Materials Science Codes on Innovative HPC Architectures

Prace-MaX Training

Cineca – Casalecchio di Reno (BO, Italy)

5-7 December 2016

Tutorial: The Siesta Code

Oswaldo Diéguez

Department of Materials Science and Engineering

Tel Aviv University

7/12/2016

Siesta: A Density-Functional Theory Code

Main Difference between Siesta and other DFT codes (such as Quantum Espresso):
Siesta Was Designed to Scale Linearly with the Number of Atoms in the Simulation Cell

Implications:

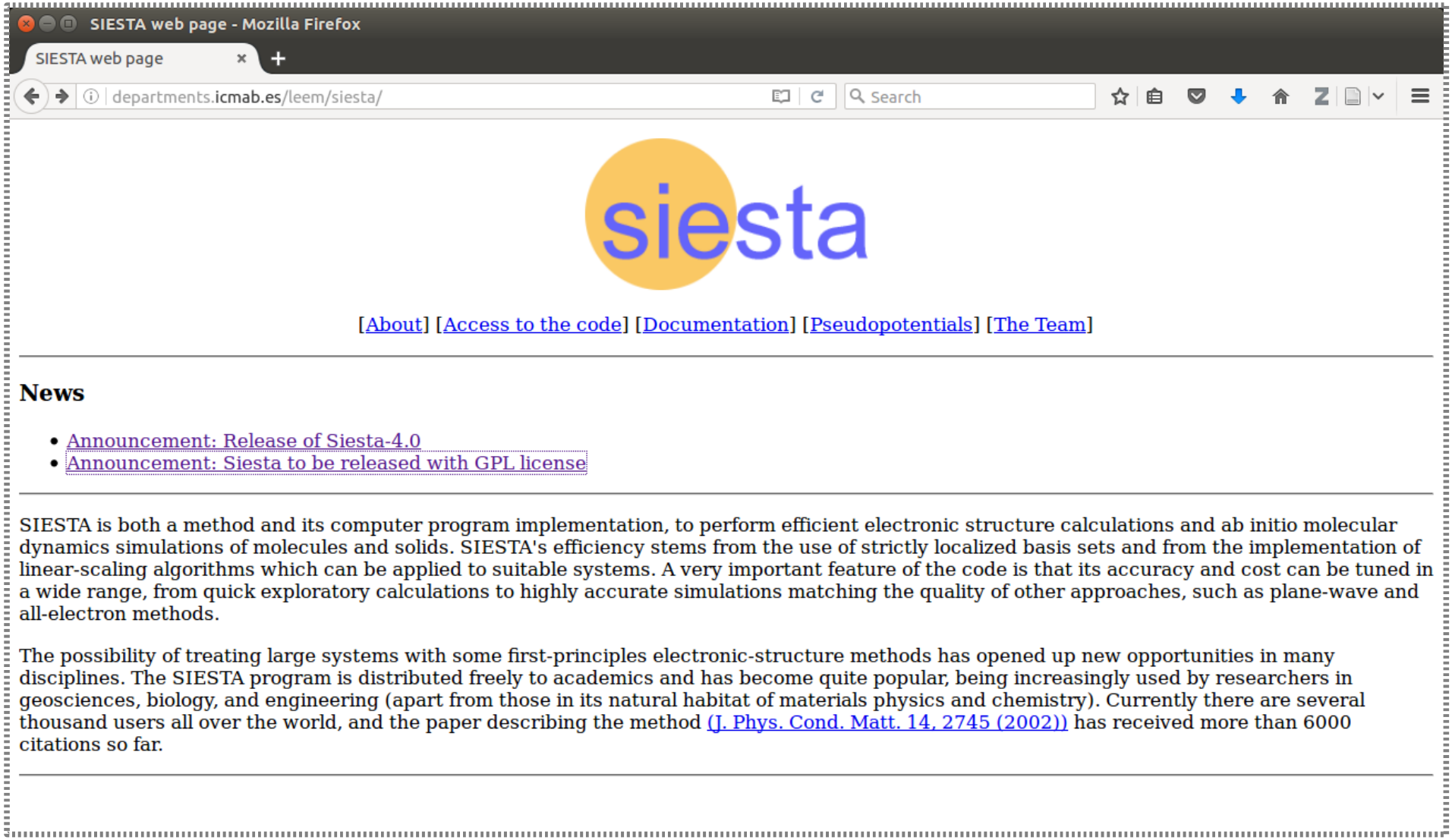
- **Siesta uses numerical localized atomic orbitals as basis set:** this means that the design of the basis set requires more testing than in the case of plane-wave codes
- ***Personal Statement*: Siesta runs faster than any plane-wave code that I have used** for the units cells that I have used (between 20 and 120 atoms; the larger the number of atoms, the larger the speed difference)

Today

- How to Get, Install, and Run Siesta
- Tutorials (by Javier Junquera)
- A Word about Parallelism (by a Non-Expert)

