

Castep & Real Space Integrals

MJR

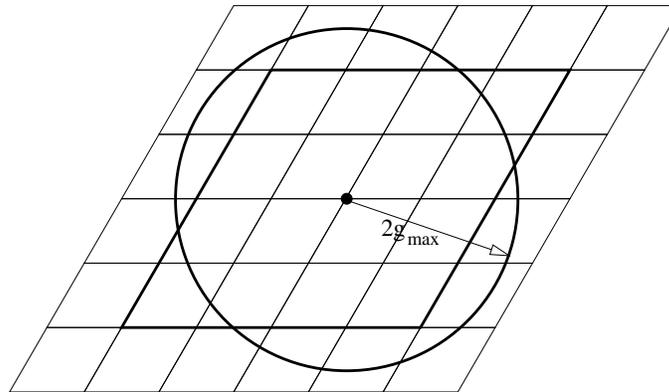
March 3, 2013

1 Real and Reciprocal Representations

The wavefunction in Castep is expressed in a plane wave basis. That the system is periodic means that the permitted g -vectors are discrete with a constant spacing, rather than continuous. The kinetic energy cut-off then imposes a maximum value for the modulus of a g -vector, g_{max} , leading to a finite number of permitted g -vectors.

The g -vectors in the basis set fill a sphere. The basic cell of the reciprocal space lattice is a rhombohedron (i.e. a cube, perhaps distorted to remove restrictions on side lengths and angles). The reciprocal space FFT grid should be large enough to contain all the sphere filled by the basis set. If so, it can exactly represent any function expressed in the wavefunction basis set, for it contains every function in that basis set (and a few more besides, the extra being orthogonal to the others, and being of an energy above the cut-off energy).

The charge density, being defined as $\psi\psi^*$, is the product of two functions containing g -vectors of moduli from zero to g_{max} . It will therefore contain g -vectors of moduli from zero to $2g_{max}$. Whether the reciprocal space FFT grid is sufficient to hold this function exactly depends on the choice of grid-size. Conventionally it does not, and the components discarded are assumed to be too small to impact the result. Choosing an FFT grid which encloses a sphere of radius $1.75g_{max}$ rather than $2g_{max}$ reduces the memory requirements for the FFT grid to $(\frac{7}{8})^3$ of what they would otherwise be, and CPU time requirements are reduced by a slightly larger amount. This saving of one third comes at little cost in accuracy.



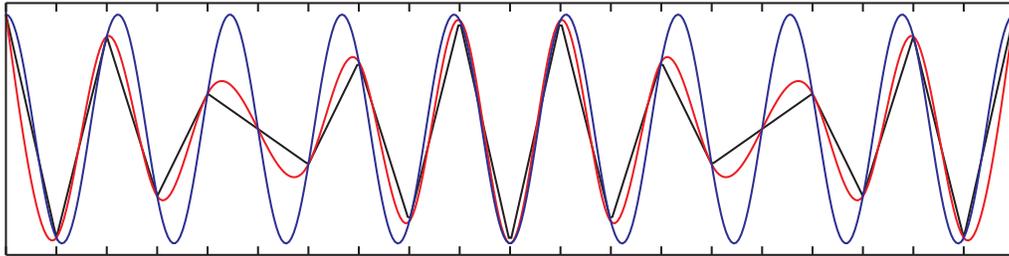
An FFT grid which contains all components to $2g_{max}$ (whole figure)
and all to $1.75g_{max}$ (bold lines)

The size of the cell in g -space is determined by the reciprocal of the size of the unit cell in real space. The size of the FFT grid then determines the extent of the volume of g -space used. Doubling the cell in real space halves the cell in reciprocal space, and thus requires the FFT grid size to be doubled to keep the same extent in g -space, i.e. the same cut-off.

1.1 Real Space Accuracy

The real space representation of the wavefunction is the FFT of the exact reciprocal space representation. Its values on the real space grid will be exact, its sum will be exact, and the result of FFTing it back to reciprocal space will be exact.

Very little else is exact. The real space grid is surprisingly coarse, and the wavefunction will vary considerably between its points. Interpolation is hard – linear, or even polynomial, interpolation does not really reflect the correct form of the wavefunction.



The function $\cos(9 \times 2\pi x)$ evaluated at 20 points from zero to 0.95. The grid is sufficient that it can be FFTed exactly onto a g -space grid of 20 points. All g -space components will be zero save those at ± 9 . But linear (black) and cubic spline (red) interpolation differ significantly from the correct function (blue). The grid points are the points at which all three functions meet.

