvf g95-bin.tar -C /**. It installs everything in **/usr/local**. You invoke g95 by typing **gfortran**. Note that you may need to add the flag **-lstdc++** for getting it recognize some of the unimplemented standard functions. You will also need to have Apple's Developer Tools installed. They are included on a separate CD in the retail version of OS X and also available as a free download from **Apple's Developer Site**. Please install the most current version of Developer Tools. Finally, note that there is an **alternate g95 project**, that is not part of GCC. They seem to have recently released their own official version of g95 for OS X which you may wish to try. Again, note that that g95 is not the same as this version here.

Note: You may need to **ranlib** some libs after you install. The compiler will tell you which ones when you try to use it. In that case, simply do a **sudo ranlib -s** on each such library.

*Binaries: **g95-bin.tar.gz** (Jaguar/Panther), updated July 15th, 2004.*
*Documentation: **click here!***

g77 3.4 :

Recently compiled using source code from the GNU CVS servers. This is the FINAL release of g77 (version 3.4.1 compiler). Future versions of GCC will have g95 (see above). Download my binaries, and **cd** to the

download folder. Then **gunzip g77v3.4-bin.tar.gz** (if your browser didn't do so already) and **sudo tar -xvf g77v3.4-bin.tar -C /**. It installs everything in **/usr/local**. Try it! You will need to have Apple's Developer Tools installed. They are included on a separate CD in the retail version of OS X and also available as a free download from **Apple's Developer Site**. Please install the most current version of Developer Tools.

Note: You may need to **ranlib** some libs after you install. The compiler will tell you which ones when you try to use it. In that case, simply do a **sudo ranlib -s** on each such library.

*Binaries: **g77v3.4-bin.tar.gz** (Jaguar/Panther), updated July 15th, 2004.*
*Documentation: **click here!***

Commercial Fortran:

ABSOFT: A commercial option for Fortran on OS X. It works very well, I have used it extensively. It optimizes better than g77. Includes **f77**, **f90**, **f95** and a rather nice IDE. It also includes AltiVec optimized Math and BLAS libraries. Moreover, along with **CBS VAST** it can vectorize and parallelize your Fortran codes! In other words, it can automatically optimize your code for AltiVec and dual processors! More details about this feature available at **this link**. For more information, try the **Absoft OS X product page**.

NAG: A commercial **f95** for OS X. I recently played with it. Does excellent optimization .. yielding much faster executables compared with g77. Very affordable commercial solution. For information, try the **NAG Fortran page**.

IBM: A **f90, C, C++** suite for OS X. IBM released its highly optimized PPC compilers (including Fortran!) for OS X. On initial tests they seem to provide a speed gain of upto 2X on current G4/G5's over other compilers.

*Absoft website: **click here!***
*IBM: **click here!***
*NAG website: **click here!***

F2C based Fortran:

This is the oldest Fortran compiler available for OS X. Its been around since OS X was in a public beta state. Download this shell script: **buildf2c**, type **chmod +x buildf2c** and then **sudo ./buildf2c**. The script will grab f2c source from Netlib repositories and install a f2c based compiler in **/usr/local/**. You are done! The compiler can be envoked by the commands **fc** or **f2c**.

*Install script: **buildf2c***
*Documentation: **click here!***

ADAPTOR (High Performance Fortran) Compiler:

*Note: You will also need **g77** 3.4 and **MPICH**.*
This is a compiler wrapper that converts high performance fortran code (.hpf) to F77 with MPI parallelization. Download the source with binaries. You will need to choose and installation directory and assign the environmental variable **PHOME** its name. Then **gunzip adaptor.tar.gz** (if your browser didn't do so already) and **tar -xvf adaptor.tar** in this directory. Remember to add the **bin** directory to your **PATH**. You may need to add the flag **-lcc_dynamic** for your hpf source to compile, depending on what version of GCC you're using.

