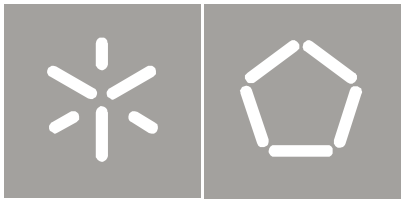
Universidade do Minho
Escola de Engenharia

Pedro Miguel Linhares Miranda

Efficient computation of CPW2000 using a CPU-GPU heterogeneous platform

Outubro de 2011



Universidade do Minho
Escola de Engenharia

Pedro Miguel Linhares Miranda

Efficient computation of CPW2000 using a CPU-GPU heterogeneous platform

Tese de Mestrado
Informática
Trabalho efectuado sob a orientação de
Alberto José Proença
José Luís Martins

Outubro de 2011

Anexo 3

DECLARAÇÃO

Nome

_____

Endereço electrónico: _____ Telefone: _____ / _____

Número do Bilhete de Identidade: _____

Título dissertação □/tese □

_____

_____

_____

Orientador(es):

_____

_____ Ano de conclusão: _____

Designação do Mestrado ou do Ramo de Conhecimento do Doutoramento:

_____

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respectiva, deve constar uma das seguintes declarações:

1. É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

2. É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA TESE/TRABALHO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, , MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

3. DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA TESE/TRABALHO

Universidade do Minho, ___/___/_____

Assinatura: _____

To my father for guidance, help, and love.

Acknowledgements

This dissertation result is a product of the continuous support and encouragement of my advisor Alberto José Proença who has provided impeccable guidance and maintained a high standard of excellence in developing my scientific career. My other adviser Jośe Luís Martins also deserves a special mention for his guidance and important support through this work.

I want to acknowledge my laboratory colleagues and all of my friends for the support and comprehension during the development of this project.
I owe a special gratitude to my family: my father Joaquim, my mother Maria, and my brothers Paulo, Nuno and Tiago. Thanks for all the love, support and good times that made my personal life so grateful and for making my academic career possible.

Special thanks to all the students, staff members, and professors of University of Minho for providing me some of the best moments of my life along these 5 years.

Abstract

The modelling and simulation of complex systems in natural science usually require powerfull and expensive computational resources. The study of the plane wave properties in crystals, based on quantum mechanichs pose challenging questions to computer scientists to improve the efficiency of the numerical methods and algorithms. Numerical libraries had a significant boost in recent years, taking advantage of multi-threaded environments.

This dissertation work addresses efficiency improvements in a plane wave package, CPW2000, developed by a physicist scientist, targeted to a heterogeneous platform with multicore CPU and CUDA enabled GPU devices. The performance botlenecks were previously identifed as being the module functions with FFT computations, and the study started with the application analysis and profiling. This study shows that (i)over 90% of the code execution time was spent in two functions, DGEMM and FFT, (ii) code efficiency of current numerical libraries is hard to improve, and (iii) DGEMM function calls were spread in the code, while FFT was concentrated in a single function.

These features were adequately explored to develop a new code version where parts of the code are computed on a multicore CPU with others taking advantage of the GPU multistreaming and parallel computing power. Experimental results show that CPU-GPU combined solutions offer near 10x speedup on the program routines that we proposed to improve, giving us a promising future work.

Computação eficiente do software CPW2000 em plataformas heterogêneas CPU-GPU

Resumo

A modelação e simulação de sistemas complexos em áreas cientificas geralmente necessita de enormes e dispendiosos recursos computacionais de processamento. O estudo das propriedades de cristais em ondas planas, com base na mecânica quântica, oferece alguns desafios aos cientistas da computação para melhorar a eficiência dos métodos numéricos e algoritmos. As bibliotecas numéricas evoluiram muito tirando vantagem de ambientes *multithreading* de computação.

O trabalho apresentado nesta dissertação baseia-se na melhoria da eficiência de um programa de ondas planas, o CPW2000, desenvolvido por um investigador da área da física, orientado para uma plataforma heterogênea de computação com um CPU multicore e um GPU com suporte á plataforma CUDA. As principais causas da deterioração da eficiência foram identificadas no módulo que contêm os cálculos de FFT, e o estudo começou com a análise dos tempos de execução de cada componente da aplicação. Este estudo mostra que (i) mais de 90% do tempo total de computação é dividido por duas funções, DGEMM e FFT, (ii) é dificil de melhorar a eficiência das bibliotecas numéricas atuais, e (iii) que as funções DGEMM estão distribuídas pela aplicação enquanto as funções FFT estão concentradas numa função.

Estas caracteristicas foram devidamente exploradas de forma a desenvolver código em que partes deste executa num CPU multicore e outras aproveitam o paralelismo e multistreaming presente nos GPU. Resultados experimentais mostram que as soluções combinadas de CPU-GPU oferecem uma melhoria de aproximadamente 10x nas funções que nos proposemos a melhorar a eficiência, culminando num trabalho futuro promissor.

Contents

List of Figures

LIST OF FIGURES

List of Tables

Glossary

BLAS Basic Linear Algebra Subprograms; API standard for publishing libraries to perform basic linear algebra operations such as vector and matrix multiplication.

