eCSE09-3: Computing infra-red spectra using finite differencing in CASTEP

Joseph C.A. Prentice[1] and Jonathan R. Yates[1]

[1] *Department of Materials, University of Oxford, Parks Road, Oxford, OX1 3PH*

November 8, 2024

Abstract

Infra-red (IR) spectroscopy is a vitally important tool for elucidating materials' structure, and computational simulation of IR spectra using density functional theory has a key role to play in understanding experimental data. The usual method for computing IR spectra, density functional perturbation theory, is powerful, but is often limited to a subset of the wide array of possible electronic structure methods due to its complexity. In this work, we implement an alternative approach to computing IR spectra – finite differencing of electric polarisation – in the UK flagship code CASTEP, making simulation of IR spectra possible using all electronic structure methods implemented in the code. As part of this, we implemented corrections to the electric polarisation arising from the use of ultra-soft pseudopotentials, and parallelised our implementation over k-points, bands, and task farms. We also extended the method to compute piezoelectric tensors via a similar approach. With this work, the simulation of IR spectra is now made possible in systems where simple electronic structure methods do not provide a good description of the system, as well as increasing the efficiency of these calculations, significantly increasing the utility of computation in this field.

Keywords: ARCHER2, eCSE, density functional theory, infra-red spectroscopy, Berry phase, polarisation

1 Introduction

Throughout this work we will use atomic units, where $e = \hbar = m_e = 4\pi\epsilon_0 = 1$. Ω_0 is the volume of the unit cell.

1.1 Density functional theory

Computational modelling is now a key facet of research into materials from small molecules to complex bulk systems, providing an underlying understanding for experimental results. In particular, for the many systems where the electrons dominate the behaviour, *ab initio* simulation of electronic structure is vital, as it allows us to understand the system in terms of the behaviour of the electrons directly from the principles of quantum mechanics. In theory, this can be done by solving the Schrödinger equation for the many-electron wavefunction, which contains all the information knowable about the system. Such a wavefunction is a $3N$-dimensional object – this, coupled with the fact that the electrons strongly interact with one another, means that in practice this approach is intractable for all but the smallest systems. To deal with larger systems, we must instead use other methods.

The most popular of these is density functional theory (DFT), which is used in approximately 30,000 published papers each year. DFT makes use of the fact that all properties of the system (including its energy) can be written as functionals of the electron density $n(\mathbf{r})$, a 3-dimensional object, significantly decreasing the complexity of the problem. DFT is formally exact: given the appropriate functional for the energy, we can find the true ground state of the system – exactly the same result as we would have found by solving the Schrödinger equation – by minimising this functional with respect to the density. The exact form of this functional is unknown; however, it can be divided into several contributions which are well-understood, leaving one part still unknown, commonly called the exchange-correlation (XC) functional. DFT has risen to prominence thanks to the existence of many approximate XC functionals (of varying degrees of complexity), even the simplest of which can give accurate results in many cases.

There are several different software packages implementing DFT, each with different capabilities and focuses. Among the most popular of these is `CASTEP`[1], one of the UK's flagship codes which is used heavily on ARCHER2. In the following, we outline some of the key features of `CASTEP` that are relevant to the work in this eCSE.

`CASTEP` utilises the Kohn-Sham (KS) variant of DFT[2]. This approaches the problem of minimising the energy with respect to the density by actually treating an auxiliary system of *non-interacting* pseudo-electrons, which are defined to have the same density as the full system. By removing the interactions, the problem can be reduced to a single-particle problem, and the 'wavefunctions' of the eigenstates of these pseudo-electrons, known as KS orbitals or states, can be computed. Any resulting error, for example in the kinetic energy, is swept into the XC functional. These KS orbitals are then used to construct the density.

Because of its focus on solid materials with periodic structures, `CASTEP` utilises a plane wave basis set, meaning that quantities such as the KS orbitals are expressed as a linear combination of plane waves with wavevectors commensurate with the unit cell of the system – the periodicity of these plane waves means that the periodicity of the system is therefore 'baked in'. Practically, only plane waves below a given cut-off are included – a larger cut-off means that properties are sampled more finely, but at the cost of greater computational expense.

This cut-off becomes particularly relevant when considering how the KS orbitals behave close to atomic nuclei. Typically, only some of the KS states are appreciably delocalised across the system – these 'valence' states correspond to electrons that are involved in bonding. States that lie lower in energy will typically be very tightly localised around the nuclei – these 'core' states correspond to electrons too far below the Fermi level to be involved in bonding. The KS orbitals must be orthogonal to one another: in the region around the nuclei, this requirement means that the valence states must oscillate rapidly to maintain orthogonality with the localised core states. These rapid oscillations therefore require a high cut-off to accurately describe them, significantly increasing the cost of the calculation. However, the core electrons are irrelevant to many properties of interest. This means we can remove them from the calculation, merging their influence on the valence electrons with that of the nuclei, creating a 'pseudopotential'. These are designed so that they produce the correct behaviour away from the core region, but have significantly less complex behaviour within the core, reducing the cut-off required for a given accuracy.

The two most common types of pseudopotentials are known as norm-conserving and ultra-soft pseudopotentials (NCPs and USPs respectively). NCPs are the most basic form of pseudopotentials, whilst USPs introduce some additional complexity to allow for an even lower cut-off energy. USPs are typically not only more efficient, but also more accurate, making them the pseudopotential of choice, but the additional layer of complexity often introduces new contributions to quantities calculated within DFT. This means that many capabilities of DFT codes are often incompatible with USPs, simply because the theory describing these new contributions has not yet been outlined, or it has not yet been implemented in the code.

1.2 Modelling infra-red spectra

One of the many key capabilities of `CASTEP`, and DFT more generally, that makes it such a popular tool is its ability to model the vibrations of molecules and solids. Of particular relevance here is the ability to simulate the results of vibrational spectroscopy experiments, such as infra-red (IR) spectra.

Like other techniques in the field of vibrational spectroscopy, such as Raman spectroscopy and neutron/X-ray scattering, IR spectroscopy is an important experimental technique. It (indirectly) allows the structure of a material to be probed, by providing information about the vibrational (or phonon) modes of the material, which are strongly influenced by its structure. Importantly, it does this cheaply and non-destructively, making IR spectroscopy a key tool across many branches of science, from pharmaceuticals and biotechnology through to ceramics and glasses.

Because IR spectroscopy probes the vibrations of a material, not its structure directly, a combined experimental and computational approach is key. By simulating IR spectra of candidate structures and comparing to experimental data, we can interpret the experimental results and ascertain the structure of the material much more precisely than experiment could do alone. This makes developing code able to produce accurate simulations of IR spectra efficiently an extremely important goal.

