

Master in Nanoscience 2020-2021

Advanced Theoretical Methods in Nanoscience

Iñaki Juaristi, Aran Garcia-Lekue, Maite Alducin

Density Functional Theory (DFT)

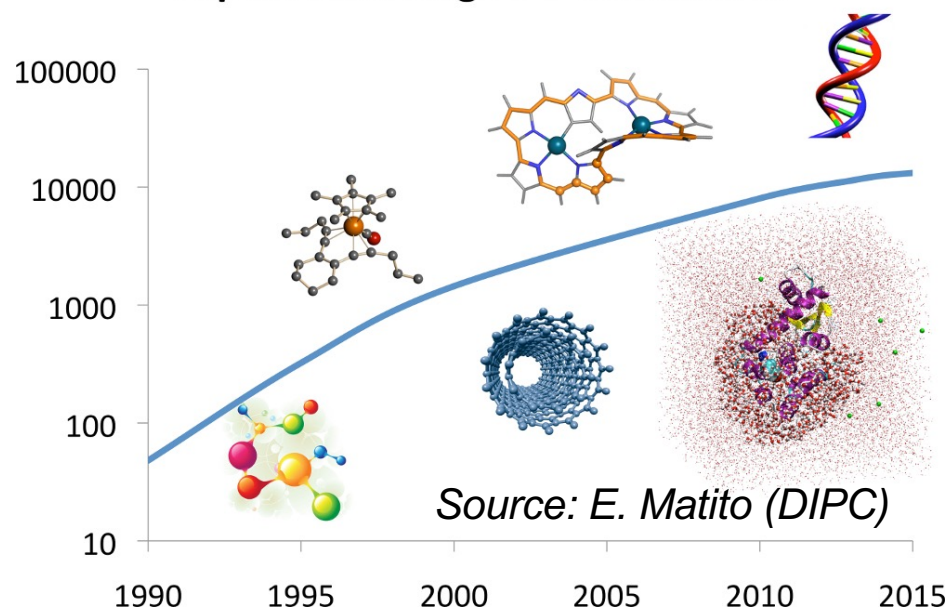
Based on two theorems
Hohenberg and Kohn (1964), Kohn and Sham (1965)