DGEMM General Matrix Multiply (GEMM). The first letter identifies the precision of the operation, in this case *d* stands for double-precision.

C_{60} A fullerene molecule composed entirely of 60 carbon atoms.

GPa GigaPascal, where Pascal is the SI derived unit of pressure, internal pressure, stress, Young's modulus and tensile.(unit Pa)

1

Introduction

1.1 Context

Fields of study such as physics and chemistry use large systems of linear equations to model and describe the forces of the universe and the most common method to model and describe these large systems is using matrix operations of linear algebra. We may state that the core of many scientific software is based on BLAS and other numerical routines. And we can conclude that the performance offered by this software is conditioned by the performance offered by the different adopted libraries.

In this document we present the **CPW2000** software, a complete plane wave package (quantum mechanics field of study) for new hardware generations. CPW2000 was developed by Prof José Luís Martins[1] and, like most of the scientific software, is strongly dependent on the performance offered by BLAS and other numerical libraries. This application has assisted, beyond the software author, some researchers on their work (1, 2) and for now is only available by author permission.

Our work will be focused on improving the efficiency of the most computational intensive algorithms and libraries used on CPW2000 targeted for current CPU and GPU architectures, and not focused on benchmarking CPW2000 against similar software from competitors. Since quantum mechanics is beyond the goals of this dissertation,

[1] JLM is the author of the CPW2000 simulation package, that he developed during his stay at USA. He is currently a Professor at IST/UTL in Lisbon, and Senior Researcher at INESC-MM.

we will not discuss the efficiency of the quantum mechanics algorithms in CPW2000 and we will not detail the theory behind quantum mechanics. Instead, we will focus our expertise to improve CPW2000 overall performance by identifying the most intensive numerical functions and the best approaches to code these algorithms to execute in current CPU-GPU based computer systems.

The end-user perspective

Quantum mechanics is a field of study that aims to explain how matter and energy change in time at different levels: sub-atomic (the smaller particles composing the atom), atomic and molecular. The basis of quantum mechanics is the Schr̈odinger equation (3):

$$Time\ dependent: \quad i\hbar\frac{\partial\psi}{\partial t} = \hat{H}\psi \tag{1.1}$$

$$Time\ independent: \quad E\psi = \hat{H}\psi \tag{1.2}$$

where i is the imaginary unit, $\hbar$ is the reduced Planck constant, $i\hbar\frac{\partial\psi}{\partial t}$ is the energy operator, and $\hat{H}$ is the Hamiltonian operator (3, 4).

Solving the Schr̈odinger equation is hard due to the complexity of the system and different approaches exist: the many body problem, mean-field-based ab initio methods, many body ab initio methods and density functional theory (DFT) (1). CPW2000 uses the DFT approach, one of the most successful and popular electronic methods available to compute materials properties, with a pseudo-potential plane wave basis set (1, 5).

The computing perspective

The computing perspective can be divided into two different and strongly interconnected views: the hardware and the software view. Concerning hardware, we want to take advantage of the computing power/capabilities of new GPU devices as computing units oriented to massive multi-threading and with very large number of cores units to perform parallel computations under the SIMD approach. We also aim to obtain the best performance of recent CPU architectures ,characterized by multi-core parallel

computing devices (which is escalating to the many-core age (6)) with each core having its specific features (e.g.,SIMD extensions and multi-level caches).
The software view is strongly connected with the hardware view since we aim to achieve the best efficiency of each library at each different hardware configuration. To achieve the best efficiency several factors must be underlined for a selected algorithm/library: scalability and workload, communications latency, memory accesses and cache management.

1.2 Motivation and objectives

Our motivation is strongly related with the computing perspective described before. Given an application that has intensive computational requirements, an effective use of the resources and an efficient distribution of the workload among different devices in a system, may increase the software performance and consequently decrease the total run time of the referred software that can last for minutes or hours.
Our gold objective is to minimize the total run time of a software application. Considering that quantum mechanics is an intensive field of research, our work can be classified as the first phase of a larger project. We will try to identify the most intensive computational routines, improve their efficiency, and give some impressions for what can be done in future work to further improve the execution performance. The second phase of the project, having a stable improved version, is to build a software library ready to be available to other scientists. In this initial work we did not explore the second phase of the project, which will require other aspects beyond the computational ones.

1.3 Dissertation structure

Next chapter describes the DFT plane wave pseudo-potentials theory and how CPW2000 implements this theory, with the main focus on the numerical algorithms and libraries. We will also identify and characterize the most intensive algorithm/libraries, describe their design and discuss how their efficiency can be improved. This chapter concludes with a presentation of the target platforms that CPW2000 is currently tuned to execute on.
Chapter 3 presents our approach to improve the intensive algorithms and libraries

detected on chapter 2, and their design and implementation on current computing architectures and presents and assesses the performance improvements.
The last chapter concludes the the work focusing on the obtained results an presents suggestions for future work.

