Development of a code for optimally tuned range-separated density functional calculations

Master's Thesis

UCL Chemistry Department

Supervisor: Dr. Martijn Zwijnenburg

Email: m.zwijnenburg@ucl.ac.uk

FACULTY OF MATHEMATICAL AND PHYSICAL SCIENCES

Rafiq Hilali

Email: rafiq.hilali.13@ucl.ac.uk

28th March 2017

Abstract

The aim of this project was to develop a code that would efficiently and accurately optimise the range-separation parameter, γ, in range-separated hybrid functionals, in a view to providing better estimates of excitation gaps. To do this a reference set of simple, photo-actively important molecules was selected, for which the optimally-tuned range separated hybrid functional (OT-RSH) approach had already been applied and values of optimal γ available.

A code was developed that took a minimum number of data points and then fit a polynomial to them in order to calculate the minimum of a cost function $J^2(\gamma)$. This proved to be a successful method of determining an optimised value of γ.

The approach was then applied to solvated systems using the COSMO model but appeared to be inaccurate. It was also applied to a selection of small titanium dioxide clusters and provided results that agree well with $qsGW$ calculations, although more investigation is needed to determine the accuracy of OT-RSH when applied to these systems.

Acknowledgements

I would like to thank Martijn for the help and guidance he has given me in supervising my project. I would also like to acknowledge Jörg Saßmannshausen as well as the UCL Research Computing Services team for their upkeep of the research clusters, without which this project would not be possible.

Declaration of Contributions

Martijn Zwijnenburg provided calculations for the titanium dioxide clusters as accredited in the main body of the report.

Table of Contents

Abstract ii

Acknowledgements iii

Declaration of Contributions iii

Table of Contents iv

1 Introduction 1

1.1 Excitation gaps 1

1.1.1 What are fundamental and optical gaps? 1

1.1.2 Computation of fundamental and optical gaps 2

1.2 Fundamental gaps from density functional theory 2

1.2.1 Kohn-Sham approach 2

1.2.2 Conventional hybrid functionals 3

1.2.3 Range-separated hybrid functionals 4

1.3 Reference molecules 5

1.4 Solvation models 6

1.4.1 The conductor-like polarisable continuum solvation model . 6

1.4.2 Previous work involving c-PCM and OT-RSH 7

1.5 Titanium dioxide clusters 7

1.5.1 Selection of molecules 7

1.5.2 Applications and importance 7

1.6 Optical gaps from time-dependent density functional theory 9

2 Methodology **11**

2.1 Software 11

2.2 Programming Language 11

2.3 Reference Molecules 11

2.4 Solvated Species 12

2.5 Titania Clusters 12

3 Results and Discussion 13
3.1 Understanding the $J^2(\gamma)$ cost function 13
3.1.1 The $J^2(\gamma)$ cost function 13
3.1.2 Grid method 13
3.2 Development of code 14
3.2.1 The polynomial approach 14
3.2.2 Trials of fitting to a polynomial 15
3.2.3 Algorithm for locating the minimum 15
3.2.4 Structure of Code 17
3.3 Reference molecules 18
3.3.1 Optimisation of γ 18
3.3.2 Excitation gaps 19
3.4 Solvation models 21
3.4.1 Optimisation of γ 21
3.4.2 Fundamental Gaps 23
3.5 Titanium dioxide clusters 23
3.5.1 Optimisation of γ 23
3.5.2 Fundamental Gaps 25
3.5.3 Optical gaps 29
3.6 Effect of optimally tuning γ 30
3.6.1 Reference molecules 30
3.6.2 Solvation Model 31
3.6.3 Titanium dioxide clusters 31

4 Concluding Remarks 35

5 References 36

Appendices I

Appendix A Functions definition of γ optimistation code II

Appendix B Part 1 of γ optimistation code VIII

Appendix C Part 2 of γ optimistation code IX

Appendix D Part 3 of γ optimistation code XI

Appendix E Analysis of γ optimistation code XIII

Appendix F Geometries XVI

Appendix G OT-BNL Tables XX

1. *Introduction*

1.1 Excitation gaps

1.1.1 What are fundamental and optical gaps?

There are several properties of molecules that are of importance to fields such as photovoltaics and photocatalysis. Excitation gaps are of particular interest, and have proved to be somewhat difficult to model in an accurate and computationally efficient fashion. The fundamental gap, E_g, is defined as the difference between the ionisation potential, I, and the first electron affinity, A. In essence this can be thought to be the HOMO-LUMO gap. The optical gap, or alternatively, the first singlet excitation, is defined to be the neutral excitation between the ground state of the molecule and the lowest dipole-allowed excited state.

These two scenarios can also be thought of in terms of quasi-particles, whereby the addition or ejection of an electron can be represented by the excitation of a quasi-electron or quasi-hole respectively. These quasi-particles are hypothetical particles that are 'dressed' with the subsequent relaxation effects of other electrons in the molecule. The ionisation potential, I, can be thought of as the lowest energy quasi-hole excitation, and the first electron affinity, A, as the lowest energy quasi-electron excitation. Thus the fundamental gap is the difference in energy between the ground-state and a state containing the first quasi-electron excitation, whilst the optical gap is the difference between the ground state and a state containing both the first quasi-electron excitation and the first quasi-hole excitation. In the hypothetical scenario where there is no interaction between the quasi-hole and quasi-electron, then the optical gap will be equal to the fundamental gap. In reality the quasi-hole and quasi-electron do interact, and the energy difference between the fundamental and optical gaps is due to this interaction. This quantity of interaction is known as the exciton binding energy.

1.1.2 Computation of fundamental and optical gaps

Both *ab inito*[1,2,3] and Monte Carlo methods[4] have been successfully implemented to calculate optical and fundamental gaps. Whilst these have been shown to provide accurate approximations, they are computationally expensive, and are therefore unsuitable for the screening of large numbers of systems. An approach that has been utilised in the calculation of excitation gaps is many-body perbutation theory. The use of GW calculations, where the G represents the Green function and W the screened coulomb potential has been shown to be reliable in the calculation of fundamental gaps, although assumptions are brought into the process which can affect its accuracy.[5,6,7,8] When this approach is coupled with the Bathe-Salpeter equation (BSE) it can also provide a powerful method for the calculation of optical gaps. This method too has been shown to be reliable in practice but again, is computationally expensive.[6]

1.2 Fundamental gaps from density functional theory

1.2.1 Kohn-Sham approach

Density functional theory (DFT) is a relatively computationally inexpensive means of modelling quantum mechanical many-body systems. The fundamentals for density functional theory were laid down by the Hohenberg-Kohn theorem.[9] This states that for a many-electron system subjected to external potential, the ground-state density will uniquely determine this potential and the energy.

The Hohenberg-Kohn theorem is utilised in the Kohn-Sham (KS) approach to DFT,[10] in which the question of computing quantum mechanical equations is simplified by stating that the ground state of a physically interacting electron system, can be simplified by mapping it to a non-interacting electron system which is subject to a common local external potential. This can be represented by

$$\Big(\frac{-\nabla^2}{2} + v_{ext}(r) + v_H([n];r) + v_{xc}([n];r)\Big)\varphi_i(r) = \varepsilon_i\varphi_i(r) \tag{1.1}$$

where $v_{ext}(r)$ is the nuclear-electron potential, $v_H([n];r)$ the Hartree potential and $v_{xc}([n];r)$ is the exchange-correlation potential. Both classical and non-classical interactions are described by the functional, by the Hartree and exchange-correlation terms respectively, which describe the electron-electron interatctions.[11]

Unfortunately the KS approach is inherently flawed in the predication of fundamental gaps. The KS approach is in theory exact, although it is required to make approximations in reality, primarily in the determination of the exchange correlation potential.[11]

In order to draw physical meaning from the quantities calculated using KS, Koopman's theory can be used. Whilst Koopman's theorem applies to Hartree-Fock theory, an analogue exists for KS. Koopman's theory states

$$\varepsilon_H = -I \tag{1.2}$$

that is that the HOMO eigenvalue, ε_H, is equal and opposite to the ionisation energy, I.[12,13,14] This allows one to extract *physical* meaning from the fictitious KS system.

In an ideal world there would be an analogue of this relationship for the LUMO eigenvalue and the first electron affinity, unfortunately this is not the case. The nature of the KS approach leads to a derivative discontinuity in the exchange correlation, Δ_{xc}.[15] This causes the eigenvalue for the LUMO to be[16]

$$\varepsilon_L = -A - \Delta_{xc} \tag{1.3}$$

studies have shown that the size of Δ_{xc} can be as large as several eV. When this result is applied to the calculation of fundamental gaps from KS theory, we obtain[12]

$$E_g \equiv I - A = \varepsilon_L - \varepsilon_H + \Delta_{xc} \tag{1.4}$$

The exact degree of Δ_{xc} cannot be determined. As this error cannot be accounted for, the determination of fundamental gaps from KS theory is inherently flawed.

1.2.2 Conventional hybrid functionals

An improvement to the KS approach was that of conventional hybrid functionals, whereby the exchange potential is described as a fraction of a Fock non-local exchange operator and a local exchange potential.[17] this is described mathematically by

$$\Big(\frac{-\nabla^2}{2}+v_{ext}(r)+v_H([n];r)+a\hat{V}_F(r,\varphi_{1\ldots n})+(1-a)v_x^{sl}([n];r)+v_c^{sl}([n];r)\Big)\varphi_i(r) = \varepsilon_i\varphi_i(r) \tag{1.5}$$

Where the fraction of the Fock exchange potential, denoted by $a\hat{V}_F$, and of the semi-local exchange potential, $(1-a)v_x^{sl}([n];r)$, is determined by a, with the term $v_c^{sl}([n];r)$ describing the semi-local correlation potential. The size of a can be determined via a variety of techniques, from theoretical considerations to fitting to a reference set.[11]

This is type of Generalised Kohn-Sham (GKS) approach and theoretically should lead to a better estimate of the fundamental gap. This is because the value of Δ_{xc} should be smaller, as the effect of the discontinuity can also be shared by the non-local Fock term.[18] Unfortunately this fractional approach still does not successfully describe the fundamental gap found in complex molecules. In the case of full Fock exchange, The Fock operator cancels the fictitious electrostatic interaction of an electron with itself that is an artefact of this methodology. In the case of fractional Fock-exchange this self-interaction error is only partially corrected, leading to errors in the prediction of the fundamental gap.[19]

1.2.3 Range-separated hybrid functionals

This brings us to another class of hybrid functionals, know as range-separated hybrid functionals. In these functionals the repulsive coulomb potential is split in to a long-range (LR) and short-range (SR) term. Both LR and SR components are treated as usual for the Hartree component, but for the exchange potential the LR componenet is represented by an exact Fock potential and the SR by a semi-local potential.[20,21] This results in a full LR Fock term, allowing this type of functional to overcome the disadvantages of conventional hybrid functionals by having no self-interaction error. The functional is represented as follows

$$\Big(\frac{-\nabla^2}{2}+v_{ext}(r)+v_H([n];r)+\hat{V}_F^{lr,\gamma}(r,\varphi_{1\ldots n})+v_x^{sr,\gamma}([n];r)+v_c^{sl}([n];r)\Big)\varphi_i(r)=\varepsilon_i\varphi_i(r) \tag{1.6}$$

where $\hat{V}_F^{lr,\gamma}$ is the LR Fock-like operator and $v_x^{sr,\gamma}([n];r)$ describes the SR semi-local exchange potential. In this functional the variable γ is known as the range-separation parameter. Physically, $1/\gamma$ is the inter-electronic spearation at which the exchange correlation potential is dominated by the SR component, and above it by the LR component.[11,20,21,22,23]

In previous work the selection of γ has mainly concerned semi-empirical techniques,[24,25,26] in which the value of γ is trained to best fit with a set of reference data. This approach assumes that one can apply a single value of γ universally, to any system and still achieve meaningful results. Unfortunately this is not always the case, as several studies have shown that an optimal γ can vary widely from system to system.[20,21]

This leads one to question if it is possible to determine an optimal value of γ from the properties of the molecules themselves. Livshits and Baer have developed a critera for the optimisation of γ.[20,22] This is done by actively enforcing Koopman's theorem, such that

$$-\varepsilon^{\gamma}_{H(N)} = I^{\gamma}(N) \equiv E_{gs}(N-1;\gamma) - E_{gs}(N;\gamma) \tag{1.7}$$

where the eigenvalue of the HOMO of the N-electron system, $\varepsilon^{\gamma}_{H(N)}$, is made to be equal to the difference in ground-state energy of the N-electron system and the N-1-electron system, i.e. equal and opposite of the ionisation energy of the N-electron system, $I^{\gamma}(N)$.

In order to predict fundamental gaps, it is desirable to have some analogous condition for optimising γ for the LUMO. As previously mentioned, a version of Koopman's theory for the LUMO energy does no exist. Fortunately this problem can be circumvented by instead considering the ionisation potential of the N+1-electron system. This is equivalent to the electron affinity of the N electron system, barring relaxation effects. This gives us with a second criterion, whereby[11]

$$-\varepsilon^{\gamma}_{H(N+1)} = I^{\gamma}(N+1) \equiv E_{gs}(N;\gamma) - E_{gs}(N+1;\gamma) \tag{1.8}$$

If Δ_{xc} is small, the optimal γ for both equations should be similar. Combining equations 1.7 and 1.8 gives us a cost function which one can minimse to optimise γ, given by[22]

$$J^2(\gamma) = (\varepsilon^{\gamma}_{H(N)} + I^{\gamma}(N))^2 + (\varepsilon^{\gamma}_{H(N+1)} + I^{\gamma}(N+1))^2 \tag{1.9}$$

By minimising the target function, $J^2(\gamma)$, one has a criterion for optimising γ by simultaneously enforcing Koopman's theorem for the N-electron system, $\varepsilon_H = -I$, and the N+1-electron system, such that, $\varepsilon_L = A$, allowing for a more accurate prediction of the fundamental gap. This approach is known as optimally-tuned range separated hybrid functionals (OT-RSH), and it is the optimisation of the variable γ, that the aim of this project is to create a efficient and accurate algorithm for.

1.3 Reference molecules

This technique has been shown to successfully predict ionisation energies and fundamental gaps for a range of photovoltaically relevant molecules as well as single atom calculations.[22] In order to trial a method for the optimisation of γ it is important to have a set of reference molecules from which to be able to check the progress of the methodology.

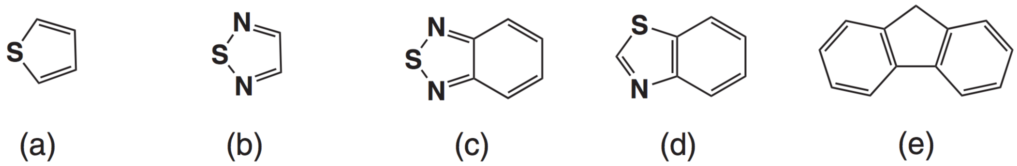


Figure 1.1: The reference molecules used to test the optimisation of γ algorithm: a) thiophene; b) thiadiazole; c) benzothiadiazole; d) benzothiazole; e) fluorene.

Refaely-Abramson[23] et al. have carried out OT-RSH calculations on a number of molecules, and provided data for the optimal γ, ionisation energies and fundamental gaps. GW calculations have also been carried out on this set of molecules by Blase et al.,[8], and experimental determinations of ionisation potentials are also available (variety of sources, adapted from Refaely-Abramson[23]). This wealth of data makes this an ideal set to test any γ optimisation algorithm on. An additional criterion to consider is that the molecules be sufficiently small that computation time is not excessive. As calculations on these molecules have already been undertaken with OT-RSH, any calculations will be unlikely to glean any novel findings, rather they should be used primarily to ensure the optimisation algorithm works. Thus, the molecules should be small enough that computation time does not take up a significant portion of the time spent to develop the code, such that it hinders the development of the code itself. With these considerations, the molecules given in Fig. 1.1 were chosen as a reference set.

1.4 Solvation models

1.4.1 The conductor-like polarisable continuum solvation model

The determination of excitation gaps in solvated systems is an area of considerable interest. The majority of DFT calculations are carried out in vacuum. Whilst this can provide comparable results, in practice, it is of more value to understand the properties of these systems under solvation, as this is the conditions they would most likely be under in the cases of photo-catalysis and, in some cases of photo-voltaics. To understand the behaviour of the systems of interest a simplified solvation model will be used. First put forward by Klamt et al.[27] and later refined by York et al.[28], the model in question is COSMO (COnductor-like Screening MOdel), a type of conductor-like polarisable continuum solvation model (c-PCM). It works by creating a smooth cavity that is defined by the atomic radii of the atoms in the molecule that it contains. The solvent is not treated as individual molecules but as a continuous medium with dielectric constant, ε_r. The solute and the solvent are partitioned and a smooth potential energy surface is created via the dielectric screening effect. In this process the effect of a solvent is simulated, without the computational expense of having to optimise and calculate several coordination shells of solvent molecules.

1.4.2 Previous work involving c-PCM and OT-RSH

Several studies have investigated the use of OT-RSH with the c-PCM model. de Queiroz et al.[29] have investigated optimising the range-separation parameter (which they call ω) whilst applying c-PCM to oligo-thiophenes. These investigations have shown that the approach leads to artificially low γ values and ionisation potentials. Zheng et al.[30] have also investigated using OT-RSH with c-PCM, but have taken the approach of using a value of γ that was optimised in the vacuum and then applying c-PCM. This approach leads to a calculation whereby Koopman's theorem is not enforced for the solvated molecule, so already stands on theoretically unstable ground. As far as to the researchers' knowledge, no direct comparison between these two techniques has been drawn by operating on the same set of molecules, and these concepts will be investigated further.