Many body problem

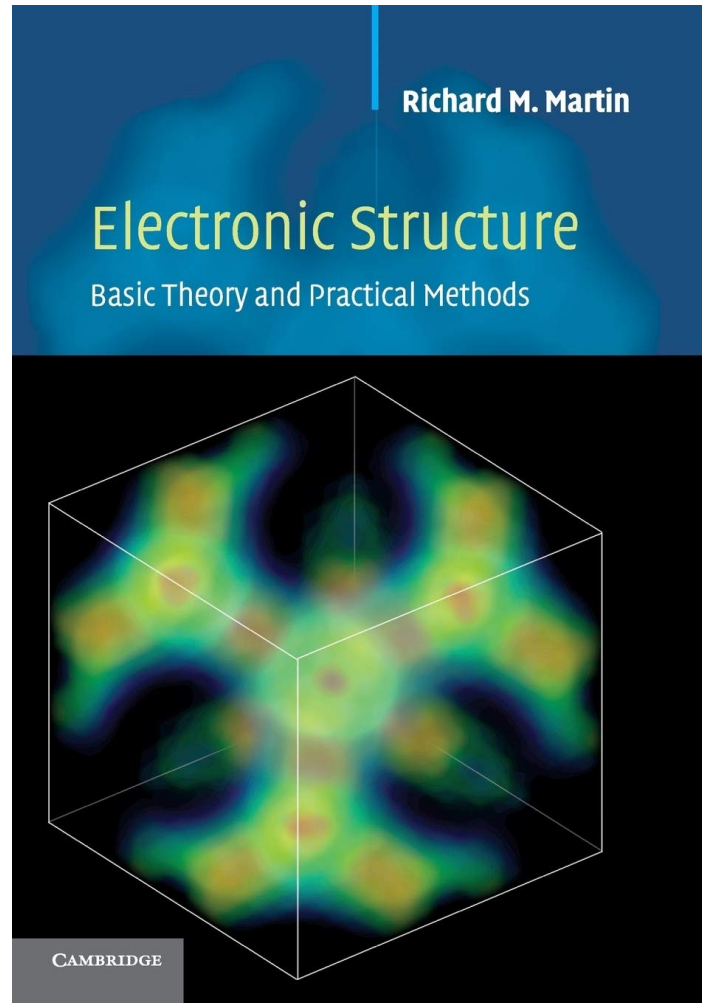


Set of single-particle equations

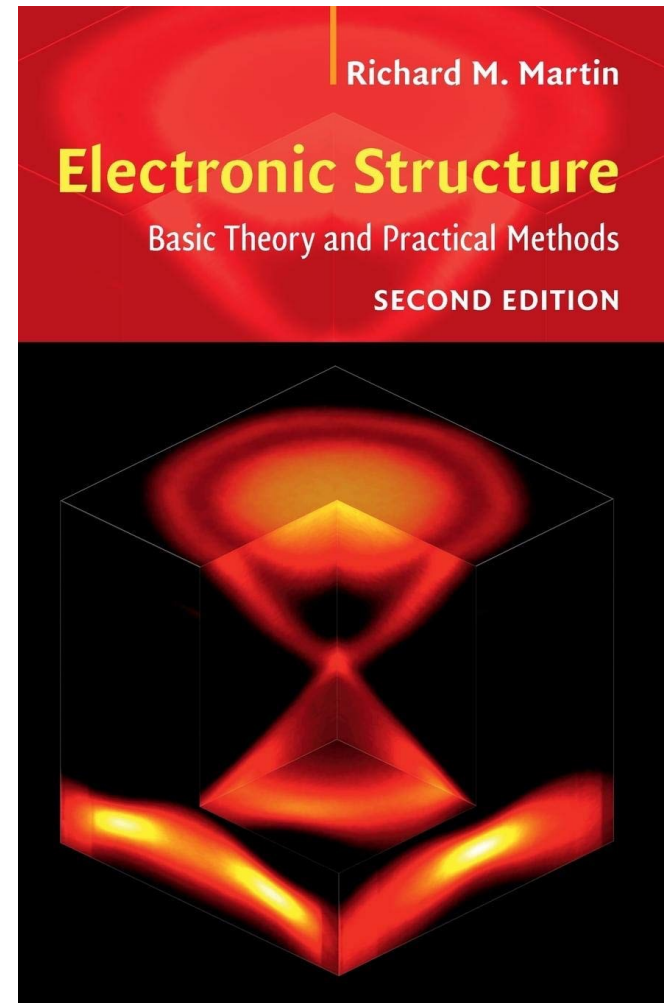
Papers including DFT calculations



Density Functional Theory (DFT)



1st edition: Comprehensive review of DFT, including most relevant references and exercises



2nd edition: advances in DFT, introduction to Berry phases and topological insulators, and many new examples of applications.

Kohn-Sham equations

Kohn-Sham equations at constant electronic charge for an external potential v

$$\left(-\frac{\nabla^2}{2} + v_{\text{eff}}[n; \vec{r}]\right) \Psi_i(\vec{r}) = \underbrace{E_i \Psi_i(\vec{r})}_{\text{Kohn-Sham energies \& states}} \rightarrow n(\vec{r}) = \sum_{i=1}^N |\Psi_i(\vec{r})|^2$$

$$v_{\text{eff}}[n; \vec{r}] = v(\vec{r}) + \underbrace{\int \frac{n(\vec{r}')}{|\vec{r} - \vec{r}'|} d\vec{r}'}_{\text{Hartree potential}} + v_{\text{xc}}[n; \vec{r}] \rightarrow \underbrace{v_{\text{xc}}[n; \vec{r}] = \frac{\delta E_{\text{xc}}[n]}{\delta n(\vec{r})}}_{\text{Exchange-correlation potential}}$$

This determines the density (and ground-state energy)
as long as we know how to write xc functional

Exchange-correlation functional

Exchange and correlation potential

$$v_{xc}(\mathbf{r}) = \frac{\delta E_{xc}[n(\mathbf{r})]}{\delta n(\mathbf{r})} = \frac{\partial [n(\mathbf{r})\epsilon_{xc}(\mathbf{r})]}{\partial n(\mathbf{r})}$$

