1 Troubleshooting / Frequently Asked Questions

1.1 Download and setup

1.1.1 Download forbidden

A: Both Google Code and SourceForge are banned from countries on the United States Office of Foreign Assets Control sanction list, including Cuba, Iran, Libya, North Korea, Sudan and Syria; see `http://en.wikipedia.org/wiki/Google_Code` and `http://en.wikipedia.org/wiki/SourceForge`. As a remedy, you might download a tarball from `http://pencil-code.nordita.org/`; see also Sectionx**??**.

1.1.2 When sourcing the '`sourceme.sh`'/'`sourceme.csh`' file or running `pc_setupsrc`, I get error messages from the shell, like 'if: Expression Syntax.' or 'set: Variable name must begin with a letter.'

A: This sounds like a buggy shell setup, either by yourself or your system administrator — or a shell that is even more idiosyncratic than the ones we have been working with.

To better diagnose the problem, collect the following information before filing a bug report to us:

```
uname -a
```

```
/bin/csh -v
```

```
echo $version

echo $SHELL

ps -p $$
```

If you have problems while sourcing the ‘`sourceme`’ script,

unset the `PENCIL_HOME` variable:

for *csh* and similar: `unsetenv PENCIL_HOME`

for *bash* and similar: `unexport PENCIL_HOME; unset PENCIL_HOME`

switch your shell in verbose mode,

for *csh* and similar: `set verbose; set echo`

for *bash* and similar: `set -v; set -x`

then source again.

If you have problems with `pc_setupsrc`, run it with `csh` in verbose mode:

```
xxxx/bin/cshx-vx-xx$PENCIL_HOME/bin/pc_setupsrc
xx
```

1.2 Compilation

1.2.1 Linker can’t find the syscalls functions:

```
xxxxxxxxld:x0711-317xERROR:xUndefinedxsymbol:x.is_nan_c
xxxxxxxxld:x0711-317xERROR:xUndefinedxsymbol:x.sizeof_real_c
xxxxxxxxld:x0711-317xERROR:xUndefinedxsymbol:x.system_c
xxxxxxxxld:x0711-317xERROR:xUndefinedxsymbol:x.get_env_var_c
xxxxxxxxld:x0711-317xERROR:xUndefinedxsymbol:x.get_pid_c
xxxxxxxxld:x0711-317xERROR:xUndefinedxsymbol:x.file_size_c
```

A: The Pencil Code needs a working combination of a Fortran- and a C-compiler. If this is not correctly set up, usually the linker won't find the functions inside the *syscalls* module. If that happens, either the combination of C- and Fortran-compiler is inappropriate (e.g. *ifort* needs *icc*), or the compiler needs additional flags, like *g95* might need the option '`-fno-second-underscore`' and *xlf* might need the option '`-qextname`'. Please refer to Sect.x**??**, Tablex**??**.

1.2.2 Make gives the following error now:

```
xxPGF90-S-0017-Unablextoxopenxincludexfile:xchemistry.hx(nochemistry.f90:x43)
xxxx0xinform,xxx0xwarnings,xxx1xseveres,x0xfatalxforxchemistryxx
xxLinex43xofxthexnochemistryxroutine,xonlyxhasx'contains'.
```

A: This is because somebody added a new module (together with a corresponding *nomodule.f90* and a *module.h* file (chemistry in this case). These files didn't exist before, so you need to say:

```
xxxxxxxxpc_setupsrc
```

If this does not help, say first `make clean` and then `pc_setupsrc`.

1.2.3 How do I compile the PENCIL CODE with the Intel (*ifc*) compiler under *Linux*?

A: The PENCIL CODE should compile successfully with *ifc*x6.x, *ifc*x7.0, sufficiently recent versions of *ifc* 7.1 (you should get the latest version; if yours is too old, you will typically get an 'internal compiler error' during compilation of '`src/hydro.f90`'), as well as with recent versions of *ifort* 8.1 (8.0 may also work).

You can find the *ifort* compiler at `ftp://download.intel.com/software/products/compilers/do`

On many current (as of November 2003) Linux systems, there is a mismatch between the *glibc* versions used by the compiler and the linker. To work around this, use the following flag for compiling

```
xxxxFC=ifcx-i_dynamic
```

and set the environment variable

```
xxxxLD_ASSUME_KERNEL=2.4.1;xexportxLD_ASSUME_KERNEL
```

or

```
xxxxsetenvxLD_ASSUME_KERNELx2.4.1
```

This has solved the problems e.g.xon a system with *glibc-2.3.2* and kernel *2.4.22*.

Thanks to Leonardo J. Milano (`http://udel.edu/~lmilano/`) for part of this info.

1.2.4 I keep getting segmentation faults with '`start.x`' when compiling with *ifort* 8.0

A: There was/is a number of issues with *ifort* 8.0. Make sure you have the latest patches applied to the compiler. A number of things to consider or try are:

Compile with the the '`-static -nothreads`' flags.