To simulate an IR spectra, two main quantities must be computed: the vibrational/phonon frequencies of the system, and the intensity with which each vibrational/phonon mode absorbs electromagnetic radiation at that frequency (typically in the infra-red). To obtain the frequencies, it is necessary to compute the second derivatives of the energy with respect to atomic positions, often known as the matrix of force constants: $\frac{\partial^2 E}{\partial x_{i\alpha} \partial x_{j\beta}}$. For the intensities, it is necessary to compute the mixed second derivative of energy with respect to atomic positions and applied electric field, often known as the Born effective charges: $Z^*_{i\alpha\beta} = -\frac{\partial^2 E}{\partial x_{i\alpha} \partial \mathcal{E}_\beta}$.

This can be done in DFT by utilising the well-established, efficient, and accurate density functional perturbation theory (DFPT) approach. This is already implemented in `CASTEP`, and has been used successfully for some time. However, the DFPT approach has a major drawback – as new methods for computing electronic structure are introduced, DFPT requires that the required derivatives are explicitly derived and implemented. As these methods become more advanced, DFPT can become prohibitively complicated to implement and use. In `CASTEP`, DFPT for vibrational calculations is only implemented for relatively simple methods. Only common semi-local XC functionals, based on the local density approximation (LDAs) and generalised gradient approximation (GGAs), are included, and only NCPs can be used, rather than the more accurate and efficient USPs.

1.3 Finite differencing

Instead, we can compute these derivatives with a finite differencing method. This method allows derivatives to be computed by explicitly perturbing the system and recomputing the quantity in question for the perturbed system. We can do this for any degree of derivative, although it is most commonly done for first derivatives. For second derivatives, we will often compute one of the derivatives by other means, and then apply finite differencing to that quantity. This is the approach taken in this work. For example, to obtain the matrix of force constants $\frac{\partial^2 E}{\partial x_{i\alpha} \partial x_{j\beta}}$, we can take advantage of the fact that the forces on the atoms (which can be computed separately at essentially no cost) are the first derivative of the energy with respect to atomic position: $F_{j\beta} = -\frac{\partial E}{\partial x_{j\beta}}$, so $\frac{\partial^2 E}{\partial x_{i\alpha} \partial x_{j\beta}} = \frac{\partial F}{\partial x_{i\alpha}}$. We can therefore shift atom α by a small distance along direction i, and recompute the forces on all atoms $F_{j\beta}$. The second derivative we want is then approximated as the change in the force on atom β in direction j, divided by the change in position of atom α in direction i (multiplied by -1).

The quality of this approximation depends on the size of the perturbation – too large and the derivative could potentially include unwanted non-linearities, but too small and any signal will be overwhelmed by numerical noise. To make the calculation more robust, perturbations are typically made in both the positive and negative directions, and the results are combined appropriately.

The key advantages of the finite differencing method are its simplicitly, and therefore its applicability to any electronic structure method. Any quantity that can be computed by a given electronic structure method can also be differentiated with respect to any perturbation that can be applied to the system, without requiring the explicit computation of derivatives. In materials that are poorly described by the methods compatible with DFPT, such as some magnetic materials, materials under high pressure, and optoelectronic systems, finite differencing enables the use of alternative methods that describe such systems much better, such as hybrid or meta-GGA XC functionals, and DFT+U methods. It also effectively means that computations of these derivatives are, to a significant extent, future-proofed against the introduction of new and more complex electronic structure methods. This makes finite differencing a vital tool in DFT's kit, and is why this eCSE has focused on finite differencing as a method, rather than DFPT.

1.4 Polarisation

As noted above, the matrix of force constants can be obtained by taking the first derivative of the forces on the atoms with respect to atomic positions. To obtain the Born effective charges, we need to identify and compute a quantity that is equivalent to a first derivative of the energy (with respect to either electric field or atomic positions). We could compute the forces under the perturbation of a small applied electric field, but dealing with electric fields can be challenging in periodic boundary conditions, as they usually must have a unphysical sawtooth shape. Instead, we can take advantage of the fact that the electric polarisation

$P_\alpha = -\frac{1}{\Omega_0}\frac{\partial E}{\partial \mathcal{E}_\alpha}$. This means that the Born effective charges are equal to the derivative of the polarisation with respect to atomic positions:

$$Z^*_{i\alpha\beta} = \Omega_0 \frac{\partial P_\beta}{\partial x_{i\alpha}} \ . \tag{1}$$

Importantly, this derivative can be done with finite differencing using the same perturbations to atomic position that are used to compute the matrix of force constants, maximising the efficiency of the calculation.

Computing the polarisation is therefore key to simulating IR spectra with finite differencing; of course, the polarisation is in itself a useful observable quantity, beyond simply enabling the calculation of Born effective charges. One subtlety is that, according to the modern theory of polarisation, the electric polarisation of a periodic system is not single-valued, but can take many values depending on how the unit cell is drawn[3]. These values are equally spaced on what is known as the 'polarisation lattice', and when comparing two values of polarisation it is vital to ensure they are on the same branch of this lattice.

There are two contributions to the electric polarisation – from the ions, and from the electrons. The ionic contribution is easy to compute by treating the ions as point charges and multiplying them by their position. The electronic polarisation is a little harder to compute, but can be calculated within DFT directly from the KS orbitals using a Berry phase (BP) approach, which at its simplest takes the form[4]:

$$\mathbf{P} = \frac{\Omega_0}{(2\pi)^3} \sum_n \int_{\mathrm{BZ}} d^3\mathbf{k}\ i \left\langle u_{n\mathbf{k}} \,\middle|\, \nabla_{\mathbf{k}} \,\middle|\, u_{n\mathbf{k}} \right\rangle \tag{2}$$

where n runs over all occupied bands, and $u_{n\mathbf{k}}$ is defined by $\phi_{n\mathbf{k}}(\mathbf{r}) = u_{n\mathbf{k}}(\mathbf{r}) e^{i\mathbf{k}\cdot\mathbf{r}}$, where $\phi_{n\mathbf{k}}$ is the nth KS orbital at k-point $\mathbf{k}$. By Bloch's theorem, $u_{n\mathbf{k}}(\mathbf{r})$ is a function with the periodicity of the lattice. In practice, we usually have a grid of k-points arranged along each of the reciprocal lattice vectors, and we compute the component of the polarisation parallel to a given reciprocal lattice vector ($\mathbf{P}_\parallel$), considering each lattice vector in turn[4]:

$$\mathbf{P}_\parallel = -\frac{2}{\Omega_0} \frac{1}{G N_{k_\perp}} \sum_{\mathbf{k}_\perp} \mathrm{Im}\ \ln \prod_{\mathbf{k}_\parallel} \det\ M^{\mathbf{k}}_{mn} \ , \tag{3}$$

where

$$M^{\mathbf{k}}_{mn} = \left\langle u_{m\mathbf{k}} \,\middle|\, u_{n\mathbf{k}+\mathbf{b}} \right\rangle \ . \tag{4}$$

