

# *The Towler Institute*

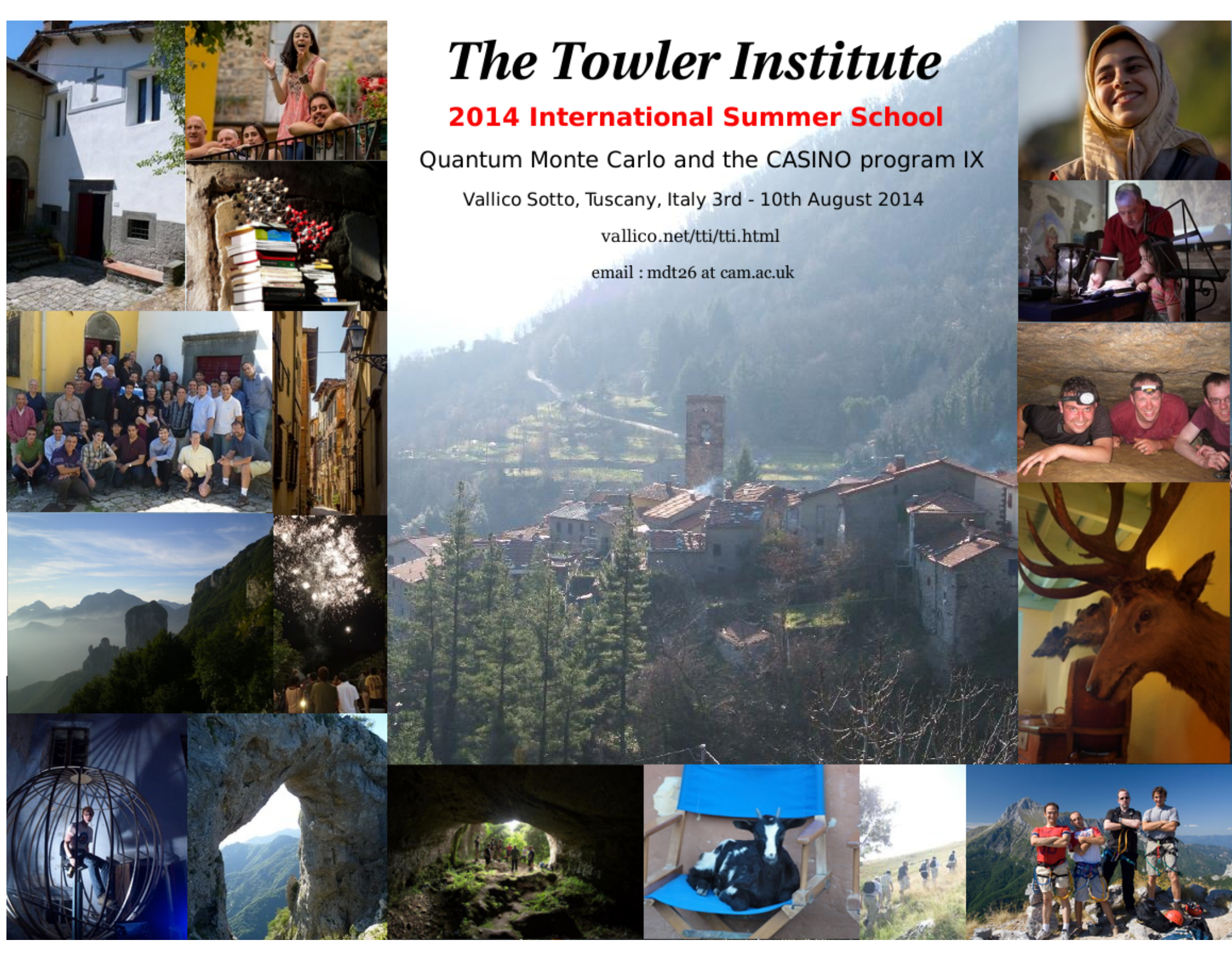
## **2014 International Summer School**

Quantum Monte Carlo and the CASINO program IX

Vallico Sotto, Tuscany, Italy 3rd - 10th August 2014

[vallico.net/tti/tti.html](http://vallico.net/tti/tti.html)