How To Get Siesta

Siesta is Now under the GNU General Public License

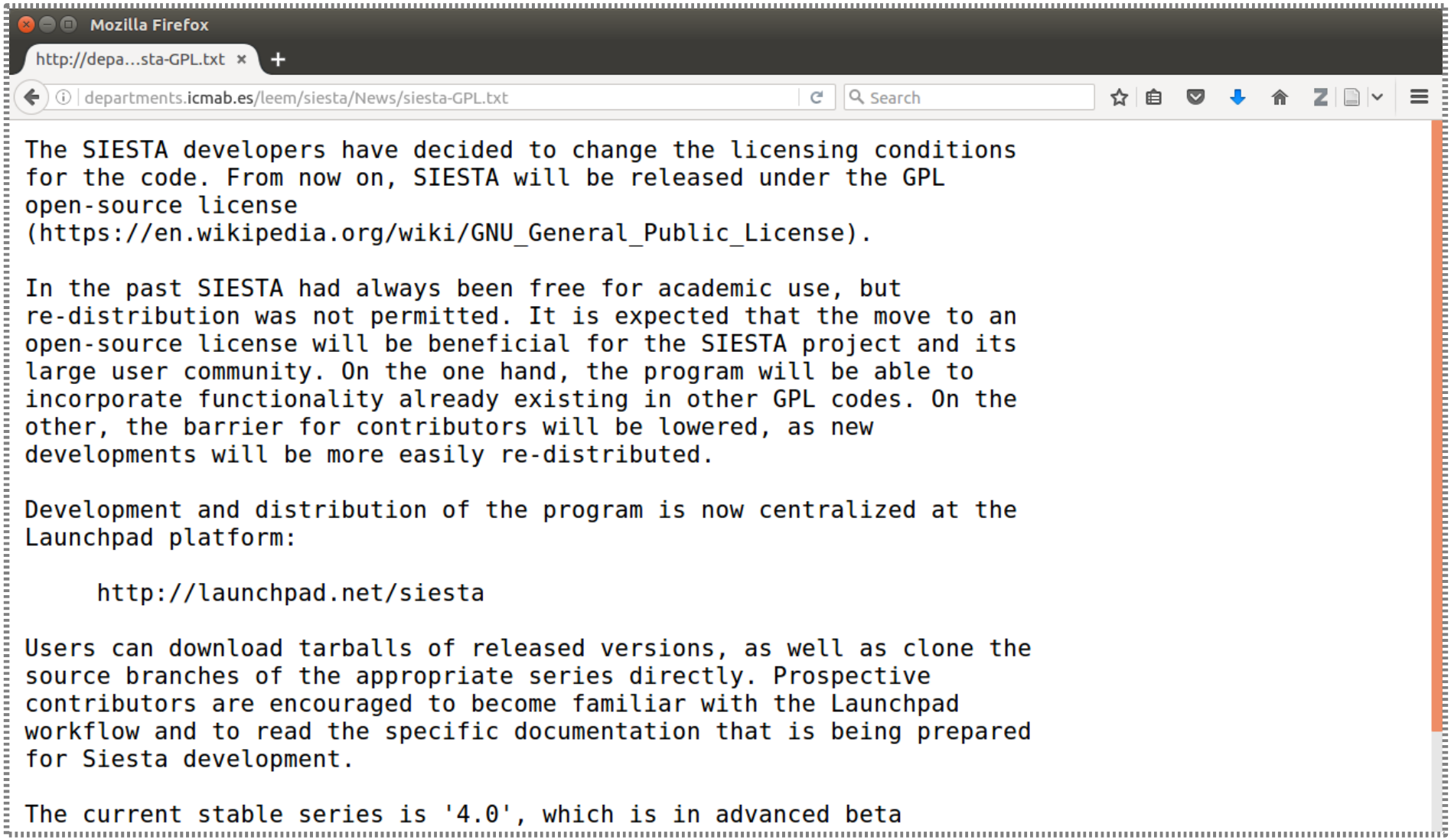


How To Get Siesta

Siesta is Now under the GNU General Public License

How To Get Siesta

Siesta is Now under the GNU General Public License



The SIESTA developers have decided to change the licensing conditions
for the code. From now on, SIESTA will be released under the GPL
open-source license
(https://en.wikipedia.org/wiki/GNU_General_Public_License).

In the past SIESTA had always been free for academic use, but
re-distribution was not permitted. It is expected that the move to an
open-source license will be beneficial for the SIESTA project and its
large user community. On the one hand, the program will be able to
incorporate functionality already existing in other GPL codes. On the
other, the barrier for contributors will be lowered, as new
developments will be more easily re-distributed.

Development and distribution of the program is now centralized at the
Launchpad platform:

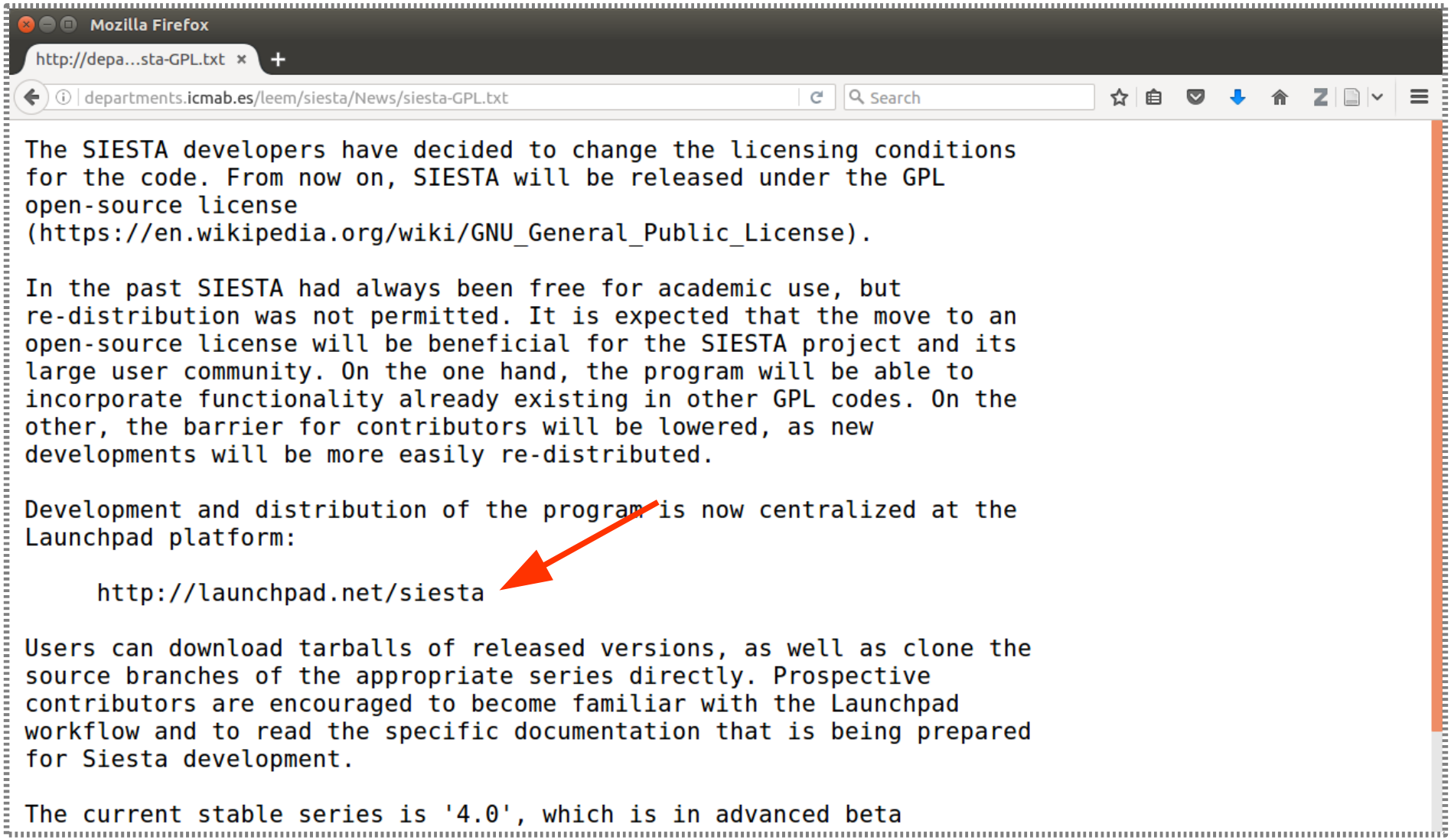
     http://launchpad.net/siesta

Users can download tarballs of released versions, as well as clone the
source branches of the appropriate series directly. Prospective
contributors are encouraged to become familiar with the Launchpad
workflow and to read the specific documentation that is being prepared
for Siesta development.