Set your stacksize to a large value (but a far too large value may be problematic, too), e. g.

```
xxlimitxstacksizex256mxxulimitx-sx256000
```

Set the environment variable KMP˙STACKSIZE to a large value (like `100M`)

See also `http://softwareforums.intel.com/ids/board/message?board.id=11&message.id=1375`

1.2.5 When compiling with MPI on a Linux system, the linker complains:

```
xxxxmpicomm.o:xInxfunctionx‘mpicomm_mpicomm_init_’:
xxxxmpicomm.o(.text+0x36):xundefinedxreferencextox‘mpi_init_’
xxxxmpicomm.o(.text+0x55):xundefinedxreferencextox‘mpi_comm_size_’
xxxxmpicomm.o(.text+0x6f):xundefinedxreferencextox‘mpi_comm_rank_’xxxx[...]
```

A: This is the infamous *underscore problem*. Your *MPI* libraries have been compiled with *G77* without the option '`-fno-second-underscore`', which makes the *MPI* symbol names incompatible with other Fortran compilers.

As a workaround, use

```
xxMPICOMMx=xmpicomm_
```

in ‘Makefile.local’. Or, even better, you can set this globally (for the given computer) by inserting that line into the file ‘~/.adapt-mkfile.inc’ (see perldoc adapt-mkfile for more details).

1.2.6 Compilation stops with the cryptic error message:

```
xxxxf95xx-O3x-ux-cx.f90.f90xxxxErrorx:xCouldxnotxopenxsourcefilex.f90.f90
xxxxcompilationxabortedxforx.f90.f90x(codex1)xxxxmake[1]:x***x[.f90.o]xErrorx1
```

What is the problem?

A: There are two possibilities:

One of the variables for *make* has not been set, so *make* expands it to the empty string. Most probably you forgot to specify a module in ‘src/Makefile.local’. One possibility is that you have upgraded from an older version of the code that did not have some of the modules the new version has.

Compare your ‘src/Makefile.local’ to one of the examples that work.

One of the variables for *make* has a space appended to it, e. g.xif you use the line

```
MPICOMM = mpicomm_␣
```

(see §x1.2.5) with a trailing blank, you will encounter this error message. Remove the blank. This problem can also occur if you added a new module (and have an empty space after the module name in ‘src/Makefile.src’, i.e.xCHIRAL=nochiral␣), in which case the compiler will talk about “circular dependence” for the file ‘nochiral’.

1.2.7 The code doesn’t compile,

xthere is a problem with *mvar*:

```
xxxxmakexstart.xxrun.xxxxxf95x-O4x-uxxx-cxcdata.f90xxxxError:xcdata.f90,xlinex71:xImp
xxxxxxxxxxxdetectedxatxMVAR@)xxxx[f95xterminatedx-xerrorsxfoundxbyxpassx1]xxxxmake[1]
```

A: Check and make sure that 'mkcparam' (directory '$PENCIL_HOME/bin') is in your path. If this doesn't help, there may be an *empty* 'cparam.inc' file in your 'src' directory. Remove 'cparam.inc' and try again (Note that 'cparam.inc' is automatically generated from the 'Makefile').

1.2.8 Some samples don't even compile,

as you can see on the web, http://www.nordita.org/software/pencil-code/tests.html.

```
xxsamples/helical-MHDturb:xxxxxxCompiling..xxxxxxxxxxxnotxok:xxxxmakexstart.xxrun.xxr
xxmake[1]:xEnteringxdirectoryx'/home/dobler/f90/pencil-code/samples/helical-MHDturb/s
xx/usr/lib/lam/bin/mpif95xx-O3xxx-cxinitcond.f90xx/usr/lib/lam/bin/mpif95xx-O3xxx-cxd
xxxxxxxxusexGravity,xonly:xgravz,xnu_epicyclexxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx^
xxErrorx208xatx(467:density.f90)x:xNoxsuchxentityxinxthexmodule
xxErrorx355x:xInxprocedurexINIT_LNRHOxvariablexNU_EPICYCLExhasxnotxbeenxgivenxaxtype
xxErrorx355x:xInxprocedurexPOLYTROPIC_LNRHO_DISCxvariablexNU_EPICYCLExhasxnotxbeenxgi
xx3xErrorsxxcompilationxabortedxforxdensity.f90x(codex1)xxmake[1]:x***x[density.o]xEr
xxmake[1]:xLeavingxdirectoryx'/home/dobler/f90/pencil-code/samples/helical-MHDturb/sr
xxmake:x***x[code]xErrorx2
```

A: Somebody may have checked in something without having run auto-test beforehand. The problem here is that something has been added in one module, but not in the corresponding no-module. You can of course check with *svn* who it wasx