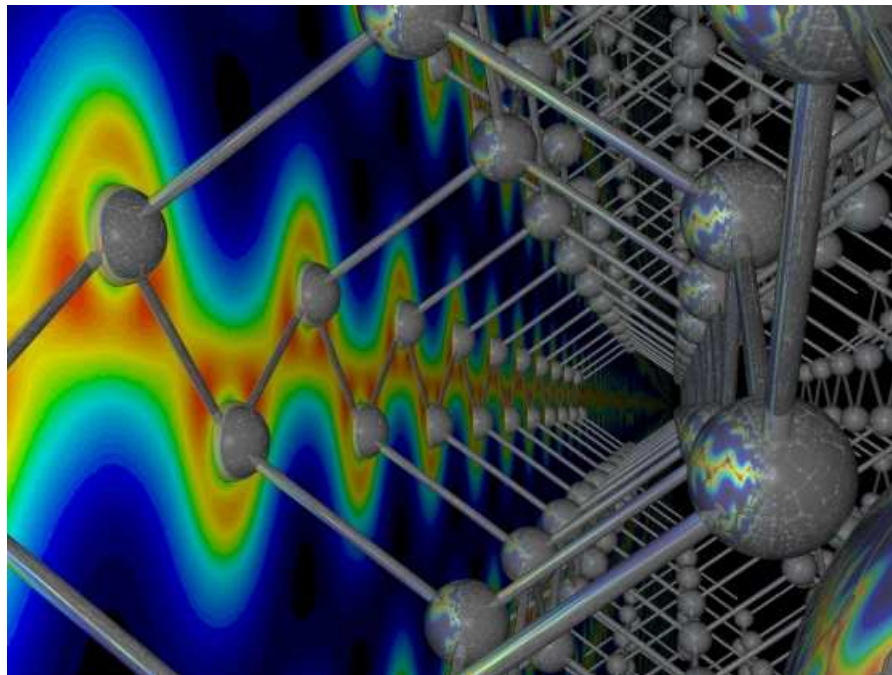
email : mdt26 at cam.ac.uk





# The CASINO program

Distribution, setup, and compilation



*Mike Towler*

*TCM Group, Cavendish Laboratory, University of Cambridge*

QMC web page: [vallico.net/casinoqmc](http://vallico.net/casinoqmc)

Email: [mdt26@cam.ac.uk](mailto:mdt26@cam.ac.uk)

## Download the code



### Download CASINO



First time users who do not have a CASINO login ID (if you do, click [here](#)) can reveal a temporary link allowing *immediate download* of the standard version of the Cambridge University quantum Monte Carlo program 'CASINO' by completing the form below then pressing the big orange "Send form" button at the bottom of the page.

Get the code by filling in the form at [vallico.net/casinoqmc/download-casino](http://vallico.net/casinoqmc/download-casino).

CASINO Central (me) will subsequently send you a 'CASINO login ID' giving you access using the same login ID and password to requested services. This includes the Discussion forum, the CASINOQMC site, and future downloads without filling in a form from: [vallico.net/casinoqmc/update-casino](http://vallico.net/casinoqmc/update-casino).

The form also allows you to request 'developer status' (given some justification and evidence of some level of competence).

## Installation

- CASINO was designed to run on machines running a Linux/Unix operating systems (this in principle includes Macs, since Mac OSX is based on BSD Unix and has a functioning bash command line).
- CASINO can also be run on Windows machines (in principle for all modern versions of Windows from XP onwards) if you first install the 'Cygwin' software which emulates a Linux-like environment to compile and run in. See the manual and FAQ.