The current stable series is '4.0', which is in advanced beta

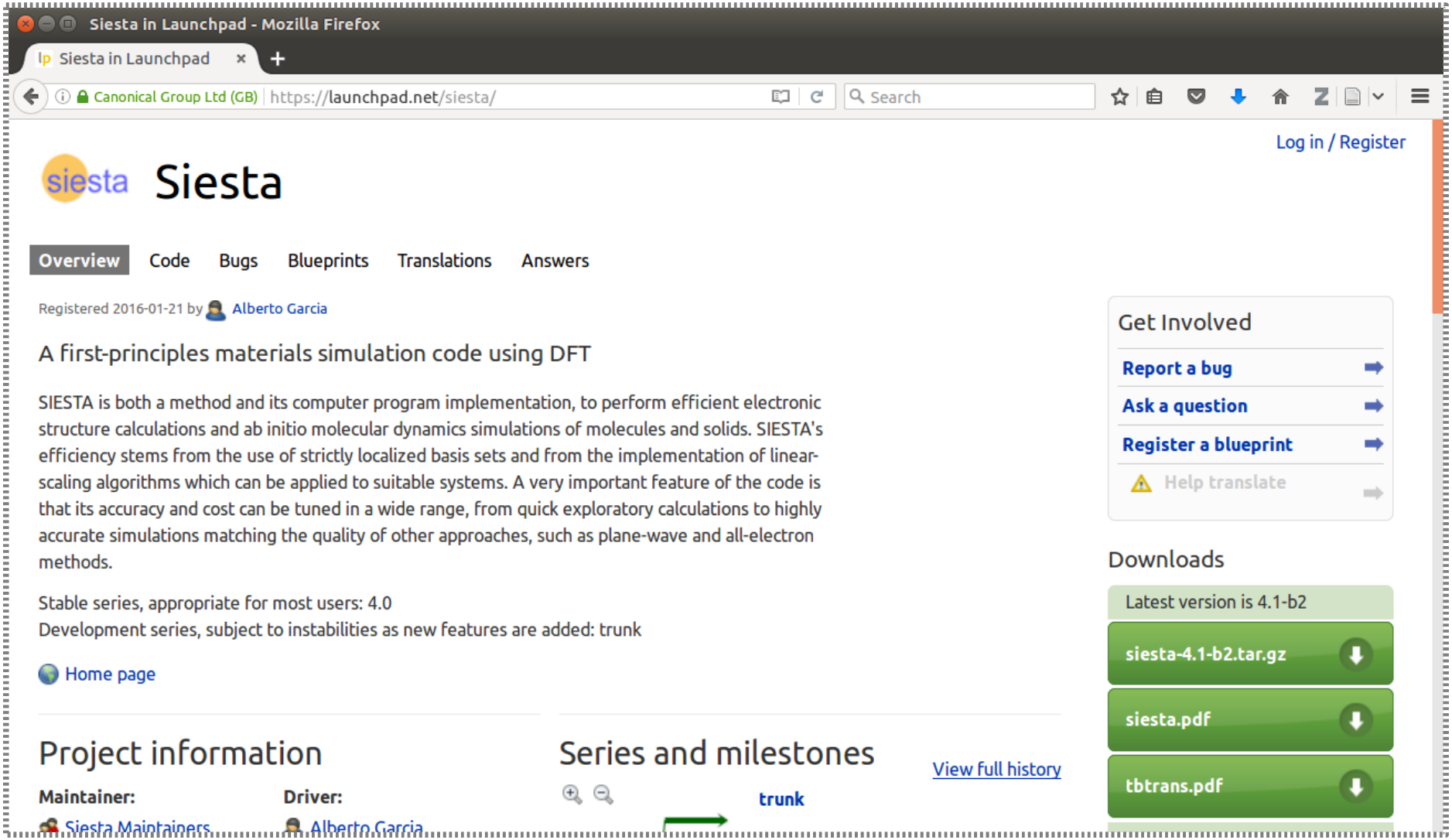
How To Get Siesta

Siesta is Now under the GNU General Public License



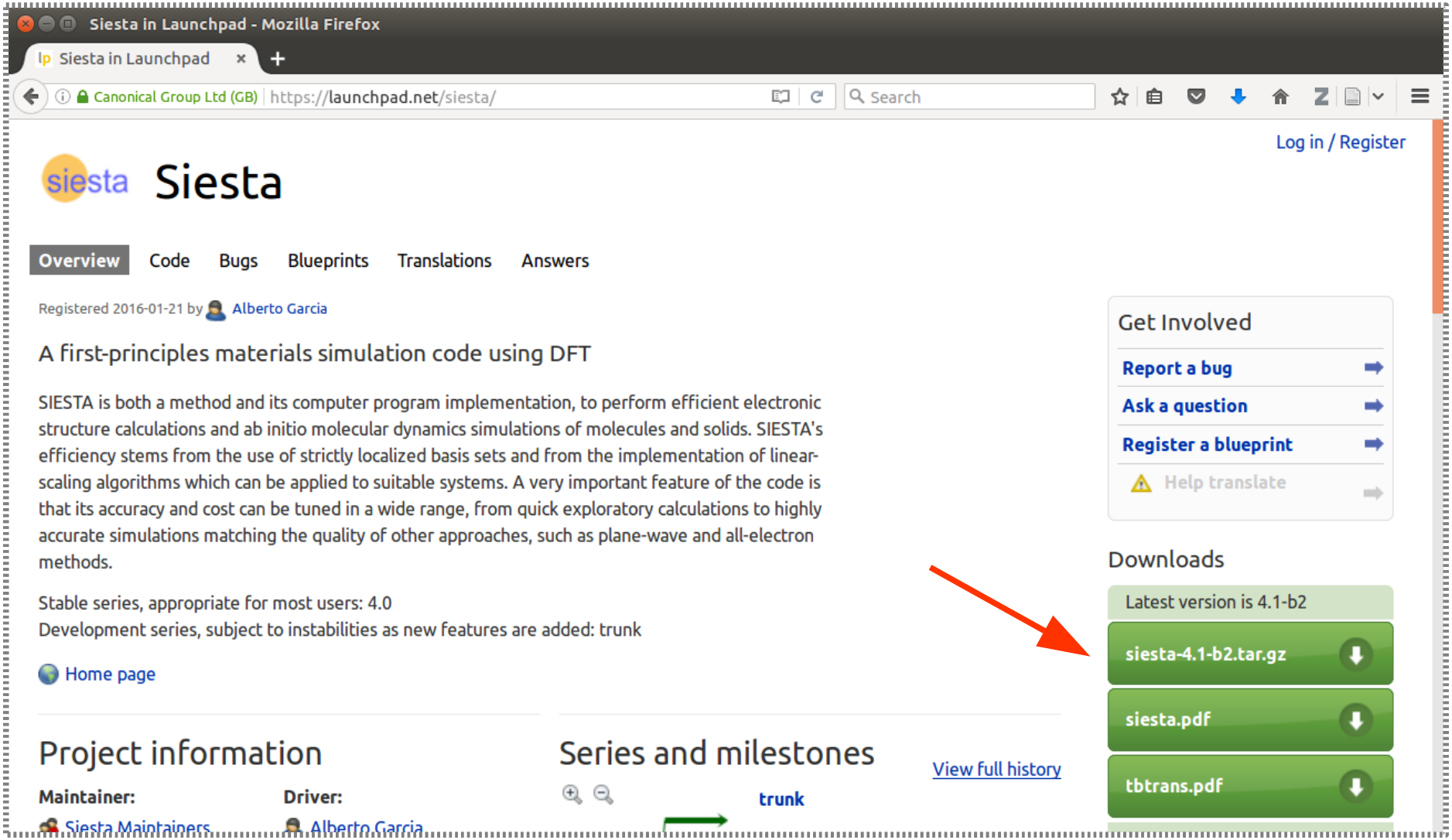
How To Get Siesta

Siesta is Now under the GNU General Public License



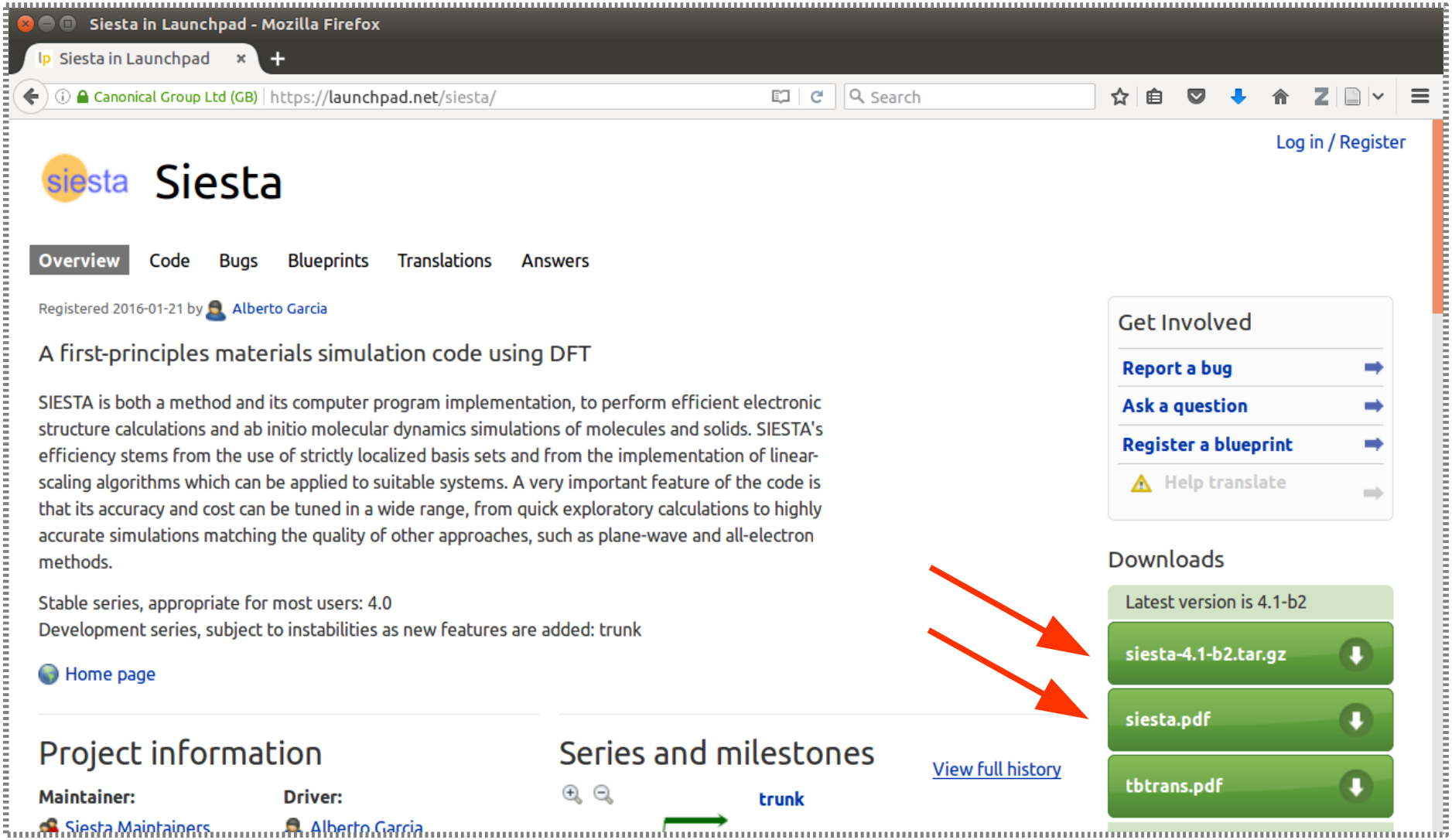
How To Get Siesta

Siesta is Now under the GNU General Public License



How To Get Siesta

Siesta is Now under the GNU General Public License



How To Get Siesta

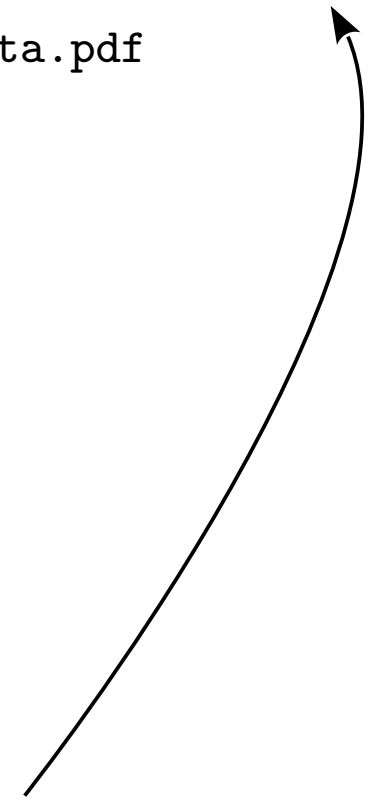
Files that you should now have in your computer:

```
-rwxrwxrwx 1 dieguez dieguez 13132430 Dec  6 09:13 siesta-4.1-b2.tar
-rwxrwxrwx 1 dieguez dieguez   989293 Dec  6 09:14 siesta.pdf
```

How To Get Siesta

Files that you should now have in your computer:

```
-rwxrwxrwx 1 dieguez dieguez 13132430 Dec  6 09:13 siesta-4.1-b2.tar
-rwxrwxrwx 1 dieguez dieguez   989293 Dec  6 09:14 siesta.pdf
```



tar ball containing
all the Siesta files
(source code,
documentation,
examples, etc)

How To Get Siesta

Files that you should now have in your computer:

```
-rwxrwxrwx 1 dieguez dieguez 13132430 Dec  6 09:13 siesta-4.1-b2.tar
-rwxrwxrwx 1 dieguez dieguez   989293 Dec  6 09:14 siesta.pdf
```

Siesta manual

(open it...)

tar ball containing
all the Siesta files
(source code,
documentation,
examples, etc)