2

The computational simulation of materials using plane waves

2.1 General framework for a description of materials

The starting point to model the behaviour of a material is a system energy function:

$$\varepsilon(s_i, \psi_{nj}, \lambda_k) \tag{2.1}$$

where s_i are the atomic coordinates, ψ_{nj} is the jth component of the electrons wave function n in a given basis, and λ_k are exterior variables of the problem (for example: temperature, pressure or magnetic field).

From a thermodynamic view, λ_k are the macroscopic variables, while the others are microscopic. The macroscopic variables are defined by environmental conditions, and for each set of macroscopic variables there is a probability distribution to observe a certain value of microscopic variables that is proportional to the Boltzmann factor at a given temperature T, which is given by

$$e^{\frac{\varepsilon}{k_B T}} \tag{2.2}$$

where k_B is the Boltzmann constant. In a simulation we may be interested in several experimental conditions. For example a study of minerals of Earth's mantle (which we do not have direct access) we may be interested in pressures between 0 and 350

GPa[1]and could, for example, perform 36 simulations with 10 GPa intervals. Each one of these simulations would be independent. If we want to find the equilibrium structure of the mineral we must find its minimum of energy, that is, for each pressure λ_k we do the minimization

$$E(\lambda_k) = \min_{s_i} \min_{\psi_{nj}} \boldsymbol{\varepsilon}(s_i, \psi_{nj}, \lambda_k) \tag{2.3}$$

For simulations at finite temperature we must sample $\boldsymbol{\varepsilon}$ which can be done with Monte-Carlo or molecular dynamics methods. However, the numerical challenges are the same for a minimization of the function or its sampling, namely we must efficiently compute the value of the function and some of its derivatives; we only discuss here the case of the minimization (4, 5, 7).

System complexity

The quantity that governs the number of variables that describe the system, e.g. the system complexity in eq. 2.1 is the number of atoms N_{at} in the system. For an isolated molecule, that is the number of its atoms. For a crystal that is a periodic repetition of an atomic arrangement, it is the number of atoms in the arrangement that is being repeated (called unit cell).
Since we are working in a 3 dimensional space, the number of s_i variables is $3N_{at}$ plus, in the case of a crystal, the 6 extra variables which describe the crystal primitive unit cell.

For the parameters that describe the electron wave ψ_{nj}, we have that the number n depends on the type of atom, is proportional to N_{at} and if we just want a rough estimate we can use a typical value of $4N_{at}$. The number of components j needed to describe each wave function also depends on the system and method, but again for a rough estimate we may assume a typical value of $200N_{at}$. Which gives us a rough estimate of $800N_{at}^2$ coefficients to describe electrons. Systems currently analysed can have up to 50 atoms when studied on desktop system or 1000 atoms on supercomputers, which means we are talking between 10^6 to 10^9 variables. A minimization of a function with such large number of variables is not a trivial task (5).
However, we can use the same algorithm to minimize the energy with respect to both

[1]For all abbreviations see the glossary on page xv.

types of variables as is done in the Car-Parrinelo method (4, 5, 7). As the two types of variables have very different physical meanings we can split the minimization (eq.2.3) in two steps:

$$\begin{aligned} E(\lambda_k) &= \min_{s_i} B(s_i, \lambda_k) \\ B(s_i, \lambda_k) &= \min_{\psi_{nj}} \boldsymbol{\varepsilon}(s_i, \psi_{nj}, \lambda_k) \end{aligned} \tag{2.4}$$

where $B(s_i, \lambda_k)$ (with fixed λ_k) is known as the Born-Oppenheimer surface (8). The advantage of this minimization split is that each one has different properties. While $B(s_i, \lambda_k)$ has many local minima, $B(s_i, \lambda_k)$ is almost quadratic with fixed λ_k and s_i. Assuming that ψ^0_{nj} minimizes ε, then:

$$\frac{\partial B}{\partial s_\ell}(s_i, \lambda_k) = \frac{\partial \boldsymbol{\varepsilon}}{\partial s_\ell}(s_i, \psi^0_{nj}(s_i, \lambda_k), \lambda_k) + \frac{\partial \boldsymbol{\varepsilon}}{\partial \psi_{nj}} \frac{\partial \psi_{nj}}{\partial_\ell}(s_i, \psi^0_{nj}(s_i, \lambda_k), \lambda_k) = \frac{\partial \boldsymbol{\varepsilon}}{\partial s_\ell} \tag{2.5}$$

and solving this gradient is relatively simple compared to find $\psi^0_{nj}(s_i, \lambda_k)$ (5).
Since the gradient is "costless", typical ways to solve it use methods that are based on the derivatives to minimize or sample $B(s_i, \lambda_k)$. For example, molecular dynamics is usually more efficient than Monte-Carlo and BFGS and related methods are more efficient than the usual conjugate gradient methods, since the minimization based on lines wastes the gradient "costless" property.
Summarizing, the $B(s_i, \lambda_k)$ minimization problem in electronic structures is essentially the problem of a function with many local minima, but having a "costless" gradient (4, 5, 7, 9, 10).

