Joint ICTP-TWAS Caribbean School on Electronic Structure Fundamentals and Methodologies, Cartagena, Colombia (2012).

# Installing the Quantum ESPRESSO distribution

Coordinator: A. D. Hernández-Nieves

# Installing the Quantum ESPRESSO distribution

## Outline:

1) How to get the QE distribution

2) how to install it, i.e. produce the executables;

3) what you need for a successful installation, and

4) what to do if the installation is NOT successful!

# Requirements for Quantum ESPRESSO installation

Things that you MUST have on your machine:

- Unix, or a Unix-like environment (a shell and the Make utility): Mac OS-X ok, Windows with "cygwin" ok
- a working fortran-95 compiler (most "fortran-90" compilers are ok, "g77" is not)
- a working C compiler ("gcc" is ok)

*All hardware is supported, as long as it has what is listed above.*

# To run in parallel

In order to run in parallel, you MUST have at least one of the following:

- working MPI (Message-passing Interface): parallel compiler ("mpif90" or similar scripts), MPI libraries, run-time environment ("mpirun" or similar launchers)

- OpenMP-capable compiler and autothreading mathematical libraries (for multi-core CPUs)

# Libraries:

Quantum ESPRESSO uses and provides a copy of the following external libraries:

- BLAS (Basic Linear Algebra Subroutines): http://www.netlib.org/blas
- LAPACK (Linear Algebra Package): http://www.netlib.org/lapack
- FFTW (Fast Fourier-Transform package): http://www.fftw.org

Things that you SHOULD have on your machine for real-life usage:

- Fast mathematical libraries
- For parallel execution: fast interprocess communication hardware and software

# Basic Installation

1) Choose and create a directory where to install QE.

*It should be on a file system that is local to the PC you are using: sometimes the home directory in a PC cluster is accessed via the network (NFS). You will need enough disk space and a large enough disk quota: sometimes the home directory is small, or has a quota enforced. You will need a few tens of Mb just to compile the distribution.*

2) Download in the chosen directory the package

espresso-5.0.1.tar.gz

*The suffix ".gz" means "compressed by gzip" (a free utility found on most Unix machines).*
*The suffix ".tar" means "archived by tar" (the standard Unix command for archiving and retrieving files)*

# Basic Installation

3) Uncompress and unpackage the file:

```
$ tar -zxvf espresso-5.0.1.tar.gz
```

Some machines may not support the "z" flag (meaning "uncompress files compressed by gzip"). In this case:

```
$ gunzip espresso-5.0.1.tar.gz
$ tar -xvf espresso-5.0.1.tar
```

A directory "espresso-5.0/" will be created, containing many files and other directories. In the following this directory will be referred to as $espresso_dir.

# Basic Installation

4) Execute initialization steps, if needed.

For instance, in order to enable the usage of Intel compiler and MKL libraries, you should set a number of variables; for parallel execution, you should add a parallel compiler to your path.

How to do this *depends on the specific machine*. Sometimes one has to use the command "module", as in the following example:

```
$ module load intel/11.1
$ module load mkl/11.1
```

# Basic Installation

5) Enter the "espresso-5.0/" directory and execute "./configure":

```
$ cd espresso-5.0/
$ ./configure
```

"configure" is a wrapper, calling the "install/configure" script, that tries to guess your machine and to choose compilation and linking options accordingly.

"install/configure" is generated from "install/configure.ac" using the complex but well-known Unix utility "autoconf".

The result of "configure" is a file called "make.sys" containing the compilation and linking options.

You may want to have a look at it to verify what "configure" thinks about your system. Several options to "configure" may (and sometimes, have to) be specified (see below).

# Basic Installation

- If you get a message "architecture xxx not recognized", specify option "ARCH=...", such as e.g. (for a BlueGene machine):

  $ ./configure ARCH=ppc64-bg

- If "configure" selects a compiler you don't like (or one that doesn't like Quantum ESPRESSO) specify option "F90=..." or "MPIF90=...", e.g.:

  $ ./configure F90=gfortran

- If everything is fine you should get a bunch of (mostly obscure and irrelevant) messages but no error. Read the last lines: you may need to understand them if something goes wrong at compilation stage, or if you need to boost performances. BEWARE: if you have a parallel compiler in you path, "configure" will choose it. If the parallel and serial compilers do not match, you may run into trouble (check for the presence of "WARNING" in the output of "configure"!). to produce a serial executable use

  $ ./configure --disable-parallel

- to produce a parallel executable with OpenMP enabled use

  $ ./configure --enable-openmp

# Basic Installation

6) Compile the packages you need. Let us start from PWscf:

```
$ make pw
```

"make" is another complex but standard unix utility that compiles what is needed in the way it is needed and in the order in which is needed (in Fortran-90 you need to compile modules before programs that use them, for instance).