How To Get Siesta

The Siesta Manual

User's Guide

SIESTA 4.1-b2

August 31, 2016

http://www.uam.es/siesta

Contributors to SIESTA

Emilio Artacho	*CIC-Nanogune and University of Cambridge*
José María Cela	*Barcelona Supercomputing Center*
Julian D. Gale	*Curtin University of Technology, Perth*
Alberto García	*Institut de Ciència de Materials, CSIC, Barcelona*
Javier Junquera	*Universidad de Cantabria, Santander*
Richard M. Martin	*University of Illinois at Urbana-Champaign*
Pablo Ordejón	*Centre de Investigació en Nanociència i Nanotecnologia, (CSIC-ICN), Barcelona*
Daniel Sánchez-Portal	*Unidad de Física de Materiales, Centro Mixto CSIC-UPV/EHU, San Sebastián*
José M. Soler	*Universidad Autónoma de Madrid*

The SIESTA project was initiated by Pablo Ordejon (then at the Univ. de Oviedo), and Jose M. Soler and Emilio Artacho (Univ. Autonoma de Madrid, UAM). The development team was then joined by Alberto Garcia (then at Univ. del Pais Vasco, Bilbao), Daniel Sanchez-Portal (UAM), and Javier Junquera (Univ. de Oviedo and later UAM), and sometime later by Julian Gale (then at Imperial College, London). In 2007 Jose M. Cela (Barcelona Supercomputing Center, BSC) became a core developer.

The current TranSIESTA module within SIESTA is developed by Nick R. Papior (then at Technical University of Denmark) and Mads Brandbygo. The original TranSIESTA module was developed by Pablo Ordejon and Jose L. Mozos (then at ICMAB-CSIC), and Mads Brandbygo, Kurt Stokbro, and Jeremy Taylor (Technical Univ. of Denmark).

Other contributors (we apologize for any omissions):

Eduardo Anglada, Thomas Archer, Luis C. Balbas, Xavier Blase, Ramon Cuadrado, Michele Cerlotti, Raul de la Cruz, Gabriel Fabricius, Marivi Fernandez-Serra, Jaime Ferrer, Chu-Chun Fu, Sandra Garcia, Victor M. Garcia-Suarez, Georg Huhs, Rogeli Grima, Rainer Hoft, Jorge Kohanoff, Richard Korytar, In-Ho Lee, Lin Lin, Nicolas Lorente, Miguel Llunell, Eduardo Machado, Maider Machado, Jose Luis Martins, Volodymyr Maslyuk, Juana Moreno, Frederico Dutilh Novaes, Micael Oliveira, Nick Rübner Papior, Magnus Paulsson, Oscar Paz, Andrei Postnikov, Tristana Sondon, Andrew Walker, Andrew Walkingshaw, Toby White, Francois Willaime, Chao Yang.

O.F. Sankey, D.J. Niklewski and D.A. Drabold made the FIREBALL code available to P. Ordejon. Although we no longer use the routines in that code, it was essential in the initial development of SIESTA, which still uses many of the algorithms developed by them.

How To Get Siesta

The Siesta Manual

SIESTA (Spanish Initiative for Electronic Simulations with Thousands of Atoms) is both a method and its computer program implementation, to perform electronic structure calculations and *ab initio* molecular dynamics simulations of molecules and solids. Its main characteristics are:

- It uses the standard Kohn-Sham selfconsistent density functional method in the local density (LDA-LSD) and generalized gradient (GGA) approximations, as well as in a non local functional that includes van der Waals interactions (VDW-DF).
- It uses norm-conserving pseudopotentials in their fully nonlocal (Kleinman-Bylander) form.
- It uses atomic orbitals as a basis set, allowing unlimited multiple-zeta and angular momenta, polarization and off-site orbitals. The radial shape of every orbital is numerical and any shape can be used and provided by the user, with the only condition that it has to be of finite support, i.e., it has to be strictly zero beyond a user-provided distance from the corresponding nucleus. Finite-support basis sets are the key for calculating the Hamiltonian and overlap matrices in $O(N)$ operations.
- Projects the electron wavefunctions and density onto a real-space grid in order to calculate the Hartree and exchange-correlation potentials and their matrix elements.
- Besides the standard Rayleigh-Ritz eigenstate method, it allows the use of localized linear combinations of the occupied orbitals (valence-bond or Wannier-like functions), making the computer time and memory scale linearly with the number of atoms. Simulations with several hundred atoms are feasible with modest workstations.
- It is written in Fortran 95 and memory is allocated dynamically.
- It may be compiled for serial or parallel execution (under MPI).

How To Get Siesta

The Siesta Manual

Pages 8-9

It routinely provides:

- Total and partial energies.
- Atomic forces.
- Stress tensor.
- Electric dipole moment.
- Atomic, orbital and bond populations (Mulliken).
- Electron density.

And also (though not all options are compatible):

- Geometry relaxation, fixed or variable cell.
- Constant-temperature molecular dynamics (Nose thermostat).
- Variable cell dynamics (Parrinello-Rahman).
- Spin polarized calculations (collinear or not).
- k-sampling of the Brillouin zone.
- Local and orbital-projected density of states.
- COOP and COHP curves for chemical bonding analysis.
- Dielectric polarization.
- Vibrations (phonons).
- Band structure.
- Ballistic electron transport under non-equilibrium (through TranSIESTA)

How To Install Siesta: marconi

Transfer the Siesta tar ball to marconi:

- this is typically done using a scp utility
- for example, from my linux machine, I did:
  `scp /home/dieguez/Downloads/siesta-4.1-b2.tar.gz a08tra74@login.marconi.cineca.it:`

How To Install Siesta: **marconi**

Transfer the Siesta tar ball to **marconi**:

- this is typically done using a scp utility
- for example, from my linux machine, I did:
  `scp /home/dieguez/Downloads/siesta-4.1-b2.tar.gz a08tra74@login.marconi.cineca.it:`

Log in to **marconi** and check that the tar ball is there:

[a08tra74@r000u06l01 ~]$ **ls**

`siesta-4.1-b2.tar.gz`

How To Install Siesta: marconi

Create new directory, move there the tar ball, and uncompress it:

```
[a08tra74@r000u06l01 ~]$ mkdir SIESTA
```

How To Install Siesta: marconi

Create new directory, move there the tar ball, and uncompress it:

```
[a08tra74@r000u06l01 ~]$ mkdir SIESTA
[a08tra74@r000u06l01 ~]$ mv siesta-4.1-b2.tar.gz SIESTA
```

How To Install Siesta: marconi

Create new directory, move there the tar ball, and uncompress it:

```
[a08tra74@r000u06l01 ~]$ mkdir SIESTA
[a08tra74@r000u06l01 ~]$ mv siesta-4.1-b2.tar.gz SIESTA
[a08tra74@r000u06l01 ~]$ cd SIESTA
```

How To Install Siesta: marconi

Create new directory, move there the tar ball, and uncompress it:

```
[a08tra74@r000u06l01 ~]$ mkdir SIESTA
[a08tra74@r000u06l01 ~]$ mv siesta-4.1-b2.tar.gz SIESTA
[a08tra74@r000u06l01 ~]$ cd SIESTA
[a08tra74@r000u06l01 SIESTA]$ gunzip siesta-4.1-b2.tar.gz
```