1.2.9 Internal compiler error with Compaq/Dec F90

The Dec Fortran optimizer has occasional problems with 'nompicomm.f90':

```
xxxxmakexstart.xxrun.xxread_videofiles.xxxxxf90xx-fastx-O5x-tunexev6x-archxev6xx-cxcp
xxxx[...]xxxxf90xx-fastx-O5x-tunexev6x-archxev6xx-cxnompicomm.f90
xxxxotalxvmxx2755568xxxxxxotalxvmxx2765296xxxxxxxxotalxvmxx2775024xxxxotalxvmxx278475
xxxxAssertionxfailure:xxCompilerxinternalxerrorx-xpleasexsubmitxproblemxr...
xxxxxxGEMxASSERTION,xCompilerxinternalxerrorx-xpleasexsubmitxproblemxreport
xxxxFatalxerrorxin:x/usr/lib/cmplrs/fort90_540/decfort90xTerminatedxxxx***xExitx3
xxxxStop.xxxx***xExitx1xxxxStop.
```

A: The occurrence of this problem depends upon the grid size; and the problem never seems to occur with 'mpicomm.f90', except when ncpus=1. The problem can be avoided by switching off the loop transformation optimization (part of the '-O5' optimization), via:

```
xxxx#OPTFLAGS=-fastx-O5x-notransform_loops
```

This is currently the default compiler setting in 'Makefile', although it has a measurable performance impact (some 8% slowdown).

1.2.10 Assertion failure under SunOS

Under SunOS, I get an error message like

```
xxxxuser@sun>xf90x-cxparam_io.f90xxxxAssertionxfailed:xat_handle_table[at_idx].tagx==
xxxxxxxxxxxxxxxxxxxxxxxfilex../srcfw/FWcvrt.c,xlinex4018xxxxf90:xFatalxerrorxinxf90com
```

A: This is a compiler bug that we find at least with Sun's WorkShop Compiler version '5.0 00/05/17 FORTRAN 90 2.0 Patch 107356-05'. Upgrade the compiler version (and possibly also the operating system): we find that the code compiles and works with version 'Sun WorkShop 6 update 2 Fortran 95 6.2 Patch 111690-05 2002/01/17' under SunOS version '5.8 Generic˙108528-11'.

1.2.11 After some dirty tricks I got pencil code to compile with MPI, ...

```
xxx>xBeforexthatxixinstalledxlam-7.1.4xfromxsource.xx
xxGoodnessxgraciousxme,xyouxshouldn't xhavextoxcompilexyourxownxMPIxlibrary.
```

A: Then don't use the old LAM-MPI. It is long superseded by open-mpi now. Open-mpi doesn't need a daemon to be running. I am using the version that ships with Ubuntu (e.g. 9.04):

```
xxxxfrenesi:~>xaptitudex-wx210xsearchxopenmpix|xgrepx'^i'xx
xxixxxlibopenmpi-devx-xhighxperformancexmessagexpassingxlibraryx--xheaderxfiles
xxixAxlibopenmpi1xxxx-xhighxperformancexmessagexpassingxlibraryx--xsharedxlibrary
xxixxxopenmpi-binxxxx-xhighxperformancexmessagexpassingxlibraryx--xbinaries
xxixAxopenmpi-commonx-xhighxperformancexmessagexpassingxlibraryx--xcommonxfiles
xxixxxopenmpi-docxxxx-xhighxperformancexmessagexpassingxlibraryx--xmanxpages
```

Install that and keep your configuration (Makefile.src and getconf.csh) close to that for 'frenesi' or 'norlx50'. That should work.

1.2.12 Error: Symbol 'mpi˙comm˙world' at (1) has no IMPLICIT type

```
xxIxinstalledxthexpencilxcodexonxUbuntuxsystemxandxtestedx"run.csh"
xxinx...\samples\conv-slab.xxHerexthexcodexworkedxprettyxwell.
xxNevertheless,xrunningx(auto-test),xIxfoundxtherexarexsomexerrors.xxxxThexmessagesxa
xxError:xSymbolx'mpi_comm_world'xatx(1)xhasxnoxIMPLICITxtypexxFatalxError:xErrorxcoun
xxmake[2]:x***x[mpicomm_double.o]xErrorx1xxmake[2]:xLeavingxdirectory
xx`/home/pkiwan/Desktop/pencil-code/samples/2d-tests/selfgravitating-shearwave/src'
xxmake[1]:x***x[code]xErrorx2xxmake[1]:xLeavingxdirectory
xx`/home/pkiwan/Desktop/pencil-code/samples/2d-tests/selfgravitating-shearwave/src'
xxmake:x***x[default]xErrorx2xxxxFinally,x###xauto-testxfailedx###xxxxWillxitxbexOK?x
```

A: Thanks for letting me know about the status, and congratulations on your progress! Those tests that fail are those that use MPI. If your machine is a dual or multi core machine, you could run faster by running under MPI. But this is probably not crucial for you at this point. (I just noticed that there is a ToDo listed in the auto-test command to implement the option not to run the MPI tests, but this hasn't been done yet. So I guess you can start with the science next.