2 Real-valued Functions and Even Grids

The FFT of a real-valued function has the property that $f(-g) = f^*(g)$. Furthermore, the $g = 0$ component, being the sum of the components on the real grid, is itself real.

If the FFT grid size is odd, then one has a $g = 0$ component, and an even number of components which are in conjugate pairs. Note that there were real numbers on the original grid, and on the complex grid there are the same number of degrees of freedom, as there are just two degrees of freedom per conjugate pair, and one for the $g = 0$ component.

If the FFT grid size is even, then the last g component which has no conjugate pair. This is at $g = \pi$, so the factor $\exp(igx)$ alternates between $+1$ and -1 on the real space grid. The coefficient of this g component will be real, and it is sometimes said that it forms a conjugate pair with itself. Again one has the expected number of degrees of freedom.

However, whereas for the components which form conjugate pairs one can indeed pair them to form

$$\begin{aligned} f(g) \exp(igx) + f(-g) \exp(-igx) &= f(g) \exp(igx) + (f(g) \exp(igx))^* & (1) \\ &= \mathcal{R}e(f(g) \exp(igx)) & (2) \end{aligned}$$

which is a function which is real everywhere, this is not the case for the unpaired $g = \pi$ component. The function $f(\pi) \exp(i\pi x)$, if $f(\pi)$ is real, is real for integer x , but complex for non-integer x .

In other words, expressing a real-valued function (such as a charge density) on an even grid and then FFTing it results in a representation which is real on the original grid, but for which evaluation between the grid points results in non-zero imaginary parts.

There are ways of eliminating this problem, which involve treating this last component specially and always considering it not as $f(\pi) \exp(i\pi x)$ but as

$$\frac{1}{2}f(\pi) \exp(i\pi x) + \frac{1}{2}f(\pi) \exp(-i\pi x) \quad (3)$$

which, as $f(\pi)$ is real, is real everywhere, and can be written equivalently as $f(\pi) \cos(\pi x)$.

Another method is to argue that the modulus of the Fourier coefficient for this last \mathbf{g} -vector must be insignificant. If it is not, then there are concerns about its neighbouring, slightly larger, \mathbf{g} -vectors which are being ignored completely. If it is insignificant, then it hardly matters how it is treated.

A final method is to ban even grid sizes.

This section is equivalent to worrying that for odd grids, with $ngx = 2n + 1$, one has a $g = 0$ component, and then \mathbf{g} components up to $\pm 2\pi n/ngx$. For even grids with $ngx = 2n + 2$, the \mathbf{g} components extend up to $+pi$, but only to $-2\pi n/ngx$, or $-\pi + 2\pi/ngx$. The grid is not at all symmetric under a transform taking g to $-g$.

3 Real Tricks

3.1 High Density Grids

One can take the reciprocal space grid and extend it to a larger size by padding with zeros. This changes nothing – one is adding \mathbf{g} -vectors far outside one's basis states, and orthogonal to them. It does then allow one to FFT back onto a higher density real space grid, which might be more appropriate for linear or polynomial interpolation. One may wish to take care when padding even grids, to ensure that the unpaired component becomes paired with its conjugate.

One should also note that whereas mathematically one is padding with high valued \mathbf{g} -vectors being set to zero, in terms of array presentation it is the middle of the array which should be set to zero. An FFT is invariably stored with the first half of the array storing the components from zero to g_{max} , and the second half from $-g_{max}$ to (but excluding) zero.

3.2 High Density Grids and the Charge Density

As

$$\rho(\mathbf{r}) = \psi(\mathbf{r})\psi^*(\mathbf{r}) \quad (4)$$

$$= \left(\sum_{\mathbf{g}} \psi(\mathbf{g}) \exp(i\mathbf{g}\mathbf{r}) \right) \left(\sum_{\mathbf{g}} \psi^*(\mathbf{g}) \exp(-i\mathbf{g}\mathbf{r}) \right) \quad (5)$$

the charge density contains \mathbf{g} -vectors of up to twice the modulus of the greatest \mathbf{g} -vectors in the wavefunction. It can thus be represented on a reciprocal space grid which contains the sphere of radius $2g_{max}$.

The following recipe therefore produces the charge density exactly.

- Place wavefunction on \mathbf{g} -space grid sufficiently large to accommodate components of twice the modulus of the largest non-zero component present.
- FFT to real space.
- Calculate $\psi\psi^*$ on the real-space grid.
- FFT back to reciprocal space.

Mathematically this is identical to the product of sums above, but computationally it is order $N \log N$, not order N^2 . The result is a set of coefficients of g -space to generate the charge density which exactly corresponds to the original wavefunction, and also that charge density evaluated on a real-space grid.