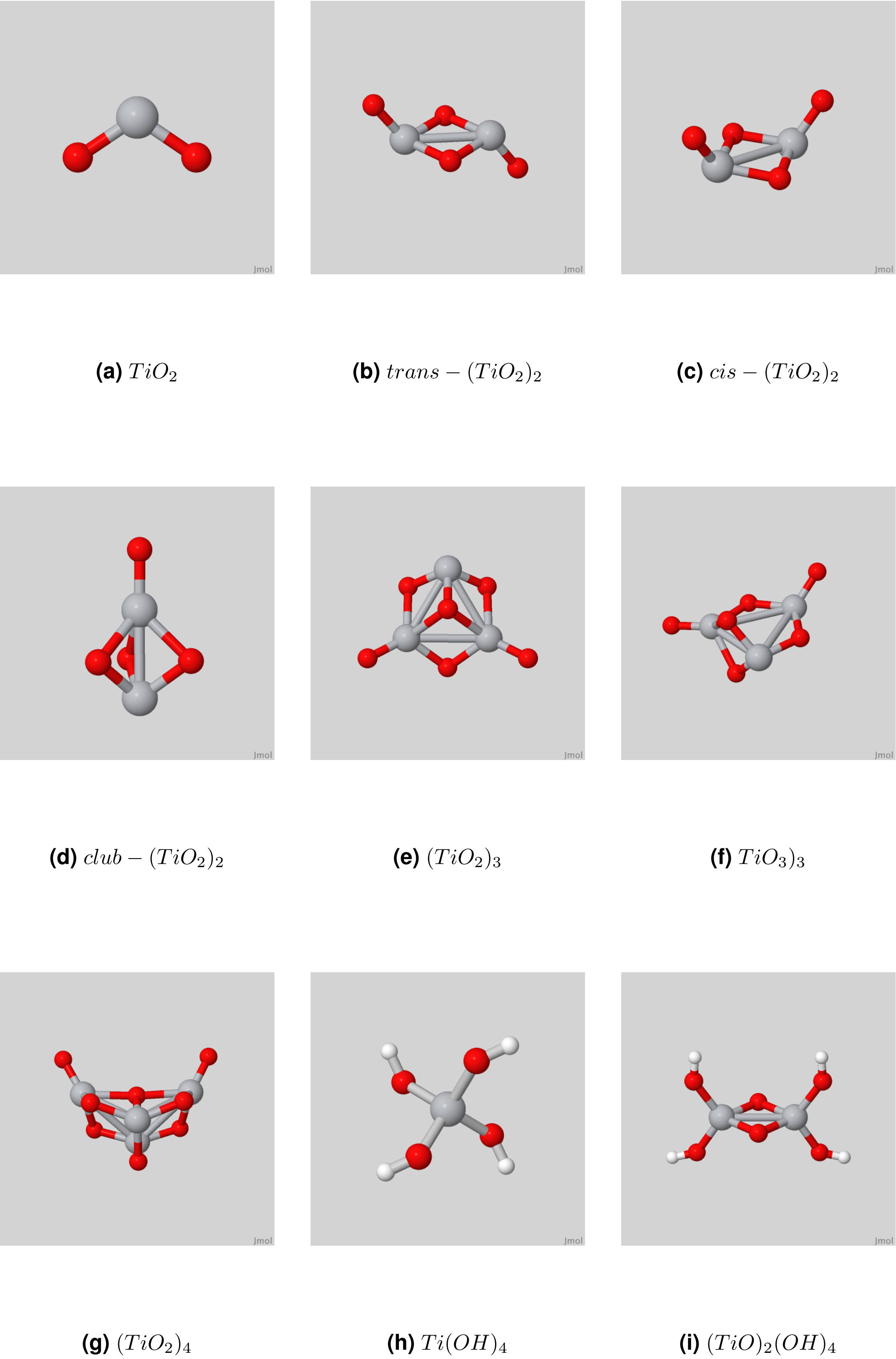
1.5 Titanium dioxide clusters

1.5.1 Selection of molecules

A selection of small titanium dioxide clusters were chosen to apply the OT-RSH approach to. These clusters were selected as a range of calculations have already been carried out on them, including B3LYP and G_0W_0 by Zwijnenburg[31]. Several studies into the stability,[32] optical properties, and behaviour of excited states[33,34,35] have been conducted on these clusters. As far as to the researchers knowledge, the OT-RSH approach has not been applied to these titanium dioxide clusters prior to these investigations. The properties of nine clusters were evaluated using OT-RSH. Of these clusters five are of the form $(TiO_2)_n$ where n ranges from 1 to 4, two are hydrated clusters, and the final cluster is of formula $(TiO_3)_3$. The shape and structure of the titanium dioxide clusters is given in Fig. 1.2.

1.5.2 Applications and importance

Titanium dioxide is an important material in the photo-catalytic and photo-voltaic fields. In its various forms it has been used to a wide range of photo-catalysis applications,[36] such as to degrade contaminants,[37] and also in dye-sensitised solar cells, where alterations in nano-scale structure has been shown to result in



(a) TiO_2 **(b)** $trans-(TiO_2)_2$ **(c)** $cis-(TiO_2)_2$

(d) $club-(TiO_2)_2$ **(e)** $(TiO_2)_3$ **(f)** $TiO_3)_3$

(g) $(TiO_2)_4$ **(h)** $Ti(OH)_4$ **(i)** $(TiO)_2(OH)_4$

Figure 1.2: The titanium dioxide clusters from which fundamental and optical gaps were calculated using OT-RSH.

large increases in efficiency.[38,39] TiO_2 materials are an important research area in the photo-catalytic splitting of water in electro-chemical cells,[40] a process that if refined has the potential to end dependence on fossil fuels. Accurate predictions of fundamental and optical gaps are vital to this field, and reliable, computationally efficient methods in no doubt accelerate the research velocity in what is a fast growing field. From combating harmful pollutants, to generating electricity and fuels from solar energy, to even the application of self-cleaning glass,[41,42] TiO_2 based materials are likely to be at the forefront of chemistry for this century and beyond.

1.6 Optical gaps from time-dependent density functional theory

Whilst the bulk of this project will mainly on the calculation of fundamental gaps, optical gaps are also of great importance to photo-active applications. DFT cannot be used to calculate optical gaps, as it is a ground-state theory. Instead an extension of DFT is used, in the form of time-dependent density functional theory (TDDFT). TDDFT provides a means of calculating the excited states of molecules, and is represented by:[43]

$$\Big(\frac{-\nabla^2}{2} + v_{ext}(r,t) + v_H([n];r,t) + v_{xc}([n];r,t)\Big)\varphi_i(r,t) = \varepsilon_i\varphi_i(r,t) \tag{1.10}$$

As can immediately be seen, this equation shows stiking similarity to the KS approach (eq. 1.1), except all quantities now exhibit an additional time dependence. Although, in practice the equations are solved using linear-response rather than explicit time evolution, which is a good approximation for weak fields.

In general both TDDFT using KS and GKS approaches have been shown to provide good estimates of the optical gap, and are generally in good agreement with experimental data.[11] Where these approaches have been shown to break down is in the case of charge-transfer excitations.[44,45] These excitations are typically seen in donor-acceptor complexes, where the donor and acceptor molecules are well separated and interactions between the entities can effectively be ignored. In these complexes the size of the optical gap is determined by the Mulliken limit, given by[46]

$$hv_{CT} = I_D - A_A - 1/R \tag{1.11}$$

where $I_D - A_A$ is the fundamental gap of the complex and R is the distance between the donor and acceptor.

To accurately model the charge-transfer optical gaps the form of the functional must correctly describe the fundamental gap and the $1/R$ dependence seen in the Mulliken limit. Both semi-local functionals and conventional hybrid functionals fail in this respect.[11] Thankfully, the OT-RSH approach provides exactly the form that is needed, with the lowest excitation energy being described by

$$\varepsilon_H - \varepsilon_L - 1/R \tag{1.12}$$

As a result of the optimal-tuning procedure, $\varepsilon_H - \varepsilon_L$ is equivalent to the fundamental gap, and the required $1/R$ dependence is clear to see. For these theoretical reasons the use of OT-RSH has been shown to provide much better predictions of optical gaps in these charge-transfer processes than other DFT functionals.[11]

2. *Methodology*

2.1 Software

All calculations were carried out using Northwest Computational Chemistry Package 6.5 (NWChem). OT-RSH calculations were carried out using the Baer-Neuhauser-Livshits functional (BNL)(2007)[21,20] and some TDDFT calculations were carried out using the B3LYP functional[47]. The Jmol software package was used to visualise the molecules and molecular orbitals.

2.2 Programming Language

All code for this project was written in Python 2.7. The NumPy module was used extensively to assist with mathematical calculations including the numpy.polyfit function to fit polynomials to data points. The Matplotlib Python module was used to produce all the figures seen within this thesis.

2.3 Reference Molecules

The coordinate geometries for the reference molecules (given in supplementary information) were kindly provided by Xavier Blase and are identical to those used by Blase et al.[8] and Refaely-Abramson et al.[23] in their calculations of this set of molecules. The reference molecules were calculated using the cc-pvtz basis set, allowing for direct comparison with the published optimised γ values.[23] Calculations were considered converged when one or more of the following three criteria were met:

1. Energy converged to $1.0 \times 10^{-6}\,E_{\mathrm{h}}$;

2. Density converged to 1.0×10^{-5};

3. Gradient converged to 5.0×10^{-4}

The variable γ was varied to enable minimisation of the $J^2(\gamma)$ cost function. The convergence aids Damping and DIIS were invoked when required for two iterations and one hundred iterations respectively. For all calculations symmetry was locked.

2.4 Solvated Species

The solvated species were calculated using the cc-pvtz basis set and the same convergence critera as for the gaseous species to allow for comparison. To calculate the solvated species c-PCM was used by using the directive *cosmo* (Conductor-like Screening Model) within the NWChem framework. This implements the solvation model developed by York et al.[48] The model calculates the polarisation of the molecule as a whole in a dielectric field. A dielectric constant of 2.0 and 80.1 were used to approximate a non-polar organic solvent and water respectively.

2.5 Titania Clusters

The coordinate geometries (given in supplementary information) were kindly provided by Martijn Zwijnenburg for comparison with previous G_0W_0 and B3LYP calculations.[31] The titanium dioxide clusters were calculated using the Def2-TZVPP basis set to allow for the previous comparison. Although not all G_0W_0 and B3LYP calculations were carried out with this basis set, as some used Def2-SVPD, the previous calculations carried out by Zwijnenburg have shown that both approaches gave similar results, and as Def2-TZVPP is generally more accurate than Def2-SVPD, it was used. To aid convergence, the quadratic convergence algorithm (CGMIN) was used for all calculations, except for the TiO_2 molecule, which was converged using the same default setting used for the reference molecules. For this method the gradient is used as the convergence criteria. The required gradient ranged from 5.0×10^{-4} to 2.0×10^{-3} dependant on what was needed to make the the molecule converge. All molecules were within the 1.0×10^{-6} E_h energy convergence criteria when converged used as the default when the CGMIN method is not invoked.

3. *Results and Discussion*

3.1 Understanding the $J^2(\gamma)$ cost function

3.1.1 The $J^2(\gamma)$ cost function

The $J^2(\gamma)$ cost function provides a means of tuning the BNL functional such that Koopman's theorem is obeyed. The cost function is as follows[11]

$$J^2(\gamma) = (\varepsilon^{\gamma}_{H(N)} + I^{\gamma}(N))^2 + (\varepsilon^{\gamma}_{H(N+1)} + I^{\gamma}(N+1))^2 \tag{3.1}$$

where $\varepsilon^{\gamma}_{H(N)}$ is the eigenvalue of the N-electron state, and $I^{\gamma}(N)$ is the ionisation energy of the N-electron state, where the ionisation energy is calculated as the total energy difference between the N and $N-1$ electron states. When the cost function is minimised, γ is said to be optimised.

To develop a code that will efficiently optimise γ, one first needs to understand the behaviour of the J^2 function. This is important in choosing what method to use to try determine the location of the global minimum. The presence of local minima can add considerable difficulty to finding the global minimum so knowledge of the behaviour of the function is vital to developing a technique to implement.

3.1.2 Grid method

To map the $J^2(\gamma)$ function a grid method was employed. A review of the literature showed that for a wide range of molecules, γ was found to be almost always between 0.0 and 0.5, including for the reference molecules chosen develop the code. For single atom calculations γ has been shown to be as high as 1.88 for a sodium atom,[22] but that is beyond the scope that the code is designed for, although the same techniques implemented in this optimisation should be applicable to other cases.

To initially map $J^2(\gamma)$ a grid of γ values between 0.0 and 0.5 at steps of 0.05 for each of the reference molecules was evaluated. The initial results showed what appeared to be a well behaved curve with a single, global minimum. To accurately determine the location of the minimum it was necessary to map the area surrounding the minimum at greater resolution. To do this another grid of γ values was evaluated. This grid of γ was determined denoting the γ value with the lowest value of $J^2(\gamma)$ as γ_{min}. The new grid was then defined to the values of γ between $\gamma_{min} - 0.05 \leq \gamma \leq \gamma_{min} + 0.05$, taken at steps of 0.005.

Once the molecules were evaluated using this new grid of γ values, the same technique was applied again to determine the optimised value of γ, γ_{opt}. The value of γ_{opt} was to be determined to three decimal places, in accordance to the accuracy of γ values given in the literature. To do this, the value of γ from this new set of γ values with the lowest value of $J^2(\gamma)$ was once again denoted γ_{min}. A new grid of γ values was created such that, $\gamma_{min} - 0.005 \leq \gamma \leq \gamma_{min} + 0.005$ at steps of 0.001. This benefits of this approach were two-fold, it would determine the value of γ to three decimal places and would also map to a high resolution the area directly surrounding the minimum.

The results of this method showed that for all of the reference molecules the $J^2\gamma$ function was well behaved and had only one, global, minimum. What was also demonstrated was that the convergence criteria was sufficient that the $J^2(\gamma)$ curve stayed well defined at steps of 0.001, such that variations in energy resulting from the iterative process implemented were not significant in the overall determination of γ.

3.2 Development of code

3.2.1 The polynomial approach

The initial plotting of the $J^2(\gamma)$ cost function showed encouragement that the function could be approximated by a simple polynomial function. As the function con-

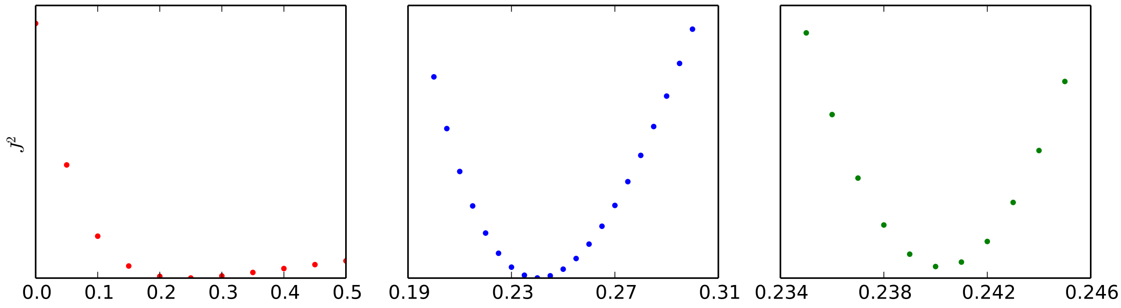


Figure 3.1: Plots of the $J^2(\gamma)$ against γ for fluorene with data points from the first scan (left), at which γ is increased in increments of 0.05 between 0.0 and 0.5, the second scan (centre), in which γ is increased by 0.005 between 0.20 and 0.30, and the final scan (right) in which γ is increased by 0.001 between 0.235 and 0.245.

tains only one, global minimum, and also as the area immediately around the minimum is close to symmetric, one should hopefully be able to predict the location of the minimum with only a small amount of data. In order to make the process as efficient as possible, the code should be optimised to use the minimum number of DFT calculations possible while still finding an accurate approximation of the minimum.

To do this polynomials of differing orders were fitted to a varying number of data points. For high-order polynomials, especially when the number of data points it is being fitted to is relatively low, over-fitting of the data can occur, known as the Runge's phenomenon. The cost function is asymmetric, and due to the constraint of attempting to fit the function to as few data points as possible, whilst also avoiding over-fitting, is apparent that a 3rd or 4th order polynomial would be most suitable to approximate the minimum.

3.2.2 Trials of fitting to a polynomial

Several different spreads of γ values were used in order to ascertain the optimal number of data points to fit to. The spread of γ values from the third stage of the grid method, where there are ten values of γ around the minimum at steps of 0.001, was fit to a third and fourth order polynomial to evaluate an accurate value of γ_{opt} to more than 3 decimal places.

Attempting to fit the polynomial to a range of γ values between 0.0 and 0.5 was shown to be inaccurate at determining the minimum, due to the asymmetric character of the $J^2(\gamma)$ function. When a smaller range of γ values were used, in which the value of γ only ranged by 0.1 far more accurate results were achieved. A polynomial was fit to the data points where the step-size in γ was 0.005, 0.01, 0.02, and 0.025 (see Fig 3.2 for a fluorene example). Remarkably for every step-size in γ, when fitted to a 4th-order polynomial, the γ value of the minimum agreed with that of the 'true' γ_{opt} value to at least four decimal places for all of the reference molecules considered. This showed that the location of the minimum could be successfully predicted using just five γ values at steps of 0.025.

3.2.3 Algorithm for locating the minimum

In order to find the optimised value of γ, γ_{opt}, the following algorithm was devised:

1. Start with a range of γ values where $0.0 \leq \gamma \leq 0.5$ at steps of 0.1 and evaluate $J^2(\gamma)$.
2. Denote the γ with the lowest $J^2(\gamma)$ value γ_{min}. Set a new range of γ values from $\gamma_{min} - 0.1 \leq \gamma \leq \gamma_{min} + 0.1$ at steps of 0.0025 and evaluate $J^2(\gamma)$.

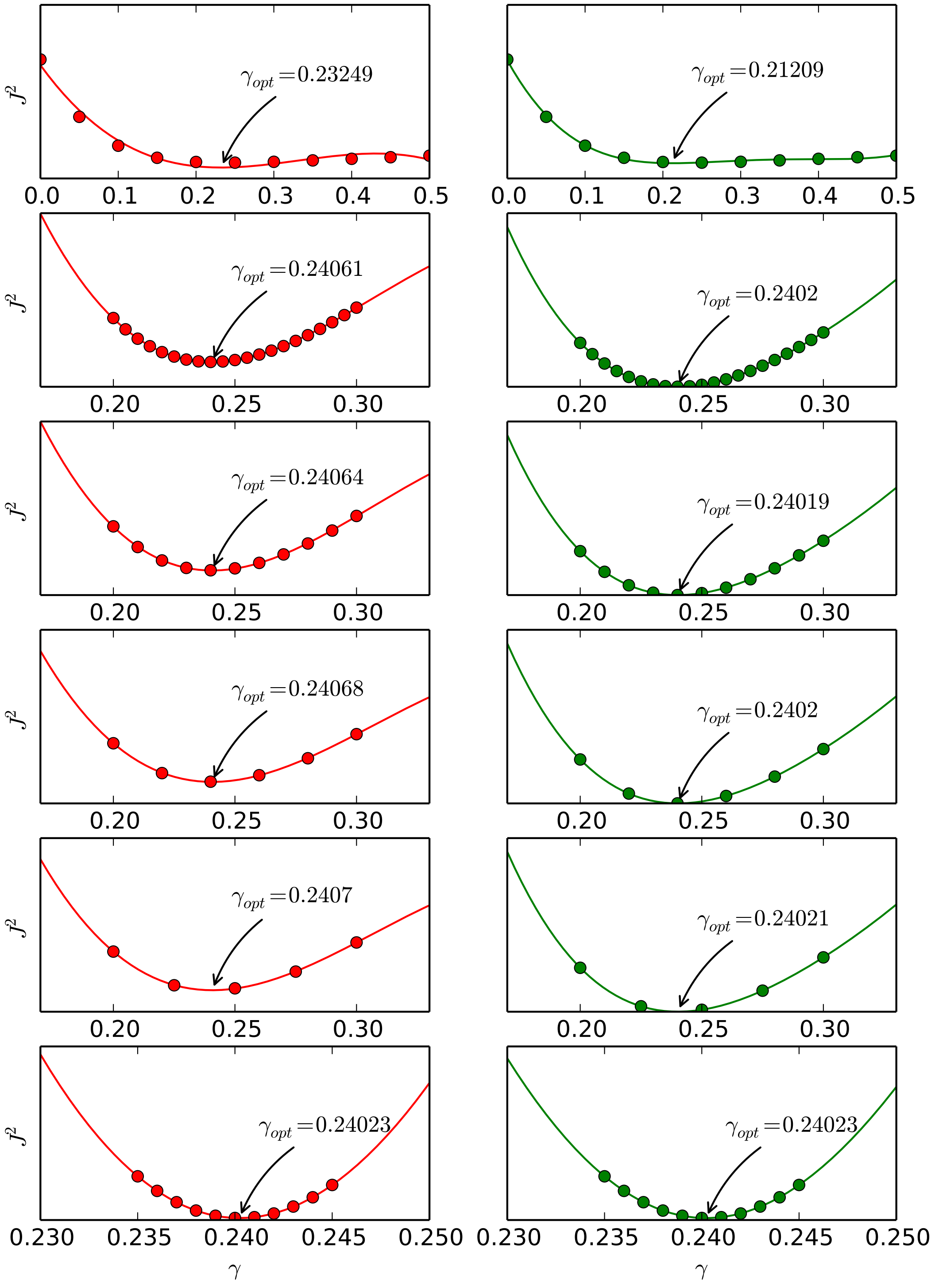


Figure 3.2: Attempts of fitting a polynomial to the $J^2(\gamma)$ data points for fluorene; The line represents the polynomial fit to the data points represented by circles. On the left (red) the data is fit to a 3rd-order polynomial and on the right (green) the a 4th-order polynomial. The value of γ at the minimum is indicated for each fitting procedure is displayed on the plot.