Local density approximation (LDA)

$$E_{XC}^{LDA}[n] = \int d^3\mathbf{r} n(\mathbf{r}) \epsilon_{XC}^{unif}[n(\mathbf{r})].$$

$$\epsilon_x^{hom} = -\frac{3}{4} \left(\frac{3n(\mathbf{r})}{\pi} \right)^{1/3},$$

- PZ (Perdew-Zunger parametrization for the correlation part)
- Useful approximation for solids with slowly varying electron density
- Tends to over-bind atoms

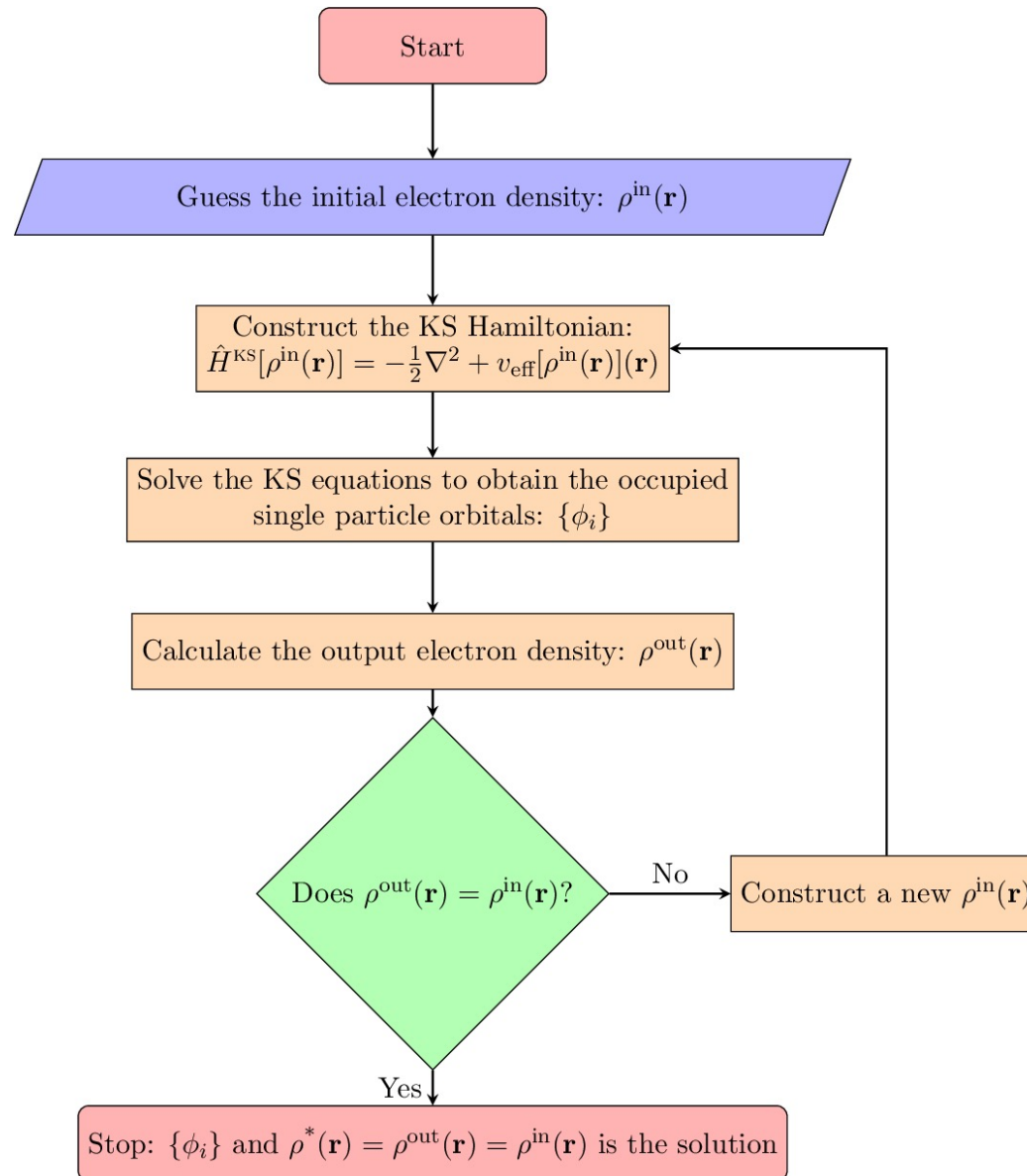
Generalized gradient approximation (GGA)

$$E_{XC}^{GGA}[n] = \int d^3\mathbf{r} n(\mathbf{r}) \epsilon_{XC}[n(\mathbf{r}), |\nabla n(\mathbf{r})|].$$