1.2.13 Error: Can't open included file 'mpif.h'

It always worked, but now, after some systems upgrade, I get

```
xxxxxgfortranx-O3x-oxmpicomm.ox-cxmpicomm.f90xxxxxError:xCan'txopenxincludedxfilex'mp
```

When I say `locate mpif.h` I only get things like

```
xxxxx/scratch/ntest/1.2.7p1-intel/include/mpif.h
```

But since I use `FC=mpif90` I thought I don't need to worry.

A: Since you use `FC=mpif90` there must definitely be something wrong with their setup. Try `mpif90 -showme` or `mpif90 -show`; the '-I' option should say where it looks for 'mpif.h'. If those directories don't exist, it's no wonder that it doesn't work, and it is time to complain.

1.3 Pencil check

1.3.1 The pencil check complains for no reason.

A: The pencil check only complains for a reason.

1.3.2 The pencil check reports MISSING PENCILS and quits

A: This could point to a serious problem in the code. Check where the missing pencil is used in the code. Request the right pencils, likely based on input parameters, by adapting one or more of the `pencil_criteria_MODULE` subroutines.

1.3.3 The pencil check reports unnecessary pencils

The pencil check reports `possible overcalculation... pencil rho ( 43) is requested, but does not appear to be required!`

A: Such warnings show that your simulation is possibly running too slowly because it is calculating pencils that are not actually needed. Check in the code where the unnecessary pencils are used and adapt one or more of the `pencil_criteria_MODULE` subroutines to request pencils only when they are actually needed.

1.3.4 The pencil check reports that most or all pencils are missing

A: This is typically a thing that can happen when testing new code development for the first time. It is usually an indication that the reference `df` changes every time you call `pde`. Check whether any newly implemented subroutines or functionality has a "memory", i.e. if calling the subroutine twice with the same `f` gives different output `df`.

1.3.5 Running the pencil check triggers mathematical errors in the code

A: The pencil check puts random numbers in `f` before checking the dependence of `df` on the chosen set of pencils. Sometimes these random numbers are inconsistent with the physics and cause errors. In that case you can set `lrandom_f_pencil_check=F` in `&run_pars` in '`run.in`'. The initial condition may contain many idealized states (zeros or ones) which then do not trigger pencil check errors when `lrandom_f_pencil_check=F`, even if pencils are missing. But it does prevent mathematical inconsistencies.

1.3.6 The pencil check still complains

A: Then you need to look into the how the code and the pencil check operate. Reduce the problem in size and dimensions to find the smallest problem that makes the pencil check fail (e.g. 1x1x8 grid points). At the line of '`pencil_check.f90`' when a difference is found between `df_ref` and `df`, add some debug lines telling you which variable is inconsistent and in what place. Often you will be surprised that the pencil check has correctly found a problem in the simulation.

1.3.7 The pencil check is annoying so I turned it off

A: Then you are taking a major risk. If one or more pencils are not calculated properly, then the results will be wrong.

1.4 Running

1.4.1 Why does '`start.x`' / '`start.csh`' write data with periodic boundary conditions?

A: Because you are setting the boundary conditions in '`run.in`', not in '`start.in`'; see Sect.x**??**. There is nothing wrong with the initial data — the ghost-zone values will be re-calculated during the very first time step.

1.4.2 csh problem?

Q: On some rare occasions we have problems with csh not being supported on other machines. (We hope to fix this by contacting the responsible person, but may not be that trivial today!) Oliver says this is a well known bug of some years ago, etc. But maybe in the long run it would be good to avoid csh.

A: These occasions will become increasingly frequent, and eventually for some architectures, there may not even be a csh variant that can be installed.

We never pushed people to use `pc_run` and friends (and to report corresponding bugs and get them fixed), but if we don't spend a bit of effort (or annoy users) now, we create a future emergency, where someone needs to run on some machine, but there is no csh and he or she just gets stuck.

We don't have that many csh files, and for years now it should be possible to compile run without csh (using `bin/pc_run`) — except that people still fall back on the old way of doing things. This is both cause and consequence of the 'new' way not being tested that much, at least for the corner cases like '`RERUN`', '`NEWDIR`', '`SCRATCH_DIR`'.

1.4.3 '`run.csh`' doesn't work:

```
xxxxInvalidxcharacterx’’’xinxNAMELISTxinputxxxxProgramxterminatedxbyxfatalxI/Oxerrorx
```

A: The string array for the boundary condition, e.g.x*bcx* or *bcz* is too long. Make sure it has exactly as many elements as *nvar* is big.

1.4.4 Namelist problem under IRIX

Under IRIX, I get

```
xxxxlib-4001x:xUNRECOVERABLExlibraryxerrorxxxxxxEncounteredxduringxaxnamelistxREADxfr
xxxxFortranxunitx1xisxconnectedxtoxaxsequentialxformattedxtextxfile:x"run.in"xxxxIOTx
```

A: This is a compiler bug that has been found at least with the MIPSpro F90 compiler version 7.3.1.3m. The problem seems to have been fixed in version 7.4.20m.