3. Denote the γ with the lowest $J^2(\gamma)$ value γ_{min}. Set a new range of γ values from $\gamma_{min} - 0.05 \leq \gamma \leq \gamma_{min} + 0.05$ at steps of 0.0025 and fit a 4th-order polynomial to these points.
4. Differentiate the polynomial and set equal to zero. Find the roots of this equation and call the real root γ_{opt}.

This ensured that γ_{opt} could in a computationally efficient manner.

3.2.4 Structure of Code

The code for optimising γ was split into three parts. An description of the algorithmic function of these parts is given in the following sections.

Part 1

This part of the code submits jobs for the initial scan of γ values, where $0.0 \leq \gamma \leq 0.5$ at steps of 0.1. It is necessary to submit jobs for the neutral, -1 anionic and +1 cationic states of the molecule to determine ionisation potentials and electron affinities.

Part 2

This script reads the output files from the initial scan and calculates the value of $J^2(\gamma)$ for each value of γ. It then submits jobs for γ values from $\gamma_{min} - 0.075 \leq \gamma \leq \gamma_{min} + 0.075$ at steps of 0.0025, not including γ_{min}, again for the neutral, anionic and cationic states. In the special case where $\gamma_{min} = 0.0$ the algorithm presented above does not give an accurate prediction of γ_{opt} as the step size is too large to accurately map the minimum when γ_{opt} is small. For this case a set of jobs is submitted for γ values where $0.01 \leq \gamma \leq 0.1$ at step size of 0.01.

Part 3

In this code values of $J^2(\gamma)$ are calculated for γ values where $\gamma_{min} - 0.1 \leq \gamma \leq \gamma_{min} + 0.1$ at steps of 0.0025. A new γ_{min} is denoted as the value of γ from this set of γ values that has the lowest value of $J^2(\gamma)$. A 4th-order polynomial is fitted the the $J^2(\gamma)$ values from $\gamma_{min} - 0.05 \leq \gamma \leq \gamma_{min} + 0.05$. The real root of the derivative of the polynomial is found and denoted γ_{opt}. Jobs are submitted for γ_{opt} to 4 decimal places, and also $\gamma_{opt} - 0.002$, $\gamma_{opt} - 0.001$, γ_{opt}, $\gamma_{opt} + 0.001$, $\gamma_{opt} + 0.002$, evaluated to three decimal places. These extra jobs are submitted to ensure the minimum is truly the minimum, and can be omitted if desired.

3.3 Reference molecules

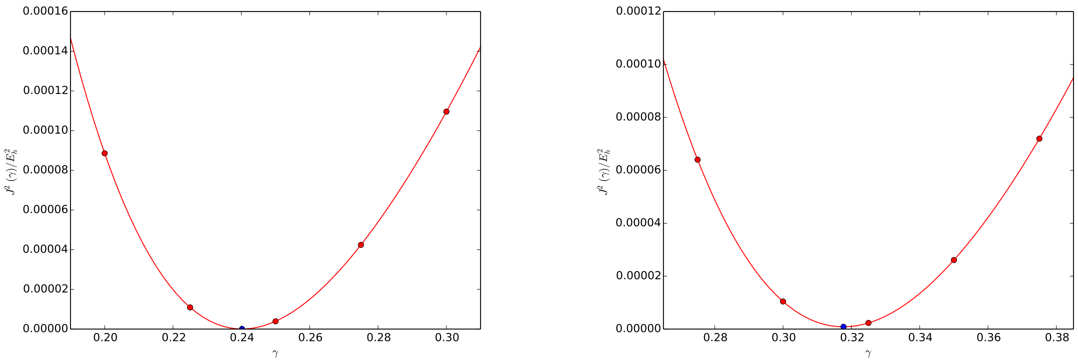
3.3.1 Optimisation of γ

Optimisation of γ for the reference molecules selected gave γ_{opt} values ranging from 0.242 and 0.353. All γ_{opt} values calculated were within 0.0047 of those calculated by Refaely-Abramson et al., (see Table 3.1) with the exception of benzothiazole, which differed by 0.0247. The small variations in γ are likely due to differences in the density of integration grid utilised by the software, and are within a reasonable error margin. The discrepancy seen in the benzothiazole result is unusual and as of yet undetermined. The other key values from the DFT calculation, such as the fundamental gaps, are still in good agreement with both experimental and computational figures, and so this difference should not cause too much concern.

	Minimisation Protocol		Literature Values[23]	
Molecule	γ_{opt}	Fundamental Gap / eV	γ_{opt}	Fundamental Gap / eV
thiophene	0.3177	10.47	0.313	10.45
thiadiazole	0.3530	10.56	0.355	10.56
benzothiadiazole	0.2915	8.21	0.288	8.16
benzothiazole	0.3177	9.49	0.293	9.36
fluorene	0.2402	8.36	0.24	8.38

Table 3.1: Comparison of calculated optimal γ and fundamental gaps of those calculated using the optimisation algorithm and those provided by Refaely-Abramson et al.[23]

The optimisation algorithm gave a polynomial with only one real root for all molecules (see Fig 3.3 for examples of fluorene and thiophene). Comparison of the root with the calculated values of $J^2(\gamma)$ for the γ values within $\pm 0.001, 0.002$ showed that it was truly a minimum and that the minimisation process was successful.



(a)

(b)

Figure 3.3: The γ optimisation procedure for a) fluorene; and b)thiophene. The red markers represent the $J^2(\gamma)$ values of the data points used for the fitting procedure and the blue marker the location of $J^2(\gamma_{opt})$.

3.3.2 Excitation gaps

All excitation gaps were in good agreement with those found in literature. As we have enforced Koopman's theory, the HOMO eigenvalue should be equivalent to the ionisation potential. The energies of the experimental ionisation potential and the HOMO eigenvalue have a standard deviation of $\sigma = \pm 0.314$ eV, with a maximum difference of ± 0.52 eV (see Table 3.2).

Due to a lack of experimental data concerning fundamental gaps, and thus LUMO energies, the most accurate method to compare to for the reference molecules is G_0W_0. B3LYP values are also provided for comparison. It is shown that the fundamental gaps from the OT-BNL calculations are in good agreement with those G_0W_0, as well as the those of the reference OT-BNL calculations from literature (see Fig. 3.4).[23] In fact the calculated OT-BNL results show slightly better agreement with G_0W_0 than the reference OT-BNL calculations, with a standard deviation of the HOMO and LUMO eigenvalues from G_0W_0 of $\sigma = \pm 0.090$ eV and $\sigma = \pm 0.360$ eV respectively.

As expected, due to the well-known deficiencies of B3LYP when it comes to calculating fundamental gaps, the correlation between the fundamental gaps of B3LYP and those calculated with the other methods is poor, being typically underestimated by $\sim 45\%$ to those from G_0W_0. As well as this, the B3LYP calculated HOMO eigenvalues are consistently less than ionisation potentials from experimental data, typically by $\sim 25\%$.

Evaluation of the optical gaps (see Table 3.3) using the OT-BNL functional with TDDFT yielded results that were typically within 0.19 eV of those found experimentally. They were also in good agreement with the OT-BNL calculations of Refaely-Abramson et al. and also with B3LYP, which is known to be accurate for these types of molecules

Overall both the fundamental gaps and optical gaps were in good agreement with both experimental and other computational methods. The similarity between the

Molecule	γ_{opt}	HOMO e / eV	LUMO e / eV	Fund. Gap / eV	$J(\gamma)$/eV	Exp. I.P. / eV
thiophene	0.3177	-8.72	1.75	10.47	0.025	-8.89
thiadiazole	0.3530	-9.91	0.65	10.56	0.049	-10.11
benzothiadiazole	0.2915	-8.63	-0.42	8.21	0.001	-8.99
benzothiazole	0.3177	-8.58	0.91	9.57	0.165	-8.74
fluorene	0.2402	-7.51	0.85	8.37	0.006	-8.03

Table 3.2: Quantities of importance from the OT-BNL calculations. Experimental ionisation potentials from the literature[23] are given for comparison.

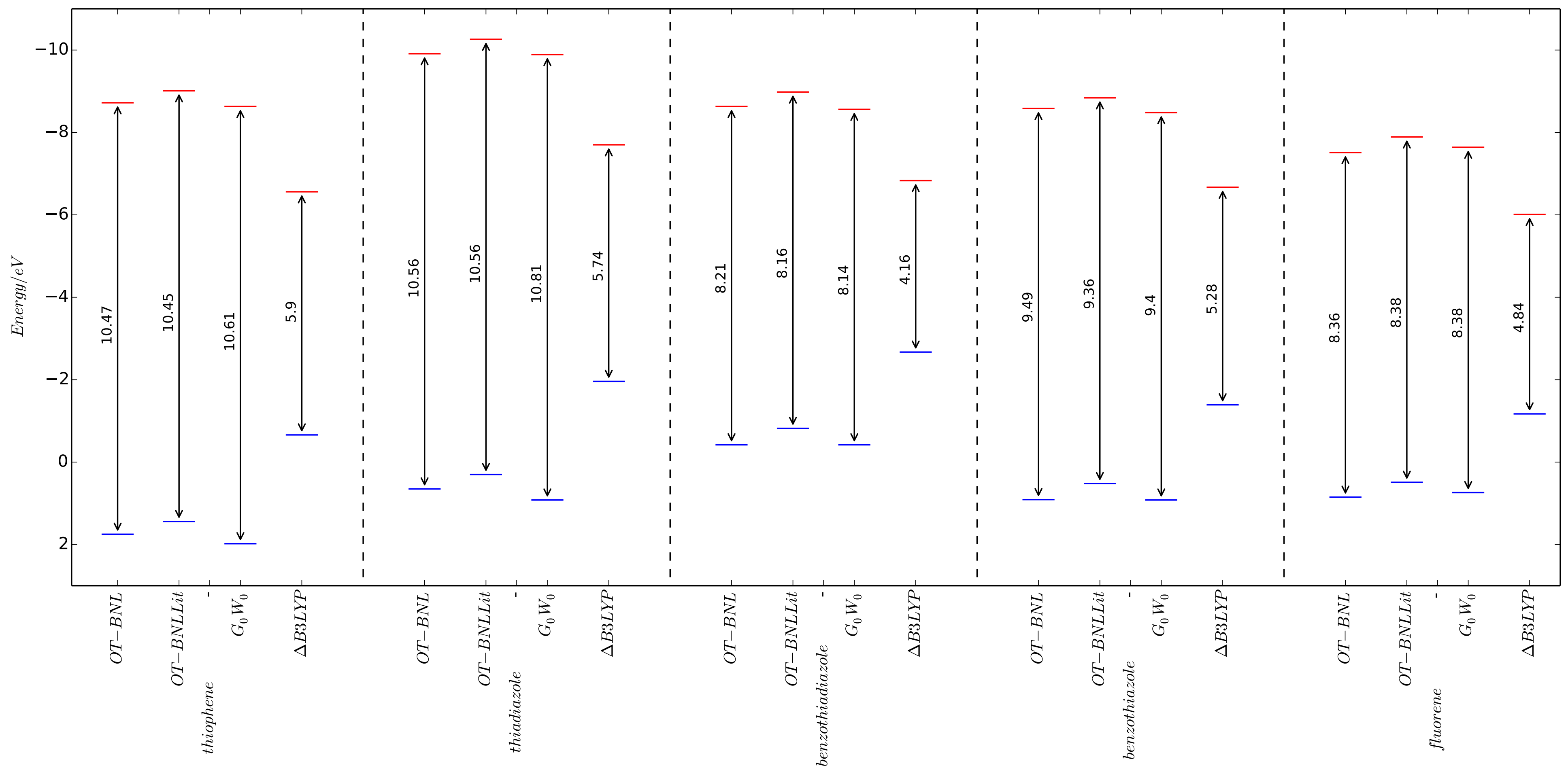


Figure 3.4: Fundamental gaps for the reference molecules. The blue line represents the energy of the HOMO and the red line the energy of the LUMO. The energy of the HOMO is taken to be ε_H, and the energy of the LUMO as ε_L. The OT-BNL-Literature values and KS-B3LYP are taken from Refaely-Abramson et al.[23] and the G_0W_0 values from Blase et al.[8]

	First singlet excitation energies			
Molecule	OT-BNL / eV	EXP / eV	Lit - OT-BNL / eV	B3LYP / eV
thiophene	5.81	5.52	5.79	5.75
thiadiazole	5.18	5	5.22	5.05
benzothiadiazole	4.18	4.05	4.15	3.71
benzothiazole	4.85	-	4.81	4.61
fluorene	-	4.19	4.65	4.43

Table 3.3: First singlet excitation energies (optical gaps) for the set of reference molecules. OT-BNL calculated quantities are provided, as well as experimental, OT-BNL values and B3LYP all from literature.[23]

values found and those calculated by Refaely-Abramson et al. shows that the optimisation algorithm has functioned as intended, and gives similar values to those found in literature. Now that it has been established that the procedure is successful, the algorithm can be directed as an approach to the evaluation of other systems.

3.4 Solvation models

3.4.1 Optimisation of γ

Non-Polar Organic solvation

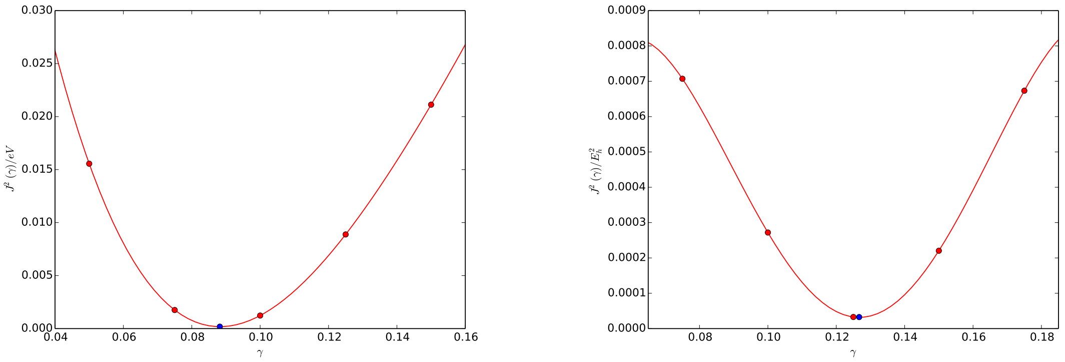
To simulate the effect of a non-polar organic solvent, the COSMO model was used with dielectric constant, ε_r = 2.0. Two methods were applied in the evaluation of solvated systems. In the first instance, γ was optimised with the COSMO model applied (see Fig. 3.5). This resulted in a significant lowering of the value of γ_{opt} for all molecules (Table 3.4). Along with the reduction of γ_{opt}, there was also a significant increase in $J^2(\gamma)$ for all molecules. This suggests that the the OT-BNL approach may not be as accurate, as Koopman's theorem is not as strictly obeyed.

	Vacuum, $\varepsilon_r = 1.0$		Organic solvent, $\varepsilon_r = 2.0$		Water, $\varepsilon_r = 80.1$	
Molecule	γ_{opt}	$J(\gamma)$/eV	γ_{opt}	$J(\gamma)$/eV	γ_{opt}	$J(\gamma)$/eV
thiophene	0.3177	0.025	0.1267	0.155	0.0149	0.431
thiadiazole	0.3530	0.049	0.1367	0.116	0.0371	0.106
benzothiadiazole	0.2915	0.001	0.1088	0.046	0.0231	0.040
benzothiazole	0.3177	0.165	0.1183	0.243	$-$	$-$
fluorene	0.2402	0.006	0.0882	0.070	0.0126	0.071

Table 3.4: The OT-BNL optimised γ and $J^2(\gamma)$ values for the reference set of molecules calculated in vacuum, in a non-polar organic solvent of dielectric constant $\varepsilon_r = 2.0$ using COSMO, and in water of dielectric constant $\varepsilon_r = 80.1$.

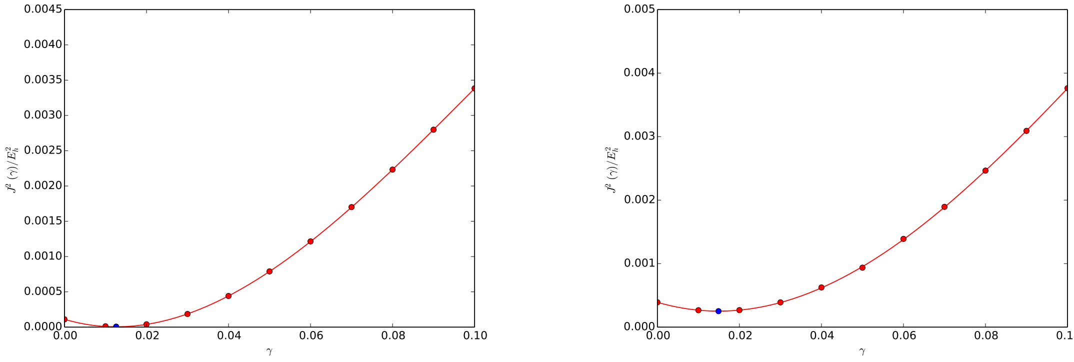
Water solvation

This trend continued when the molecules were treated as being in solvated by water, by increasing the relative permittivity to $\varepsilon_r = 80.1$. As can be seen in Fig 3.6, the value of γ_{opt} is now close to zero, and the $J^2(\gamma = 0)$ value is not significantly larger than that of γ_{opt}. This shows that the contribution of the LR part of the exchange functional is no longer being as effective in accurately enforcing Koopman's theorem, and the system is being described by almost solely SR exchange. The $J^2(\gamma)$ values are much larger for all molecules than of those calculated in the vacuum, but do not show any clear increase or decrease on those for the organic solvated system. (see Table 3.4) This suggests both systems may be less accurate than the results found for the molecule in a vacuum, as Koopman's theorem is not as strictly enforced in either case, but the two solvated models are likely to have similar performances in comparison to eachother.



(a) **(b)**

Figure 3.5: The γ optimisation procedure for a) fluorene; and b) thiophene undertaken using COSMO to replicate a non-polar organic solvent with dielectric constant $\varepsilon_r = 2.0$.



(a) **(b)**

Figure 3.6: The γ optimisation procedure for a) fluorene; and b) thiophene undertaken using COSMO to replicate water as a solvent using dielectric constant $\varepsilon_r = 80.1$.