Enhancement factor
(several recipes)

- PBE (Perdew-Burke-Ernzerhof)
- Tends to under-bind atoms

Typical DFT program flow chart



Periodic boundary conditions

Simulation of periodic, infinite systems



Unit cell (defined by lattice vectors \mathbf{a}_i) + periodic boundary conditions



Bloch theorem: $\psi_{n,\mathbf{k}}(\mathbf{r}) = \underbrace{e^{i\mathbf{k}\cdot\mathbf{r}}}_{\text{Plane wave}} \underbrace{u_{n,\mathbf{k}}(\mathbf{r})}_{\text{Periodic function}}$

n: band index

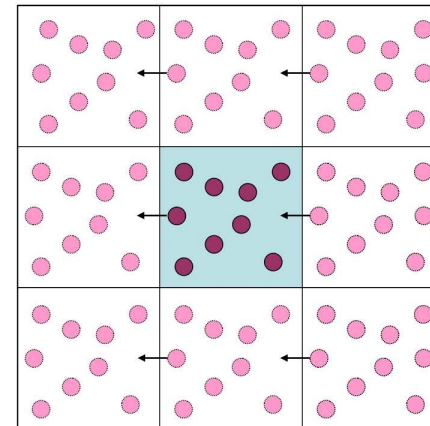
k: crystal wave vector



The wave-function of a system only has to be calculated within the unit cell

$$\psi_{n,\mathbf{k}+\mathbf{K}}(\mathbf{r}) = \psi_{n,\mathbf{k}}(\mathbf{r})$$

↑
Reciprocal lattice vector
(within 1st Brillouin zone)



Periodic boundary conditions

Instead of computing an **infinite** number of electronic **wave functions**

Finite number of **wave functions** at an infinite number of k-points.

In practice: electronic wave functions at k-points that are very close together will be almost identical \Rightarrow

It is possible to represent electronic wave functions over a region of k-space by the wave function at a single k-point.

$$\int d\vec{k} \longrightarrow \sum_{\vec{k}} \Delta\vec{k}$$

Source: J. Junquera (Univ. Cantabria)

Pseudopotentials

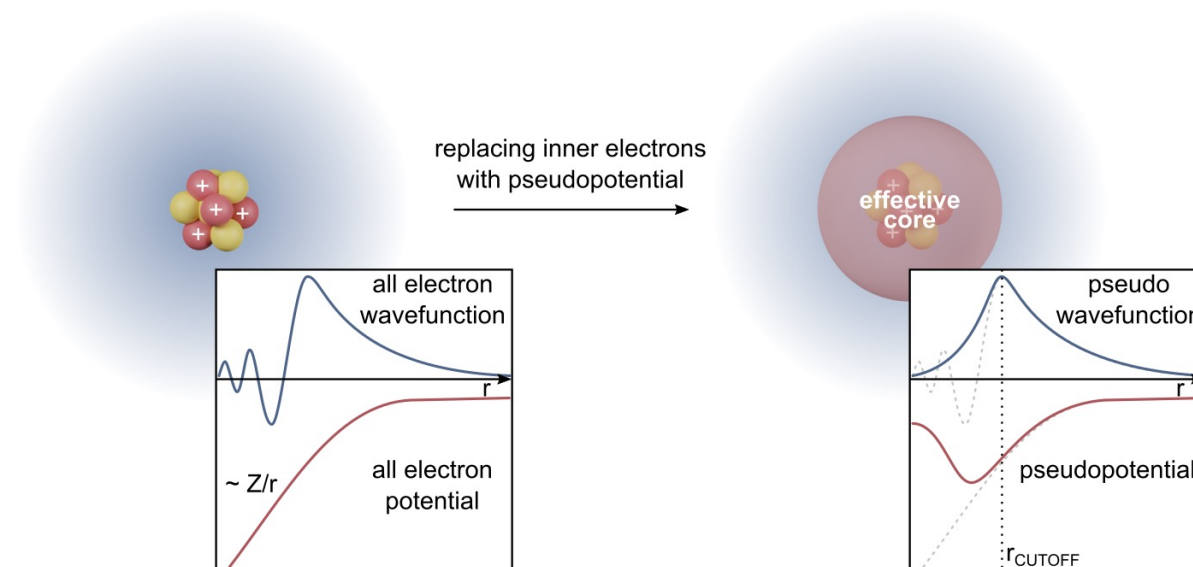
Core electrons are chemically inert (only valence electrons involved in bonding)