The error comes and goes, depending on the configuration (and possibly even the input parameters) you are using. Until SGI fix their compiler, you can experiment with adding new variables to the module *Param˙IO*; this has solved the problem once for us. If this trick does not help, you will need to turn your namelist input (at least '`run.in`' into Fortran statements, include them into a replacement version of '`param_io.f90`', and recompile each time you make changes.

1.4.5 Code crashes after restarting

```
xx>xx>xremovingxmu_rxfromxthexnamelistxjustx‘likexthat’xmakesxthexcodexx>xx>xbackward
xx>xThatxmeansxthatxwexcanxneverxgetxridxofxaxparameterxinxstart.inxoncexwexx>xhavexi
```

A: In the current implementation, without a corresponding cleaning procedure, unfortunately yes.

Of course, this does not affect users' private changes outside the central svn tree.

1.4.6 auto-test gone mad...?

Q: Have you ever seen this before:

```
xxxxgiga01:/home/pg/n7026413/cvs-src/pencil-code/samples/conv-slab>xauto-testxxxx.xx
xxxx/home/pg/n7026413/cvs-src/pencil-code/samples/conv-slab:xxxxxxxCompiling..xxxxxx
xxxxxxxxxxxxNoxdataxdirectory;xgeneratingxdatax->x/var/tmp/pencil-tmp-25318
xxxxxxxxStarting..xxxxxxxxxxxxokxxxxxxxxRunning..xxxxxxxxxxxxxok
xxxxxxxxValidatingxresults..MalformedxUTF-8xcharacterx(unexpectedxcontinuation
xxxxbytex0x80,xwithxnoxprecedingxstartxbyte)xinxsplitxat
xxxx/home/pg/n7026413/cvs-src/pencil-code/bin/auto-testxlinex263.
xxxxMalformedxUTF-8xcharacterx(unexpectedxcontinuationxbytex0x80,xwithxno
xxxxprecedingxstartxbyte)xinxsplitxatxxxx/home/pg/n7026413/cvs-src/pencil-code/bin/au
```

A: You are running on a RedHat 8 or 9 system, right?

Set `LANG=POSIX` in your shell's startup script and life will be much better.

1.4.7 Can I restart with a different number of cpus?

Q: I am running a simulation of nonhelical turbulence on the cluster using MPI. Suppose if I am running a 128^3 simulation on 32 cpus/cores i.e.

```
xxxxinteger,xparameterx::xncpus=32,nprocy=2,nprocz=ncpus/nprocy,nprocx=1
xxxxinteger,xparameterx::xnxgrid=128,nygrid=nxgrid,nzgrid=nxgrid
```

And I stop the run after a bit. Is there a way to resume this run with different number of cpus like this :

```
xxxxinteger,xparameterx::xncpus=16,nprocy=2,nprocz=ncpus/nprocy,nprocx=1
xxxxinteger,xparameterx::xnxgrid=128,nygrid=nxgrid,nzgrid=nxgrid
```

I understand it has to be so in a new directory but making sure that the run starts from where I left it off in the previous directory.

A: The answer is no, if you use the standard distributed io. There is also parallel io, but I never used it. That would write the data in a single file, and then you could use the data for restart in another processor layout.

1.4.8 Can I restart with a different number of cpus?

Q: Is it right that once the simulation is resumed, pencil-code takes the last data from var.dat (which is the current snapshot of the fields)? If that is true, then, is it not possible to give that as the initial condition for the run in the second directory (with changed ”ncpus”)? Is there a mechanism already in place for that?

A: Yes, the code restarts from the last var.dat. It is written after a successful completion of the run, but it crashes or you hit a time-out, there will be a var.dat that is overwritten every isave timesteps. If the system stops during writing, some var.dat files may be corrupt or have the wrong time. In that case you could restart from a good VAR file, if you have one, using, e.g.,

```
xxxxrestart-new-dir-VARx.x46
```

where 46 is the number of your VAR file, i.e., VAR46 im this case. To restart in another directory, you say, from the old run directory,

```
xxxxrestart-new-dirx../another_directory
```

Hope this helps. Look into pencil-code/bin/restart-new-dir to see what it is doing.

1.4.9 fftxxyzxparallelx3D: nygrid needs to be an integer multiple...

Q: I just got an:

```
xxxxfft_xyz_parallel_3D:xnygridxneedsxtoxbexanxintegerxmultiplexofxnprocy*nprocz
```

In my case, nygrid=2048, nprocy=32, and nprocz=128, so nprocy*nprocz=4096. In other words, 2048 needs to be a multiple of 4096. But isn’t this the case then?

A: No, because 2048 = 0.5 * 4096 and 0.5 is not an integer. Maybe try either setting nprocz=64 or nprocy=64. You could compensate the change of ncpus with the x-direction. For 2048^3 simulations, nprocy=32 and nprocz=64 would be good.