Here, we have grouped the grid of k-points into a set of strings which run parallel to the reciprocal lattice vector in question, $\mathbf{G}$ ($G = |\mathbf{G}|$). Each k-point is therefore labelled with two indices – $\mathbf{k}_\perp$ runs over the first k-point in each of the strings (i.e. it runs over the strings), whilst $\mathbf{k}_\parallel$ iterates over k-points within a string. $N_{k_\perp}$ therefore counts the number of strings. $\mathbf{b}$ is the spacing between grid-points along the direction of the reciprocal lattice vector in question, so $M^{\mathbf{k}}_{mn}$ is the overlap between the periodic part of the Kohn-Sham orbitals at k-points along the string. Due to periodic boundary conditions, the first and last k-points in a string are adjacent in this manner (although there is an additional phase factor of $e^{-i\mathbf{G}\cdot\mathbf{r}}$ in this case – periodicity demands that $\phi_{n\mathbf{k}}(\mathbf{r}) = \phi_{n\mathbf{k}+\mathbf{G}}(\mathbf{r})$, which implies that $u_{n\mathbf{k}+\mathbf{G}}(\mathbf{r}) = u_{n\mathbf{k}}(\mathbf{r}) e^{-i\mathbf{G}\cdot\mathbf{r}}$).

Before this eCSE, the computation of the electric polarisation through this BP formalism was already implemented within `CASTEP`, but only as a post-processing step after a ground state single point energy calculation. To enable the computation of Born effective charges through finite differencing, the structure of the code needed to be adjusted, to allow the routines calculating the electric polarisation to be called for each perturbation of atomic positions. This also needed to be compatible with the recently implemented task farming approach for accelerating phonon calculations, where different perturbations are treated almost as separate calculations and calculated independently, before bringing the results back together for the final computation of the IR spectrum.

However, the computation of the electric polarisation itself had some issues restricting its utility. The most important of these was that the Berry phase formalism was only compatible with NCPs, as it did not include important contributions that arise when using USPs. It was also only parallelised over G-vectors (the plane wave basis functions), whereas most of the rest of the code makes use of two additional parallelisms, over k-points and over bands. To make simulation of IR spectra truly accessible across all the electronic structure methods implemented in `CASTEP`, these issues needed to be addressed as well.

1.5 Aims of eCSE

This eCSE therefore aimed to address these issues through the following work packages (WPs):

- WP1: Implementing finite difference calculations of IR strengths
  - Implement the additional contribution to the BP polarisation from USPs
  - Implementing the calculation of Born effective charges, and thus IR absorption strengths, using a finite differencing method to obtain the derivative of the polarisation with respect to atomic position
- WP2: Improving parallelisation of computation of BP polarisation
  - Parallelise the calculation of polarisation over k-points
  - Parallelise the calculation of polarisation over bands
- WP3: Implementing task farm parallelisation for finite difference IR calculations
  - Implement task farm parallelisation for finite difference IR calculations

2 Work done

2.1 USP contributions to polarisation

Before implementing the additional terms arising in the BP polarisation from USPs, we need to establish what these are. These terms have been the subject of several pieces of previous work[4, 5, 6] due to their relevance in the calculation of polarisation, and in the computation of Wannier functions. However, to the best of our knowledge, no work has brought the full correct expressions together in one place – this is what we seek to do here.

There are in fact two additional USP contributions to the electric polarisation. The first of these is the more important, and corrects the overlap matrix $M^{\mathbf{k}}_{mn}$ defined in Eq. (4), so that it is now redefined as[4, 6]:

$$M^{\mathbf{k}}_{mn} = \langle u_{m\mathbf{k}} \,|\, u_{n\mathbf{k+b}} \rangle + \sum_{\alpha} \sum_{ij} Q_{\alpha;ij}(\mathbf{b}) \left\langle u_{m\mathbf{k}} \,\middle|\, \beta^{(\mathbf{k})}_{\alpha;i} \right\rangle \left\langle \beta^{(\mathbf{k+b})}_{\alpha;j} \,\middle|\, u_{n\mathbf{k+b}} \right\rangle \ . \tag{5}$$

Here, we have introduced two additional objects associated with USPs that are typically absent in NCPs: the projectors $|\beta\rangle$, and the augmentation charge Q. In both cases, we actually require these properties in reciprocal space (i.e. their Fourier transforms), hence the dependence on $\mathbf{k}$ and $\mathbf{b}$. α runs over atoms in the unit cell, whilst i, j run over the projectors. All other symbols have the same meaning as in Eq. (4). (Note that Eq. 23 in Ref. 4 has an error – the integral of the augmentation charge is used where its Fourier transform should be, as seen correctly in Ref. 6.) We refer to this new term as the Berry phase (BP) correction.

The second of the two correction terms is referred to as the expectation value (EV) correction, which is simply added to the polarisation calculated through Eq. (3). This is a simpler object, with the form:

$$\mathbf{P}^{\mathrm{BP}} = -\frac{2}{\Omega_0} \sum_{n} \frac{1}{N_k} \sum_{\mathbf{k}} \sum_{\alpha} \sum_{ij} \mathbf{d}_{\alpha;ij} \left\langle u_{m\mathbf{k}} \,\middle|\, \beta^{(\mathbf{k})}_{\alpha;i} \right\rangle \left\langle \beta^{(\mathbf{k})}_{\alpha;j} \,\middle|\, u_{n\mathbf{k}} \right\rangle \ . \tag{6}$$

n runs over occupied bands, α runs over atoms in the unit cell, and i, j run over projectors. All objects have already been introduced, except for $\mathbf{d}$, which is defined as:

$$\mathbf{d}_{\alpha;ij} = \int \mathrm{d}^3\mathbf{r}\ \mathbf{r}\, Q_{\alpha;ij}(\mathbf{r}) \ . \tag{7}$$

It only remains to compute these corrections. Some of the objects required were already computed within `CASTEP`, such as the 'beta-phis', as $\left\langle \beta^{(\mathbf{k})}_{\alpha;j} \,\middle|\, u_{n\mathbf{k}} \right\rangle$ are usually known. Others needed to be implemented, such as $Q_{\alpha;ij}(\mathbf{b})$ and $\mathbf{d}_{\alpha;ij}$. The fact that the augmentation charge is expressed as a radial function multiplied by a product of two spherical harmonics makes these integrals convenient to compute, as they each reduce to a single radial integral. The radial integral required for $\mathbf{d}_{\alpha;ij}$ is pre-computed at the beginning of the calculation, as part of the initialisation of the pseudopotentials; the radial integration for $Q_{\alpha;ij}(\mathbf{b})$ is computed on the fly, as the required value for $\mathbf{b}$ is unknown beforehand.

We also corrected a more general issue with performing electric polarisation calculations where the only k-point is the Γ-point, i.e., the centre of the Brillouin zone. It is still possible to perform BP calculations in this case (only 1 overlap enters into M_{mn} – that of the Γ-point with its periodic image). However, to speed up calculations using the Γ-point only, `CASTEP` makes some assumptions, in particular that the KS orbitals will be real. This is generally true, but poses problems when considering the KS orbitals at the periodic image of Γ. These orbitals are related to those at Γ by a phase factor of $e^{-i\mathbf{G}\cdot\mathbf{r}}$, as noted previously, which means they are now complex, not purely real. It therefore proved necessary to turn off these Γ-point optimisations when performing electric polarisation calculations.