+

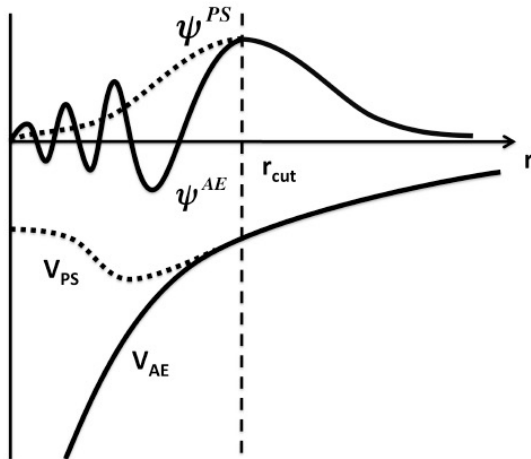
Core electrons make the calculation more expensive



Idea: Ignore the dynamics of the core electrons (freeze them)
And replace their effects by an effective potential



Pseudopotentials



Required:

- Reproduce atomic scattering properties
- Transferable

Desirable

- Norm-conserving (for better transferability)
- Soft
- Non-locality

Different types of pseudopotentials:

- Norm-conserving inside cut-off sphere (Kleynman-Bylander)

$$V_{\text{ion}} = V_{\text{LOC}} + \sum_{lm} \frac{|\phi_m^0 \delta V_l \rangle \langle \delta V_l \phi_{lm}^0|}{\langle \phi_{lm}^0 | \delta V_l | \phi_{lm}^0 \rangle}$$

*We will use these
in SIESTA*

- Ultrasoft (Vanderbilt)
- Projected Augment Waves (PAW) (Blöchl)

Basis sets: localized orbitals or plane waves

Basis set: choice of functions for the representation of the wave-function

Plane waves (VASP code)

$$\psi_{n,\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{n,\mathbf{k}}^{\mathbf{G}} e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}}$$

*Infinite sum over
reciprocal lattice vectors*

In practice: use of an **energy-cutoff**

$$E_{\text{cut}} = \frac{\hbar^2}{2m} G_{\text{cut}}^2$$

$$\psi_{n,\mathbf{k}}(\mathbf{r}) = \sum_{|\mathbf{k}+\mathbf{G}| < G_{\text{cut}}} c_{n,\mathbf{k}}^{\mathbf{G}} e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}}$$

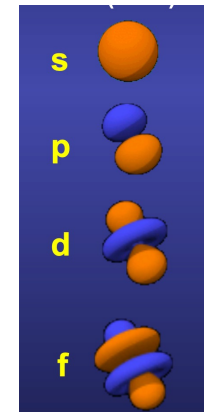
Localized atomic orbitals (SIESTA code)

Basis set built up by a linear combination of orbitals centered on the positions of the ions

$$\phi_{Inlm}(\mathbf{r}) = \phi_{I nl}(r_I) * Y_{lm}(\mathbf{r}_I),$$

Index of an atom Angular momentum Radial part Spherical harmonics

- Single- ζ (SZ): single radial function l per angular momentum
- Double- ζ polarized (DZP): double radial function + add shells with different symmetry (different l)



Advantages/Disadvantages??

What shall we be doing...

