**Quantum espresso manual pdf**

The intended audience of the Developers’ Manual is everybody who wants to: know how Quantum ESPRESSO works, including its internals modify/customize/add/extend/improve/clean up Quantum ESPRESSO know how to read data produced by the two DFT engines PWscf and CPV (follow this link to download Quantum ESPRESSO xml schemas) The same category of people should also write this manual, of course. People who want to know about the capabilities of Quantum ESPRESSO, or who want just to use it, should read the General documentation and the package-specific documentation. People who want to know about the methods or the physics behind Quantum ESPRESSO should read first the relevant literature. Developers manual PHonon developers manual (HTML or pdf) Next: Contents  Contents Available Input file description for the following executables: PWscf: pw.x, hp.x, band_interpolation.x, bgw2pw.x, pw2bgw.x, pwcond.x, pprism.x PHonon: ph.x, dynmat.x, matdyn.x, postahc.x, q2r.x PWneb: neb.x CP: cp.x, cppp.x TurboTDDFT: turbo_lanczos.x, turbo_spectrum.x, turbo_davidson.x TurboMAGNON: turbo_magnon.x TurboEELS: turbo_eels.x XSpectra: xspectra.x atomic: ld1.x QEHeat: all_currents.x KCW: kcw.x Postprocessings: pp.x dos.x bands.x projwfc.x molecularpdos.x ppp.x ppacf.x Other executables have a brief documentation in the header of the main *.f90 file. For other packages please refer to their respective user documentation. Last updated: June 2022 The 7.0 version of Quantum ESPRESSO is available for download. For more information please see the release notes inside the downloaded directory under Docs. Read More Bravais-lattice index. Optional only if space_group is set. If ibrav /= 0, specify EITHER [ celldm(1)-celldm(6) ] OR [ A, B, C, cosAB, cosAC, cosBC ] but NOT both. The lattice parameter "alat" is set to alat = celldm(1) (in a.u.) or alat = A (in Angstrom); see below for the other parameters. For ibrav=0 specify the lattice vectors in CELL_PARAMETERS, optionally the lattice parameter alat = celldm(1) (in a.u.) or = A (in Angstrom). If not specified, the lattice parameter is taken from CELL_PARAMETERS IMPORTANT NOTICE 1: with ibrav=0 lattice vectors must be given with a sufficiently large number of digits and with the correct symmetry, or else symmetry detection may fail and strange problems may arise in symmetrization. IMPORTANT NOTICE 2: do not use celldm(1) or A as a.u. to Ang conversion factor, use the true lattice parameters or nothing, specify units in CELL_PARAMETERS and ATOMIC_POSITIONS ibrav structure celldm(2)-celldm(6) or: b,c,cosbc,cosac,cosab 0 free crystal axis provided in input: see card CELL_PARAMETERS 1 cubic P (sc) v1 = a(1,0,0), v2 = a(0,1,0), v3 = a(0,0,1) 2 cubic F (fcc) v1 = (a/2)(-1,0,1), v2 = (a/2)(0,1,1), v3 = (a/2)(-1,1,0) 3 cubic I (bcc) v1 = (a/2)(1,1,1), v2 = (a/2)(-1,1,1), v3 = (a/2)(-1,-1,1) -3 cubic I (bcc), more symmetric axis: v1 = (a/2)(-1,1,1), v2 = (a/2)(1,-1,1), v3 = (a/2)(1,1,-1) 4 Hexagonal and Trigonal P celldm(3)=c/a v1 = a(1,0,0), v2 = a(-1/2,sqrt(3)/2,0), v3 = a(0,0,c/a) 5 Trigonal R, 3fold axis c celldm(4)=cos(gamma) The crystallographic vectors form a three-fold star around the z-axis, the primitive cell is a simple rhombohedron: v1 = a(tx,-ty,tz), v2 = a(0,2ty,tz), v3 = a(-tx,-ty,tz) where c=cos(gamma) is the cosine of the angle gamma between any pair of crystallographic vectors, tx, ty, tz are: tx=sqrt((1-c)/2), ty=sqrt((1-c)/6), tz=sqrt((1+2c)/3) -5 Trigonal R, 3fold axis celldm(4)=cos(gamma) The crystallographic vectors form a three-fold star around . Defining a' = a/sqrt(3) : v1 = a' (u,v,v), v2 = a' (v,u,v), v3 = a' (v,v,u) where u and v are defined as u = tz - 2*sqrt(2)*ty, v = tz + sqrt(2)*ty and tx, ty, tz as for case ibrav=5 Note: if you prefer x,y,z as axis in the cubic limit, set u = tz + 2*sqrt(2)*ty, v = tz - sqrt(2)*ty See also the note in Modules/latgen.f90 6 Tetragonal P (st) celldm(3)=c/a v1 = a(1,0,0), v2 = a(0,1,0), v3 = a(0,0,c/a) 7 Tetragonal I (bct) celldm(3)=c/a v1=(a/2)(1,-1,c/a), v2=(a/2)(1,1,c/a), v3=(a/2)(-1,-1,c/a) 