3.4.2 Fundamental Gaps

As well as the calculations for which γ has been optimised within the COSMO solvation model, the fundamental gaps of the molecules have also been calculated using the γ_{opt} optimised in the vacuum, as previously carried out by Zheng et al.[30] (see Fig. 3.7). Theoretically it is predicted that the size of the fundamental gap will be reduced under solvation, as the solvent stabilises the LUMO of the molecule. This trend is seen across all molecules for which the value of γ was optimised whilst COSMO was applied. It is generally seen for the molecules that use γ optimised in vacuum as well, although for benzothiadiazole there is in fact an increase in the size of the fundamental gap. This result does not make physical sense, and suggests that that this approach is unreliable.

Queiroz et al. have investigated the optimisation of the range-separation parameter under c-PCM.[29] They have shown that the method leads to an artificial lowering of γ to unrealistic values. In this case it would appear this is true, for the molecules solvated in water γ is almost 0, and the value of $J^2(\gamma)$ is little lower than if it was. This means that there is almost no contribution from the long-range Fock-like operator in the exchange term. Queiroz et al. have shown that there is a decrease in the ionisation energies when the range-separation parameter is optimised using c-PCM for thiophene oligomers, which is seen in the solvated models for the reference molecules. They argue that the reason for the reduction in γ is due to the calculation of the ionisation potentials being vertical in vacuum, but only vertical with respect to the solute and not the solvent when in solution. This leads to ionisation potentials that are unrealistically low. They have overcome this by explicitly representing the solvent molecules within the system, and have shown that by gradually extending the solvent system one can provide a estimate of γ_{opt} that does not suffer from the same artificial lowering seen whilst using c-PCM.[49]

3.5 Titanium dioxide clusters

3.5.1 Optimisation of γ

The process of optimising γ for the titanium dioxide clusters produced similar results to the reference molecules, aside from a quadratic convergence algorithm being required to ensure the calculations converged to the lowest energy state. The $J^2(\gamma)$ functions of all the clusters were well-behaved and only had a global minimum (see Fig. 3.8). There is a general trend as cluster size increases for the value of γ_{opt} to decrease, as SR effects begin to dominate due to delocalisation of MOs across the molecule. Indeed, it has been demonstrated by Stein et al.[22] that

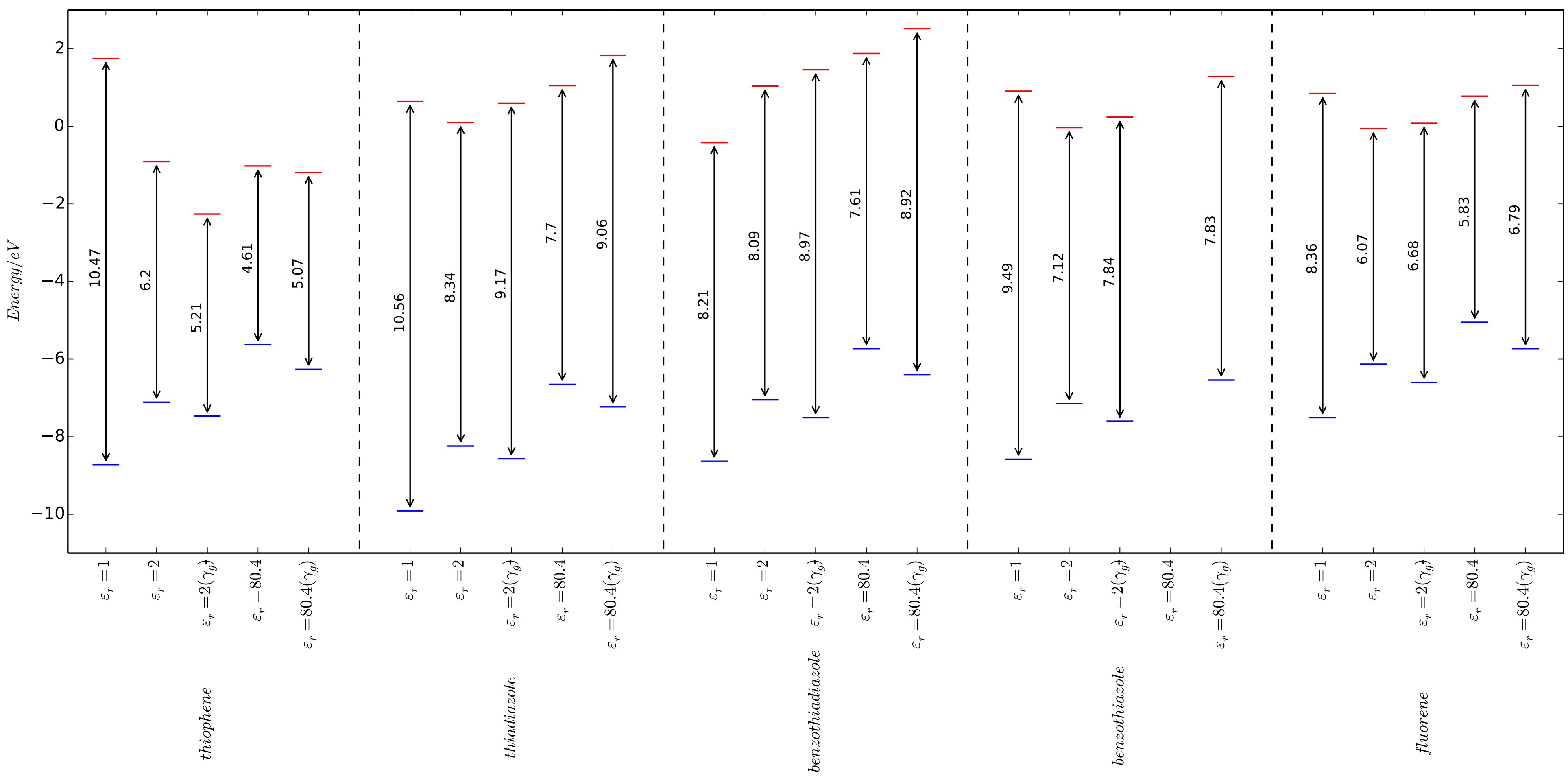
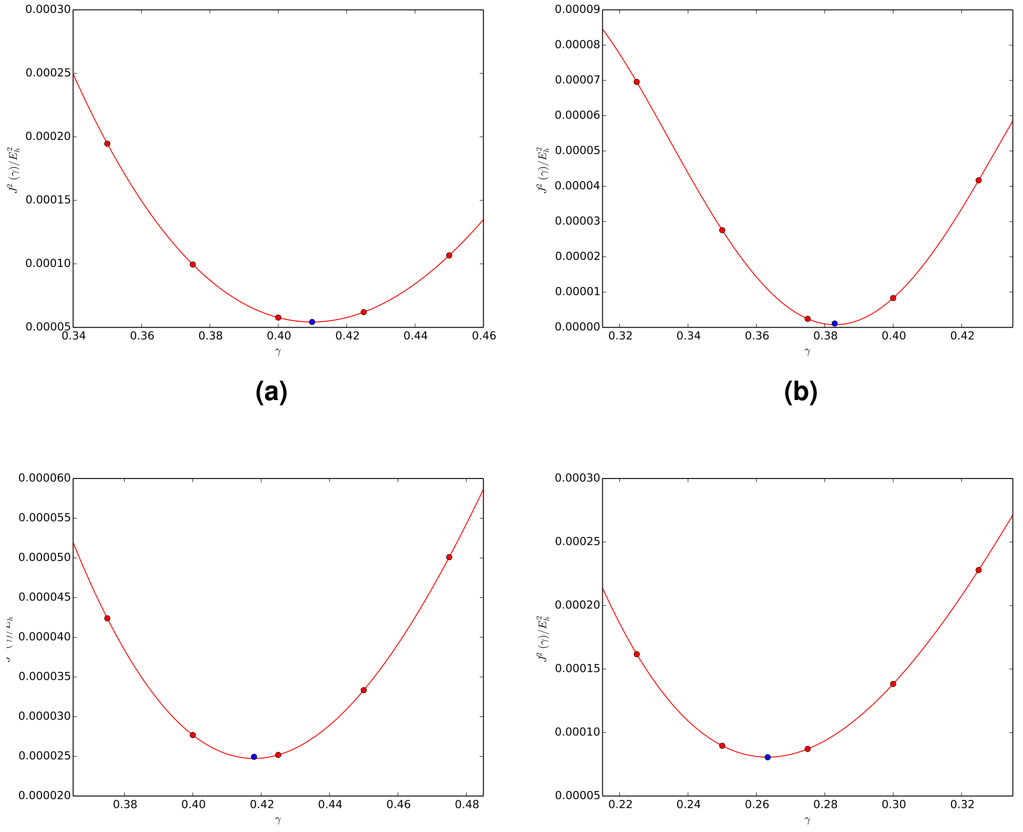


Figure 3.7: Fundamental gaps from OT-BNL for the reference molecules with COSMO applied at dielectric constants of 1.0 (vacuum), 2.0 (non-polar organic solvent) and 80.1 (water). The blue line represents the energy of the HOMO and the red line the energy of the LUMO. The energy of the HOMO is taken to be ε_H, and the energy of the LUMO as ε_L. γ_g represents that γ was optimised in vacuum whilst for the other plots γ was optimised with COSMO applied.

for both oligoacenes and silicon crystals there is a trend that as size is increased there is a reduction in γ_{opt}, so this result is as expected.

Whilst the optimisation of γ was generally as simple as with the reference molecules, the $J^2(\gamma)$ values produced were considerably higher. This indicates that the approach is unlikely to be as effective as for the reference molecules due to the greater disparity between the N, and N+1 HOMO eigenvalues, and their corresponding ionisation potentials, meaning that Koopman's theorem is not as strictly enforced.



(a)

(b)

(c)

(d)

Figure 3.8: The γ optimisation procedure for a) TiO_2; b) $Ti(OH)_4$; c) $trans-(TiO_2)_2$ and d)$(TiO_2)_4$. The red markers represent the $J^2(\gamma)$ values of the data points used for the fitting procedure and the blue marker the location of $J^2(\gamma_{opt})$.

3.5.2 Fundamental Gaps

In order to gauge the effectiveness of the OT-RSH approach, fundamental gaps calculated using G_0W_0 and B3LYP have been included for comparison (see Fig. 3.9 and Table 3.5). Whilst G_0W_0 is not an exact method, as approximations are introduced that affect the accuracy if its results, it does have a tendency to correctly predict trends across a set of molecules.

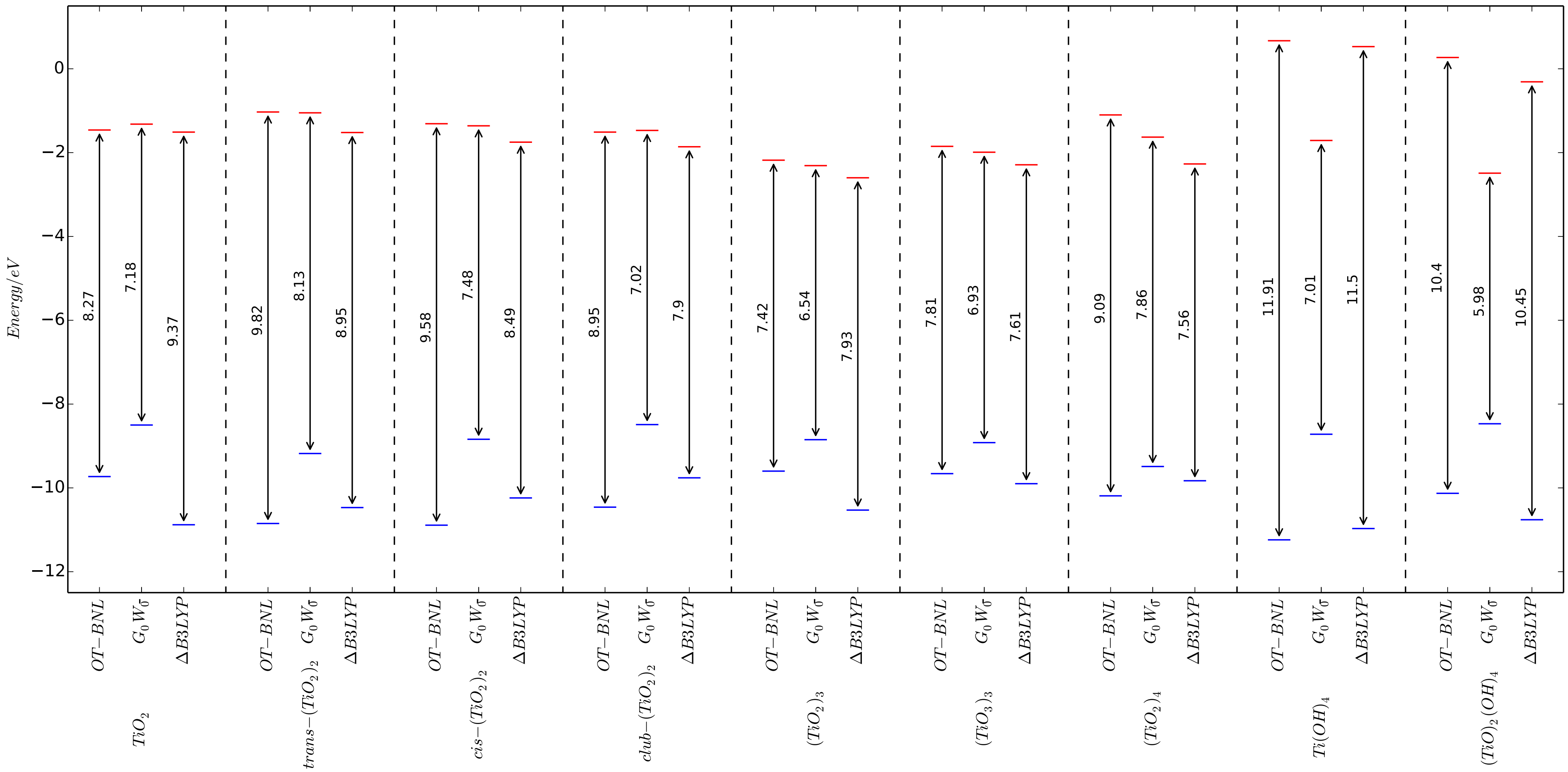


Figure 3.9: Fundamental gaps for the titanium dioxide clusters. The blue line represents the energy of the HOMO and the red line the energy of the LUMO. The energy of the HOMO is taken to be ε_H, and the energy of the LUMO as ε_L for OT-BNL, which were calculated using Def2-TZVPP. G_0W_0 and $\Delta B3LYP$ values were calculated using the Def2-SVPD basis set and provided by M. A. Zwijnenburg/[31] The $\Delta B3LYP$ are calculated by the difference in ionisation potential and electron affinity of the molecule.

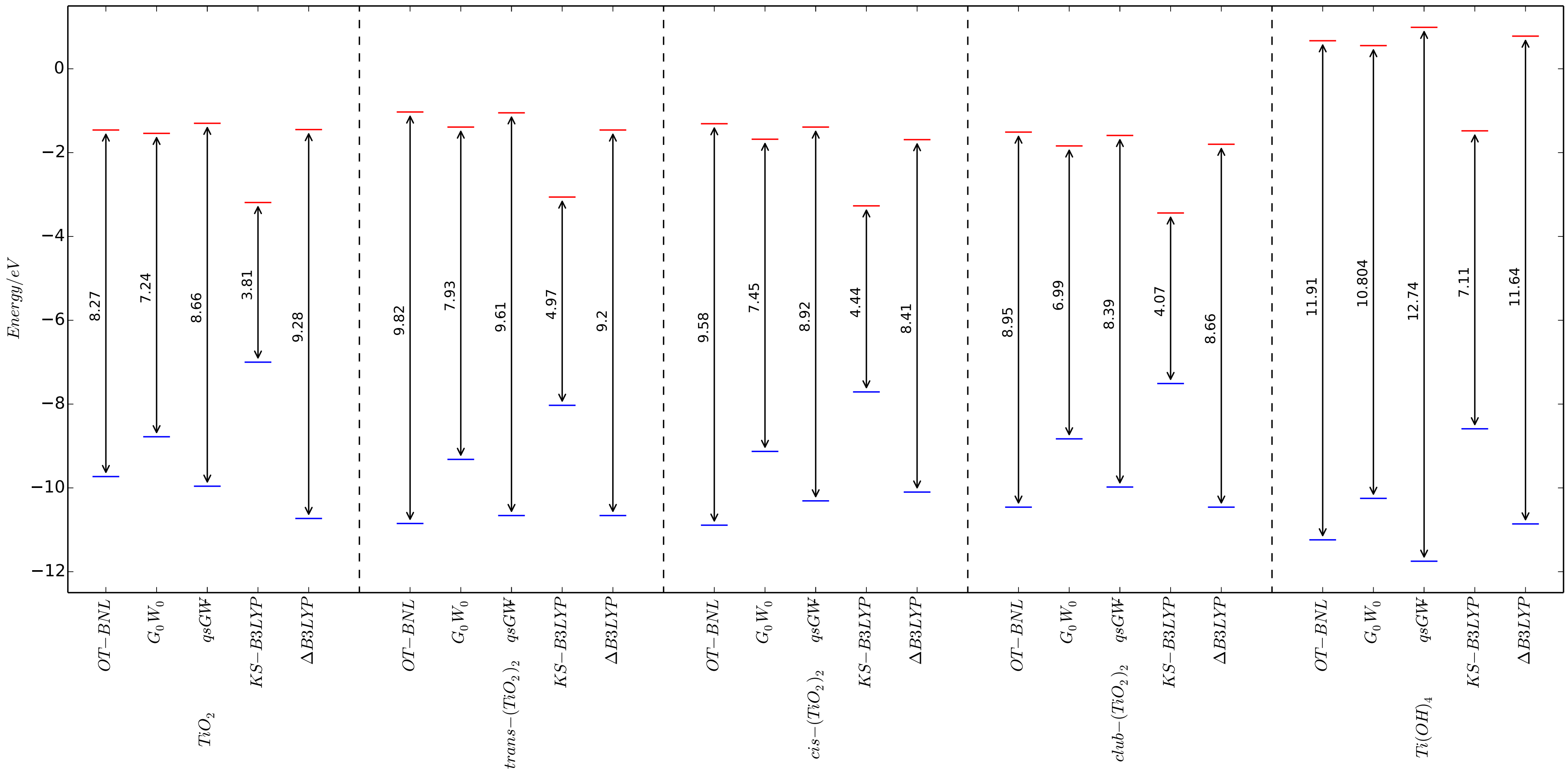


Figure 3.10: Fundamental gaps for the titanium dioxide clusters. The blue line represents the energy of the HOMO and the red line the energy of the LUMO. The energy of the HOMO is taken to be ε_H, and the energy of the LUMO as ε_L for OT-BNL. G_0W_0, $qsGW$, $\Delta B3LYP$ and $KS - B3LYP$ values were provided by M. A. Zwijnenburg/[31] The $\Delta B3LYP$ are calculated from the ionisation potential and electron affinity of the molecule, whilst the $KS - B3LYP$ values were calculated from ε_H and ε_L. All calculations were using the Def2-TZVPP basis set.