We will use the SIESTA code

(<https://departments.icmab.es/leem/siesta/>)

to calculate DFT-based structural and electronic properties

3D



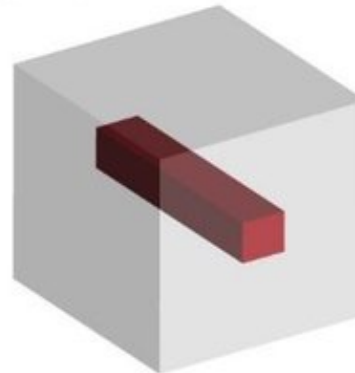
Bulk crystals

2D



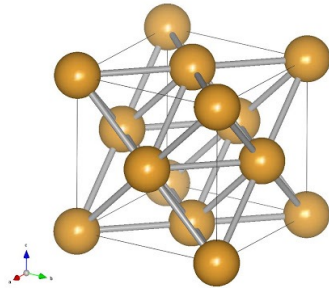
Graphene

1D



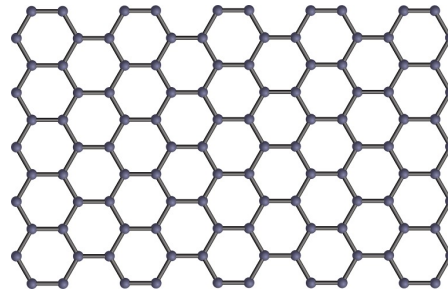
*Graphene
nanoribbons*

Bulk Au



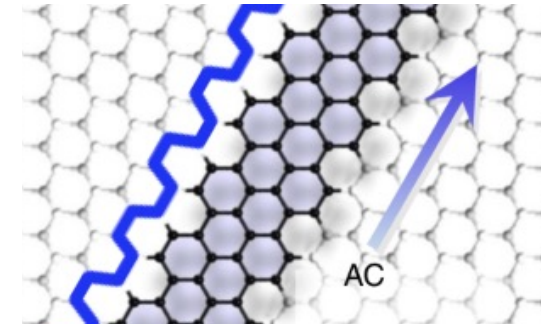
- Getting familiar with input/output
- Convergence (k-points)
- Obtain lattice constant for Au(fcc)
- Effect of basis set, xc functional

Graphene



- Construct unit cell and Brillouin zone
- Convergence (k-points)
- Calculate electronic band structure
- Calculate electronic density of states (DoS)

Graphene nanoribbons



- Relax atomic structure
- Calculate electronic band structure and DoS
- Obtain bandgap for GNRs of different width
- Obtain wavefunctions of frontier orbitals

How to log in, scripts...

Calculations to be carried out in the cluster ATLAS at the DIPC computing center

```
>> ssh -X username@ac-02.sw.ehu.es
```

```
>> ssh -X atlas-fdr-login-01
```

Basic unix commands

- **cd** *directory 1* **Change current directory to *directory 1***
- **cd** **Change back to your home directory**
- **cp** *file1 file2* **Copy *file1* to *file2***
- **mv** *file1 file2* **Move *file1* to *file2***
- **mv** *dir1 dir2* **Move *dir1* to *dir2***
- **rm** *file1* **Remove *file1***
- **rm -r** *dir1* **Remove *directory1***
- **mkdir** *dir1* **Make new *directory* named *dir1***
- **ls** **List the files in the current directory**
- **ls** */dipc/user* **List the files in */dipc/user* directory**
- **more** *file1* **Look at the content of *file1*, use 'q' to get out**
- **cat** *file1* **Type out *file***

How to log in, scripts...

Calculations to be carried out in the cluster ATLAS at the DIPC computing center

```
>> ssh -X username@ac-02.sw.ehu.es
```

```
>> ssh -X atlas-fdr-login-01
```

Launch jobs from /scratch/username/... with a script.

You can find a sample job.pbs, input files, etc. in /scratch/username

Job submission:

```
>> sbatch job.pbs
```

Job tracking:

```
>> squeue
```

Delete jobs:

```
>> scancel <job.id>
```


Sample ATLAS script (job .pbs) for SIESTA jobs

```
#!/bin/bash
#SBATCH --partition=esc
#SBATCH --job-name=TEST_SIESTA
#SBATCH --cpus-per-task=1
#SBATCH --mem=20gb
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
```

Maximum allowed tasks
per node = 4 !!!

Launch jobs from your
/scratch/\$USER directory

```
module load SIESTA
```

Load siesta version to be used

```
mpirun -np $SLURM_NTASKS siesta < RUN.fdf >& OUTPUT
```

This is the executable of the
loaded siesta version

Main input file

Main output file