How To Install Siesta: marconi

Create new directory, move there the tar ball, and uncompress it:

```
[a08tra74@r000u06l01 ~]$ mkdir SIESTA
[a08tra74@r000u06l01 ~]$ mv siesta-4.1-b2.tar.gz SIESTA
[a08tra74@r000u06l01 ~]$ cd SIESTA
[a08tra74@r000u06l01 SIESTA]$ gunzip siesta-4.1-b2.tar.gz
[a08tra74@r000u06l01 SIESTA]$ tar -xvf siesta-4.1-b2.tar
```

How To Install Siesta: marconi

Create new directory, move there the tar ball, and uncompress it:

```
[a08tra74@r000u06l01 ~]$ mkdir SIESTA
[a08tra74@r000u06l01 ~]$ mv siesta-4.1-b2.tar.gz SIESTA
[a08tra74@r000u06l01 ~]$ cd SIESTA
[a08tra74@r000u06l01 SIESTA]$ gunzip siesta-4.1-b2.tar.gz
[a08tra74@r000u06l01 SIESTA]$ tar -xvf siesta-4.1-b2.tar
[a08tra74@r000u06l01 SIESTA]$ ls -l
total 46400
drwxr-xr-x 10 a08tra74 corsi     4096 Nov 28 10:35 siesta-4.1-b2
-rw-r--r--  1 a08tra74 corsi 47513600 Dec  6 12:09 siesta-4.1-b2.tar
```

How To Install Siesta: marconi

What is in the newly created directory?

[a08tra74@r000u06l01 SIESTA]$ **cd siesta-4.1-b2/**

How To Install Siesta: **marconi**

What is in the newly created directory?

[a08tra74@r000u06l01 SIESTA]$ **cd siesta-4.1-b2/**

[a08tra74@r000u06l01 siesta-4.1-b2]$ **ls**

```
AUTHORS            Docs        Pseudo              Src          version.info
CHANGES            Examples    README              Tests
CHANGES_DETAILED   NOTICE.txt  README_TRANSIESTA   Tutorials
COPYING            Obj         RELEASE_NOTES       Util
```

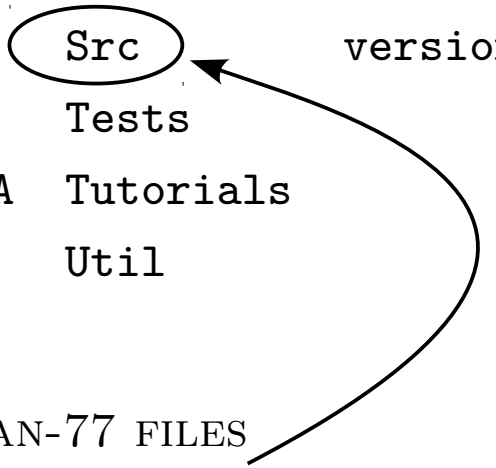
How To Install Siesta: **marconi**

What is in the newly created directory?

[a08tra74@r000u06l01 SIESTA]$ **cd siesta-4.1-b2/**

[a08tra74@r000u06l01 siesta-4.1-b2]$ **ls**

```
AUTHORS             Docs         Pseudo              Src          version.info
CHANGES             Examples     README              Tests
CHANGES_DETAILED    NOTICE.txt   README_TRANSIESTA   Tutorials
COPYING             Obj          RELEASE_NOTES       Util
```



Version in mid-2014: 269 Fortran-77 files
207 Fortran-90 files
15 shell scripts
23 m4 files
3 Matlab files
1 C file
185,263 lines of code

How To Install Siesta: marconi

How to Compile a Serial Executable?

(see section 2 of The Siesta Manual)

```
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj/
```

How To Install Siesta: **marconi**

How to Compile a Serial Executable?

(see section 2 of The Siesta Manual)

```
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj/
[a08tra74@r000u06l01 Obj]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
```

How To Install Siesta: **marconi**

How to Compile a Serial Executable?

(see section 2 of The Siesta Manual)

```
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj/
[a08tra74@r000u06l01 Obj]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
[a08tra74@r000u06l01 Obj]$ cp gfortran.make arch.make
```

How To Install Siesta: **marconi**

How to Compile a Serial Executable?

(see section 2 of The Siesta Manual)

```
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj/
[a08tra74@r000u06l01 Obj]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
[a08tra74@r000u06l01 Obj]$ cp gfortran.make arch.make
[a08tra74@r000u06l01 Obj]$ make
[...lots of output messages about the compilation...]
```

How To Install Siesta: **marconi**

How to Compile a Serial Executable?

(see section 2 of The Siesta Manual)

```
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj/
[a08tra74@r000u06l01 Obj]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
[a08tra74@r000u06l01 Obj]$ cp gfortran.make arch.make
[a08tra74@r000u06l01 Obj]$ make
[...lots of output messages about the compilation...]
[a08tra74@r000u06l01 Obj]$ ls -l siesta
-rwxr-xr-x 1 a08tra74 corsi 7298284 Dec  6 15:06 siesta
```

How To Install Siesta: marconi

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
```

How To Install Siesta: marconi

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ mkdir Obj_intel/
```

How To Install Siesta: **marconi**

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ mkdir Obj_intel/
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj_intel/
```

How To Install Siesta: **marconi**

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ mkdir Obj_intel/
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj_intel/
[a08tra74@r000u06l01 Obj_intel]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
```

How To Install Siesta: **marconi**

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ mkdir Obj_intel/
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj_intel/
[a08tra74@r000u06l01 Obj_intel]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
[a08tra74@r000u06l01 Obj_intel]$ cp ../Obj/intel.make arch.make
```

How To Install Siesta: **marconi**

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ mkdir Obj_intel/
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj_intel/
[a08tra74@r000u06l01 Obj_intel]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
[a08tra74@r000u06l01 Obj_intel]$ cp ../Obj/intel.make arch.make
[a08tra74@r000u06l01 Obj_intel]$ module load intel
```

How To Install Siesta: **marconi**

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ mkdir Obj_intel/
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj_intel/
[a08tra74@r000u06l01 Obj_intel]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
[a08tra74@r000u06l01 Obj_intel]$ cp ../Obj/intel.make arch.make
[a08tra74@r000u06l01 Obj_intel]$ module load intel
[a08tra74@r000u06l01 Obj_intel]$ make
[...lots of output messages about the compilation...]
```

How To Install Siesta: **marconi**

How to Compile another Serial Executable?