To reduce the cost of the polarisation calculations, we also implemented a routine that used the symmetry operations of the system to ascertain if particular components of the polarisation should be identical, or even zero. In systems with high symmetry, this allows the number of polarisation calculations to be reduced. **NB** – when the symmetry enforces that a component of the polarisation must be zero, this applies to the *total* polarisation. Generally, the electronic and ionic contributions to this component will be non-zero, but will cancel out. However, to allow for the reduction of computational time, the electronic and ionic contributions are not calculated for such components, and are assumed to be zero. This means that if such symmetry optimisations are used, only the total polarisation remains completely reliable.

Upon making these changes, we were able to fulfil the first of our goals, WP1a, allowing us to perform electric polarisation calculations with USPs. The code was tested and benchmarked by considering two systems – a HF molecule in a large box, and the ferroelectric material $\mathrm{BaTiO_3}$. We chose HF because we expect the USP contributions in F to be among the largest, if not the largest, in the periodic table, making this a worst case scenario. To test the robustness of our approach for systems with different symmetries, we looked at a HF molecule in both a cubic box and a trigonal box. By considering a molecule, we were also able to check our results for the polarisation against existing functionality in `CASTEP` that can compute the dipole moment of isolated molecules using an finite position operator approach[7]. This approach considers the final charge density directly, and is therefore already compatible with USPs, making this a useful check. To make this comparison, we convert the polarisations calculated using the BP method into molecular dipole moments, by multiplying by the cell volume. To demonstrate the code's utility in periodic systems as well, $\mathrm{BaTiO_3}$ was chosen due to its well-known spontaneous polarisation.

The parameters used for the HF calculations were as follows: the cut-off energy was 800 eV; the XC functional was PBE[8]; the lattice parameter of the cubic box was 10 Å; the lattice parameters of the trigonal box were $a = 8$ Å, $b = 9$ Å, $c = 10$ Å, $\alpha = 85°$, $\beta = 80°$, $\gamma = 93°$. The parameters used for the $\mathrm{BaTiO_3}$ calculations were as follows: the cut-off energy was 1000 eV; the XC functional was PBE; a $9 \times 9 \times 9$ k-point grid was used; the lattice parameters for the orthorhombic cell were $a = 3.9988$ Å, $c = 4.0222$ Å. All pseudopotentials used were `CASTEP`'s own on-the-fly-generated (OTFG) pseudopotentials.

First, we demonstrate that the calculation of the dipole moment using the BP approach converges as the k-point grid size increases. This is shown in Fig. 1, with grids ranging in size from $1 \times 1 \times 1$ (the Γ-point), to $8 \times 8 \times 8$. We can see that all results converge rapidly, with a $3 \times 3 \times 3$ grid providing a result within 0.1 Debye of the converged value. The USP results converge slightly faster. We can see that the values obtained using the same method for the two different systems typically agree to within 0.002 Debye, an extremely high level of agreement, demonstrating the method's insensitivity to different symmetries. Some difference between the NCP and USP methods is to be expected, as they will produce slightly different electronic structures – however, they still agree extremely well on the dipole moment, to within 0.015 Debye, well within the bounds of acceptable accuracy.

We also broke the results down further in the USP case, examining the individual contributions of the BP and EV contributions to the polarisation. We do not present the results here, but it can be seen that the EV correction reduces rapidly in size as the k-point grid increases, meaning that for most practical purposes, the BP correction is all that matters. Indeed, this is the approach taken in other plane-wave pseudopotential DFT codes, such as `Quantum ESPRESSO`[9], where the EV correction is neglected.

Table 1 then compares the converged results from the BP method to those obtained from the alternative finite position operator (FPO) method in a cubic box. If we compare the results obtained with the cubic box using the same pseudopotentials, we can see that the methods agree to within 0.01 Debye, well within the

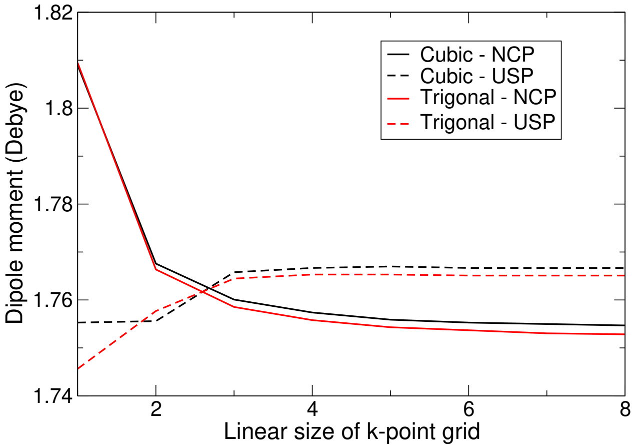

Figure 1: Dependence of dipole moment of HF, as calculated via the Berry phase polarisation approach, on size of k-point grid used. The solid and dashed curves were computed with norm-conserving and ultrasoft pseudopotentials (NCPs and USPs) respectively; the black and red curves were computed using a cubic and trigonal unit cell respectively.

	NCP		USP	
	Cubic	Trigonal	Cubic	Trigonal
FPO approach	1.754	–	1.759	–
BP approach	1.755	1.753	1.767	1.765

Table 1: Dipole moment of HF as calculated by different methods (either the finite position operator (FPO) method, or the Berry phase (BP) approach) and in different periodic boxes (either cubic or trigonal), with either NCPs or USPs. The FPO approach does not work for non-cubic boxes, so a value is not reported.

bounds of acceptable accuracy. We can also see that the values for the trigonal structure calculated using the BP approach agree well with those calculated for the cubic structure.

If we now compute the polarisation of $BaTiO_3$ using both NCPs and USPs, we obtain 0.0468 C m^{-2} and 0.0490 C m^{-2} respectively. These agree to within 5%, again demonstrating good agreement.

2.2 IR calculations with finite differences

To fulfil WP1b, we implemented the computation of Born effective charges using finite differencing, by adding an additional call to compute the electric polarisation for each perturbation. Each perturbation corresponds to the displacement of a particular atom – using the polarisations calculated for positive and negative perturbations of this atom (as discussed previously), a row of the Born effective charge tensor can be computed. This calculation is performed at the same time as the generation of the corresponding row of the force constant matrix. The resulting Born effective charge tensor is then passed back through to be used in the computation of the IR absorption strengths.