The electronic problem with plane waves

The most critical computational resources is minimizing ε to fixed s_i and λ_k. There are many basis sets for wave functions, each one leading to different methods (7). Here we are considering the very popular pseudo-potential plane wave method, where the wave function dependency in the coefficients is given by: (5)

$$\Psi_n(\vec{r}) = \sum_j^{M_{txd}} \psi_{nj} e^{i\vec{G}_j \cdot \vec{r}} \tag{2.6}$$

that is, the coefficients ψ_{nj} are the Fourier expansion coefficients of the wave function. For the sake of simplicity of notation, we will drop the $e^{i\vec{G}\cdot\vec{r}}$ pre-factor in the remaining discussion, since it does not change the computational problem.
For every j-index we have a vector associated:

$$\vec{G}_j = m_1(j)\vec{b}_1 + m_2(j)\vec{b}_2 + m_3(j)\vec{b}_3, \quad m \in \mathbb{Z} \tag{2.7}$$

The complete basis would be infinite with all the integer values of m_k. This basis is currently truncated to $|\vec{G}| < G_{max}$ and is ordered so that $|\vec{G}_j| < |\vec{G}_\ell|$, if $j < \ell$. The $\vec{b}_j$ usually are not orthogonal or unitary. For each G_{max} value there are maximum values:

$$M_i = \max_{|\vec{G}_j|<G_{max}} m_i(j) \tag{2.8}$$

In the particular case of the plane wave functions, we will use a simplified equation to compute the energy function in eq. 2.1, where the microscopic variables are kept constant ($\boldsymbol{\varepsilon}(\psi_{nj})$ and the energy only depends on ψ_{nj}:

$$\begin{aligned}
\boldsymbol{\varepsilon}(\psi_{nj}) = &\sum_{n}^{N_{eig}} \sum_{j}^{M_{txd}} \frac{1}{2}|\vec{G}|^2|\psi_{nj}|^2 + \\
&\sum_{n}^{N_{eig}} \sum_{j}^{M_{txd}} \sum_{\ell}^{M_{txd}} \bar{\psi_{nj}} v(\vec{G}_j - \vec{G}_\ell)\psi_{n\ell} + \\
&\sum_{n}^{N_{eig}} \sum_{j}^{M_{txd}} \sum_{k}^{N_{anl}} \sum_{i}^{N_{anl}} \sum_{\ell}^{M_{txd}} \psi_{nj}\bar{A}_{kj}B_{ki}A_{i\ell}\psi_{n\ell} + \\
&\sum_{n}^{N_{eig}} \sum_{j}^{M_{txd}} \sum_{\ell}^{M_{txd}} \bar{\psi_{nj}} w(\vec{G}_j - \vec{G}_\ell, \rho_i(\psi))\psi_{n\ell}
\end{aligned} \tag{2.9}$$

where the wave function coefficients must obey to the following rule:

$$\sum_j |\psi_{nj}|^2 = 1 \tag{2.10}$$

On eq.(2.9), $v(\vec{G}_j - \vec{G}_\ell)$ are the Fourier coefficients of the local potential:

$$V(\vec{r}) = \sum_j \psi_n e^{i\vec{G}_j\cdot\vec{r}}$$

and A_{kj} and B_{ki} are the projectors of the non-local part of the pseudo-potential. The w potential has a functional dependency in $\rho_i(\psi_{nj})$ which introduces non-linearity into the function. But since w is weak the non-linearity is also weak (5).

2.2 CPW2000 structure

CPW2000 is a software package developed in late 80's, that was improved several times during the following decades. The starting version for this work has a complex structure as shown in fig. 2.1. This call graph was generated with Doxygen 1.8, and the modifications that were introduced (boxes with a thicker border) aimed to show the functions that took longer execution times: SCFKB, OUT_DOS and OUT_BAND.

These three functions consumed over 99% of the overall execution time on more than one computing platform. An analysis of the caller graph for function DITSPC (Fig. 2.2) shows that almost all execution time is spent on this function, which is performing the following actions, according to the comments on the original program version that we have used:

```
C      ITERATIVE DIAGONALIZATION OF A MATRIX WITH A LEADING
C      SUBMATRIX. IT IS INSPIRED ON THE RITZIT PROCEDURE
C      (H. RUTHISHAUSER NUMER. MATH. 16, 205 (1970))
C      BUT USES JACOBI RELAXATION. COMPLEX VERSION.
C      THE NON-LOCAL PSEUDOPOTENTIAL IS SEPARABLE.
C      WRITTEN FEBRUARY 1990. JLM
C      MODIFIED 17 JANUARY 2007 FOR NEW ORTHOGONALIZATION
C      AND DGEMM HAMILTONIAN CALCULATION. JLM
```