What can be seen is that the trends in fundamental gaps seen from OT-RSH do not match those seen in G_0W_0 for the unhydrated clusters. (see Fig. 3.10) The fundamental gaps have been universally calculated to be much larger than those seen in G_0W_0 for both non-hydrated and hydrated clusters, although the effect is more pronounced for those that are hydrated. As well as this, they show less agreement with the G_0W_0 values than those calculated with ∆B3LYP, a method which has well documented deficiencies when it comes to calculating fundamental gaps, although these values are calculated by finding the difference between the ionisation energy and the electron affinity, rather than from the HOMO and LUMO eigenvalues, with the latter being a less accurate method.

Interestingly though, for those particles for which $qsGW$ values are available, the HOMO and LUMO eigenvalues and fundamental gaps of the OT-RSH approach are seen to be much closer in value to those from the $qsGW$ approach than those of G_0W_0, with a standard deviation of the fundamental gap of $\sigma \pm 0.62$eV and $\sigma \pm 1.57$eV respectively (see Fig 3.10). It has been shown that G_0W_0 consistently underestimates the size of the fundamental gap for bulk solids, whilst $qsGW$ slightly overestimates it.[5] Whilst the application of GW to clusters may result in different trends to those seen in bulk materials, this does suggest that the results from OT-RSH could be a improvement of those found with G_0W_0.

For both $cis-(TiO_2)_2$ and $club-TiO_2$, there is a large difference in the value of γ that enforces Koopman's theorem for the N and N+1-electron systems respectively (explained further in the next Section 3.6.3). Interestingly this does not result in significantly higher values in the $J^2(\gamma)$ function than for the other clusters but it does appear to have an effect on the predictive power of the OT-RSH approach for them. The trend in fundamental gaps exhibited by both $qsGW$ and G_0W_0 is replicated if the $cis-(TiO_2)_2$ and $club-TiO_2$ are ignored. What's more it can be seen that this disparity between the optimal γ for the two versions of Koopman's theorem causes an overestimation of the fundamental gap in both cases in, when compared to the trends seen in both $G@$ calculations.

Molecule	γ_{opt}	HOMO e / eV	IP / eV	LUMO e / eV	EA / eV	$J^2(\gamma)$/eV
TiO_2	0.4099	-9.73	9.77	-1.46	1.51	1.045
$trans-(TiO_2)_2$	0.4179	-10.85	10.89	-1.03	1.03	0.709
$cis-(TiO_2)_2$	0.4974	-10.89	10.93	-1.31	1.37	1.126
$club-(TiO_2)_2$	0.4592	-10.46	10.73	-1.51	1.89	3.304
$(TiO_2)_3$	0.2878	-9.60	9.80	-2.18	2.35	1.484
$(TiO_3)_3$	0.282	-9.66	9.83	-1.85	2.01	1.368
$(TiO_2)_4$	0.2633	-10.19	10.41	-1.10	1.21	1.274
$Ti(OH)_4$	0.3829	-11.24	11.24	0.67	-0.75	0.148
$(TiO)_2(OH)_4$	0.2916	-10.13	10.20	0.27	-0.21	0.499

Table 3.5: Quantities of importance from the OT-BNL calculations for the titanium clusters

Although G_0W_0 and B3LYP for all clusters calculated with basis set Def2-TZVPP are not available, those that are show a similar relation to the values from OT-RSH to those that are seen using def2-SVPD. Furthermore, a large change is not seen when changing basis set for the G_0W_0 and B3LYP calculations, so it would seem unlikely that changing the basis set for the OT-RSH calculations would cause a substantial difference to the fundamental gaps calculated. More data is needed to determine the accuracy of the fundamental gaps predicted by as OT-RSH. Fundamental gaps for all systems determined by $qsGW$, coupled with experimental data, would provide a means of drawing comparisons with better defined and more reliable quantities. OT-RSH seem to work as long as the value of γ that enforces Koopman's theorem for the N and N+1 electron systems is similar. This can be easily checked when undertaking calculations, and suggests a means of checking the validity of the OT-RSH approach for a specific molecule beyond analysis of the $J^2(\gamma)$ function.

3.5.3 Optical gaps

Optical gaps were calculated using OT-BNL and B3LYP with TDDFT (Table 3.6). Further values for optical gaps calculated using CAM-B3LYP and Equation-of-Motion Coupled Cluster (EOM-CC) by Berado et al. were found in the literature.[34] EOM-CC is computationally more expensive than TDDFT and is generally considered as more accurate, and will be considered as a bench-mark from which to compare results against. CAM-B3LYP is a modified version of B3LYP that performs better than it at describing optical excitations, particularly charge-transfer excitations, due to the nature of its exchange correlation.

Analysis of the optical gaps from OT-RSH is inconclusive. The OT-BNL functional does perform well in the case of $(TiO_2)_3$ functional, which due to the charge transfer-like nature of the excitation is underestimated by B3LYP. For the other clusters OT-BNL generally gives results that are closer to EOM-CC than B3LYP, but never closer than those given by CAM-B3LYP. For the hydrated clusters OT-RSH performs better than both B3LYP and CAM-B3LYP, with respect to optical gaps given by EOM-CC. Once again, if the optical gaps calculated for the $cis-(TiO_2)_2$ and $club-TiO_2$ are ignored, a much better agreement with the other computational methods is observed, and OT-BNL behaves in a manner that is much superior to B3LYP. It is becoming more clear it is very important that the optimal values of γ that satisfy Koopman's theorem for the N and N+1 systems are close together, and that this seems to affect the predictive power of the method more than just the size of $J^2(\gamma)$.

Molecule	First singlet excitation / eV			
	$OT-BNL$	$B3LYP$	$CAM-B3LYP$	$EOM-CC$
TiO_2	2.74	2.63	2.86	–
$trans-(TiO_2)_2$	4.15	3.74	4.02	–
$cis-(TiO_2)_2$	4.03	3.27	3.63	3.49
$club-(TiO_2)_2$	3.22	2.78	2.99	2.78
$(TiO_2)_3$	3.82	2.97	4.09	4.09
$(TiO_3)_3$	3.49	3.34	3.78	3.69
$(TiO_2)_4$	–	–	4.32	4.14
$Ti(OH)_4$	6.10	5.64	5.93	6.51
$(TiO)_2(OH)_4$	4.77	4.72	5.02	4.88*

Table 3.6: First singlet excitation energies (optical gaps) for the titanium clusters. OT-BNL and B3LYP calculated quantities are provided, as well as CAM-B3LYP and EOM-CC from literature. All values were calculated using the Def2-TZVPP basis set, with the exception of * which was SVPD.[34]

3.6 Effect of optimally tuning γ

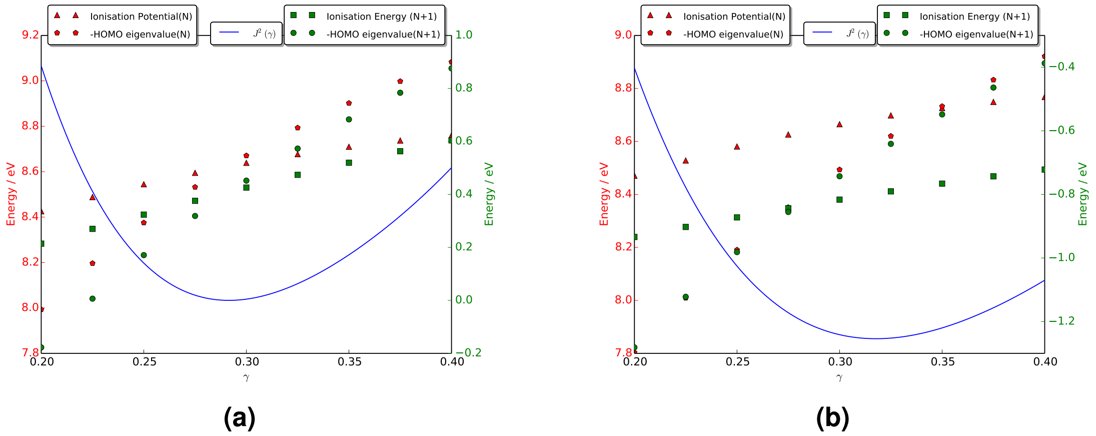
3.6.1 Reference molecules

By tuning γ one is hoping to minimise the equation

$$J^2(\gamma) = (\varepsilon^{\gamma}_{H(N)} + I^{\gamma}(N))^2 + (\varepsilon^{\gamma}_{H(N+1)} + I^{\gamma}(N+1))^2 \tag{3.2}$$

The hope is is if the value of the derivative discontinuity, Δ_{xc}, is small, then the the same value of γ will minimise both terms within the equation. This can be seen in the case on benzothiadiazole (Fig. 3.11 (a)). When the ionisation potentials and HOMO eigenvalues of the N and N+1 electron systems are plotted as functions of γ, we can see that in the case of a molecule that has displayed a very low value of the cost function, with $J(\gamma) = 0.001$ eV, the γ value at which the ionisation potential and HOMO eigenvalues of the N and N+1-electrons respectively are equal is almost the same.

Now to view the opposing case within the reference set whereby the value of $J(\gamma)$ was relatively high, for example benzothiazole, with $J(\gamma) = 0.165$ eV (Fig. 3.11 (b)). The N-electron system HOMO eigenvalue is equivalent to the ionisation potential at around $\gamma \approx 0.350$, while the equivalent criterion for the N+1-electron system is only fulfilled at $\gamma \approx 0.275$. This represents a large difference in optimal γ values for the two criteria, and this means Koopman's theorem can not be strongly enforced for both the N and N+1-electron HOMO eigenvalues simultaneously. What occurs is a 'best fit' of γ between the two criteria. Whilst this approach may provide successful predictions when the difference in optima γ for the two criteria is small, when the gap grows larger it is bound to have a negative effect on the predictive power of this method.



(a) (b)

Figure 3.11: Plots of ionisation potentials and HOMO eigenvalues for the N and N+1-electron systems as a function of γ calculated using OT-BNL for a) benzothiadiazole; and b) benzothiazole.

For the set of reference molecules one can also see that the effect the number of electrons in the system does not have a large effect on how the ionisation potentials and HOMO eigenvalues behave with respect to γ. In the case of benzothiazole and benzothiadiazole (Fig. 3.11) one can see that the behaviour of the ionisation potential compared to the HOMO eigenvalue for the N and N+1-electron systems very similar. This similarity of behaviour between the N and N+1-electron systems is likely what leads to similar values of optimal γ when enforcing Koopman's theorem for both systems.

3.6.2 Solvation Model

When comparing the same quantities for the solvated systems, more precisely for fluorene (see Fig. 3.12), it is clear that the occupation of the molecule is still not having a large effect on the behaviour of the ionisation potential and the HOMO eigenvalues with respect to γ. The respective functions still display similar slopes, except where we find the optimal γ is much lower than in the unsolvated systems. Koopman's theorem is valid at reasonable similar γ values for both systems, resulting in the low J^2 value seen in comparison to the titanium dioxide clusters, but still higher than those seen in the unsolvated systems.

3.6.3 Titanium dioxide clusters

For the reference molecules the values of $J^2(\gamma)$ seen were substantially smaller than those seen for the titanium dioxide clusters. In the case of $cis-(TiO_2)_2$ (Fig. 3.13 (a)) the values of optimal γ to enforce Koopman's theorem for the N-electron and N+1-electron systems are approximately 0.515 and 0.275 respectively. This represents a huge difference in optimal γ values for the two criteria and points to

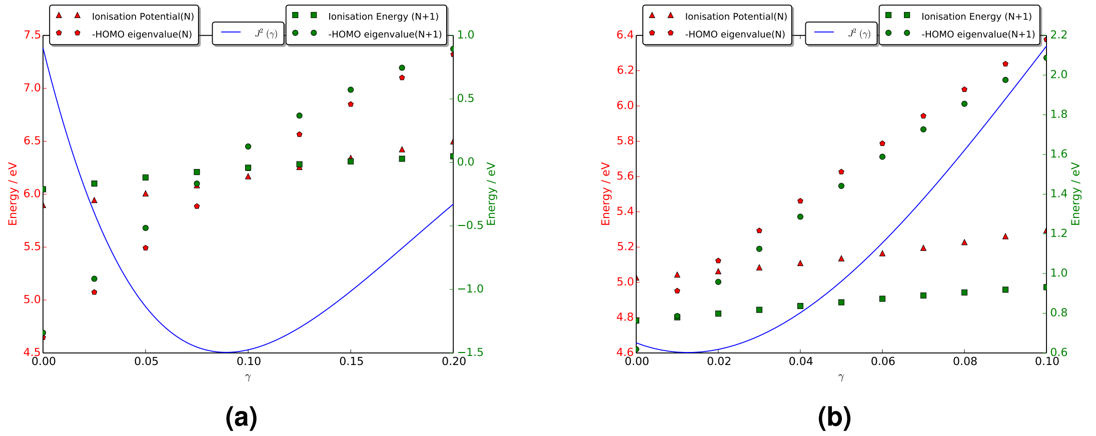


Figure 3.12: Plots of ionisation potentials and HOMO eigenvalues for the N and N+1-electron systems as a function of γ calculated using OT-BNL for fluorene solvated using COSMO in solvent of dielectric constant a) $\varepsilon_r = 2.0$; and b) $\varepsilon_r = 80.1$.

a large value of the derivative discontinuity. This large value of Δ_{xc} will alter the size of the fundamental gap, and is likely a severe blow to the predictive power of these functionals. One can only hope that the compromise found between the two criteria introduces an error to both the HOMO eigenvalue and the LUMO eigenvalue in such a way that they cancel when it comes to calculating the fundamental gap. Whether this is the case is unlikely as in both the $cis-(TiO_2)_2$ and $club-(TiO_2)_2$ systems, where this effect is most pronounced, the trend seen in fundamental gaps from G_0W_0 and $qsGW$ calculations is not replicated, as well as displaying substantially worse estimates of optical gaps.

The cluster $(TiO_2)_2(OH)_4$ exhibited interesting behaviour with respect to the other clusters (Fig. 3.13 (b)). The ionisation potential and the HOMO eigenvalue of the N-electron system behaved in much the same way as seen in other clusters. The the behaviour of those of the N+1 electron system on the other hand was unique. In all cases as γ has risen both the ionisation energy and the HOMO eigenvalue

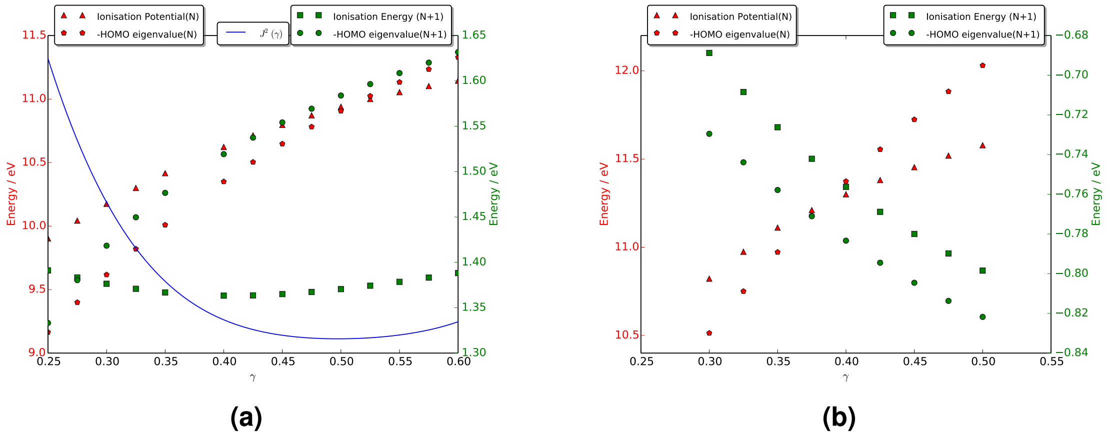


Figure 3.13: Plots of ionisation potentials and HOMO eigenvalues for the N and N+1-electron systems as a function of γ calculated using OT-BNL for a) $cis-(TiO_2)_2$; and b) $Ti(OH)_4$.

have risen, with the ionisation potential being higher in energy at low values of γ before being overtaken by the energy of the HOMO eigenvalue at higher values of γ. In the $(TiO_2)_2(OH)_4$ the behaviour is completely different. As the value of γ rises both the ionisation potential and the HOMO eigenvalue fall. They also do not cross, instead running parallel in the area close to the calculated γ_{opt}. The most likely reasoning for this behaviour is the S4 symmetry the neutral molecule contains. This symmetry means the molecule contains a triply degenerate molecular orbital. Addition of an electron means that this orbital is partially occupied, and this causes symmetry to be broken. The significantly different chemistries of the N-electron and N+1-electron states are likely the cause for this peculiar behaviour.

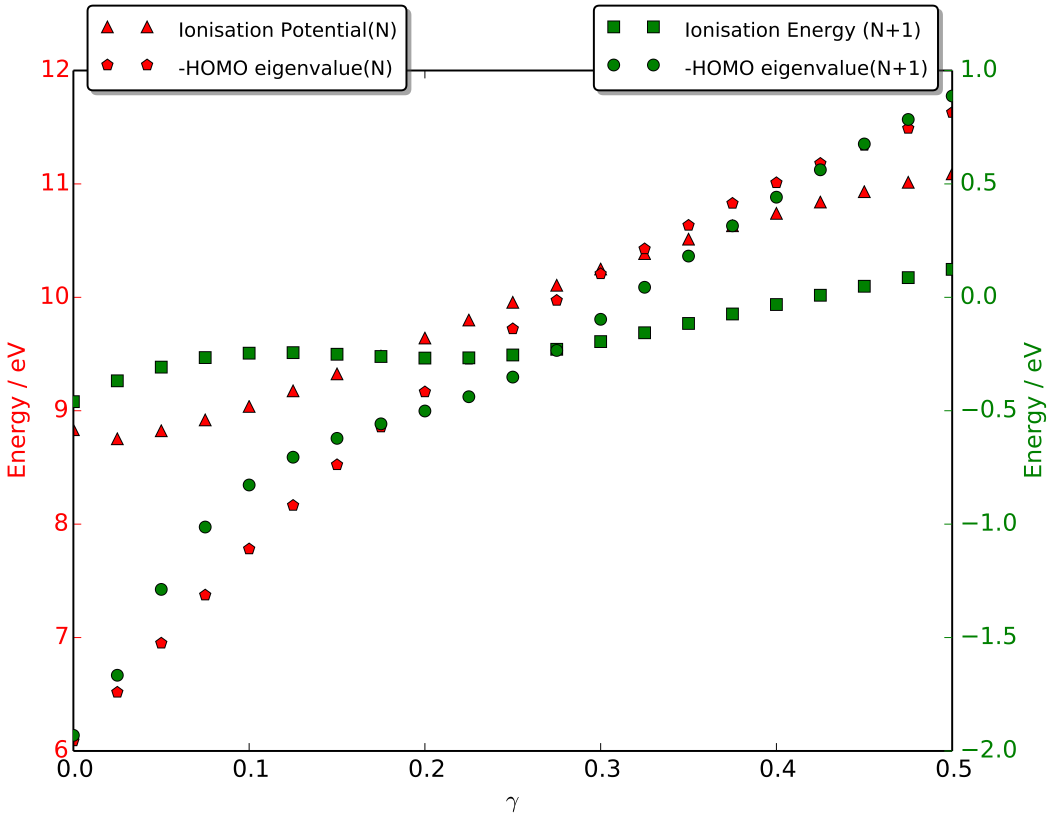


Figure 3.14: Plots of ionisation potentials and HOMO eigenvalues for the N and N+1-electron systems as a function of γ calculated using OT-BNL for $(TiO_2)(H_2O)_2$.