The main subtlety here arises from the fact that in periodic systems, polarisation is a multi-valued quantity, as described by the modern theory of polarisation[3]. Possible polarisation values sit on a lattice, separated by a polarisation quantum dependent on the size of the system. The derivative of polarisation is therefore defined as the derivative of the values within a single branch of the polarisation lattice. It can happen, however, that the polarisation computed for a negative perturbation ends up on a different branch to the polarisation computed for the corresponding positive perturbation. In this case, the derivative computed by finite differencing these two values will be much too large. To deal with this issue, we compare the two perturbed polarisations to the polarisation in the unperturbed structure, and identify if all three lie roughly on a straight line. If one does not, it is shifted by the polarisation quantum to bring it onto the same branch of the polarisation lattice as the other two, before the finite differencing procedure takes place.

A few minor points should also be noted about the implementation. Firstly, the use of symmetry to

Pseudopotential	NCP			USP
MFC method	DFPT	FD	FD	FD
BEC method	DFPT	DFPT	FD	FD
Frequency (cm^{-1})	3941	3946	3948	4022
Intensity (D^2 Å^{-2} amu^{-1})	2.2347	2.2347	2.2346	2.1279

Table 2: Frequency and intensity of the bond stretching mode of the HF molecule, as calculated by a variety of different combinations of methods. The matrix of force constants (MFC) and the Born effective charges (BECs) can be computed by either DFPT or by finite differencing (FD). The frequency is then dependent on the MFC, and the intensity on the MFC and BECs.

Pseudopotential	NCP			USP
MFC method	DFPT	FD	FD	FD
BEC method	DFPT	DFPT	FD	FD
Frequency (cm^{-1})	1045	1045	1045	1041
Intensity (D^2 Å^{-2} amu^{-1})	42.4297	42.4275	42.4265	41.2604

Table 3: Frequency and intensity of the mode with the highest computed IR intensity in quartz, as calculated by a variety of different combinations of methods. The matrix of force constants (MFC) and the Born effective charges (BECs) can be computed by either DFPT or by finite differencing (FD). The frequency is then dependent on the MFC, and the intensity on the MFC and BECs.

reduce the number of polarisation calculations is turned off within an IR calculation, to avoid interfering with the procedure for confirming everything is on the same branch of the polarisation lattice. Secondly, the Born effective charges are only computed for the unit cell – this means that the computation of IR spectra by finite differencing is not compatible with the largely deprecated `SUPERCELL` method for computing phonons (or, at least, the supercell in question must just be the unit cell to obtain an IR spectrum). Thirdly, the requirement for a regular grid of k-points for the polarisation calculations required a small modification to the way k-point grids are generated for each phonon perturbation, as the previous algorithm generated a non-regular grid in the case of centred lattices.

With these changes, we were able to fulfil WP1b. The code was tested on the isolated HF molecule and quartz. Quartz was chosen as it is often a challenging material for phonon calculations, demonstrating the robustness of our implementation.

For the HF calculations, a cubic box was used, with all other parameters identical to the previous calculations performed on HF. The parameters used for the quartz calculations were as follows: the cut-off energy was 844 eV; the XC functional was PBE[8]; a $5 \times 5 \times 4$ k-point grid was used; the lattice parameters of the hexagonal unit cell were fixed at $a = 4.9137$ Å, $c = 5.4047$ Å. Both the HF and quartz were geometry optimised using the appropriate pseudopotentials, with the lattice constants fixed. Again, all pseudopotentials used were `CASTEP`'s own on-the-fly-generated (OTFG) pseudopotentials.

Table 2 shows the numerical results for the frequency and IR intensity of the only non-trivial mode in HF, calculated using several different combinations of methods. The matrix of force constants can be computed using either DFPT or finite differencing, as can the Born effective charges, with the method chosen for each independent of the choice for the other. The results in Table 2 show that, when using NCPs, the results are effectively identical for all combinations of methods, demonstrating that they are all consistent and giving confidence in our implementation. The values obtained when using USPs, which must be obtained using finite differencing, are slightly different, as the electronic ground state will be slightly different, but the values still match very well with the NCP values.

Table 3 shows a similar set of results for quartz, but for the mode with the maximum IR intensity. Again, we can see exceptional consistency across all combinations of methods and pseudopotentials. To show this is not just true for one mode, Fig. 2 shows the full approximate IR spectra for all the combinations of methods and pseudopotentials. We can see that all methods are extremely consistent with one another, reinforcing the robustness of our implementation.

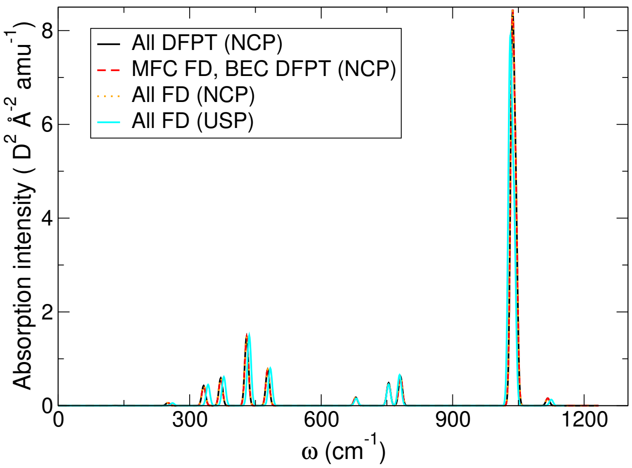

Figure 2: Computed IR spectrum of quartz, as calculated by a variety of different combinations of methods. The matrix of force constants (MFC) and the Born effective charges (BECs) can be computed by either DFPT or by finite differencing (FD). The frequencies of the peaks are then dependent on the MFC, and the intensities of the peaks on the MFC and BECs. The solid black line represents both MFC and BECs being computed with DFPT, the dashed red line the MFC being computed with FD and the BECs with DFPT, and the dotted orange line both MFC and BECs being computed with FD; all of the above were computed with NCPs. The solid cyan line represents both MFC and BECs being computed with FD and USPs.

2.3 Parallelisation

As noted above, there are three main types of parallelism within `CASTEP` – G-vector, k-point, and band parallelism. (Task farm parallelism, which is only relevant for phonon/IR calculations, will also be discussed below, but initially we deal with the more generally applicable parallelisms.) By making electric polarisation calculations take advantage of these, they can be made faster and more efficient, particularly on high performance computing facilities such as ARCHER2. G-vector parallelism was already implemented in the BP approach to computing polarisation. G-vectors do not explicitly arise in any of the equations defining the various contributions to the polarisation above, merely acting as a basis set over which other properties are expressed as a linear combination. This means that the pre-existing parallelisation over G-vectors works very similarly to that used elsewhere in the code.

Given a set of G-vectors, bands, and k-points, as well as the number of MPI processes requested, `CASTEP` automatically determines an optimal parallelisation strategy, deciding on the relative amounts of G-vector, k-point, and band parallelism. Each MPI process is assigned to a G-vector, k-point, and band 'group', with the number and size of these groups depending on the parallelisation strategy employed. The optimal parallelisation strategy for the ground state energy calculation is chosen at the beginning; however, a reparallelisation can take place if the structure or symmetry of the system changes, for example when atomic positions are perturbed as part of a phonon calculation.