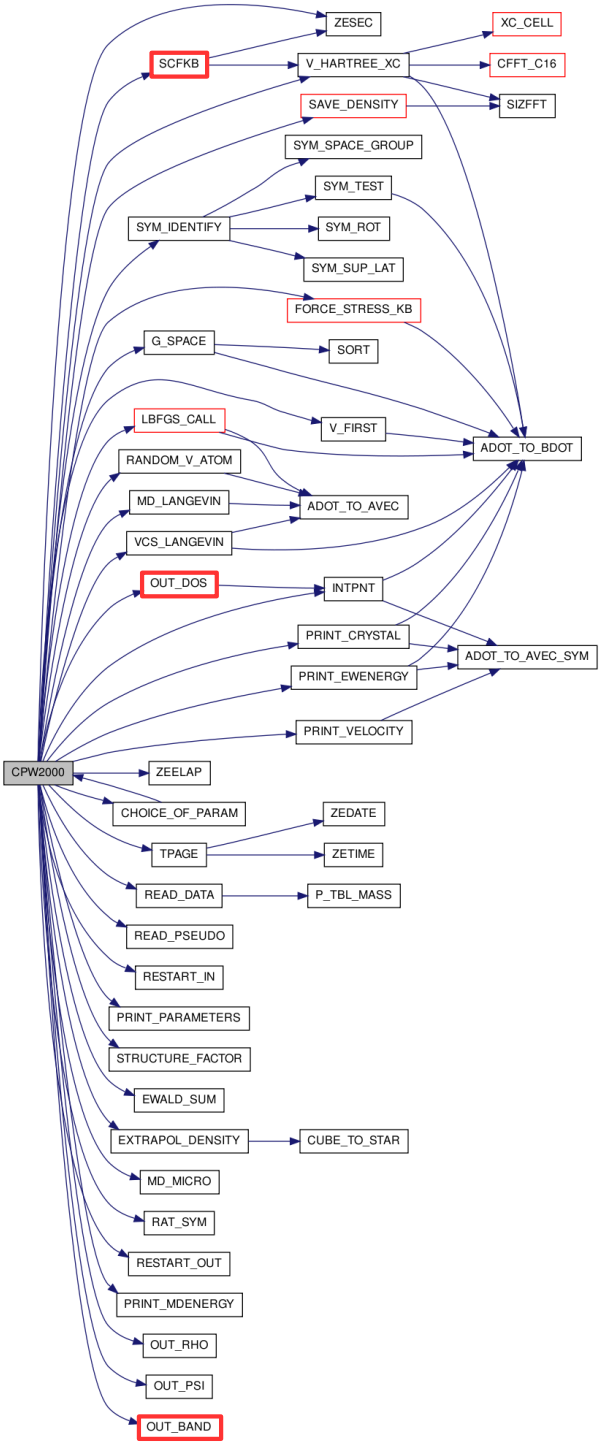


Figure 2.1: CPW2000 call graph. - Red boxes indicate that those functions call other functions not represented in the call graph.