3.3 Special Points

If one wants a real-space value at a particular point, then one can evaluate

$$F(\mathbf{r}) = \sum_{\mathbf{g}} f(\mathbf{g}) \exp(i\mathbf{g} \cdot \mathbf{r}) \quad (6)$$

directly.

4 Integrals

This section assumes that the cell is orthorhombic, and treats just a single dimension, z , in which the cell is assumed to run from 0 to c . Generalisation is left to the reader.

4.1 Charge

To find the charge in part of the cell, a part not necessarily starting or ending on a real-space grid point, one considers

$$\int_{c_1}^{c_2} \rho(z) dz = \int_{c_1}^{c_2} \sum_{g_z} \rho(g_z) \exp(i g_z z) dz \quad (7)$$

$$= \sum_{g_z} \int_{c_1}^{c_2} \rho(g_z) \exp(i g_z z) dz \quad (8)$$

$$= \sum_{g_z} \left[\rho(g_z) \frac{1}{i g_z} \exp(i g_z z) \right]_{c_1}^{c_2} \quad (9)$$

$$= \sum_{g_z} (\exp(i g_z c_2) - \exp(i g_z c_1)) \frac{1}{i g_z} \rho(g_z) \quad (10)$$

This is exact, with no worries about how to interpolate on the real space grid, or to perform numeric integration on the real space grid.

There is a slight problem about the $g_z = 0$ component, so one needs to consider the expansion of that term in small g_z , and soon concludes that

$$(c_2 - c_1) \rho(g_z)|_{g_z=0} + \sum_{g_z \neq 0} (\exp(i g_z c_2) - \exp(i g_z c_1)) \frac{1}{i g_z} \rho(g_z) \quad (11)$$

is correct. Of course if $c_2 = c$ and $c_1 = 0$, this reduces to $c \rho(g_z = 0)$.

4.1.1 Dipoles

Consider a dipole moment

$$\mathbf{p}(\mathbf{r}_0) = \int_{cell} (\mathbf{r} - \mathbf{r}_0) \rho(\mathbf{r}) d\mathbf{r} \quad (12)$$

It is tempting to write this as

$$\mathbf{p}(\mathbf{r}_0) = \sum_{\mathbf{r}} (\mathbf{r} - \mathbf{r}_0) \rho(\mathbf{r}) \quad (13)$$

and then calculate it in this manner

However, if instead one considers the z component of $\mathbf{p}(\mathbf{r}_0)$, $p_z(z_0)$, and writes $\rho_z(z)$ for the linear charge density along z , i.e.

$$\rho(z) = \int_{cell} \rho(\mathbf{r}) dx dy \quad (14)$$

with c being the extent of the cell in the z direction one can write

$$p_z(z_0) = \int_0^c (z - z_0) \rho(z) dz \quad (15)$$

$$= \int_0^c (z - z_0) \sum_{g_z} \rho(g_z) \exp(i g_z z) dz \quad (16)$$

$$= \sum_{g_z} \rho(g_z) \int_0^c (z - z_0) \exp(i g_z z) dz \quad (17)$$

$$= \sum_{g_z} \rho(g_z) \left[\frac{1}{i g_z} z \exp(i g_z z) + \frac{1}{g_z^2} \exp(i g_z z) - \frac{1}{i g_z} z_0 \exp(i g_z z) \right]_0^c \quad (18)$$

$$= \sum_{g_z} \frac{c}{i g_z} \rho(g_z) \quad (19)$$

noting that $\exp(i g_z z) = 1$ at $z = 0$ and $z = c$. Again one must consider the $g_z = 0$ term separately, and, after expansion, one obtains $\frac{1}{2}c^2 - z_0 c$ for the multiplier for this term.

5 Summary

Performing the above integrals analytically removes concerns about the accuracy of numerical integration on a rather coarse real-space grid. However, the code to do so is generally more complicated than the real space code, so there is more scope for errors. In particular, nothing is said here about normalisation.

6 Appendix

Pyxplot code for interpolation figure. This will create a file called 'dat' holding the data, and produce the plot given.

```
f(x)=cos(2*pi*x*9)
set output 'dat'
set samples 21
tabulate [0:1] f(x)
interpolate linear f_linear() 'dat'
interpolate spline f_spline() 'dat'
set width 10
set size ratio 1/4
set samples 400
set nokey
set noxtics
set noytics
set mxtics 0.05
plot [0:1][-1.1:1.1] f_linear(x), f_spline(x), f(x)
# set output 'interpolate.eps'
# set terminal eps
# replot
```