1.4.10 Unit-agnostic calculations?

Q: The manual speaks about unit-agnostic calculations, stating that one may choose to interpret the results in any (consistent) units, depending on the problem that is solved at hand. So, for example, if I chose to run the ‘`2d-tests/battery_term`’ simulation for an arbitrary number of time-steps and then choose to examine the diagnostics, am I correct in assuming the following:

1)x[Brms]x=xGaussx(asxoutputxbyxunit_magnetic,xbeforexthexrunxbegins)
x2)x[t]x=xsx(sincexthexdefaultxunitxsystemxisxleftxasxCGS)
x3)x[urms]x=xcm/sx(again,xasxoutputxbyxunit_velocity,xbeforexthexrunxbegins)
x4)xandxetc.xforxthexunitsxofxthexotherxdiagnostics

A: Detailed correspondence on this item can be found on:
https://groups.google.com/forum/?fromgroups#!topic/pencil-code-discuss/zek-uYNbgXI Th
material on unit systems under http://www.nordita.org/~brandenb/teach/PencilCode/MixedTo
with a link to http://www.nordita.org/~brandenb/teach/PencilCode/material/AlfvenWave_SIu
Below is a pedagogical response from Wlad Lyra:

Inxthexsamplexbattery-term,xthexsoundxspeedxcs0=1xsetsxthexunitxof
xvelocity.xTogetherxwithxthexunitxofxlength,xthatxsetsxyourxunitxof
xtime.xThexunitxofxmagneticxfieldxfollowsxfromxthexunitxofxvelocity,
xdensity,xandxyourxchoicexofxmagneticxpermittivity,xaccordingxtoxthe
xdefinitionxofxthexAlfvenxvelocity.

Ifxyouxarexassumingxcgs,xyouxarexsayingxthatxyourxsoundxspeedxcs0=1
xactuallyxmeansx[U]=1xcm/s.xYourxunitxofxlengthxisxequivalentlyx1xcm,
xandxthereforexthexunitxofxtimexisx[t]x=x[L]/[U]=1xs.xThexunitxof
xdensityxisx[rho]x=x1xg/cm^3.xSincexinxcgsxvA=B/sqrt(4*pix*xrho),xyour
xunitxofxmagneticxfieldxisx[B]x=x[U]x*xsqrt([rho]x*x4*pi)x~=x3.5
xsqrt(g/cm)x/xsx=x3.5xGauss.

IfxinsteadxyouxarexassumingxSI,xyouxhavexcs0=1xassumingxthatxmeans
x[U]=1xm/sxandxrho0=1xassumingxthatxtoxmeanx[rho]=1xkg/m^3.xUsingx[L]=1
xm,xyouxhavexstillx[t]=1xs,xbutxnowxwhatxappearsxasxB=1xinxyourxoutput
xisxactuallyx[B]x=x[U]x*xsqrtx(mux*x[rho])x=x1xm/sx*xsqrt(4*pix*x1e-7
xN*A-2xx1xkg/m^3)xx~=x0.0011210xkg/(s^2*A)x~x11xGauss.

Youxcanxmakexitxmorexinterestingxandxusexunitsxrelevantxtoxthe xproblem.xSayxyouxarexatxthexphotospherexofxthexSun.xYouxmayxwantxto xusexdimensionlessxcs0=1xmeaningxaxsoundxspeedxofx10xkm/s.xxYour xappropriatexlengthxcanxbexaxmegameter.xNowxyourxtimexunitxis x[t]=[L]/[U]x=x1e3xkm/x10xkm/sx=x10^2xs,xi.e.,xroughlyx1.5xminute.xFor xdensity,xassumexrho=2x10-4xkg/m^3,xtypicalxofxthexsolarxphotosphere. xYourxunitxofxmagneticxfieldxisxthereforex[B]x=x[U]x*xsqrt([rho]x* x4*pi)x=x1e6xcm/sx*xsqrt(4*pix*x2e-7xg/cm^3)x~x1585.33xGauss.

Noticexthatxforxmu0=1xandxrho0=1xyouxsimplyxhavexvA=B.xThenxyouxcan
xconvenientlyxsetxthexfieldxstrengthxbyxyourxchoicexofxplasmaxbetax(=
x2*cs^2/vA^2).xThere'sxaxreasonxwhyxwexlikexdimensionlessxquantities!

1.5 Visualization

1.5.1 'start.pro' doesn't work:

xxxxReadingxgrid.dat..xxxxReadingxparam.nml..xxxx\%xExpressionxmustxbexaxstructurexin
xxxx\%xExecutionxhaltedxat:xx\$MAIN\$xxxxxxxxxxx104
xxxx/home/brandenb/pencil-code/runs/forced/hel1/../../../idl/start.pro

A: You don't have the subdirectory 'data' in your IDL variable *!path*. Make sure you source 'sourceme.csh'/'sourceme.sh' or set a sufficient IDL path otherwise.