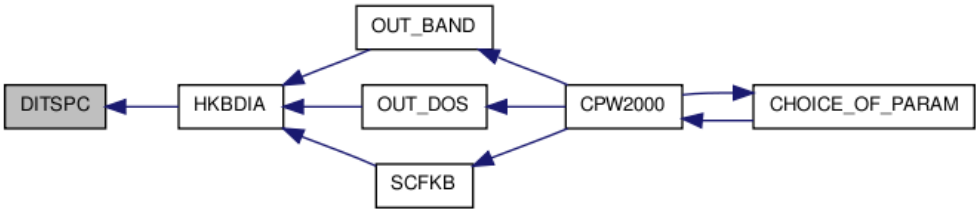


Figure 2.2: DITSPC caller graph. -

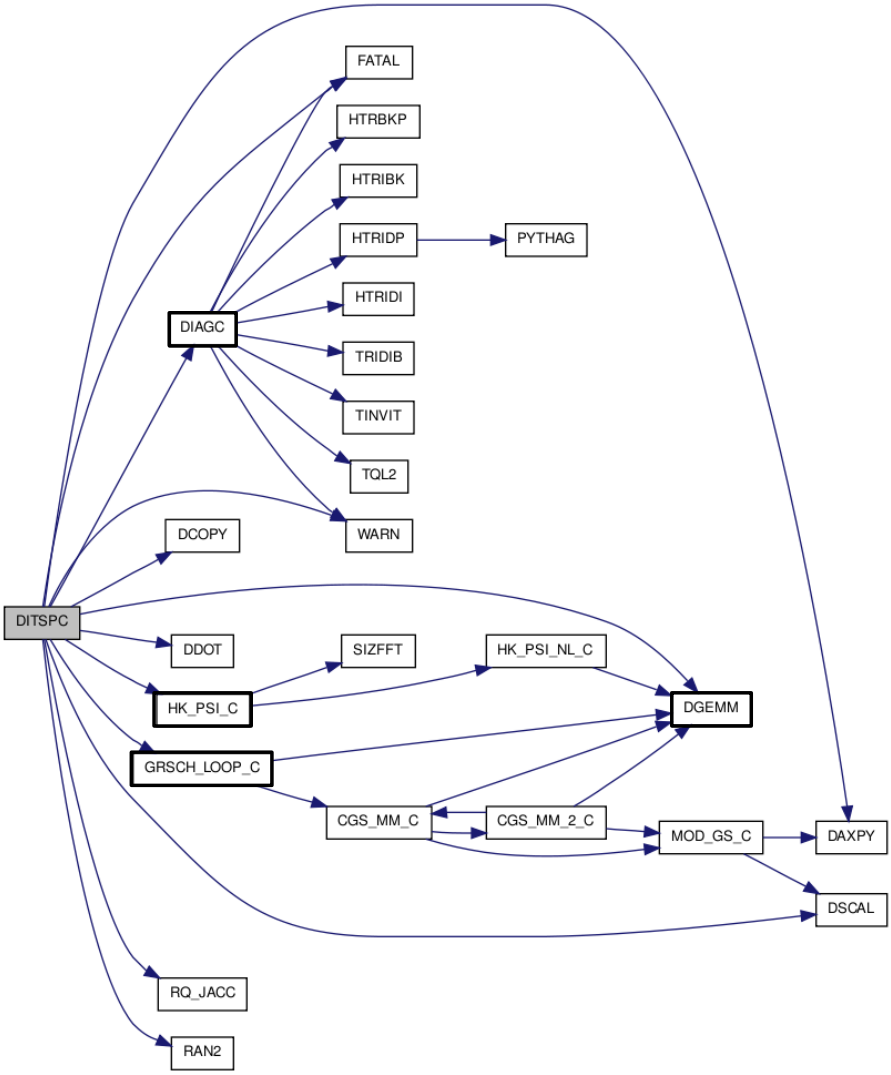


Figure 2.3: DITSPC call graph.

The call graph for DITSPC is represented in fig. 2.3, where boxes drawn with a thicker border were identified by a profiler as takers of almost 100% of the execution time of DITSPC.

These functions are:

- **DIAGC**, which according to the program comments:

```
C      DIAGONALIZES THE HERMITIAN HAMILTONIAN
C      STORED IN LOWER TRIANGULAR FORM
C      IN PACKED OR UNPACKED STORAGE.
C      WRITTEN OCTOBER 11 1989. JLM
```

- **HK_PSI_C**, which according to the program comments:

```
C      CALCULATES THE PRODUCT OF THE HAMILTONIAN TIMES
C      NEIG WAVEVECTORS. THE NON-LOCAL PSEUDOPOTENTIAL
C      IS SEPARABLE. THE LOCAL POTENTIAL IS DEALT WITH
C      FAST FOURIER TRANSFORMS. COMPLEX VERSION
```

- **GRSCH_LOOP_C**, which according to the program comments:

```
C      PERFORMS A GRAM-SCHMIDT ORTHOGONALIZATION STEP OF
C      NVEC XVEC VECTORS. THE FIRST NCONV VECTORS ARE SUPPOSED
C      TO BE ALREADY ORTHOGONAL.
C      CALLS BLAS SUBROUTINES FOR SPEED
C      IF ALSO='H' PERFORMS THE CORRESPONDING
C      TRANSFORMATION IN HXVEC
C
C      REAL VECTORS VERSION
C
C      IT USES A MIXTURE OF BLOCK CLASSICAL GRAM-SCHMIDT (USING BLAS3)
C      AND LOCAL MODIFIED GRAM-SCHMIDT. ALL STEPS ARE REPEATED
C      ACCORDING TO THE "TWICE IS ENOUGH" PRINCIPLE.
```

- **DGEMM**, which is a level 3 BLAS[1]routine to perform a matrix-matrix operation:

$$C := alpha * op(A) * op(B) + beta * C$$

  Where alpha and beta are scalars. A,B and C are matrices. And $op(X) = X$ or $op(X) = X'$;

As can be seen from the short description of these computational intensive functions, the main focus of this dissertation work is on these issues.

[1]For all abbreviations see the glossary on page xv.

2.3 Computing intensive numerical methods

Given the formal description of the scientific theory on CPW2000, we will now continue this formal description introducing and relating with the computational aspects and reaching the motivation of our work: (i) which are the most intensive numerical methods and how they are used, and (ii) what can be done to improve their efficiency using heavily tuned available library functions in a platform with CPU and GPU devices. From a computational point of view, (i) and (ii) may support a successful problem comprehension and a better implementation in current computing platforms.
We identify matrix diagonalization as a computing intensive method. However, we should consider this method as a "black box" that requires a matrix and a function as input, and computes the resulting matrix, containing the eigenvectors. For a detailed problem description, we will continue describing the CPW2000 behavior to achieve the core of our work: the function that matrix diagonalization requires and that will prove to be one of the most intensive and resources consumer.

2.3.1 Matrix diagonalization on CPW2000

Without the dependence of w in ρ, we recognize in eq. 2.4 a quadratic form. There are two approaches to deal with the non-linearity in the minimization of $\varepsilon(\psi_{nj})$. The first is to use a conjugate gradient to do the full minimization, knowing that we are close to a quadratic form, and therefore should quickly converge to the minimum. The second is to have an iterative method. We assume that a given function $\rho(\vec{r})$ minimizes the quadratic form, recalculates $\rho(\vec{r})$ and iterates until convergence (using convergence acceleration techniques). Usually 5 to 15 iterations are sufficient to converge using Broyden methods. We will be using the iterative method. But again the computational intensive step is the same in both cases, so our discussion is quite general (11).