*Source and Binaries: **adaptor.tar.gz**, updated Spring 2004.*
*Documentation: **click here!***

Computation Tools :: MPI and OpenMP

MPI (Message Passing Interface):

MPI is the most widely used framework for parallel computing. With this you can parallel compute over almost any kind of cluster .. Mac's, SGI's, SUN's, Linux .. even over a hybrid or heterogeneous cluster. If you want to use MPI with **ssh** try **this link** for instructions.

MPICH: Get the source from the **MPICH website**, configure, make and install. Use a UFS filesystem if you want to compile the C++ extensions. Should have no problems! Or get my binaries, **gunzip mpich-bin.tar.gz** (if your browser didn't do so already) and **sudo tar -xvf mpich-bin.tar -C /** and you are done! It installs in **/usr/local/mpich/**.

LAM MPI: OS X is now officially supported! Download their binaries for OS X from their site. An initial release of **XMPI for OS X** has recently appeared. XMPI graphical user interface for running, debugging and visualizing MPI programs.

There is also UCLA's **AppleSeed and MacMPI** for Mac ONLY clusters that has been a big success. It indeed is a very easy way to build and run a cluster. In that sense it is very *mac-like* in spirit! Running distributed simulations is also very easy using their GUI based **Pooch Application**. They also provide useful tutorials and detailed documentation on writing and running parallel codes. Check those out at **this link**.

*Source: **LAM MPI**, **MPICH***
*Binaries: **click here!**, **mpich-bin.tar.gz**, updated Spring 2004.*
*Documentation: **LAM MPI**, **MPICH***

OpenMP (Shared Memory Multiprocessing):

These tools compile code optimized for dual processors based on OpenMP directives. SMP is much easier to learn, compared to MPI .. so this could be useful to some. Get the latest version (1.6) of Omni MP compiler source and install. You should have no problems. Or get my binary distribution. To install, **gunzip openmp-bin.tar.gz** (if your browser didn't do so already) and then **sudo tar -xvf openmp-bin.tar -C /** and you are done! It installs OpenMP in **/usr/local/openmp/**. If you are interested in a tool that auto-parallelizes your code, **click here**.

*Source: **click here!***
*Binaries: **openmp-bin.tar.gz**, updated Spring 2004.*
*Documentation: **click here!***

PVM (Parallel Virtual Machine):

An alternate infrastructure to run a cluster of computers. PVM compiles on OS X with minimal modifications to the current source. Just get my source and binary distribution. To install, **gunzip pvm3.tar.gz** (if your browser didn't do so already) and then **tar -xvf pvm3.tar** and you are done! It installs PVM in your current directory. You'll need to set the environmental variable **PVM_ROOT** before you start to use it.

*Source and Binaries: **pvm3.tar.gz**, updated Spring 2004.*
*Documentation: **click here!***

Computation Tools :: Octave

Octave:

OCTAVE: Octave is an open source, Matlab-like numerical analysis software package that is very popular among scientists and engineers. I compiled Octave 2.1.55 with **octave-forge** extensions using Apple's GCC

and the g77 3.4 above. Download my binaries, **gunzip octave-forge-bin.tar.gz** (if your browser didn't do so already) and then **sudo tar -xvf octave-forge-bin.tar -C /** and you are done! It installs in **/usr/local/**. Note that this Octave installation does NOT install **gnuplot**. Please download a version of gnuplot on OS X, of your choice (Aqua, X11, AquaTerm) from **this link**. For the octave-forge extensions, remember to create a file called **.octaverc** with **these lines** in it and place it in your home directory.

OCTAVE-MPI: I was also able to compile the **source** for **Octave-MPI**! This version of Octave can run in parallel over a cluster of machines and/or multiple processors using MPICH MPI. For Octave-MPI, download my binaries, **gunzip octave-mpi.tar.gz** (if your browser didn't do so already) and then **sudo tar -xvf octave-mpi.tar -C /** and you are done! It installs in **/usr/local**.