What is apparent is that in the case of the titanium dioxide clusters, whether the molecule is N or N+1 occupied now does exhibit an effect on the slope of the ionisation potentials and the HOMO eigenvalues with respect to γ. It is likely this is the reason that the optimal γ for enforcement of Koopman's theorem for the two systems is much more variable than seen in the reference set of molecules.

The $(TiO_2)_2(OH)_4$ cluster has been evaluated with γ taking steps of 0.025 between 0.0 and 0.5. The plots of ionisation potentials and HOMO eigenvalues are shown in Fig. 3.14. From the range of γ shown, it appears that there is likely only one minimum in the $J^2(\gamma)$ function. With the ionisation potentials and HOMO eigenvalues of the N and N+1 electron systems respectively diverging at both low and high γ.

4. *Concluding Remarks*

The main aim of the project has been accomplished, with a computer code that can optimise γ for range-separated hybrid functionals accurately and efficiently having been developed. The application of this to a variety of systems has provided a mix of interesting results. For the reference set of organic molecules the OT-RSH approach has been shown to be highly succesful, with its predictive power closely matching what is seen experimentally, and by other computational methods.

When applied to c-PCM to simulate solvation, the accuracy of the results appears to be compromised. That said there is a theoretical basis for this,[29] and a way of applying OT-RSH to solvated systems has been developed,[49] although the application of this technique to more systems is required.

Application to small titanium clusters was promising, with OT-RSH giving fundamental gap values that were closer to those calculated with $qsGW$ than G_0W_0, which has been shown to be more accurate for bulk materials.[5]. These results are by no means conclusive however and the trends were not strictly followed, although in the variation in optimal γ needed to enforce Koopman's theorem for the N and N+1 system we have a reasoning for this. More investigation is required to truly understand the accuracy and feasibility of the method. Optical gap studies were also inconclusive, again with more data required to draw solid conclusions.

Overall the OT-RSH method has performed well in comparison to other methods. It does appear to provide a useful method of calculating fundamental gaps from DFT and optical gaps from TDDFT. Only continuous investigation and the application of OT-RSH to a wide range of systems will give us a better understanding of the behaviour of this functional. The gap in optimal γ for enforcing Koopman's theorem in the N and N+1-electron system for some molecules provides an obstacle to the application of OT-RSH, but there are other variables with which these range-separated functionals that can be tuned. Perhaps the tuning of these variables will allow this difference to be minimised, and allow more accurate predictions will follow.

5. *References*

[1] Andreas Dreuw and Martin Head-Gordon. Single-Reference ab Initio Methods for the Calculation of Excited States of Large Molecules. *Chem. Rev.*, 105(11):4009–4037, November 2005.

[2] Hideo Sekino and Rodney J. Bartlett. Relativistic coupled cluster calculations on neutral and highly ionized atoms. *Int. J. Quantum Chem.*, 38(S24):241–244, March 1990.

[3] Jonathan C. Rienstra-Kiracofe, Gregory S. Tschumper, Henry F. Schaefer, Sreela Nandi, and G. Barney Ellison. Atomic and molecular electron affinities: photoelectron experiments and theoretical computations. *Chem. Rev.*, 102(1):231–282, January 2002.

[4] Andrew J. Williamson. Quantum Monte Carlo Calculations of Nanostructure Optical Gaps: Application to Silicon Quantum Dots. *Phys. Rev. Lett.*, 89(19), 2002.

[5] Cristiana Di Valentin, Silvana Botti, and Matteo Cococcioni. *First Principles Approaches to Spectroscopic Properties of Complex Materials*. Springer, September 2014. (p.123-124).

[6] Georgy Samsonidze, Manish Jain, Jack Deslippe, Marvin L. Cohen, and Steven G. Louie. Simple approximate physical orbitals for GW quasiparticle calculations. *Phys. Rev. Lett.*, 107(18):186404, October 2011.

[7] Giovanni Onida, Lucia Reining, and Angel Rubio. Electronic excitations: density-functional versus many-body Green's-function approaches. *Rev. Mod. Phys.*, 74(2):601–659, June 2002.

[8] X. Blase, C. Attaccalite, and V. Olevano. First-principles gw calculations for fullerenes, porphyrins, phtalocyanine, and other molecules of interest for organic photovoltaic applications. *Phys. Rev. B*, 83(11):115103, March 2011.

[9] P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Phys. Rev.*, 136:B864–B871, November 1964.

[10] W. Kohn. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.*, 140(4A):A1133–A1138, 1965.

[11] Leeor Kronik, Tamar Stein, Sivan Refaely-Abramson, and Roi Baer. Excitation Gaps of Finite-Sized Systems from Optimally Tuned Range-Separated Hybrid Functionals. *J. Chem. Theory Comput.*, 8:1515–1531, May 2012.

[12] John P. Perdew. Density-Functional Theory for Fractional Particle Number: Derivative Discontinuities of the Energy. *Phys. Rev. Lett.*, 49(23):1691–1694, 1982.

[13] C.-O. Almbladh. Exact results for the charge and spin densities, exchange-correlation potentials, and density-functional eigenvalues. *Phys. Rev. B*, 31(6):3231–3244, 1985.

[14] John P. Perdew. Comment on “Significance of the highest occupied Kohn-Sham eigenvalue”. *Phys. Rev. B*, 56(24):16021–16028, 1997.

[15] Andrew M. Teale, Frank De Proft, and David J. Tozer. Orbital energies and negative electron affinities from density functional theory: Insight from the integer discontinuity. *The Journal of Chemical Physics*, 129(4):044110, July 2008.

[16] Alex Borgoo, Andrew M Teale, and David J Tozer. Effective homogeneity of the exchange–correlation and non-interacting kinetic energy functionals under density scaling. *The Journal of chemical physics*, 136(3):034101, 2012.

[17] Axel D. Becke. A new mixing of Hartree-Fock and local density-functional theories. *The Journal of Chemical Physics*, 98:1372–1377, January 1993.

[18] A. Seidl, A. Görling, P. Vogl, J. A. Majewski, and M. Levy. Generalized Kohn-Sham schemes and the band-gap problem. *Phys. Rev. B*, 53:3764–3774, February 1996.

[19] J. P. Perdew. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23(10):5048–5079, 1981.

[20] Ester Livshits and Roi Baer. A well-tempered density functional theory of electrons in molecules. 9(23):2932–2941, June 2007.

[21] Roi Baer and Daniel Neuhauser. Density Functional Theory with Correct Long-Range Asymptotic Behavior. *Phys. Rev. Lett.*, 94(4):043002, February 2005.

[22] Tamar Stein, Helen Eisenberg, Leeor Kronik, and Roi Baer. Fundamental Gaps in Finite Systems from Eigenvalues of a Generalized Kohn-Sham Method. *Phys. Rev. Lett.*, 105(26):266802, December 2010.

[23] Sivan Refaely-Abramson, Roi Baer, and Leeor Kronik. Fundamental and excitation gaps in molecules of relevance for organic photovoltaics from an optimally tuned range-separated hybrid functional. *Phys. Rev. B*, 84(7):075144, August 2011.

[24] Jeng-Da Chai and Martin Head-Gordon. Systematic optimization of long-range corrected hybrid density functionals. *J Chem Phys*, 128(8):084106, February 2008.

[25] Oleg A. Vydrov and Gustavo E. Scuseria. Assessment of a long-range corrected hybrid functional. *J Chem Phys*, 125(23):234109, December 2006.

[26] Mary A. Rohrdanz, Katie M. Martins, and John M. Herbert. A long-range-corrected density functional that performs well for both ground-state properties and time-dependent density functional theory excitation energies, including charge-transfer excited states. *J Chem Phys*, 130(5):054112, February 2009.

[27] Andreas Klamt and GJGJ Schüürmann. Cosmo: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *Journal of the Chemical Society, Perkin Transactions 2*, (5):799–805, 1993.

[28] Darrin M York and Martin Karplus. A smooth solvation potential based on the conductor-like screening model. *The Journal of Physical Chemistry A*, 103(50):11060–11079, 1999.

[29] Thiago B. de Queiroz and Stephan Kümmel. Charge-transfer excitations in low-gap systems under the influence of solvation and conformational disorder: exploring range-separation tuning. *J Chem Phys*, 141(8):084303, August 2014.

[30] Shaohui Zheng, Eitan Geva, and Barry D. Dunietz. Solvated Charge Transfer States of Functionalized Anthracene and Tetracyanoethylene Dimers: A Computational Study Based on a Range Separated Hybrid Functional and Charge Constrained Self-Consistent Field with Switching Gaussian Polarized Continuum Models. *J. Chem. Theory Comput.*, 9(2):1125–1131, February 2013.

[31] Martijn A Zwijnenburg. Personal correspondence.

[32] Zheng-wang Qu and Geert-Jan Kroes. Theoretical study of the electronic structure and stability of titanium dioxide clusters (tio2) n with n= 1- 9. *The Journal of Physical Chemistry B*, 110(18):8998–9007, 2006.

[33] Enrico Berardo, Han-Shi Hu, Karol Kowalski, and Martijn A Zwijnenburg.

Coupled cluster calculations on tio2 nanoclusters. *The Journal of chemical physics*, 139(6):064313, 2013.

[34] Enrico Berardo, Han-Shi Hu, Stephen A. Shevlin, Scott M. Woodley, Karol Kowalski, and Martijn A. Zwijnenburg. Modeling Excited States in TiO2 Nanoparticles: On the Accuracy of a TD-DFT Based Description. *J. Chem. Theory Comput.*, 10(3):1189–1199, March 2014.

[35] Enrico Berardo, Han-Shi Hu, Hubertus J. J. van Dam, Stephen A. Shevlin, Scott M. Woodley, Karol Kowalski, and Martijn A. Zwijnenburg. Describing Excited State Relaxation and Localization in TiO2 Nanoparticles Using TD-DFT. *J. Chem. Theory Comput.*, 10(12):5538–5548, December 2014.

[36] Shanmugasundaram Sakthivel and Horst Kisch. Daylight Photocatalysis by Carbon-Modified Titanium Dioxide. *Angewandte Chemie International Edition*, 42(40):4908–4911, October 2003.

[37] Fang Han, Venkata Subba Rao Kambala, Madapusi Srinivasan, Dharmarajan Rajarathnam, and Ravi Naidu. Tailored titanium dioxide photocatalysts for the degradation of organic dyes in wastewater treatment: A review. *Applied Catalysis A: General*, 359(1–2):25–40, May 2009.

[38] Brian O'regan and M Grfitzeli. A low-cost, high-efficiency solar cell based on dye-sensitized. *nature*, 353(6346):737–740, 1991.

[39] Gopal K Mor, Karthik Shankar, Maggie Paulose, Oomman K Varghese, and Craig A Grimes. Use of highly-ordered tio2 nanotube arrays in dye-sensitized solar cells. *Nano letters*, 6(2):215–218, 2006.

[40] Akira Fujishima, Tata N. Rao, and Donald A. Tryk. Titanium dioxide photocatalysis. *Journal of Photochemistry and Photobiology C: Photochemistry Reviews*, 1(1):1–21, June 2000.

[41] T Watanabe, A Nakajima, R Wang, M Minabe, S Koizumi, A Fujishima, and K Hashimoto. Photocatalytic activity and photoinduced hydrophilicity of titanium dioxide coated glass. *Thin Solid Films*, 351(1–2):260–263, August 1999.

[42] Ivan P. Parkin and Robert G. Palgrave. Self-cleaning coatings. *Journal of Materials Chemistry*, 15(17):1689–1695, 2005.

[43] Erich Runge and Eberhard KU Gross. Density-functional theory for time-dependent systems. *Physical Review Letters*, 52(12):997, 1984.

[44] David J Tozer. Relationship between long-range charge-transfer excitation energy error and integer discontinuity in kohn–sham theory. *The Journal of chemical physics*, 119(24):12697–12699, 2003.

[45] Andreas Dreuw, Jennifer L Weisman, and Martin Head-Gordon. Long-range charge-transfer excited states in time-dependent density functional theory require non-local exchange. *The Journal of chemical physics*, 119(6):2943–2946, 2003.

[46] Robert S Mulliken. Molecular compounds and their spectra. ii. *Journal of the American Chemical Society*, 74(3):811–824, 1952.

[47] Chengteh Lee, Weitao Yang, and Robert G Parr. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. *Physical review B*, 37(2):785, 1988.

[48] Darrin M. York and Martin Karplus. A Smooth Solvation Potential Based on the Conductor-Like Screening Model. *J. Phys. Chem. A*, 103(50):11060–11079, December 1999.

[49] Thiago B. de Queiroz and Stephan Kümmel. Tuned range separated hybrid functionals for solvated low bandgap oligomers. *J Chem Phys*, 143(3):034101, July 2015.