The minimization of the quadratic form with the the rule (2.10) requires solving N_{eig} eigenvectors. Since $N_{eig} \ll M_{txd}$ finding the eigenvectors is advantageous. The explicit quadratic form of the linearised energy equation (eq. (2.9)) is given by:

$$\varepsilon_{lin}(\psi_{nj}) = \sum_{n}^{N_{eig}} \sum_{j}^{M_{txd}} \sum_{\ell}^{M_{txd}} \bar{\psi}_{nj} H_{j\ell} \psi_{n\ell} \tag{2.11}$$

where $H_{j\ell}$ is a Hermitian matrix. Either on iterative diagonalization or gradients method we have to compute:

$$f_{nj} = \sum_{\ell}^{M_{txd}} \bar{\psi}_{nj} H_{j\ell} \psi_{n\ell} \tag{2.12}$$

i.e. the computational heavy step is the same in both methods, and what really matters is how many times we have to do that step to minimize ε with the different methods. CPW2000 uses an algorithm that mixes the Davidson method (12) with the ritzit method (13), the DIIS method (14) and other sources (15).
To give an idea of the problem size based on eq.2.9, we may consider C_{60}[1]as a problem of considerable size. The sizes are given by:

$$N_{eig} = 172,\ N_{anl} = 290,\ M_{txd} = 10^5$$

so v and w requires 12 MB in memory, ψ_{nj} 262 MB memory, A 445 MB and the Hermitian matrix H, where each dim has 10^5 length, requires approximately 150 GB of memory space. Calculating and storing H is hard, but analysing the description given so far and eq .(2.9) we do not need to resolve H to obtain f_{nj}:

$$\begin{aligned} f_{nj} &= \frac{1}{2}|\vec{G}_j|^2 \psi_{nj} + \\ &\sum_{\ell}^{M_{txd}} (v(|\vec{G}_j| - |\vec{G}_\ell|) + w(|\vec{G}_j| - |\vec{G}_\ell|, \rho_i))\psi_{nl} + \\ &\sum_{k}^{N_{anl}} \sum_{i}^{N_{anl}} \sum_{\ell}^{M_{txd}} \bar{A}_{kj} B_{ki} A_{i\ell} \psi_{n\ell} \end{aligned} \tag{2.13}$$

where the first line of the equation is diagonal and easy to compute (vector-vector multiplication), the third line is a series of matrix-multiplications of significantly smaller sizes than $M_{txd} \times M_{txd}$(efficiently executed with level 3 BLAS[2]routines). The second line of the equation is a convolution and can be efficiently implemented with Fourier

[1]For all abbreviations see the glossary on page xv.
[2]For all abbreviations see the glossary on page xv.

transforms.

From eq. 2.14, eq. 2.9 and from the problem description given so far, we conclude, analytically, that computing f_{nj} is the most is the most repeated process during CPW2000 execution (5, 15).

From now on $H\boldsymbol{\psi}$ will represent the computational representation of f_{nj}.

2.3.2 The Fast Fourier transform on CPW2000

The convolution can be efficiently implemented with Fourier transforms (16), namely using efficient FFT code in widely available libraries.

$$h_j = \sum_{\ell}^{M_{txd}} v(\vec{G}_j - \vec{G}_\ell)\psi_\ell \tag{2.14}$$

For the sake of simplicity of notation, in this expression we are not including the eigenvector n. The contribution to w is equal. To compute the convolution we need to map the ψ_ℓ values in a three dimensional grid:

$$\psi_\ell = \tilde{\psi}_{m_1(\ell)m_2(\ell)m_3(\ell)} \qquad \vec{G}_\ell = m_1(\ell)\vec{b}_1 + m_2(\ell)\vec{b}_2 + m_3(\ell)\vec{b}_3 \tag{2.15}$$

The ψ_ℓ coefficients match the values on the $\vec{G}$ frequencies space, so we perform an inverse Fourier transform to obtain the wave function $\Psi(\vec{r})$ (eq. 2.6) in an uniform grid in the "$\vec{r}$" space. The wave function is then multiplied with $V(\vec{r})$, which is the previous calculated inverse transform of $v(\vec{G})$, since is the same for every n.

$$H(\vec{r}) = V(\vec{r})\Psi(\vec{r}) \tag{2.16}$$

We set the values from "$\vec{r}$" space back to "$\vec{G}$" space using Fourier transform. The h_j (eq. 2.14) values are given by (5):

$$h_j = \tilde{h}_{m_1(\ell)m_2(\ell)m_3(\ell)} \tag{2.17}$$

This approach to compute the convolution introduces an artefact well known in computer science and signal processing, the *aliasing phenomena* that is characterized by different signals becoming indistinguishable one of the other when sampled. The adopted solution is padding the 3D structures with zeros (5, 17).

FFT description

The discrete Fourier transform (DFT) of an array X of n complex points is given by:

$$Y(k) = \sum_{j=0}^{n-1} X[j]\omega_n^{jk} \tag{2.18}$$

where $0 \leq k < n$, $\omega_n = e^{-2\pi i/n}$ is a primitive n-th root of unity, and $i = \sqrt{-1}$. Computing directly the DFT definition requires $\Theta(n^2)$ operations while the FFT algorithms get the same result in $O(n \log n)$ operations (18, 19, 20, 21).