The configuration files for "make" are either coming with the package or are generated by "configure".

If everything goes well, executables will appear in "bin/":

```
$ ls bin/*.x
```

7) Quick test (for "pw.x" executable only) to verify that things look good:

```
$ cd PW/tests
$ ./check-pw.x.j
```

# Basic Installation

8) Compile other packages. "make" with no argument yields a list of targets, i.e., packages to be compiled.

If you want to compile everything:

```
$ make all
```

Packages that are not contained in the base tarball will be download from the net. **Beware:** this will work *only* if you have direct access to the internet, and working "wget" or "curl" commands.

If not, you will need to download the required packages from http://www.quantum-espresso.org *and* to put them into "archive/".

Then you can do "make all" again.

# Requirements for installation of a FAST executable

Most of the CPU time in a typical run is spent in:

- Fast Fourier Transform
- matrix-matrix and matrix-vector multiplications (BLAS)
- solution of linear systems, diagonalizations (LAPACK)

If you want a fast executable, you MUST provide:

- machine-optimized BLAS (VERY important) and LAPACK libraries (important)
- for some architectures (in particular IBM SP): machine-optimized FFT library
- Most vendors provide highly optimized BLAS and LAPACK replacements and FFT libraries (different from FFTW)

# Machine-optimized libraries: BLAS, LAPACK

The "configure" scripts can recognize and use the following architecture-specific replacements for BLAS and LAPACK:

- MKL for Intel/AMD PCs
- ACML for AMD (e.g. Opterons) PCs
- SCSL for SGI Altix
- essl for IBM RISC and SP machines
- sunperf for Sun workstations and parallel machines

If none of these is available, you should try the ATLAS (Automatically Tuned Linear Algebra Subroutines) library: http://math-atlas.sourceforge.net

# Machine-optimized libraries: FFT

Quantum ESPRESSO can use the following architecture-specific mathematical libraries instead of FFTW:

- MKL for Intel CPUs
- ACML for AMD CPUs
- essl for IBM SP machines
- ASL for NEC-SX machines
- SCSL for SGI Altix
- sunperf for Sun workstations and parallel machines

You may want to check for the presence of preprocessing options "-D" in the definition of DFLAGS in file "make.sys".

For recent MKL versions, it is sufficient to have "-D__FFTW3" in DFLAGS, to have MKL properly linked, and nothing in FFTLIBS.

If none of the above is available, a precompiled FFTW v.3 library or the built-in FFTW library contained in the distribution will be fine. Their performances are quite decent. Only ESSL on IBM machines beats them by a large margin.

# ScaLAPACK

"configure" by default tries to compile support for ScaLAPACK (check for "-D__SCALAPACK in DFLAGS, file "make.sys").

For machines with recent Intel MKL libraries, you may need to specify

```
$ ./configure --with-scalapack=intelmpi
```

- Or

```
$ ./configure --with-scalapack=openmpi
```

depending upon whether you are linking Intel version of MPI or OpenMPI.

# Problematic cases

Most installation problems falls into one of the following categories:

1) fortran-95 compiler not installed or not working. For Intel/AMD PC's, 32- and 64-bit CPU's:

- download or buy the Intel compiler (Linux version free for academic use)
- download the GNU gfortran compiler (recent versions work quite well) or the g95 compiler (no longer maintained)
- buy a commercial compiler (PGI, Lahey, Pathscale are supported; Absoft may or may not work)

For all other machines:

- try gfortran or g95 (they are designed to be portable and in principle should work on any architecture)
- buy a commercial compiler

# Problematic cases

2) fortran-95 compiler unable to compile Quantum ESPRESSO: "internal compiler error" or some other strange errors occur. Possible solutions:

- upgrade to the latest available compiler version, or install the patches provided by the vendor (if any)
- install and try a different compiler
- try to figure out what to do: sometimes lowering the optimization level of a specific routine, or splitting a routine into pieces, or simply moving around some lines, will do the job.