Appendices

A. Functions definition of γ optimistation code

```
import os
import copy
import errno
import shutil
import numpy as np


def readnwfile(master_file):
    # reads a file and returns it as a list
    with open(master_file + '.nw') as f:
        nw_file_lines = f.readlines()
    return nw_file_lines


def Environment():
    # Stores the environment as global variables for use with other functions
    home = os.path.expanduser('~')
    global cwd
    cwd = os.getcwd()
    global scratch
    scratch = home + '/Scratch' + cwd[len(home):]



def mkdir_p(path):
    # makes a directory tree if it does not already exist
    try:
        os.makedirs(path)
    except OSError as exc:
        if exc.errno == errno.EEXIST and os.path.isdir(path):
            pass
        else:
            raise


def ChargedSpecies(nw_lines, charge):
    # This function alters the charge and mult values for the nw input file.
    # A deep copy is required otherwise both lists are altered
    new_nw_lines = copy.deepcopy(nw_lines)
    for i in range(len(new_nw_lines)):
        if new_nw_lines[i] == 'charge␣0\n':
            new_nw_lines[i] = 'charge␣%s\n' %charge
        if new_nw_lines[i] == 'dft\n':
            new_nw_lines[i] = 'dft\n␣␣mult␣2\n'
    return new_nw_lines



def CreateInputFile(master_file, nw_file_lines, gamma, opt):
    # This Function creates the input file with a certain gamma
    # It also creates a cationic and anionic state for calculations
    # the gammafile variable is to name files without periods
    # Before using this function the masters input file must be readlines() to a variable
    if opt:
        gammafile = 'OPT'
    else:
        gammafile = int(gamma * 1000)
    for i in range(len(nw_file_lines)):
        # Replace the gamma value
        if nw_file_lines[i].startswith('␣␣cam'):
            nw_file_lines[i] = '␣␣cam␣%s␣cam_alpha␣0.0␣cam_beta␣1.0\n' %gamma

    # Create the Directory and input file for netural
    os.mkdir('%s_0_%s' %(master_file, gammafile))
    f = open('%s_0_%s/%s_0_%s.nw' %(master_file, gammafile, master_file, gammafile), 'w')
    f.writelines(nw_file_lines)
    f.close()

    #Create Dir and input file for cation
    os.mkdir('%s_p1_%s' %(master_file, gammafile))
    g = open('%s_p1_%s/%s_p1_%s.nw' %(master_file, gammafile, master_file, gammafile), 'w')
    g.writelines(ChargedSpecies(nw_file_lines, '+1'))
    g.close()

    # Create Dir and input file for anion
    os.mkdir('%s_m1_%s' %(master_file, gammafile))
    h = open('%s_m1_%s/%s_m1_%s.nw' %(master_file, gammafile, master_file, gammafile), 'w')
    h.writelines(ChargedSpecies(nw_file_lines, '-1'))
    h.close()



def ErrorOutputLog(working_file, type):
    #Creates a file with a desired suffix, in the scratch space
    open('%s/%s/%s.%s' %(scratch, working_file, working_file, type), 'w').close()

def WorkingFile(master_file, gammafile, charge):
    # returns a working file variable
    return '%s_%s_%s' %(master_file, charge, gammafile)


def FileEnvironment(working_file):
    # Requires Environment() to be run prior
    # Creates the e, o, and log files in scratch space and sets up environment
    #Create Scratch Directory
    mkdir_p('%s/%s' %(scratch, working_file))
    # Copy .nw file to scratch as .nw.tmp
    shutil.copy2('%s/%s.nw' %(working_file, working_file),
                 '%s/%s/%s.nw.tmp' %(scratch, working_file, working_file))
    # Create Error, Output and log files
    ErrorOutputLog(working_file, 'e')
    ErrorOutputLog(working_file, 'o')
    ErrorOutputLog(working_file, 'log')

def CopyLog(working_file):
    # Copies the .log file from the scratch space to the permanent memory
    shutil.copy2('%s/%s/%s.log' %(scratch, working_file, working_file),
                 '%s/%s.log' %(working_file, working_file))


def MakeShell(working_file):
    # Creates a .sh script and submits it to the cluster
    # Input varables can be changed to change number of cores and clock time

    # create the nw.sh file
    f = open('%s/%s/%s.nw.sh' %(cwd, working_file, working_file), 'w')

    # Tells to excecute in bash
    f.write('#!/bin/bash␣-l\n')
    # Again forces SGE to excecute in bash
    f.write('#$␣-S␣/bin/bash␣\n')
    #set work to current working directory
    f.write('#$␣-cwd␣\n')
    # Request wallclock time, can be altered
    f.write('#$␣-l␣h_rt=5:59:0␣\n')
    # Request 12 cores, can be altered
    f.write('#$␣-pe␣mpi␣12␣\n')
    # Request 1GB of RAM per core
    f.write('#$␣-l␣mem=2G␣\n')
    #set error output file
    f.write('#$␣-e␣%s/%s/%s.e␣\n' %(scratch, working_file, working_file))
    # set output file
    f.write('#$␣-o␣%s/%s/%s.o␣\n' %(scratch, working_file, working_file))
    #run n nwchem
    f.write('#$␣-N␣%s\n' %JobNameGen(working_file))
    #load modules
    f.write("module␣load␣python/2.7.9\n")
    f.write("module␣load␣nwchem/6.5-r26243/intel-2015-update2\n")
    f.write("module␣list\n")
    # change to scratch directory
    f.write('cd␣%s/%s␣\n' %(scratch, working_file))
    # Run file in nwchem and save output to .log file
    f.write('gerun␣nwchem␣%s.nw.tmp␣>>␣%s.log\n' %(working_file, working_file))
    # change back to working directory
    f.write('cd␣%s␣\n' %cwd)
    f.close()
    # Submit job to SGE
    os.system('qsub␣%s/%s.nw.sh' %(working_file, working_file))

def SetUp(master_file, gamma, opt):
    # This function creates the scratch space and the shell and submits it for a masterfile
    # It works at one value of gamma and submits 3 jobs for charge -1, 0, +1
    # neutral species
    if opt:
        filegamma = 'OPT'
    else:
        filegamma = int(gamma * 1000)
    working_file = WorkingFile(master_file, filegamma, '0')
    FileEnvironment(working_file)
    MakeShell(working_file)
    #negative species
    working_file = WorkingFile(master_file, filegamma, 'm1')
    FileEnvironment(working_file)
    MakeShell(working_file)
    #positive species
    working_file = WorkingFile(master_file, filegamma, 'p1')
    FileEnvironment(working_file)
    MakeShell(working_file)

def CompSetUp(master_file, gamma, opt):
    # Combines the creation of the input file with the SetUp function to create a job and submit
    nw_lines = readnwfile(master_file)
    CreateInputFile(master_file, nw_lines, gamma, opt)
    SetUp(master_file, gamma, opt)


def ArrayPopulator(working_file, gamma_table, homo_vector, neutral, n):
    # This function scans the .log files and extracts the useful information to an array
    # It extracts the DFT Energy, and HOMO energy
    # It also finds the LUMO energy of the neutral species
    os.chdir(working_file)
    searchfile = open(working_file + '.log', "r")
    for line in searchfile:
        if "Total␣DFT␣energy" in line:
            DFT_energy_string = line
            gamma_table[n,1] = DFT_energy_string[31:49]
        #Assigns the Total DFT Energy to index 1 of the gamma table

        if "Vector␣␣␣%s␣␣Occ=1" %homo_vector in line:
            HOMO_String = line
            if HOMO_String[34] == '-':
                gamma_table[n,2] = HOMO_String[34:47].replace('D', 'E')
            else:
                gamma_table[n,2] = HOMO_String[35:47].replace('D', 'E')

        if "Vector␣␣␣%s␣␣Occ=2" %homo_vector in line:
            HOMO_String = line
            if HOMO_String[34] == '-':
                gamma_table[n,2] = HOMO_String[34:47].replace('D', 'E')
            else:
                gamma_table[n,2] = HOMO_String[35:47].replace('D', 'E')

    #These two loops assign the HOMO energy to index 2 of the gamma array,
    #in the closed shell system it is doubly occupied, in the open shell singlely

        if neutral:
            if "Vector␣␣␣%s␣␣Occ=0" %(homo_vector + 1) in line:
                LUMO_String = line
                if LUMO_String[34] == '-':
                    gamma_table[n,3] = LUMO_String[34:47].replace('D', 'E')
                else:
                    gamma_table[n,3] = LUMO_String[35:47].replace('D', 'E')

    #As above for the LUMO, although is only required for the molecule in its neutral state
    searchfile.close()
    os.chdir('..')


def JSqTable(master_file, gamma_values, HOMO):

    # This function implements the ArrayPopulator function to fill a J^2 Array
    # It then returns an array contains the J^2 values with gamma and ionisation energies
    data_points = len(gamma_values)
    gamma_table_minus = np.zeros((data_points , 3))
    gamma_table_neutral = np.zeros((data_points , 4))
    gamma_table_plus = np.zeros((data_points , 3))
    gamma_J_minimise = np.zeros((data_points , 4))

    """ The indicies for the gamma_tables are:

        0 - gamma value
        1 - DFT Energy
        2 - HOMO energy
        3 - LUMO energy

        and for the gamma_J_minimise table:

        0 == a.min() - gamma value
        1 - Ionisation Energy for N e-
        2 - Ionistation energy for N+1 e-
        3 - J value of function to be minimised
    """
    print gamma_values
    for i in range(len(gamma_values)):
        gamma = float('%.4f' % gamma_values[i])
        gamma_table_minus[i, 0] = gamma
        gamma_table_neutral[i, 0] = gamma
        gamma_table_plus[i, 0] = gamma
        # Populate for neutral
        working_file = WorkingFile(master_file, int(gamma * 1000), 0)
        ArrayPopulator(working_file, gamma_table_neutral, HOMO, True, i)

        # Populate for negative
        working_file = WorkingFile(master_file, int(gamma * 1000), 'm1')
        ArrayPopulator(working_file, gamma_table_minus, HOMO + 1, False, i)

        # Populate for positive
        working_file = WorkingFile(master_file, int(gamma * 1000), 'p1')
        ArrayPopulator(working_file, gamma_table_plus, HOMO, False, i)


    for n in range(data_points):
        # similar code to above to populate gamma values
        gamma_J_minimise[n,0] = gamma_table_plus[n,0]
        gamma_J_minimise[n,1] = gamma_table_plus[n,1] - gamma_table_neutral[n,1]
        #gives the ionisation energy of N electron system
        gamma_J_minimise[n,2] = gamma_table_neutral[n,1] - gamma_table_minus[n,1]
        #gives the ionisation energy of the N+1 electron system
        gamma_J_minimise[n,3] = (gamma_table_neutral[n,2] + gamma_J_minimise[n,1]) ** 2 +
        (gamma_table_minus[n,2] + gamma_J_minimise[n,2]) ** 2 #calculates the J^2
        n += 1
    return gamma_J_minimise



def JSqTable_final(master_file, gamma_values, HOMO):
    # This function implements the ArrayPopulator function to fill a J^2 Array
    # It then returns an array contains the J^2 values with gamma and ionisation energies
    data_points = len(gamma_values)
    gamma_table_minus = np.zeros((data_points , 3))
    gamma_table_neutral = np.zeros((data_points , 4))
    gamma_table_plus = np.zeros((data_points , 3))
    gamma_J_minimise = np.zeros((data_points , 4))

    """ The indicies for the gamma_tables are:
        0 - gamma value
        1 - DFT Energy
        2 - HOMO energy
        3 - LUMO energy

        and for the gamma_J_minimise table:

        0 == a.min() - gamma value
        1 - Ionisation Energy for N e-
        2 - Ionistation energy for N+1 e-
        3 - J value of function to be minimised
    """
    print gamma_values
    for i in range(len(gamma_values)):
        gamma = float('%.4f' % gamma_values[i])
        gamma_table_minus[i, 0] = gamma
        gamma_table_neutral[i, 0] = gamma
        gamma_table_plus[i, 0] = gamma

        if len(str(gamma)) == 6:
            working_file = WorkingFile(master_file, 'OPT', 0)
            ArrayPopulator(working_file, gamma_table_neutral, HOMO, True, i)

            # Populate for negative
            working_file = WorkingFile(master_file, 'OPT', 'm1')
            ArrayPopulator(working_file, gamma_table_minus, HOMO + 1, False, i)

            # Populate for positive
            working_file = WorkingFile(master_file, 'OPT', 'p1')
            ArrayPopulator(working_file, gamma_table_plus, HOMO, False, i)
        else:
            # Populate for neutral
            working_file = WorkingFile(master_file, int(gamma * 1000), 0)
            ArrayPopulator(working_file, gamma_table_neutral, HOMO, True, i)

            # Populate for negative
            working_file = WorkingFile(master_file, int(gamma * 1000), 'm1')
            ArrayPopulator(working_file, gamma_table_minus, HOMO + 1, False, i)

            # Populate for positive
            working_file = WorkingFile(master_file, int(gamma * 1000), 'p1')
            ArrayPopulator(working_file, gamma_table_plus, HOMO, False, i)


    for n in range(data_points):
        # similar code to above to populate gamma values
        gamma_J_minimise[n,0] = gamma_table_plus[n,0]
        gamma_J_minimise[n,1] = gamma_table_plus[n,1] - gamma_table_neutral[n,1]
        #gives the ionisation energy of N electron system
        gamma_J_minimise[n,2] = gamma_table_neutral[n,1] - gamma_table_minus[n,1]
        #gives the ionisation energy of the N+1 electron system
        gamma_J_minimise[n,3] = (gamma_table_neutral[n,2] + gamma_J_minimise[n,1]) ** 2 +
        (gamma_table_minus[n,2] + gamma_J_minimise[n,2]) ** 2 #calculates the J^2
        n += 1
    return gamma_J_minimise, gamma_table_minus, gamma_table_neutral, gamma_table_plus


def FindHOMO(working_file):
    # Searches through a .log file and returns the vector number of the HOMO
    searchfile = open(working_file + '/' + working_file + '.log', "r")
    for line in searchfile:
        searchline = line
        if searchline.startswith('␣Vector␣␣␣'):
            if searchline[18:21] == '0.0':
                HOMO = int(searchline[10:12]) - 1
                searchfile.close()
                return HOMO

def GammaEnv(master_file):
    # Returns a list of all gamma for the directory
    dirs = os.listdir('.')
    gammas = []
    for item in dirs:
        if item.startswith('%s_0_' %master_file):
            if item[-2] == '_':
                gammas.append(float(item[-1:])/1000)
            elif item[-3] == '_':
                gammas.append(float(item[-2:])/1000)
            elif item[-4] == '_' and item[-3:] != 'OPT':
                gammas.append(float(item[-3:])/1000)
    return gammas

def MasterFileName():
    dirs = os.listdir('.')
    for item in dirs:
        if item.endswith('.nw'):
            master_file = item[:-3]
    return master_file


def JobNameGen(working_file):
    notcomplete = True
    for i in range(len(working_file)):
        if working_file[-i] == '_':
            filegamma = working_file[-i+1:]
            if working_file[-i-1] == '0':
                charge = 'n'
                name = working_file[:-i-2]
            else:
                charge = working_file[-i-2]
                name = working_file[:-i-3]
            break

    i = 0
    if len(name) <= 5:
        jobname = name + filegamma  + charge
    else:
        while (len(name) - i + 1) % 4 != 0:
            i += 1
        jobname = name[:len(name) - i + 2: (len(name) - i + 1) / 4] + filegamma  + charge
    return jobname
```

B. *Part 1 of γ optimistation code*

```
import NWFunction_v2 as nwf
import os
# This script is part 1 of the DFT gamma optimisation
# It should be run in a directory with the .nw input file
# It will then create and run the initial jobs as laid out below

# This script creates the job environment for an nw file for
#gamma 0, 0.1, 0.2, 0.3, 0.4 0.5
# It also creates cationic, anionic and neutral nw input files
# It then submits them all to the cluster and works in the Scratch space

# Save environmental conditions to variables for other functions
os.chdir('..')
nwf.Environment()
gamma_values = []

master_file = nwf.MasterFileName()

# Make list of gamma values
for i in range(0,501,100):
    gamma_values.append(float(i)/1000)

# Use CompSetUp function to initialise all jobs
for gamma in gamma_values:
    nwf.CompSetUp(master_file, gamma, False)
```

C. *Part 2 of γ optimistation code*

```
import NWFunction_v2 as nwf
import numpy as np
import os
# This script is part 2 of the DFT gamma optimisation.
# It will first copy over the .log files from the scratch space.
# It will then extract the Total DFT Energy and HOMO values from the .log files
# It will use these to find the J^2 value for each gamma
# It will then create new jobs at steps of .025 gamma from -100 to +100
#around the lowest gamma value

os.chdir('..')
# Create list containing old gamma values
old_gamma = []

# Ask the user what file is being operated on
master_file = nwf.MasterFileName()
# Find environmental conditions
nwf.Environment()

# Make list of gamma values
for i in range(0,501,100):
    old_gamma.append(float(i)/1000)

for gamma in old_gamma:
    filegamma = int(1000 * gamma)
    # Copy .log files from scratch space for all gamma values
    working_file = nwf.WorkingFile(master_file, filegamma, '0')
    nwf.CopyLog(working_file)
    working_file = nwf.WorkingFile(master_file, filegamma, 'm1')
    nwf.CopyLog(working_file)
    working_file = nwf.WorkingFile(master_file, filegamma, 'p1')
    nwf.CopyLog(working_file)


# Now we must find a value for vector of the HOMO

working_file = nwf.WorkingFile(master_file, 100, '0')
HOMO_vector = nwf.FindHOMO(working_file)

# Now to iterate through the .log files and find the minimum J^2
gamma_J_minimise = np.zeros((len(old_gamma), 4))
gamma_J_minimise, a, b, c = nwf.JSqTable_final(master_file, old_gamma, HOMO_vector)

print a
print b
print c
print gamma_J_minimise
# Save the J^2 values to a list, find the index of the minimum values and
#then find the corresponding gamma
J2 = gamma_J_minimise[:,3]
index = np.argmin(J2)
gamma_min = old_gamma[index]
new_gamma = []

# Find a new list of gamma values to perform jobs on, steps of .025 gamma
#avoiding repeats of those divisible by 0.1
if gamma_min - 0.1 <= 0:
    new_gamma = [0.025, 0.05, 0.075, 0.125, 0.15, 0.175]
else:
    for i in range((int(gamma_min * 1000) - 75), int(gamma_min * 1000 + 76), 25):
        if i % 100 != 0:
            new_gamma.append(float(i)/1000)

# Submit new jobs to the cluster

print new_gamma
for gamma in new_gamma:
    nwf.CompSetUp(master_file, gamma, False)
```

D. *Part 3 of γ optimistation code*

```
import NWFunction_v2 as nwf
import numpy as np
import numpy.polynomial.polynomial as poly
import os

os.chdir('..')
# Find environmental conditions
nwf.Environment()
# Ask the user what file is being operated on
master_file = nwf.MasterFileName()
gammas = nwf.GammaEnv(master_file)
# Make list of gamma values that have not already been copied from
#scratch - Ones not divisible by 0.1
new_gammas = []
for item in gammas:
    if int(item * 1000) % 100 != 0:
        new_gammas.append(item)

# Copy said gammas .log files from scratch space
for gamma in new_gammas:
    filegamma = int(1000 * gamma)
    # Copy .log files from scratch space for all gamma values
    working_file = nwf.WorkingFile(master_file, filegamma, '0')
    nwf.CopyLog(working_file)
    working_file = nwf.WorkingFile(master_file, filegamma, 'm1')
    nwf.CopyLog(working_file)
    working_file = nwf.WorkingFile(master_file, filegamma, 'p1')
    nwf.CopyLog(working_file)

# Adds the gammas that are divisible by 0.1 to the new gammas list
min_gamma = min(new_gammas)
new_gammas.append(min_gamma - 0.025)
new_gammas.append(min_gamma + 0.075)
new_gammas.append(min_gamma + 0.175)
new_gammas.sort()
# Finds the HOMO vector of the gamma 0.1 file (arbitrarily chosen)
working_file = nwf.WorkingFile(master_file, 100, '0')
HOMO_vector = nwf.FindHOMO(working_file)

# Store data from JSqTable in an array
gamma_J_minimise = np.zeros((len(new_gammas) , 4))
gamma_J_minimise = nwf.JSqTable(master_file, new_gammas, HOMO_vector)
J2 = gamma_J_minimise[:,3]
index = np.argmin(J2)
gamma_min = new_gammas[index]
# Only data from the 5 points surrounding the minimum are used to
#fit the polyniomial to increase accuracy at the minimum
# Here we check if the point 0.05 above the min is already calculated,
#and if it is take from there, or move 0.025 back and plot from there
x = gamma_J_minimise[index - 2:index + 3,0]
y = gamma_J_minimise[index - 2:index + 3,3]


# Use a 4th order polynomial to fit the data
# Then find the roots of the differential to find the minimum
coefs = poly.polyfit(x, y, 4)
diff = [coefs[4] * 4, coefs[3] * 3, coefs[2] * 2, coefs[1]]
roots = np.roots(diff)
print roots

opt_gamma = 0
```

```
# Extract the roots that are real, then find the closest root to
#the gamma value with the lowest calculated value of j^2
for num in roots:
    if num.imag == 0:
        if abs(num.real - gamma_min) < abs(opt_gamma - gamma_min):
            opt_gamma = num.real

print opt_gamma
# Submit a file with the optimised gamma with gamma to 3dp.
nwf.CompSetUp(master_file, float("{0:.4f}".format(opt_gamma)), True)

# OPTIONAL - edit out if needed, also submits jobs for the gamma
#values immediatly surroundingthe opt_gamma to check if J^2 is minimised
opt_gamma = float("{0:.3f}".format(opt_gamma))
gamma_check = [opt_gamma - 0.002, opt_gamma - 0.001, opt_gamma,
               opt_gamma + 0.001, opt_gamma + 0.002]
for gamma in gamma_check:
    if int(gamma * 1000) % 25 != 0:
        nwf.CompSetUp(master_file, gamma, False)
```

E. *Analysis of γ optimistation code*

```
# This code is used to interpret the results from Part 3
# It save important values for the molecule into a text file
# it also makes two graphs, one of the fitting procedure
# and one of the relation between HOMO eigenvalues, ionisation energies and
# gamma
# all this is stored in a folder marker Analysis_*moleculename*

import os
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import NWFunction_v2 as nwf
import numpy.polynomial.polynomial as poly

os.chdir('..')

nwf.Environment()
master_file = nwf.MasterFileName()

allgammas = nwf.GammaEnv(master_file)

filegammas = ['OPT']
for gamma in allgammas:
    filegammas.append(int(1000*gamma))

working_file = nwf.WorkingFile(master_file, '100', 0)
# Get HOMO vactor from neutral, gamma = 0.1 file
HOMO_vector = nwf.FindHOMO(working_file)

filegammas.remove('OPT')
polyfitgammas = []

# Find a all gammas from the Part2 and add to list
for a in filegammas:
    if a % 100 != 0 and a % 25 == 0:
        polyfitgammas.append(float("{0:.3f}".format(float(a)/1000)))
min_gamma = min(polyfitgammas)
polyfitgammas.append(float("{0:.3f}".format(min_gamma - 0.025)))
polyfitgammas.append(float("{0:.3f}".format(min_gamma + 0.075)))
polyfitgammas.append(float("{0:.3f}".format(min_gamma + 0.175)))
polyfitgammas.sort()
print polyfitgammas

#calculate J^2 values for this lise
polyJ, polyminus, polyneutral, polyplus = nwf.JSqTable_final(master_file,
                                                             polyfitgammas, HOMO_vector)

print polyJ

# use min index to create new lists of the gamma values used in the fitting
# algorithm in part 3
J2 = polyJ[:,3]
index = np.argmin(J2)
x = polyJ[index - 2: index + 3, 0]
y = polyJ[index - 2: index + 3, 3]
gamma_min = polyJ[index,0]

# Fit to a 4th order polynomial
coefs = poly.polyfit(x, y, 4)
diff = [coefs[4] * 4, coefs[3] * 3, coefs[2] * 2, coefs[1]]
roots = np.roots(diff)
```

```
print coefs
print roots
opt_gamma = 0
# Extract the roots that are real, then find the closest root to the
#gamma value with the lowest calculated value of j^2
for num in roots:
    if num.imag == 0:
        if abs(num.real - gamma_min) < abs(opt_gamma - gamma_min):
            opt_gamma = num.real

gamma_check = []
opt_gamma = float("{0:.4f}".format(opt_gamma))
print opt_gamma
print gamma_check
gamma_check.append(opt_gamma)

# calculate values of importance for opt gamma

gamma_J_minimise, minus, neutral, plus = nwf.JSqTable_final(master_file,
                                                      gamma_check, HOMO_vector)
print minus
print neutral
print plus
print gamma_J_minimise
HOMO_LUMO = neutral[-1,3]-neutral[-1,2]
#make directory for graphs and data
os.mkdir('Analysis_%s' %master_file)

# save important quantities to text file. Listed as gamma, HOMO(N), I(N),
# LUMO (N+1), I(N+1), J^2

array = np.array((gamma_J_minimise[-1,0], neutral[-1,2], gamma_J_minimise[-1,1],
         neutral[-1,3], gamma_J_minimise[-1,2], gamma_J_minimise[-1,3]))

np.savetxt('Analysis_%s/Analysis_%s' %(master_file, master_file), array)

# create a plot of the minimisation procedure

x_new = np.arange(x[0] -0.011, x[-1] + 0.0111, 0.002)
ffit = poly.polyval(x_new, coefs)
plt.figure(1)
plt.plot(x_new, ffit, 'r-')
plt.xlim((x[0] - 0.01, x[-1] +0.01))
plt.xlabel('$\gamma$')
plt.ylabel('$J^2(\gamma)/E_h^2$')
plt.plot(x, y, 'ro')

plt.plot(gamma_J_minimise[-1,0], gamma_J_minimise[-1,3] , 'bo')

plt.savefig("Analysis_%s/%s.pdf" %(master_file, master_file))

# Create a plot of HOMO eigenvalues and I for N and N+1 e system vs gamma

x_dif = np.arange(polyJ[0,0], polyJ[-1,0]+0.001, 0.002)
ffit_dif = poly.polyval(x_dif, coefs)
plt.figure(2)
fig, ax1 = plt.subplots()
ax1.plot(polyJ[:,0], polyJ[:,1]*27.211396132, 'r^',
         label='Ionisation␣Potential(N)')
ax1.plot(polyJ[:,0], -polyneutral[:,2]*27.211396132, 'rp',
         label='-HOMO␣eigenvalue(N)')
ax1.tick_params('y', colors='r')
ax1.set_ylabel('Energy␣/␣eV', color = 'r')

ax2 = ax1.twinx()
ax2.plot((polyJ[:,0]), (polyJ[:,2]*27.211396132) , 'gs',
         label='Ionisation␣Energy␣(N+1)')
ax2.plot((polyJ[:,0]), -(polyminus[:,2]*27.211396132) , 'go',
         label='-HOMO␣eigenvalue(N+1)')
ax2.tick_params('y', colors='g')
ax2.set_ylabel('Energy␣/␣eV', color = 'g')

ax3 = ax1.twinx()
ax3.plot(x_dif, ffit_dif, 'b-', label='$J^2(\gamma)$')
ax3.set_yticks([])
ax3.set_xlim([polyJ[0,0], polyJ[-1,0]])
```

```
ax1.set_xlabel('$\gamma$', color = 'k')

legend = (
            ax1.legend(loc='lower␣left', bbox_to_anchor=(0,0.95), fancybox=True,
                    shadow=True),
            ax2.legend(loc='lower␣right', bbox_to_anchor=(1,0.95),
                       fancybox=True, shadow=True),
            ax3.legend(loc='lower␣center', bbox_to_anchor=(0.5,0.95),
                       fancybox=True, shadow=True)
         )

for a in legend:
    for label in a.get_texts():
        label.set_fontsize('small')
plt.savefig("Analysis_%s/gamma_breakdown_%s.pdf" %(master_file, master_file))
```