1.5.2 'start.pro' doesn't work:

Isn't there some clever (or even trivial) way that one can avoid the annoying error messages that one gets, when running e.g. ".r rall" after a new variable has been introduced in "idl/varcontent.pro"? Ever so often there's a new variable that can't be found in my param2.nml – this time it was IECR, IGG, and ILNTT that I had to circumventx

A: The simplest solution is to invoke 'NOERASE', i.e.xsay

xxtouchxNOERASExxstart.csh

or, alternatively, start_run.csh. What it does is that it reruns src/start.x with a new version of the code; this then produces all the necessary auxiliary files, but it doesn't overwrite or erase the 'var.dat' and other 'VAR' and 'slice' files.

1.5.3 Something about tag name undefined:

Q: In one of my older run directories I can't read the data with idl anymore. What should I do? Is says something like

```
xxReadingxparam.nml..xx%xTagxnamexLEQUIDISTxisxundefinedxforxstructurex<Anonymous>.
xx%xExecutionxhaltedxat:x$MAIN$xxxxxxxxxxxx182xxxx/people/disk2/brandenb/pencil-code/
```

A: Go into '`data/param.nml`' and add , `LEQUIDIST=T` anywhere in the file (but before the last slash).

1.5.4 Something INC in start.pro

Q: start doesn't even work:

```
xx%xCompiledxmodule:x$MAIN$.xxnname=xxxxxx11xxReadingxgrid.dat..xxReadingxparam.nml..
xxCan'txlocatexNamelist.pmxinxINCx(INCxcontains:x/etc/perlx/usr/local/lib/perl/5.8.4x
xxBEGINxfailed--compilationxabortedxatx/home/brandenb/pencil-code/bin/nl2idlxlinex49.
```

A: Go into '`$PENCIL_HOME`' and say `svn up sourceme.csh` and/or `svn up sourceme.sh`. (They were just out of date.)

1.5.5 nl2idl problem when reading param2.nml

Q: Does anybody encounter a backward problem with nl2idl? The file param*.nml files are checked in under '`pencil-code/axel/couette/SStrat128a_mu0.20_g2`' and the problem is below.

```
xxxatx/people/disk2/brandenb/pencil-code/bin/nl2idlxlinex120
xxHCOND0=x0.0,HCOND1=x1.000000,HCOND2=x1.000000,WIDTHSS=x1.192093E-06,MPOLY0=
xx^------xxHERExxxatx/people/disk2/brandenb/pencil-code/bin/nl2idlxlinex120
```

A: The problem is the stupid ifc compiler writing the following into the namelist file:

```
xxxxCOOLING_PROFILE='gaussianxxxxxxxxxxxxxxxxx',COOLTYPE='Temp
xxxx'COOL=x0.0,CS2COOL=x0.0,RCOOL=x1.000000,WCOOL=x0.1000000,FBOT=x0.0,CHI_T=x0.0
```

If you add a comma after the closing quote:

```
xxxxCOOLING_PROFILE='gaussianxxxxxxxxxxxxxxxxx',COOLTYPE='Temp
xxxx',COOL=x0.0,CS2COOL=x0.0,RCOOL=x1.000000,WCOOL=x0.1000000,FBOT=x0.0,CHI_T=x0.0
```

things will work.

Note that ifc cannot even itself read what it is writing here, so if this happened to occur in param.nml, the code would require manual intervention after each start.csh.

1.5.6 Spurious dots in the time series file

Q: Wolfgang, you explained it to me once, but I forget. How can one remove spurious dots after the timestep number if the time format overflows?

A: I don't know whether it exists anywhere, but it's easy. In Perl you'd say

```
xxxxperlx-pex's/^(\s*[-0-9]+)\.([-0-9eEdD])/$1x$2/g'
```

and in sed (but that's harder to read)

```
xxxxsedx's/^\(x*[-0-9]\+\)\.\([-0-9eEdD]\)/\1x\2/g'
```

1.6 General questions

1.6.1 "Installation" procedure

Why don't you use GNU *autoconf/automake* for installation of the PENCIL CODE?

A: What do you mean by "installation"? Unlike the applications that normally use *autoconf*, the *Pencil Code* is neither a binary executable, nor a library that you compile once and then dump somewhere in the system tree. *Autoconf* is the right tool for these applications, but not for numerical codes, where the typical compilation and usage pattern is very different:

You have different directories with different '`Makefile.local`' settings, recompile after introducing that shiny new term in your equations, etc. Moreover, you want to sometimes switch to a different compiler (but just for that run directory) or another *MPI* implementation. Our `adapt-mkfile` approach gives you this flexibility in a reasonably convenient way, while doing the same thing with *autoconf* would be using that system against most of its design principles.

Besides, it would really get on my (WD's) nerves if I had to wait two minutes for *autoconf* to finish before I can start compiling (or maybe 5–10 minutes if I worked on a NEC machinex).