*Assume Linux from now on..*

# Installation

CASINO determines what kind of computer you are running on by looking at the value of the Unix environment variable `CASINO_ARCH`, which must be defined in your shell session. This tells CASINO to look into a particular file in a set of 'arch' files—permanently stored in the `arch/data` directory of the CASINO distribution—which contains instructions that the 'make' shell command and various CASINO utilities can follow both for compiling the code and running calculations on the machine in question.

Can set this up manually or quasi-automatically, for both currently-supported architectures/machines and for new unsupported ones...

Get the code by filling in the form at [vallico.net/casinoqmc/download-casino](http://vallico.net/casinoqmc/download-casino):

## Installation: the install script

This can be run by typing `./install` then pressing Enter in the base directory of the CASINO distribution, after which you should follow the prompts. You can re-run this installer any time you like to amend your configuration.

To give you an idea of what it does, note that the install script will present you with the following options:

- [c] Compile CASINO for already-configured CASINO\_ARCHs
- [s] Sort/remove configured CASINO\_ARCHs
- [a] Auto-detect valid CASINO\_ARCHs for this machine
- [p] Pick a specific CASINO\_ARCHs for this machine
- [n] Create a new CASINO\_ARCH for this machine interactively
- [y] Install CASINO syntax highlighting for various text editors
- [i] Install required software using package manager (requires root access)
- [r] Restore the CASINO distribution directory to its original state
- [q] Save configuration and quit the installer
- [x] Quit the installer without saving

## Unpacking the distribution

- Change to the directory where you want the distribution to live (this is usually assumed to be your home directory, but it need not be).
- Remove/rename any existing CASINO directory.
- Unpack the `CASINO_vxxx.tar.gz` distribution (`'tar xvfz CASINO_vxxx.tar.gz'`). This will result in a new directory called `CASINO` containing the CASINO distribution.
- If you want to maintain different versions of the code, it may be useful to rename the `CASINO` directory to be something like `CASINO_v2.13.378` and set up a symbolic link called `CASINO` which points to it (`'ln -s CASINO_v2.13.378 CASINO'`). There is a CASINO utility *update\_src* which will do this for you—just type, e.g., `update_src 2.13.378` in a directory containing a `CASINO_v2.13.378.tar.gz` archive.

## Preliminary configuration of the machine

You need to ensure that the machine has all the relevant software installed (Fortran, C, and possibly C++ compilers, an MPI library etc.) For machines administered by other people, this should have been done for you.

Refer to the CASINO/FAQ file for notes on the preliminary configuration of machines that you administer yourself. This advice is duplicated online at [http://www.tcm.phy.cam.ac.uk/~mdt26/casino\\_faq.txt](http://www.tcm.phy.cam.ac.uk/~mdt26/casino_faq.txt) and on the main website.

The install script can help you with installing required software using the package manager (choose the [i] option).



## Finding or creating the arch file

The install script can largely do this for you. The options which concern finding or creating the arch file are [a], [p], [n] and [s].

First of all, try the 'Auto detect' [a] option. The install script may detect an exact match for the particular machine you're running on (i.e., someone else has already set CASINO up on it)—in which case, after you accept its recommendation, CASINO will simply work. The script can also suggest generic similar machines on which you can base your installation.

Alternatively, if you know which CASINO\_ARCH you want, you can simply type in its name after selection of the [p] option.

Choosing the [n] option will take you through a guided procedure for creating your own personalized arch file (make sure you have the machine's documentation handy so you can answer the questions the script will ask you). The result of this will be the arch file describing your machine that will be placed in the CASINO/arch/data directory (see below); new arch files can be emailed to the CASINO developers for permanent inclusion in the distribution. On extremely complicated machines the arch file produced by install may need to be tweaked by hand.

Multiple alternative configurations are supported. This includes setting up CASINO for use with multiple compilers—each of which will have its own arch file. One may also create set-ups where multiple machines share the installation directory, such as different-architecture queues on the same cluster, or workstations sharing their home directories over a networked file system. For the latter case, run the installer on one machine of each relevant type to set it up.