F. *Geometries*

Reference Molecules

C	1.24466441	-0.04169163	-0.00000523
C	-1.24466487	-0.04169212	-0.00000089
S	-0.00000005	-1.24297642	-0.00001925
C	0.71470471	1.23933435	0.00000525
C	-0.71470511	1.23933421	0.00000739
H	1.34451518	2.15739977	0.00000346
H	-1.34451614	2.15739924	0.00000773
H	2.31181247	-0.34842016	-0.00000433
H	-2.31181235	-0.34842100	-0.00000038

Table F.1: thiophene

C	-0.71563084	0.01767714	-0.00000164
C	0.71562942	0.01767692	0.00000601
H	-1.37241614	0.91854728	-0.00000072
H	1.37241478	0.91854733	0.00000356
N	-1.26402143	-1.19678075	-0.00001163
N	1.26401984	-1.19678068	0.00001139
S	-0.00000086	-2.27413643	-0.00000095

Table F.2: Thiadiazole

C	1.14132236	0.63862445	0.00004023
C	-1.32126555	-0.71588635	-0.00002300
C	-1.26797783	0.71607604	0.00002849
C	1.08676011	-0.81750184	-0.00002836
C	-0.06262959	1.40394172	0.00006066
C	-0.17039317	-1.49120016	-0.00005137
H	-2.31756427	-1.21558983	-0.00003667
H	-2.22460531	1.28866103	0.00004450
N	2.39808120	1.13545681	0.00008626
N	2.30254700	-1.40758625	-0.00005216
S	3.39503444	-0.17530211	0.00000217
H	-0.01231444	2.51639458	0.00010695
H	-0.20115259	-2.60456055	-0.00008570

Table F.3: Benzothiadiazole

C	1.12513572	0.61251811	0.00003022
C	-1.36414468	-0.69473860	-0.00000139
C	-1.29406235	0.71925449	0.00000655
C	1.03805271	-0.81395283	0.00000698

C	-0.06126176	1.37989443	0.00002546
C	-0.20265868	-1.47927824	-0.00000464
H	-2.36026441	-1.19427391	-0.00000601
H	-2.23681177	1.31270123	0.00000248
N	2.40640470	1.12692441	0.00003818
S	2.65090514	-1.48565876	-0.00001894
C	3.28130945	0.16242638	0.00001553
H	4.38480638	0.32174713	0.00001353
H	0.01510755	2.49118738	0.00002798
H	-0.26444139	-2.59176344	-0.00001352

Table F.4: benzothiazole

C	-0.73367061	-0.59060097	-0.00010261
C	0.73367272	-0.59061215	-0.00009699
C	1.18117310	0.75603208	-0.00008578
C	-1.18113098	0.75605514	-0.00009329
C	0.00002950	1.69161140	-0.00008610
H	0.00001890	2.36525576	0.89991026
H	0.00002779	2.36533715	-0.90002724
C	2.55208149	1.04677317	-0.00002686
C	-2.55204215	1.04682038	-0.00004539
C	3.47405819	-0.01747482	0.00000266
C	-3.47404053	-0.01739975	-0.00000458
C	3.02836307	-1.35549776	-0.00000820
C	-3.02839265	-1.35543973	-0.00000524
C	1.65522521	-1.65372468	-0.00005647
C	-1.65524870	-1.65369012	-0.00005270
H	2.91239131	2.10238404	-0.00001602
H	-2.91235173	2.10243293	-0.00003516
H	4.56720193	0.19893313	0.00003894
H	-4.56718896	0.19901620	0.00003091
H	3.77460009	-2.18350187	0.00003329
H	-3.77463327	-2.18343695	0.00004381
H	1.30419322	-2.71247112	-0.00005040
H	-1.30423301	-2.71244534	-0.00004026

Table F.5: fluorene

Titanium clusters

Ti	0.0000000	0.0000000	0.6131572
O	-1.3586178	0.0000000	-0.3066726
O	1.3586178	0.0000000	-0.3066726

Table F.6: TiO_2

Ti	-0.73300	1.14424	0.00000
Ti	0.73300	-1.14424	0.00000
O	0.10027	-2.64271	0.00000
O	0.00000	0.00000	-1.25140
O	0.00000	0.00000	1.25140
O	-0.10027	2.64271	0.00000

Table F.7: $trans-(TiO_2)_2$

Ti	0.00000	-1.36118	-0.43071
Ti	0.00000	1.36118	-0.43071
O	0.00000	2.29693	0.89501
O	-1.25846	0.00000	-0.46430
O	1.25846	0.00000	-0.46430
O	0.00000	-2.29693	0.89501

Table F.8: $cis - (TiO_2)_2$

Ti	1.5800350	0.0816697	0.0000000
O	0.6579338	-1.4182969	0.0000000
O	0.5150702	0.7437215	-1.2499304
O	0.5150702	0.7437215	1.2499304
Ti	-0.8204234	-0.0468890	0.0000000
O	-2.4476859	-0.1039269	0.0000000

Table F.9: $club - (TiO_2)_2$

Ti	-0.3652106	-0.3641201	-1.4352507
Ti	1.4766863	1.2010052	0.0000000
Ti	-0.3652106	-0.3641201	1.4352507
O	-1.3099183	-0.2907969	2.7522544
O	-0.8163564	-1.4513368	0.0000000
O	-0.3283011	1.1579020	0.0000000
O	1.5643179	0.2019928	-1.4136330
O	-1.3099183	-0.2907969	-2.7522544
O	1.5643179	0.2019928	1.4136330

Table F.10: ($(TiO_2)_3$

Ti	-0.7221843	-1.3327414	-0.0050453
Ti	1.8302321	-0.4449549	0.0112542
O	0.8604109	-1.1955768	-1.2201681
O	0.7846160	-1.0132832	1.2731841
O	-1.6427035	-2.6705908	0.0003192
O	1.4905023	1.3055795	-0.1630027
O	-1.4591783	0.3986124	-0.1919984
Ti	-0.3448112	1.8300305	-0.2830483
O	-0.7968841	3.1229248	0.5785053

Table F.11: $alt - (TiO_2)_3$

Ti	0.0000000	-2.0308706	-0.5475607
Ti	1.4114333	0.0000000	0.9536247
Ti	0.0000000	2.0308706	-0.5475607
Ti	-1.4114333	0.0000000	0.9536247
O	0.0000000	2.7809130	-1.9759412
O	-1.6889503	-1.6763018	0.3653630
O	0.0000000	0.0000000	-0.4450330
O	-1.6889503	1.6763018	0.3653630
O	1.6889503	-1.6763018	0.3653630
O	0.0000000	-2.7809130	-1.9759412
O	1.6889503	1.6763018	0.3653630
O	0.0000000	0.0000000	2.1233351

Table F.12: $(TiO_2)_4$

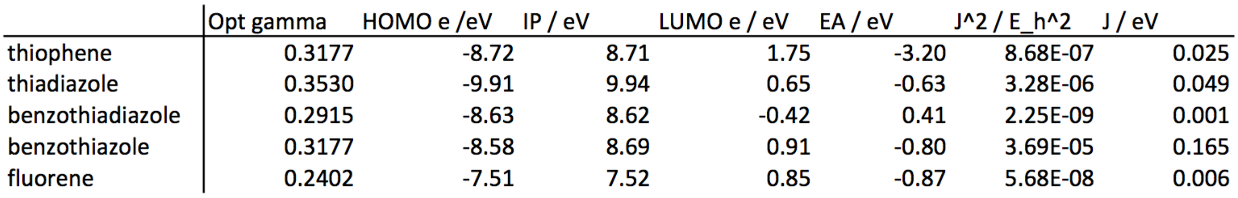
Ti	0.0000000	-0.0000000	0.0000000
O	0.0284625	1.4701427	-1.0607106
O	1.4701427	-0.0284625	1.0607106
O	-1.4701427	0.0284625	1.0607106
O	-0.0284625	-1.4701427	-1.0607106
H	0.5826058	2.2545116	-1.0227117
H	2.2545116	-0.5826058	1.0227117
H	-0.5826058	-2.2545116	-1.0227117
H	-2.2545116	0.5826058	1.0227117

Table F.13: $Ti(OH)_4$

Ti	0.0000000	-1.3559432	0.1293614
Ti	0.0000000	1.3559432	0.1293614
O	0.0000000	2.3573399	-1.3723640
O	-1.2306796	0.0000000	0.1221198
O	1.2306796	0.0000000	0.1221198
O	0.0000000	-2.3354084	1.6540984
H	0.0000000	2.2853438	-2.3283452
O	0.0000000	2.3354084	1.6540984
H	0.0000000	3.2709414	1.8703829
H	0.0000000	-3.2709414	1.8703829
H	0.0000000	-2.2853438	-2.3283452
O	0.0000000	-2.3573399	-1.3723640

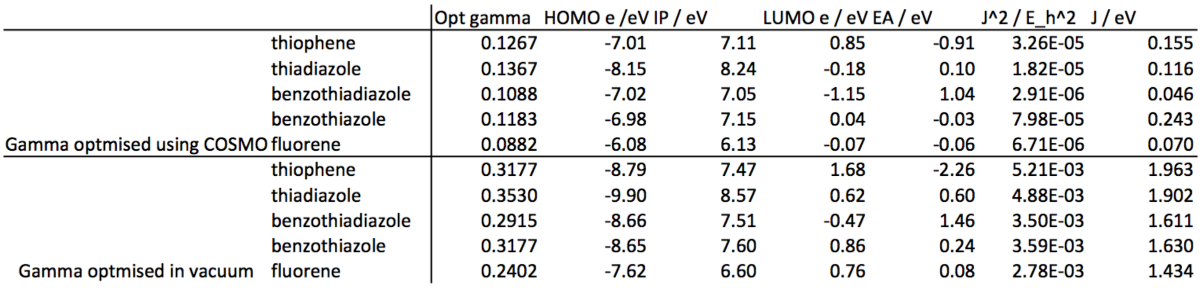
Table F.14: $(TiO_2)_4(H_2O)_2$

G. *OT-BNL Tables*



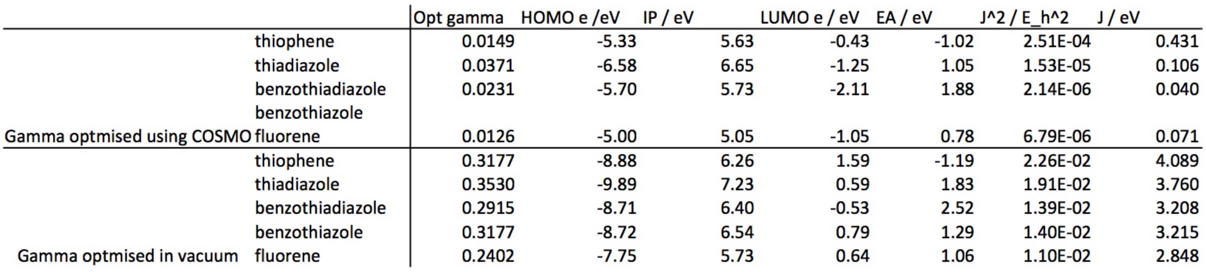
	Opt gamma	HOMO e /eV	IP / eV	LUMO e / eV	EA / eV	J^2 / E_h^2	J / eV
thiophene	0.3177	-8.72	8.71	1.75	-3.20	8.68E-07	0.025
thiadiazole	0.3530	-9.91	9.94	0.65	-0.63	3.28E-06	0.049
benzothiadiazole	0.2915	-8.63	8.62	-0.42	0.41	2.25E-09	0.001
benzothiazole	0.3177	-8.58	8.69	0.91	-0.80	3.69E-05	0.165
fluorene	0.2402	-7.51	7.52	0.85	-0.87	5.68E-08	0.006

Figure G.1: Quantities of interest from the OT-BNL calculations of the reference set.



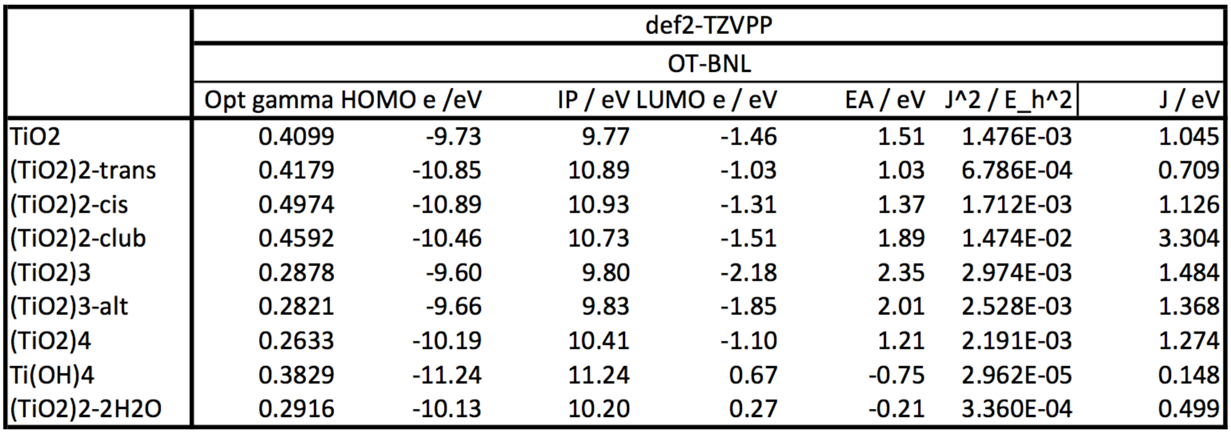
		Opt gamma	HOMO e /eV	IP / eV	LUMO e / eV	EA / eV	J^2 / E_h^2	J / eV
	thiophene	0.1267	-7.01	7.11	0.85	-0.91	3.26E-05	0.155
	thiadiazole	0.1367	-8.15	8.24	-0.18	0.10	1.82E-05	0.116
	benzothiadiazole	0.1088	-7.02	7.05	-1.15	1.04	2.91E-06	0.046
	benzothiazole	0.1183	-6.98	7.15	0.04	-0.03	7.98E-05	0.243
Gamma optmised using COSMO	fluorene	0.0882	-6.08	6.13	-0.07	-0.06	6.71E-06	0.070
	thiophene	0.3177	-8.79	7.47	1.68	-2.26	5.21E-03	1.963
	thiadiazole	0.3530	-9.90	8.57	0.62	0.60	4.88E-03	1.902
	benzothiadiazole	0.2915	-8.66	7.51	-0.47	1.46	3.50E-03	1.611
	benzothiazole	0.3177	-8.65	7.60	0.86	0.24	3.59E-03	1.630
Gamma optmised in vacuum	fluorene	0.2402	-7.62	6.60	0.76	0.08	2.78E-03	1.434

Figure G.2: Quantities of interest from the OT-BNL calculations of the organic solvated reference set.



		Opt gamma	HOMO e /eV	IP / eV	LUMO e / eV	EA / eV	J^2 / E_h^2	J / eV
	thiophene	0.0149	-5.33	5.63	-0.43	-1.02	2.51E-04	0.431
	thiadiazole	0.0371	-6.58	6.65	-1.25	1.05	1.53E-05	0.106
	benzothiadiazole	0.0231	-5.70	5.73	-2.11	1.88	2.14E-06	0.040
	benzothiazole							
Gamma optmised using COSMO	fluorene	0.0126	-5.00	5.05	-1.05	0.78	6.79E-06	0.071
	thiophene	0.3177	-8.88	6.26	1.59	-1.19	2.26E-02	4.089
	thiadiazole	0.3530	-9.89	7.23	0.59	1.83	1.91E-02	3.760
	benzothiadiazole	0.2915	-8.71	6.40	-0.53	2.52	1.39E-02	3.208
	benzothiazole	0.3177	-8.72	6.54	0.79	1.29	1.40E-02	3.215
Gamma optmised in vacuum	fluorene	0.2402	-7.75	5.73	0.64	1.06	1.10E-02	2.848

Figure G.3: Quantities of interest from the OT-BNL calculations of the water solvated reference set.



	def2-TZVPP						
	OT-BNL						
	Opt gamma	HOMO e /eV	IP / eV	LUMO e / eV	EA / eV	J^2 / E_h^2	J / eV
TiO2	0.4099	-9.73	9.77	-1.46	1.51	1.476E-03	1.045
(TiO2)2-trans	0.4179	-10.85	10.89	-1.03	1.03	6.786E-04	0.709
(TiO2)2-cis	0.4974	-10.89	10.93	-1.31	1.37	1.712E-03	1.126
(TiO2)2-club	0.4592	-10.46	10.73	-1.51	1.89	1.474E-02	3.304
(TiO2)3	0.2878	-9.60	9.80	-2.18	2.35	2.974E-03	1.484
(TiO2)3-alt	0.2821	-9.66	9.83	-1.85	2.01	2.528E-03	1.368
(TiO2)4	0.2633	-10.19	10.41	-1.10	1.21	2.191E-03	1.274
Ti(OH)4	0.3829	-11.24	11.24	0.67	-0.75	2.962E-05	0.148
(TiO2)2-2H2O	0.2916	-10.13	10.20	0.27	-0.21	3.360E-04	0.499

Figure G.4: Quantities of interest from the OT-BNL calculations of the TiO2 set.