The initial intention of this eCSE was to reparallelise the calculation before computing the electric polarisation, to ensure that the optimal parallelisation strategy was used for the polarisation calculation. Such a reparallelisation was performed in the existing BP polarisation implementation, as a one-off post-processing calculation; in fact, this was necessary, given that the BP approach was only parallelised over G-vectors, so the calculation needed to be reparallelised over G-vectors only. With the addition of band and k-point parallelism to the BP approach, such a reparallelisation would no longer be strictly necessary, but would have opened up the possibility of optimising the parallelisation for the polarisation calculation.

However, a reparallelisation prior to each polarisation calculation within an IR calculation proved not to be feasible due to the structure of the code. `CASTEP` already automatically reparallelises at each perturbation within a finite differencing calculation – to add another reparallelisation nested within this one would have required a complete overhaul of the code structure. Instead, we took the approach of maintaining the current parallelisation strategy – i.e., the relative amounts of G-vector, k-point, and band parallelisation –

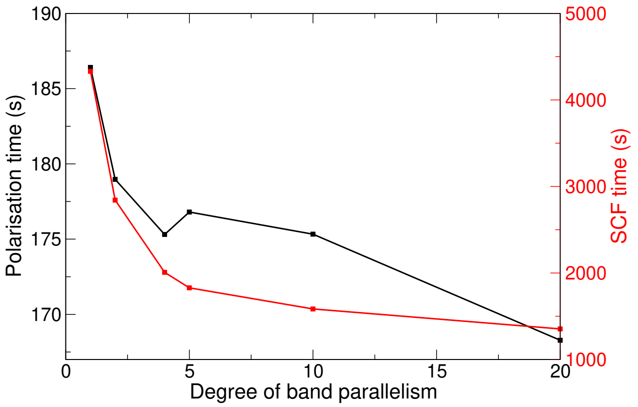

Figure 3: Time taken for two parts of a calculation of the polarisation of $BaTiO_3$ – the SCF electronic ground state calculation (red), and the calculation of the polarisation itself (black) – against the degree of band parallelism. 3-way G-vector parallelism was maintained throughout, and no band parallelism was used. A single node of ARCHER2 was used for all calculations.

but potentially changing how data was assigned to groups. This strategy is likely to be non-optimal for the polarisation calculation, but is the best possible within the code structure as it stands. This also means that the implementation of band and k-point parallelisation in the polarisation calculation, as performed in this work, is necessary for IR calculations to function properly – without being able to reparallelise for the polarisation calculations, the existing implementation's requirement for G-vector parallelism only would enforce G-vector parallelism throughout the entire IR calculation, rendering it extremely computationally inefficient. This, even setting aside the potential efficiency increases, significantly motivates the work performed in this eCSE.

2.3.1 Band parallelism

As the various contributions to the polarisation only ever involve straightforward summations over bands, implementing band parallelisation could largely follow existing code, with one caveat – the required summations are over *occupied* bands, not all bands. With small modifications to take this into account, the implementation of band parallelisation was able to follow band parallelisation elsewhere in the code.

By implementing this, we fulfilled WP2b. To demonstrate the effect of this new parallelisation, we tested it by computing the polarisation of $BaTiO_3$ using USPs and with varying levels of parallelism, and comparing the time taken for the SCF calculation finding the ground state and the polarisation calculation. The same $BaTiO_3$ structure was used as above. Calculations were performed by 1-, 2-, 4-, 5-, 10-, and 20-way band parallelism, alongside 3-way G-vector parallelism. In `CASTEP`'s implementation, the bands must be split evenly between the nodes, meaning that the degree of band parallelism must be a factor of the total number of bands. 40 bands were included in this calculation, hence the previously quoted numbers all being factors of 40. These calculations were all performed on a single node of ARCHER2.

NB: the polarisation calculations run over only occupied bands, which means that the degree of band parallelism should ideally also be a factor of the number of occupied bands. If this is not the case, our implementation finds the lowest multiple of the degree of band parallelism that is larger than the number of occupied bands, so that all occupied bands are included, potentially along with unoccupied ones as well. The unoccupied bands are then simply discarded. In the $BaTiO_3$ calculations here, there are 20 occupied bands, meaning that this issue does not arise for any of the degrees of band parallelism chosen.

As can be seen in Fig. 3, band parallelism improves the timings of the polarisation calculation a little, but not significantly. The improvement in the timings of the SCF calculation is much more significant, although the polarisation still only takes 12% of the time of the SCF calculation in the worst case, demonstrating that the calculation of the polarisation has a limited effect on the overall run time of the calculation, as intended. The bottleneck here appears to be the computation of the Fourier transform of the augmentation charge $Q_{\alpha;ij}(\mathbf{b})$ (c.f. Eq. (5)). As $\mathbf{b}$ is the same along each string, this can often be reused, but care must be taken that a previously calculated value is not used incorrectly. Our implementation attempts to do this where possible, speeding up the calculation, but this is a candidate for further optimisation.

2.3.2 K-point parallelism

As can be seen above, the BP approach to computing the electric polarisation does not merely sum over k-points, but considers overlaps between functions expressed at different k-points. As k-points are assigned to k-point groups in a round robin style, this means that it is very likely that the two k-points involved in a given overlap will be assigned to different k-point groups, thus requiring communication between processes, slowing the calculation down. The existing k-point parallelism approach in the code was therefore clearly sub-optimal for polarisation calculations, unlike G-vector and band parallelism.

Importantly, in a BP calculation, we only need to compute the overlap of each k-point with its neighbours along a string (c.f. Eq. (3)). To minimise communication, therefore, k-point groups should only contain k-points from the same string. In the ideal case, each entire k-point string would be associated with a single k-point group, as this would effectively eliminate communication between k-point groups, only requiring a communication of the final results at the end. However, as we have chosen to maintain the existing parallelisation strategy for polarisation calculations, the number of k-point groups cannot be changed to align with the number of k-point strings. We therefore took the following approach:

1. Ascertain which k-points are in which string

2. If the number of strings is equal to the number of groups, each string is assigned to its own group, with all k-points in that string being assigned to that group. No communication is necessary between groups.

3. If there are more strings than groups, some groups are assigned more than one complete string – this potentially leads to load imbalance between different groups, but maintains the absence of communication between groups.

4. If there are more groups than strings, strings are split across two (or more) groups, keeping the number of splits to a minimum. The k-points are ordered before being split across the groups, so that one group holds the first n k-points in the string, the next group holds the next m k-points, and so on. This leads to each group having to send and receive one set of data – it sends the data of its first k-point to the group assigned the previous section of the string, and receives the data corresponding to the k-point immediately after its final k-point from the group assigned the next section of the string. This includes the groups containing the first and last sections of the string, as we also need the overlap between the first and last k-points, as noted above.