```
[a08tra74@r000u06l01 Obj]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ mkdir Obj_intel/
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd Obj_intel/
[a08tra74@r000u06l01 Obj_intel]$ sh ../Src/obj_setup.sh
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
[a08tra74@r000u06l01 Obj_intel]$ cp ../Obj/intel.make arch.make
[a08tra74@r000u06l01 Obj_intel]$ module load intel
[a08tra74@r000u06l01 Obj_intel]$ make
[...lots of output messages about the compilation...]
[a08tra74@r000u06l01 Obj_intel]$ ls -l siesta
-rwxr-xr-x 1 a08tra74 corsi 12122531 Dec  6 16:33 siesta
```

How To Install Siesta: **marconi**

How to Compile a Parallel Executable?

[a08tra74@r000u06l01 Obj_intel]$ **cd ..**

[a08tra74@r000u06l01 siesta-4.1-b2]$ **mkdir Obj_parallel/**

[a08tra74@r000u06l01 siesta-4.1-b2]$ **cd Obj_parallel/**

[a08tra74@r000u06l01 Obj_parallel]$ **sh ../Src/obj_setup.sh**

```
 *** Compilation setup done.
 *** Remember to copy an arch.make file into the directory.
 *** These files are template arch.make files:
 ***    gfortran.make (for gfortran compiler)
 ***    intel.make (for intel compiler)
 ***    DOCUMENTED-TEMPLATE.make (requires customization)
```

[a08tra74@r000u06l01 Obj_parallel]$ **cp ../Obj/DOCUMENTED-TEMPLATE.make arch.make**

...and now edit the **arch.make** to suit **marconi**...

How To Run Siesta

First, prepare the input files...

```
[a08tra74@r000u06l01 Obj_intel]$ cd ..
[a08tra74@r000u06l01 siesta-4.1-b2]$ cd ..
[a08tra74@r000u06l01 SIESTA]$ mkdir h2o/
[a08tra74@r000u06l01 SIESTA]$ cd h2o/
[a08tra74@r000u06l01 h2o]$ cp ../siesta-4.1-b2/Examples/H2O/h2o.fdf .
```

How To Run Siesta

```
[a08tra74@r000u06l01 h2o]$ cat h2o.fdf
SystemName          Water molecule
SystemLabel         h2o
NumberOfAtoms       3
NumberOfSpecies     2

%block ChemicalSpeciesLabel
 1  8  O      # Species index, atomic number, species label
 2  1  H
%endblock ChemicalSpeciesLabel

AtomicCoordinatesFormat  Ang
%block AtomicCoordinatesAndAtomicSpecies
 0.000  0.000  0.000  1
 0.757  0.586  0.000  2
-0.757  0.586  0.000  2
%endblock AtomicCoordinatesAndAtomicSpecies
```

How To Run Siesta