3) "configure" unable to locate some external libraries you know are there. Possible solutions:

- bug your system manager until he/she sets up properly the libraries, or puts them in a sane location
- manually provide the correct location of libraries (you will have to use "configure" options, or to edit file "make.sys")

# Problematic cases

4) Parallel Machines (especially PC clusters):

- configure says that there is a mismatch between the serial and parallel compiler. Possible solution:
  (a) use "./configure F90=..." of "MPIF90=..." to force the choice of the correct compiler
  (b) modify your $PATH so that "configure" finds the good combination of serial and parallel compilers
- mpif90 doesn't work. Solution:
  a) bug your system manager until he/she sets up properly the script
- you manage to compile but the executable yields strange errors in parallel execution. Solution:
  bug your system manager until he/she install a working combination of libraries and of compiler

In all cases mentioned above, see also the documentation on:
http://www.quantum-espresso.org

# When everything else fails…

- ...read the manual and the documentation in directory Doc/
- search the pw_forum mailing list (follow the link in the Contacts section of http://www.quantum-espresso.org)
- post to the pw_forum mailing list (subscribe by following the link in the Contacts section of http://www.quantum-espresso.org)

# If the Installation was Successful

- ...you can start reading the manual and the documentation in directory Doc/
- For example the Input File Description: .../Doc/INPUT_PW.html

# Input File Description: …/Doc/INPUT_PW.html

INPUT_PW.html

file://…/espresso-5.0/PW/Doc/INPUT_PW.html

## Input File Description

### Program: pw.x / PWscf / Quantum Espresso

**TABLE OF CONTENTS**


INTRODUCTION

&CONTROL

calculation | title | verbosity | restart_mode | wf_collect | nstep | iprint | tstress | tprnfor | dt | outdir | wfcdir | prefix | lkpoint_dir | max_seconds | etot_conv_thr | forc_conv_thr | disk_io | pseudo_dir | tefield | dipfield | lelfield | nberrycyc | lberry | gdir | nppstr

&SYSTEM

ibrav | celldm | A | B | C | cosAB | cosAC | cosBC | nat | ntyp | nbnd | tot_charge | tot_magnetization | starting_magnetization | ecutwfc | ecutrho | ecutfock | nr1 | nr2 | nr3 | nr1s | nr2s | nr3s | nosym | nosym_evc | noinv | no_t_rev | force_symmorphic | use_all_frac | occupations | one_atom_occupations | starting_spin_angle | degauss | smearing | nspin | noncolin | ecfixed | qcutz | q2sigma | input_dft | exx_fraction | screening_parameter | exxdiv_treatment | ecutvcut | nqx1 | nqx2 | nqx3 | lda_plus_u | lda_plus_u_kind | Hubbard_U | Hubbard_alpha | Hubbard_J(i,ityp) | starting_ns_eigenvalue(m,ispin,l) | U_projection_type | edir | emaxpos | eopreg | eamp | angle1 | angle2 | constrained_magnetization | fixed_magnetization | lambda | report | lspinorb | assume_isolated | esm_bc | esm_w | esm_efield | esm_nfit | london | london_s6 | london_rcut

&ELECTRONS

electron_maxstep | conv_thr | adaptive_thr | conv_thr_init | conv_thr_multi | mixing_mode | mixing_beta | mixing_ndim | mixing_fixed_ns | diagonalization | ortho_para | diago_thr_init | diago_cg_maxiter | diago_david_ndim | diago_full_acc | efield | efield_cart | startingpot | startingwfc | tqr

&IONS

ion_dynamics | ion_positions | phase_space | pot_extrapolation | wfc_extrapolation | remove_rigid_rot | ion_temperature | tempw | tolp | delta_t | nraise | refold_pos | upscale | bfgs_ndim | trust_radius_max | trust_radius_min | trust_radius_ini | w_1 | w_2

&CELL

cell_dynamics | press | wmass | cell_factor | press_conv_thr | cell_dofree

ATOMIC_SPECIES

X | Mass_X | PseudoPot_X