It should be noted that which k-points are in which strings depends on the component of the polarisation being computed (c.f. Eq. (3)). This reorganisation of the k-point data must therefore take place as a precursor to each component of the polarisation being calculated, before reverting the data back to its original arrangement so that the calculation can continue (if necessary).

By implementing this, we were able to fulfil WP2a. We again tested this by computing the polarisation of $BaTiO_3$ with USPs. In this case, we performed calculations with 10-way G-vector parallelism, alongside k-point parallelism ranging from 1-way to 12-way. These calculations were all performed on a single node of ARCHER2.

The results can be seen in Fig. 4. We can clearly see that increasing k-point parallelism significantly reduces the time required to compute the polarisation, to much greater degree than band parallelism – 10-way k-point parallelism reduces the time by a factor of 6.6 compared to no k-point parallelism, whereas a similar comparison from the band parallelism results shows only an improvement of 1.1. This matches our expectation that the preferred parallelism for polarisation should be over k-points, and demonstrates the success of our parallelisation implementation. The improvement does plateau at large-scale k-point parallelism, similar to the SCF calculation, as serial parts of the calculation begin to dominate. Identifying the bottlenecks here (such as the computation of $Q_{\alpha;ij}(\mathbf{b})$ noted above) and optimising them would be a prime candidate for further development.

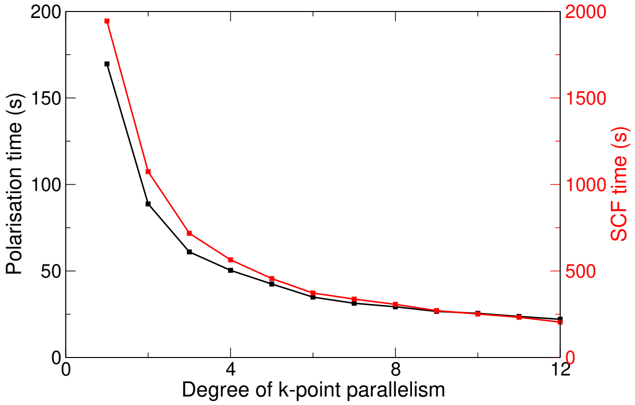

Figure 4: Time taken for two parts of a calculation of the polarisation of $BaTiO_3$ – the SCF electronic ground state calculation (red), and the calculation of the polarisation itself (black) – against the degree of k-point parallelism. 10-way G-vector parallelism was maintained throughout, and no band parallelism was used. A single node of ARCHER2 was used for all calculations.

	Degree of parallelism		
Provided cores	G-vector	k-point	Used cores
12	1	12	12
16	1	16	16
20	1	20	20
40	1	40	40
80	1	80	80
128	25	3	125
256	3	73	219

Table 4: Optimal parallelisation strategy for the electronic ground state calculation, as computed by CASTEP's existing algorithm for different numbers of cores provided for MPI parallelism. These are the parallel strategies used to obtain the data in Fig. 5.

2.3.3 CASTEP 'optimal' parallelism

To replicate what would be seen in an actual `CASTEP` calculation, we also performed calculations where the code was allowed to choose the optimal parallelisation strategy for the ground state SCF calculation (which may not use all the cores given to it), which was then maintained for the polarisation calculation. By default, `CASTEP` does not use band parallelism, so this focused on G-vector and k-point parallelism only. These calculations were again performed on $BaTiO_3$. The calculations used a variety of core counts, from 12 to 256, using 4 nodes of ARCHER2 with the allocated cores spread evenly between them. The resulting parallelisation strategies are shown in Table 4. We can see that most only use k-point parallelism, with G-vectors being used for high core counts.

The results can be seen in Fig. 5, and we can generally see that the time to perform a polarisation calculation continues to go down with increasing core count (with one exception). The large-scale k-point parallelism used for high core counts means that strings are being broken up to allow such k-point parallelism, as described in the previous section – the fact that the relative time spent in the SCF and polarisation calculation remains roughly constant at $3-4\%$ demonstrates that the k-point parallelism remains robust even in this regime, despite the increased communication required.

The exception is the calculation with 128 cores, where G-vector parallelism dominates. Here, the time to perform the polarisation calculation actually increases relative to 80 cores, even though the SCF calculation time continues to decrease. This demonstrates two things that have already been stated: 1. k-point parallelism is strongly preferred for the polarisation calculation, and 2. the parallelisation strategy that is optimal for the SCF calculation may not be optimal for the polarisation calculation. However, if we compare the 80 and 256 core cases, we see that the time taken for the polarisation calculation actually decreases, even though

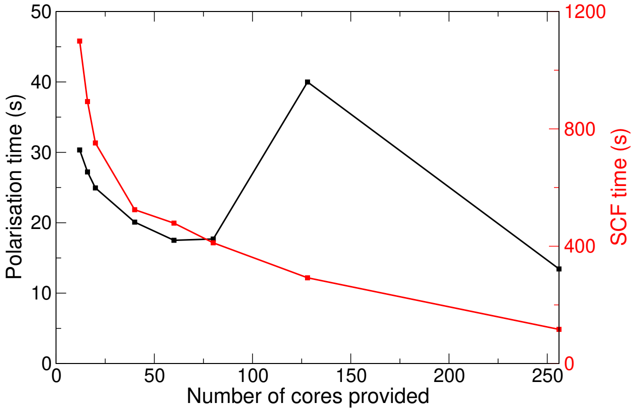

Figure 5: Time taken for two parts of a calculation of the polarisation of $BaTiO_3$ – the SCF electronic ground state calculation (red), and the calculation of the polarisation itself (black) – against the number of cores provided for MPI parallelism, using the parallelisation strategy decided on by CASTEP. The breakdown of the parallelisation strategy for each point on the plot is given in Table 4. 4 nodes of ARCHER2 were used for all calculations, with the appropriate number of cores shared evenly between them.

the degree of k-point parallelism slightly decreases. This demonstrates that G-vector parallelism can provide an improvement on pure k-point parallelism, especially for large core counts, where k-point strings must be broken up. We can also see that the polarisation still takes less than 15% of the SCF calculation time, even in the worst case, further reinforcing the fact that the polarisation has a limited effect on the total calculation time.

2.3.4 Task farm parallelism

Finally, we have task farm parallelism. This parallelism is only available for calculations where some parts of the calculation do not depend on the results of previous parts, allowing them to be effectively run as independent calculations in parallel. Phonon calculations, particularly those using finite differencing, are particularly suited to this, as the calculations of the perturbed structures are all independent of one another, so can be run independently. To make this form of parallelism compatible with finite differencing IR calculations, the main task was therefore to ensure the Born effective charges computed within each independent calculation are fed back into the main calculation appropriately, to ultimately compute the IR absorption strengths.

By implementing this, we were able to fulfil WP3a. We tested this using the quartz system used previously. We performed calculations with 1, 2, 3, and 6 farms (these numbers were chosen to partition the 6 perturbations required evenly between the farms), with 64 cores assigned to each farm. These calculations were performed on 4 nodes of ARCHER2, with the allocated cores split evenly between the nodes.