```
[a08tra74@r000u06l01 h2o]$ cp ../siesta-4.1-b2/Examples/Vps/O.psf .
[a08tra74@r000u06l01 h2o]$ cp ../siesta-4.1-b2/Examples/Vps/H.psf .
[a08tra74@r000u06l01 h2o]$ ../siesta-4.1-b2/Obj/siesta < h2o.fdf
```

Tutorials

- Tutorial 1 (by Javier Junquera):

  H2O molecule: converging the size of the unit cell for a polar molecule

- Tutorial 2 (by Javier Junquera):

  Lattice constant, bulk modulus, and equilibrium energy of solids: bulk Si

- Tutorial 3 (by Javier Junquera:

  Analysis of the charge density in real space of a solid: bulk Si

- Tutorial 4 (by Javier Junquera):

  Band structure of an ionic solid: the case of MgO

A Word About Parallelism in Siesta

From The Siesta Manual:

2.3 Parallel

To achieve a parallel build of SIESTA one should first determine which type of parallelism one requires. It is advised to use MPI for calculations with moderate number of cores. If one requires eXa-scale parallelism SIESTA provides hybrid parallelism using both MPI and OpenMP.

[...]

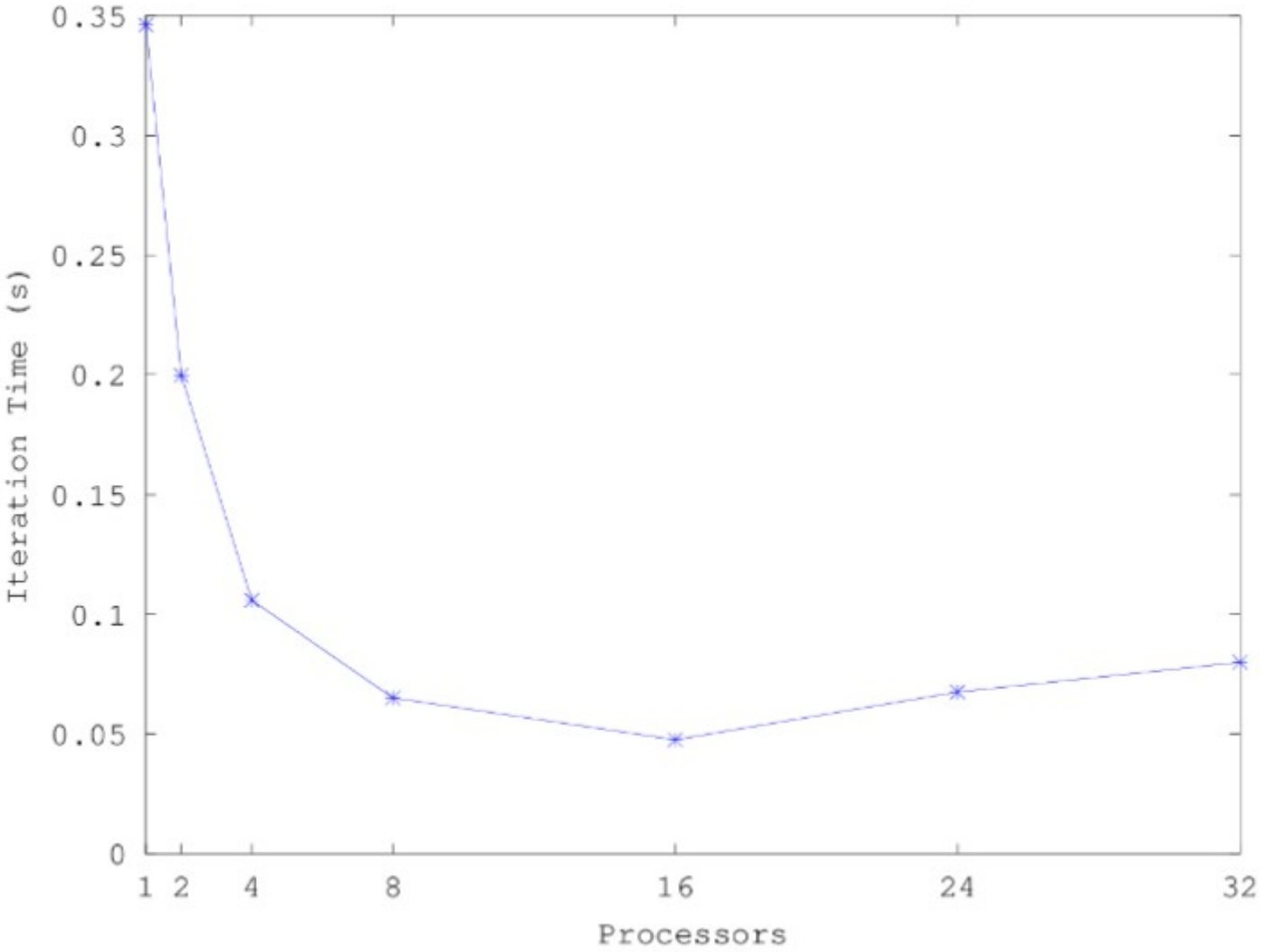
If using only 1 thread per MPI core it is advised to compile SIESTA without OpenMP. As such it may be advantageous to compile SIESTA in 3 variants; OpenMP-only (small systems), MPI-only (medium to large systems) and MPI+OpenMP (large+ systems).

Diag.ParallelOverK **false** *(logical)*

For the diagonalisation there is a choice in strategy about whether to parallelise over the K points or over the orbitals. K point diagonalisation is close to perfectly parallel but is only useful where the number of K points is much larger than the number of processors and therefore orbital parallelisation is generally preferred. The exception is for metals where the unit cell is small, but the number of K points to be sampled is very large. In this last case it is recommend that this option be used.

A Word About Parallelism in Siesta

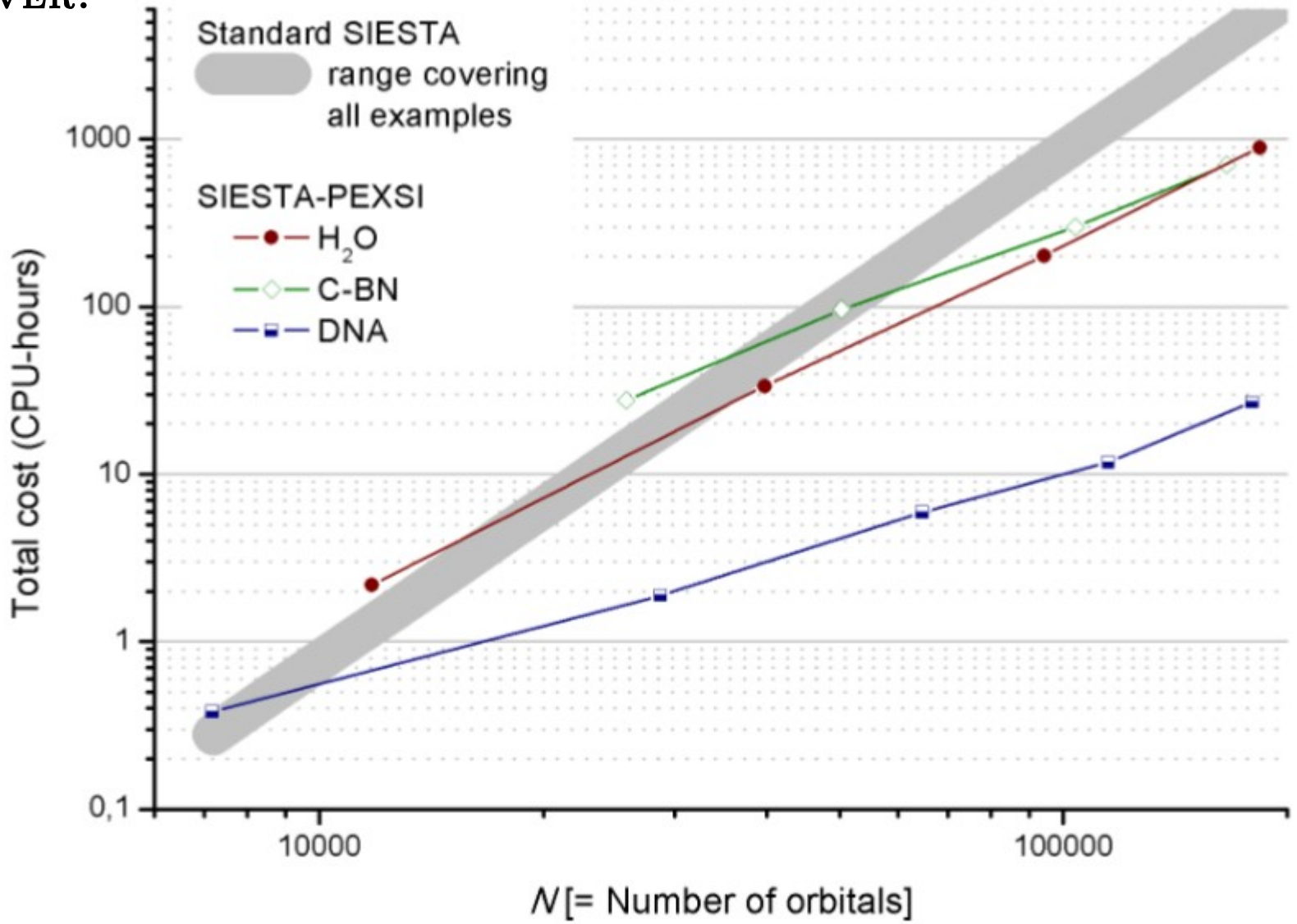
From Personal Experience:



Information from: Siesta Tutorial in Zaragoza (2014)

A Word About Parallelism in Siesta

PEXSI Solver:



Information from: Siesta Tutorial in Zaragoza (2014)

A Word About Parallelism in Siesta

IOP Publishing

Journal of Physics: Condensed Matter

J. Phys.: Condens. Matter **26** (2014) 305503 (15pp)

doi:10.1088/0953-8984/26/30/305503

SIESTA-PEXSI: massively parallel method for efficient and accurate *ab initio* materials simulation without matrix diagonalization

Lin Lin[1], Alberto García[2], Georg Huhs[3] and Chao Yang[1]

Abstract

We describe a scheme for efficient large-scale electronic-structure calculations based on the combination of the pole expansion and selected inversion (PEXSI) technique with the SIESTA method, which uses numerical atomic orbitals within the Kohn–Sham density functional theory (KSDFT) framework. The PEXSI technique can efficiently utilize the sparsity pattern of the Hamiltonian and overlap matrices generated in SIESTA, and for large systems it has a much lower computational complexity than that associated with the matrix diagonalization procedure. The PEXSI technique can be used to evaluate the electron density, free energy, atomic forces, density of states and local density of states without computing any eigenvalue or eigenvector of the Kohn–Sham Hamiltonian. It can achieve accuracy fully comparable to that obtained from a matrix diagonalization procedure for general systems, including metallic systems at low temperature. The PEXSI method is also highly scalable. With the recently developed massively parallel PEXSI technique, we can make efficient use of more than 10 000 processors on high performance machines. We demonstrate the performance and accuracy of the SIESTA-PEXSI method using several examples of large scale electronic structure calculations, including 1D, 2D and bulk problems with insulating, semi-metallic, and metallic character.

More About Siesta

- Tutorials: webpages of Javier Junquera
- Documentation (tutorials, manuals, videos): old Siesta page
- About the method: Technical Publications
- Getting Siesta and collaborating: Launchpad
- Asking questions: Siesta Mailing List

dieguez@tau.ac.il