Source: ***Octave***, ***Octave-MPI***
Binaries: ***octave-forge-bin.tar.gz***, ***octave-mpi.tar.gz***, *updated Spring 2004.*
Documentation: ***click here!***

GDL:

Gnu Data Language (GDL) is an open source implementation of Interactive Data Language (IDL) which is an expensive commercial software package developed by **Research Systems Inc.** To find out more, visit the **GDL website**. To download my binaries, **gunzip gdl-bin.tar.gz** (if your browser didn't do so already) and then **sudo tar -xvf gdl-bin.tar -C /** and you are done! It installs in **/usr/local**.

Source: ***GDL***
Binaries: ***gdl-bin.tar.gz***, *updated Spring 2004.*
Documentation: Refer to IDL's documentation.

Computation Tools :: Cactus, Globus, etc.

CactusCode (Numerical Computing Toolkit):

Cactus is an open source problem solving environment designed for scientists and engineers. Its modular structure ("thorns") easily enables parallel computation across different architectures and collaborative code development between different groups. It is mainly used for solving complicated PDE's. It highly portable, and has some really remarkable features like observing and even "controlling" or "steering" a running simulation using an ordinary web browser! Here is a link to a perpetual demo running in Albert-Einstein-Institute, MPG, Germany: **Cactus Demo**. Check it out! Cactus compiles and runs great on OS X with GCC 3.3. Get the "last stable release" source. Remember to use the make options **ARFLAGS='ruc' USE_RANLIB='yes' CC=gcc CXX=g++**. The C and F77 "thorns" compile with no trouble. For the Fortran 90 ones, you will need a commercial Fortran compiler, like the one from **Absoft**. The above MPI distributions are fully compatible with Cactus! For some more info, check out CactusCode's recent **OS X compatibility web page**.

Source: ***click here!***
Documentation: ***click here!***

Note: If you are interested in other advanced PDE solving environments, try **PETSc** from **ANL**. They now officially support Mac OS X!

Globus (Grid Computing Infrastructure):

The Globus project is about developing technology to build grids for high performance computation. Grids are environments that enable software applications to integrate instruments, displays, computational and information resources that are managed by diverse organizations in widespread locations. As an example, look at this **web page** that has information on a Cactus based grid computation of the collision of two black

holes. The OS X implementation of the Globus framework, is based on Java, called **Java Commodity Kit.**There are various efforts underway, attempting to port the entire C based Globus Toolkit to OS X. **Update!** Finally, intructions for a complete port of C-based Globus Toolkit 2.2.4 are **available here**. Get my binaries for Panther below. They'll install in **/usr/local/globus** in the usual way, **sudo tar -xvf globus-2.4-bin.tar -C /**.

A batch system for Globus, called the **Grid Engine** is an open source project sponsored by **SUN**. They have a functioning OS X port!

Source and Binaries: ***Java CoG***, ***Grid Engine***, ***Globus 2.4***, *updated Spring 2004.*
Documentation: ***Java CoG***, ***Grid Engine***

Condor (HTC):

Condor is a software system developed at the University of Wisconsin that promises to expand computing capabilities through efficient capture of cycles on idle machines. The software, operating within an **HTC (High Throughput Computing)** rather than a traditional HPC (High Performance Computing) paradigm, organizes machines into clusters, called pools, or collections of clusters called flocks, that can exchange resources. Condor then hunts for idle workstations to run jobs. When the owner resumes computing, Condor migrates the job to another machine. The Condor Project now officially supports Mac OS X!

Binaries: ***click here!***
Documentation: ***click here!***

RNPL (Rapid Numerical Prototyping Language):

This is a great tool that takes as input details about the form of an equation (usually a partial differential equation), and some parameters .. and then spits out a C or FORTRAN Code that solves the equation numerically using established iterative numerical techniques. RNPL can also be used for producing skeleton programs and for converting existing programs. Download the source, compile and install. This source has been modified to compile with OS X. Or get my binary distribution. You probably want to install this in **/usr/local** .. to do so, **gunzip rnpl-bin.tar.gz** (if your browser didn't do so already) and then **sudo tar -xvf rnpl-bin.tar -C /** and you are done! You will need to use the F77 flag: **-fno-second-underscore**, for RNPL generated F77 codes to compile properly.