Once you have all your CASINO\_ARCHs defined, you may sort them into a preferred order (perhaps to indicate a preferred compiler) or remove them using the [s] option.

## Compiling the code

Once you have sorted out the arch file, can compile the code and the utilities in two different ways.

### AUTOMATIC PROCEDURE

Run the install script as before and this time select the [c] option ('Compile casino for already-configured CASINO\_ARCHs'). The script will respond with a list of defined CASINO\_ARCHs and the following text (using the Titan machine as an example):

The following CASINO\_ARCHs are configured (it is possible that not all of them can be compiled from this machine depending on your set-up):

- [1] linuxpc-gcc-pbs-parallel.titan
- [2] linuxpc-cray-pbs-parallel.titan
- [3] linuxpc-ifort-pbs-parallel.titan

At the prompt below enter the numbers corresponding to the CASINO\_ARCHs you would like to compile, separated by spaces.

You can specify which optional compile-time features to enable appending ':<feature>' to each number. Available <features> include:

- 'Openmp' for building OpenMP support
- 'Shm' for building the SMP shared-memory facility (of most use for calculations with blip or Gaussian basis sets)
- 'OpenmpShm' for building a version with both features enabled

## Compiling the code: continued

One may thus type '1' to compile with the gcc compiler, '2' to compile with the Cray compiler, etc. One may compile special versions such as the shared-memory version of CASINO with the gcc compiler by typing '1:Shm'.

We recommend that this automatic procedure is used, essentially for three reasons:

- One can build multiple CASINO executables with a single typed command such as '1 1:Shm 1:Openmp 1:OpenmpShm 2 3:debug 4:Shm'.
- The install script will run make in parallel as far as it can over multiple cores (this can of course be done on the command line, but the user may not know how).
- On some obscure Unix machines, the default version of 'make' will work with a sufficiently different syntax that the CASINO Makefile is not interpreted correctly. The install script will know to run an alternative version of make which we know works, such as GNU's 'gmake' whereas just typing 'make' on the command line will not work.

See the manual procedure in the documentation if this doesn't work.

## Running the code

You run the code using the 'runqmc' script (see the manual or type 'runqmc--help'). This script is able to access all the run time information in the arch file to determine how to run jobs on your machine - even to the extent of loading modules, writing batch scripts, and submitting them for you. `runqmc` (which should be in your path after installing CASINO) is designed to reduce the effort of doing QMC calculations to just entering one command, and that command is essentially the same on all different kinds of computer. It is most useful when using parallel machines with a batch queueing system, since it detects most common errors that a user may make in setting up a calculation and such faults are thus detected immediately rather than when CASINO actually starts running (which may be many hours or even days later).

`runqmc` can run calculations on single- and multiprocessor workstations, and on clusters. To run the CASINO calculation set up in the current directory, simply type *runqmc*. This will automatically—and possibly a little dangerously—occupy all cores on a workstation, or the maximum permitted allocation (in both number of cores and walltime) on a cluster. To change this, or to specify other calculation parameters, there are various options available. An example on a big parallel machine:

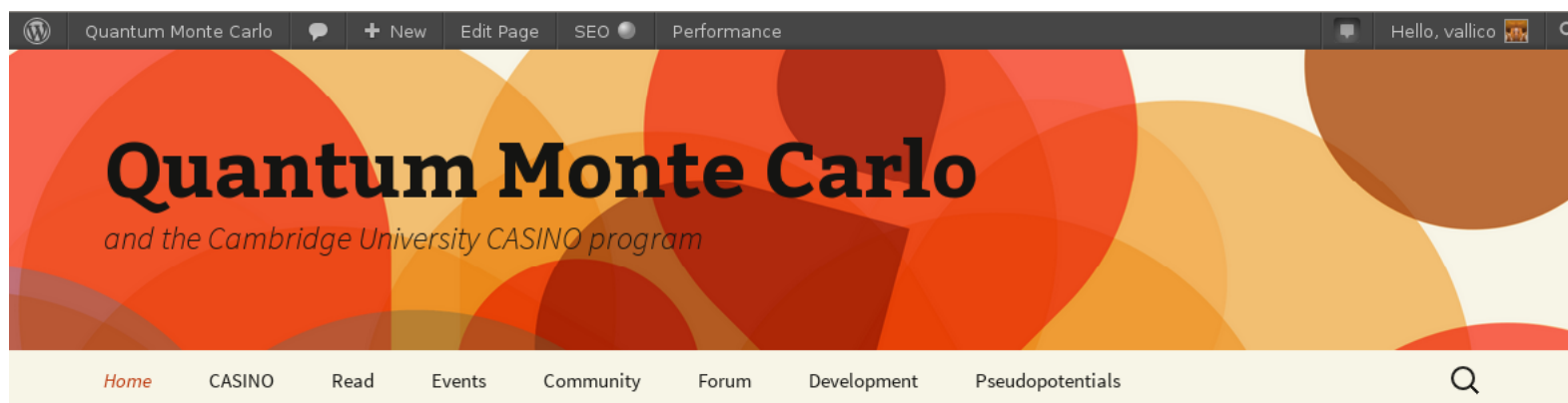
```
runqmc --nproc=224256 --walltime=10h --shm
```

runs in parallel using 224256 MPI processes (distributed among nodes depending on the number of cores per node in the machine, which `runqmc` knows about) using shared memory, and with a walltime limit of 10 hours (affects queue your job is put in). This is equivalent to

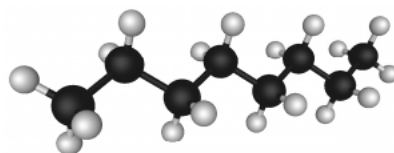
```
runqmc -p 224256 -T 10h -s
```

(provided the machine has 12 cores per node) i.e., most options come with a longer version (with two hyphens and an equals sign) and a shorter easier-to-type version (with one hyphen, and zero or more spaces before the option value).

# Web resources



## QMC in Cambridge and around the world

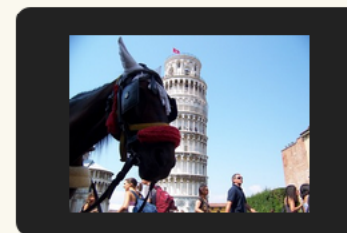


Quantum Monte Carlo (QMC) is an exciting, modern computational technique which allows us to approximately solve the equations of quantum mechanics – which are far too complicated to solve exactly – and in most cases get *essentially the right answer*. Its key advantage is that, unlike with all other known highly accurate techniques, it is still possible to do the calculations for relatively large systems with many atoms – providing you have a big enough computer (and QMC is quite capable of using the biggest). Used in combination with other cheaper methods, it provides researchers with the final building block in an atomic scale micro-laboratory on their computer which can be used to simulate small parts of the real world. Its practical application to real-life problems – via a general, widely-distributed computer program system – has been pioneered in the United Kingdom by members of the Cambridge University [Theory of Condensed Matter Group](#) working out of the [Cavendish Laboratory](#), famous throughout the world as the venue for amazing historical discoveries such as the electron and DNA. A number of their collaborators at various other universities – whose work is also discussed here – have also made fundamental contributions, both to the software, and to the field in general.

This page exists for the following reasons:

- To tell you about the research of the U.K. QMC community. The site includes introductory and review information along with a library of as many of their relevant scientific articles as we can find

### QMC in the Apuan Alps: 10 years



### Login Status

You are logged in as vallico  
[click here to logout](#)

### Recent Blog Posts

[QMC at the research frontier: useful calculations for big complicated systems](#) Mike Towler

[Mike Towler's Jerry Lee Lewis set list](#) vallico

[Report on 'QMC in the Apuan Alps VIII'](#)

# Live Demonstration!