8 Orthorhombic P celldm(2)=b/a celldm(3)=c/a v1 = (a,0,0), v2 = (0,b,0), v3 = (0,0,c) 9 Orthorhombic base-centered(bco) celldm(2)=b/a celldm(3)=c/a v1 = (a/2, b/2,0), v2 = (-a/2,b/2,0), v3 = (0,0,c) -9 as 9, alternate description v1 = (a/2,-b/2,0), v2 = (a/2, b/2,0), v3 = (0,0,c) 91 Orthorhombic one-face base-centered A-type celldm(2)=b/a celldm(3)=c/a v1 = (a, 0, 0), v2 = (0,b/2,-c/2), v3 = (0,b/2,c/2) 10 Orthorhombic face-centered celldm(2)=b/a celldm(3)=c/a v1 = (a/2,0,c/2), v2 = (a/2,b/2,0), v3 = (0,b/2,c/2) 11 Orthorhombic body-centered celldm(2)=b/a celldm(3)=c/a v1=(a/2,b/2,c/2), v2=(-a/2,b/2,c/2), v3=(-a/2,-b/2,c/2) 12 Monoclinic P, unique axis c celldm(2)=b/a celldm(3)=c/a, celldm(4)=cos(ab) v1=(a,0,0), v2=(b*cos(gamma),b*sin(gamma),0), v3 = (0,0,c) where gamma is the angle between axis a and b. -12 Monoclinic P, unique axis b celldm(2)=b/a celldm(3)=c/a, celldm(5)=cos(ac) v1 = (a,0,0), v2 = (0,b,0), v3 = (c*cos(beta),0,c*sin(beta)) where beta is the angle between axis a and c 13 Monoclinic base-centered celldm(2)=b/a (unique axis c) celldm(3)=c/a, celldm(4)=cos(gamma) v1 = ( a/2, 0, -c/2), v2 = (b*cos(gamma), b*sin(gamma), 0 ), v3 = ( a/2, 0, c/2), where gamma=angle between axis a and b projected on xy plane -13 Monoclinic base-centered celldm(2)=b/a (unique axis b) celldm(3)=c/a, celldm(5)=cos(beta) v1 = ( a/2, b/2, 0), v2 = ( -a/2, b/2, 0), v3 = (c*cos(beta), 0, c*sin(beta)), where beta=angle between axis a and c projected on xz plane IMPORTANT NOTICE: until QE v.6.4.1, axis for ibrav=-13 had a different definition: v1(old) =-v2(now), v2(old) = v1(now) 14 Triclinic celldm(2)= b/a, celldm(3)= c/a, celldm(4)= cos(bc), celldm(5)= cos(ac), celldm(6)= cos(ab) v1 = (a, 0, 0), v2 = (b*cos(gamma), b*sin(gamma), 0) v3 = (c*cos(beta), c*(cos(alpha)-cos(beta)cos(gamma))/sin(gamma), c*sqrt( 1 + 2*cos(alpha)cos(beta)cos(gamma) - cos(alpha)^2-cos(beta)^2-cos(gamma)^2 )/sin(gamma) ) where alpha is the angle between axis b and c beta is the angle between axis a and c gamma is the angle between axis a and b Next: Contents  Contents Next: Contents  Contents The general documentation (HTML or pdf) covers the installation and usage of the current stable release of Quantum ESPRESSO (opEn-Source Package for Research in Electronic Structure, Simulation, and Optimization). We refer the reader to the package-specific documentation for all the information on features and capabilities of Quantum ESPRESSO. The full Quantum ESPRESSO distribution contains the following core packages for the calculation of electronic-structure properties within Density-Functional Theory (DFT), using a Plane-Wave basis set and pseudopotentials: PWscf (PW): Plane-Wave Self-Consistent Field CP (CPV): Car-Parrinello Molecular Dynamics It also includes the following more specialized packages: PWneb (NEB): energy barriers and reaction pathways through the Nudged Elastic Band method PHonon: phonons with Density-Functional Perturbation Theory PostProc (PP): various utilities for data postprocessing PWcond: ballistic conductance GWL: GW calculations and solution of the Bethe-Salpeter Equation XSPECTRA: K-edge X-ray adsorption spectra TDDFPT: calculations of spectra using Time-Dependent Density-Functional Perturbation Theory EPW: electron-phonon calculations using Wannier functions The following auxiliary codes are included as well: PWgui: a Graphical User Interface, producing input data files for PWscf atomic: a program for atomic calculations and generation of pseudopotentials Several additional packages that exploit data produced by Quantum ESPRESSO or patch some Quantum ESPRESSO routines can be installed using the Makefile of Quantum ESPRESSO: GIPAW (Gauge-Independent Projector Augmented Waves): NMR chemical shifts and EPR g-tensor WANNIER90: maximally localized Wannier functions WanT: quantum transport properties with Wannier functions YAMBO: electronic excitations within Many-Body Perturbation Theory: GW and Bethe-Salpeter equation PLUMED: calculation of free-energy surface through metadynamics (Last update: June, 2022)