Source: ***rnpl.tar.gz***
Binaries: ***rnpl-bin.tar.gz***, *updated 2003.*
Documentation: ***click here!***

Apple's Xgrid:

This is a new tool that **Apple's ACG** is developing that makes it incredibly easy to run a cluster of Macs by using Apple's technolgies like Rendezvous, etc. It can automatically seek available resources and execute your simulation appropriately. Learn more about it at their site. The current version does not support message passing, but I understand that they are working on it.

Xgrid Site: ***click here!***

Computation Tools :: Other Miscellaneous

Great Internet Mersenne Prime Search:

These are OS X clients for the internet based, distributed computing project, GIMPS. The clients are based on the **GLUCAS** and **MLUCAS** source codes. They benchmark quite well. The sources compile without any

special modifications. The GLUCAS binaries for OS X are available from **this link** (including dual processor versions), while you can get my MLUCAS binary (compiled statically using IBM's Fortran). Keep your Mac crunching!

Sources: ***MLUCAS***
Binaries: ***mlucas.gz****, updated Spring 2004.*

The GRAVSIM Package:

GRAVSIM is a package that contains an ever evolving and hopefully growing set of gravitational N body solvers. These solvers attempt to track the motion of a set of bodies moving under the influence of gravity. They are used in many areas of astrophysical study including planetary motion, star cluster formation and stability and galactic formation, stability and interactions (i.e. colliding galaxies). This package is highly AltiVec and DP optimized!

Source and Binaries: ***click here!***
Documentation: ***click here!***

The FEYNMAN Java Framework:

The framework provides the infrastructure every scientist or engineer needs to develop simulations using the Java programming language. The framework eliminates numerous code writing tasks by allowing the developer to focus on the physical aspects of the simulation and the type of numerical algorithm being implemented. This is a new package and still under development, but has considerable promise.

Source and Binaries: ***click here!***
Documentation: ***click here!***

gcj 3.5 :

GNU Java recently compiled using source code from the GNU CVS servers. This contains gfortran (gcj) version 3.5 compiler. Download my binaries, and **cd** to the download folder. Then **gunzip gcj-bin.tar.gz** (if your browser didn't do so already) and **sudo tar -xvf gcj-bin.tar -C /**. It installs everything in **/usr/local**. You will also need to have Apple's Developer Tools installed. They are included on a separate CD in the retail version of OS X and also available as a free download from **Apple's Developer Site**. Please install the most current version of Developer Tools.

Note: You may need to **ranlib** some libs after you install. The compiler will tell you which ones when you try to use it. In that case, simply do a **sudo ranlib -s** on each such library.

Binaries: ***gcj-bin.tar.gz*** *(Jaguar/Panther), updated July 15th, 2004.*
Documentation: ***click here!***

PIXELGLOW Software's MACSTL :

This is a high performance, (mostly conforming) implementation of the Standard Template Library and associated numerics classes, optimized for the Altivec execution engine on the Power Macintosh G4, G5 and other PowerPC machines. How fast? Just take a **look here**. More on this soon!

Binaries: ***click here!***
Documentation: ***click here!***

Some Useful Commercial Tools:

Altair's PBS Pro: Workload management system.
BioTeam's iNquiry: Cluster configuration tool.
Dauger Research's Pooch: Running parallel simulations using a GUI based application.
Grid Iron's XLR8: An infrastructure to simply the task of parallelization.
Platform LSF: Clustering and workload management solution.
Scientific Paradise: Parallel and distributed computing environment.

Vectorizing and Parallelizing:

If you are interesting in harnessing the full power of a dual G4/G5, i.e. using **AltiVec** and both processors, **click here**.

If you need more, try these links: **fink**, **gnu-darwin.org** and **osxgnu.org**. Good Luck! By the way, if you are looking for a nice, easy to use, 2D and 3D, **plotting package** for OS X, try **pro Fit**. It works great. They have a free trial version, that is not time-limited and is full-featured. It just has some restrictions on the volume of data you can make it handle. Even the full version is very inexpensive!

Back to top

Physics Department
University of Massachusetts at Dartmouth
North Dartmouth, MA 02747.