Fig. 6 shows how the total calculation time (minus the time taken for the equilibrium structure ground state calculation, which needs to be done before farm parallelisation can take place) depends on the number of farms. It can be seen that increasing the number of farms significantly decreases the calculation time. For 1 to 3 farms, this decrease is almost linear, as would be expected, although for 6 farms the decrease in computational time is less significant, as serial parts of the calculation have an effect. This linear behaviour demonstrates that we can use farm parallelism to reduce the walltime of large-scale IR calculations by utilising additional computational resources, with an almost linear dependence.

2.4 Additional successes

In collaboration with other developers of `CASTEP`, we were also able to go beyond the work packages outlined above. This focused on calculating the piezoelectric tensor, which defines the strength of the piezoelectric effect in a material. Importantly, derivatives of polarisation are also relevant to the computation of the

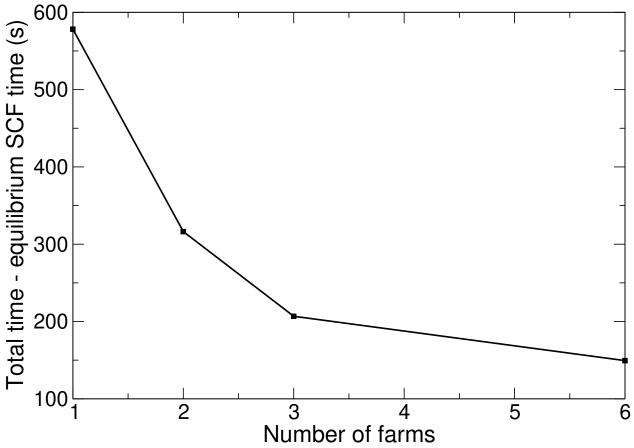

Figure 6: Total time taken for a calculation of the IR spectrum of quartz (minus the time for the initial SCF electronic ground state calculation) against the number of farms used for task farm parallelism. Each farm was given 64 cores. 4 nodes of ARCHER2 were used for all calculations, with the appropriate number of cores shared evenly between them.

	Proper piezoelectric tensor (C m^{-2})					
	xx	yy	zz	zy	zx	yx
X	-0.1208	0.1208	0.0	-0.0392	0.0	0.0
Y	0.0	0.0	0.0	0.0	0.0392	0.1208
Z	0.0	0.0	0.0	0.0	0.0	0.0

Table 5: Components of the piezoelectric tensor of quartz, as computed using the finite differencing method introduced in this work.

piezoelectric tensor, which can be written as (in Voigt notation)[10]:

$$e_{ij} = \frac{\partial P_i}{\partial \epsilon_j} \tag{8}$$

where P is the electric polarisation, and ϵ is the strain. We can therefore use finite differencing to evaluate this derivative, by applying small strains to the system and calculating the polarisation as before. (To ensure that we obtain the proper rather than improper piezoelectric tensor, the polarisation must be rescaled to the unperturbed unit cell volume[10], but this is a trivial operation to perform.) This procedure was included within the existing implementation for calculating elastic constants, in a similar way to the computation of the Born effective charges being included within a standard phonon calculation.

We demonstrated this new functionality on quartz, a well-known piezoelectric material. We used USPs only for these calculations. In addition to the geometry optimisation undertaken for the IR calculations (as described above), we additionally optimised the lattice parameters. All other calculation parameters were the same. The components of the resulting proper piezoelectric tensor are shown in Table 5.

3 Conclusions

To summarise, we successfully implemented new contributions to the computation of electric polarisation in `CASTEP` required when using USPs, making it possible to calculate electric polarisation for the vast majority of electronic structure methods within `CASTEP`. We also implemented new parallelisation strategies for the calculation of polarisation, using both k-point and band parallelism. Using this newly improved approach for calculating polarisation, we successfully implemented a finite differencing approach for computing IR spectra, making it possible to simulate IR spectra with virtually all electronic structure methods in `CASTEP`.

This work has been merged into the main source code, and is available in the most recent release of `CASTEP`, v. 25, which was released in autumn 2024. The new functionality has been fully documented, and tests covering finite differencing IR calculations and polarisation calculations with USPs have been included within the test set. A full paper making use of this functionality and comparing between different electronic structure methods is in progress.

Looking forward, the next steps would be to apply this finite differencing approach to other phonon-related calculations, such as Raman spectroscopy or phonon-loss EELS. Simulating the latter of these is very closely related to the computation of IR spectra, and indeed relies mainly on the computation of Born effective charges. Simulating Raman spectroscopy, however, is somewhat more difficult, as it requires third derivatives of the energy – twice with respect to electric field, and once with respect to atomic positions. This would therefore likely involve performing a two-dimensional finite differencing approach with respect to a finite electric field and atomic positions.

4 Acknowledgements

This work was funded under the embedded CSE programme of the ARCHER2 UK National Supercomputing Service (http://www.archer2.ac.uk).

References

[1] Stewart J. Clark, Matthew D. Segall, Chris J. Pickard, Phil J. Hasnip, Matt I. J. Probert, Keith Refson, and Mike C. Payne. First principles methods using CASTEP. *Z. Kristallogr.*, 220:567, 2005.

[2] W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.*, 140:A1133, 1965.

[3] Nicola A. Spaldin. A beginner's guide to the modern theory of polarization. *J. Solid State Chem.*, 195:2, 2012.

[4] David Vanderbilt and R. D. King-Smith. Electronic polarization in the ultrasoft pseudopotential formalism, 1998. arXiv:cond-mat/9801177, version: 1.

[5] P. Umari, Xavier Gonze, and Alfredo Pasquarello. Density-functional perturbational theory for dielectric tensors in the ultrasoft pseudopotential scheme. *Phys. Rev. B*, 69:235102, 2004.

[6] A. Ferretti, A. Calzolari, B. Bonferroni, and R. Di Felice. Maximally localized Wannier functions constructed from projector-augmented waves or ultrasoft pseudopotentials. *J. Phys.: Condens. Matter*, 19:036215, 2007.

[7] S. J. Clark, G. J. Ackland, and J. Crain. Ab initio molecular polarisabilities of liquid crystals: Application to DOBAMBC and 5CB. *Europhys. Lett.*, 44:578, 1998.

[8] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865, 1996.

[9] Ivan Carnimeo, Fabio Affinito, Stefano Baroni, Oscar Baseggio, Laura Bellentani, Riccardo Bertossa, Pietro Davide Delugas, Fabrizio Ferrari Ruffino, Sergio Orlandini, Filippo Spiga, and Paolo Giannozzi. Quantum ESPRESSO: One Further Step toward the Exascale. *J. Chem. Theory Comput.*, 19:6992, 2023.

[10] Xifan Wu, David Vanderbilt, and D. R. Hamann. Systematic treatment of displacements, strains, and electric fields in density-functional perturbation theory. *Phys. Rev. B*, 72:035105, 2005.