Finally, if you have ever tried to figure out what a '`configure`' script does, you will appreciate a comprehensible configuration system.

1.6.2 Small numbers in the code

What is actually the difference between epsi, tini and tiny?

A:

```
F90xhasxtwoxfunctionsxepsilon()xandxtiny(),xwith
x
xxxepsilon(x)x=x1.1920929e-07
xxxtiny(x)xxxx=x1.1754944e-38
x(andxthenxtherexisxhuge(x)x=x3.4028235e+38)
xforxaxsingle-precisionxnumberxx.
x
xepsilon(x)xisxthexsmallestxnumberxthatxsatisfies
xxx1+epsilon(1.)x/=x1x,
xwhilextiny(x)xisxthexsmallestxnumberxthatxcanxbexrepresentedxwithout
xprecisionxloss.
x
xInxthexcodexwexhavexvariantsxhereof,
xxxxepsi=5*epsilon(1.0)
xxxxtini=5*tiny(1.0)
xxxxhuge1=0.2*huge(1.0)
xthatxhavexaddedxsafetyxmargins,xsoxwexdon't xhavextoxthinkxaboutxdoing
xthingsxlikex1/tini.
x
xSoxinxsub.f90,
xxx-xxxxxxevrx=xevrx/xspread(r_mn+epsi,2,3)
xdidx(minimally)xaffectxthexresultxforxr_mn=0(1),xwhilexthexcorrectxversion
xxx+xxxxxxevrx=xevrx/xspread(r_mn+tini,2,3)
xonlyxavoidsxoverflow.
```

1.6.3 Why do we need a `/lphysics/` namelist in the first place?

Wolfgang answered on 29 July 2010: "'`cdata.f90`' has the explanation"

```
xxxx!xConstantx'parameters'xcannotxoccurxinxnamelists,xsoxinxorderxtoxgetxthe
xxxx!xnowxconstantxmodulexlogicalsxintoxthexlphysicsxnamexlist...
xxxx!xWexhavexsomexproxiesxthatxarexusedxtoxinitializexprivatexlocalxvariables
xxxx!xcalledxlhydroxetc,xinxthexlphysicsxnamelist!
```

So the situation is this: we want to write parameters like ldensity to param.nml so IDL (and potentially octave, python, etc.) can know whether density was on or not. To avoid confusion, we want them to have exactly their original names. But we cannot assemble the original ldensity etc. constants in a namelist, so we have to define a local ldensity variable. And to provide it with the value of the original cdata.ldensity, we need to transfer the value via *ldensity˙var*. That's pretty scary, although it seems to work fine. I can track the code back to the big *eos˙merger* commit, so it may originate from that branch. One obvious problem is that you have to add

code in a number of places (the `ldensity` → *ldensity_var* assignment and the local definition of ldensity) to really get what you need. And when adding a new boolean of that sort to '`cdata.f90`', you may not even have a clue that you need all the other voodoo.

There may be a cleaner solution involving generated code. Maybe something like

```
xxxxlogicalx::xldensityx!xINCLUDE_IN_LPHYSICS
```

could later generate code (in some param_io_extra.inc file) that looks like this:

```
xxxxwrite(unit,x*)x'ldensityx=x',xldensity
```

i.e.xwe can manually write in namelist format. But maybe there are even simpler solutions?

1.6.4 Can I run the code on a Mac?

A: Macs work well for Linux stuff, except that the file structure is slightly different. Problems when following Linux installs can usually be traced to the PATH. For general reference, if you need to set an environment variable for an entire OS-X login session, google environment.plist. That won't be needed here.

For a Mac install, the following should work:

Install Dev Tools (an optional install on the MacOS install disks). Unfortunately, last time I checked the svn version that comes with DevTools is obsolete. So:

Install MacPorts (download from web). Note that MacPorts installs to a non-standard location, and will need to be sourced. The installation normally drops an appropriate line in .profile. If it does so, make sure that that line gets sourced. Otherwise

```
xxxxexportxPATH=/opt/local/bin:/opt/local/sbin:$PATHxxxxexportxMANPATH=/opt/local
```

Install g95 (download from web). Make sure it is linked in /bin.

execute macports svn install

download the pencil-code and enjoy.

Note: the above way to get svn works. It takes a while however, so there are certainly faster ways out there. If you already have a non-obsolete svn version, use that instead.

1.6.5 Pencil Code discussion forum

Do I just need to send an email somewhere to subscribe or what?

A" The answer is yes; just go to:

`xxxxxhttp://groups.google.com/group/pencil-code-discuss`

1.6.6 The manual

It would be a good idea to add this useful information in the manual, no?

A: When you have added new stuff to the code, don't forget to mention this in the '`pencil-code/doc/manual.tex`' file.

Again, the answer is yes; just go to:

`xxxxxcdxpencil-code/doc/xxxxxvixmanual.texxxxxxsvnxcix-mx"explanationsxaboutxaxnewxmo`