The most relevant FFT algorithm is the Cooley-Tukey (18), which is a divide-and-conquer algorithm. It has the name of his authors (James W. Cooley and John W. Tukey), but it is believed that it was derived earlier by Euler in 1805 and later by other authors (20, 21, 22, 23). The Cooley-Tukey FFT algorithm re-expresses a DFT of composite size $n = n_0 \times n_1$ in terms of smaller DFT's of sizes n_0 and n_1. Conceptually we can see this as a 2-D FFT of size $n_0 \times n_1$ where the output is transposed. Assuming that n is a composite, i.e. $n = n_0 \times n_1$, and rewriting the indices *j,k* of eq. 2.18 as $j = j_0 n_2 + j_1$ and $k = k_0 + k_1 n_0$, eq. 2.18 can be written as (18, 19, 23):

$$Y(k_0 + k_1 n_0) = \sum_{j_1=0}^{n_1-1} \left[\left(\sum_{j_0=0}^{n_0-1} X(j_0 n_1 + j_1)\omega_{n_0}^{j_0 k_0} \right) \omega_n^{j_1 k_0} \right] \omega_{n_1}^{j_1 k_1} \tag{2.19}$$

The algorithm computes n_1 DFTs of size n_0 (the inner sum), multiplies the result by $\omega_n^{j_1 k_0}$ (known in the literature as the *twiddle factor*), and then computes n_0 DFTs of size n_1 (the outer sum). The algorithm continues in a divide-and-conquer strategy (recursively, but efficient implementations avoid recursion) until the smaller DFT is reached, a simple DFT of size 2 (*butterfly* in the literature).
Since n is divided by 2, this is known as *radix-2* FFT algorithm, and using different radices and different sizes from power of two lead to different variations of this algorithm (19, 20, 21). Computing a DFT of length n recursively in terms of two DFTs of size $n/2$, is the key concept of Cooley-Tukey FFT algorithm and it has its speed ($O(n \log n)$) by reusing the results of intermediate computations (*twiddle factor*) to

compute the entire FFT.

The generalization of eq. 2.19 to more than two dimensions give us the definition of multidimensional DFT of a L-dimensional array X of $N_0 \times ... \times N_L$ complex points:

$$Y(n_0...n_l) = \sum_{k_L=0}^{N_L-1} ... \sum_{k_0=0}^{N_0-1} \omega_{N_L}^{k_L n_L} \times ... \times \omega_{N_0}^{k_0 n_0} \times X(k_0...k_L) \tag{2.20}$$

where $0 < n_0, k_0 < N_0 - 1$,and $0 < n_L, k_L < N_L - 1$. In a two dimension case (matrix), we can see this as performing 1D FFTs along the rows and then perform 1D FFTs along the columns, so called row-column algorithm. In higher dimensions the reasoning is similar. There are that also compute a multidimensional FFT, such as the vector-radix FFT algorithm, but all of them have $O(n \log n)$ complexity (20, 21, 23).

Currently, many FFT libraries are available and most of them offer several algorithms to compute FFT's in special datasets with different sizes (although sizes that are powers of two tend to achieve better performance (19)). It is often recommended to use a larger grid that is a power of two rather than a smaller grid, and never a grid whose size is a prime number. In three dimensions it is better not to use a grid much larger then needed, as long as the size factors are powers of two, three or even five.
These libraries are usually offered by the hardware vendors (24, 25, 26), being specially tuned for their platform, others developed by open-source groups, aiming to achieve the best efficiency in different platforms (27). CPW2000 needs to compute several 3D complex transformations, with the 3D structures size between $64 \times 64 \times 64$ and $128 \times 128 \times 128$ double precision complex points, and for this purpose we did not develop special software to compute FFT, instead we focussed on the communications problem. We used the available efficient libraries combined with our improvements to the overall program structure and how data is transferred between memory CPU and GPU, to obtain better results and to expose some of the bottlenecks in the overall program efficiency.

2.3.3 Profiling CPW2000

To improve an application efficiency, the first approach is to profile the application to identify the functions that consume more resources. In this section we will present some

profiling results of CPW2000 program with different BLAS libraries (Netlib, ACML and MKL) and different FFT libraries (FFTW3, MKL and JLM). Detailed information about these libraries will be presented further in this document.

		MCr		2nd MCr		
	BLAS + FFT	**Routine**	**Time(%)**	**Routine**	**Time(%)**	**Time(s)**
seq.						
	Netlib + JLM	DGEMM	87.7	FFT	9.9	12464
	ACML + FFTW3	DGEMM	66.9	FFT	29.6	5256
	MKL + JLM	DGEMM	56.4	FFT	35.6	3475
par.						
	MKL + MKL	FFT	60.4	DGEMM	30.7	1858
	ACML + FFTW3	DGEMM	57.6	FFT	33.7	1773
	MKL + JLM	DGEMM	49.8	FFT	38.9	1372

Table 2.1: CPW2000 Initial profiling - *MCr* stands for "Most Consuming routine"; *Seq* stands for "sequential" execution; *Par* stands for "Parallel". On the parallel versions the number of threads is 4.

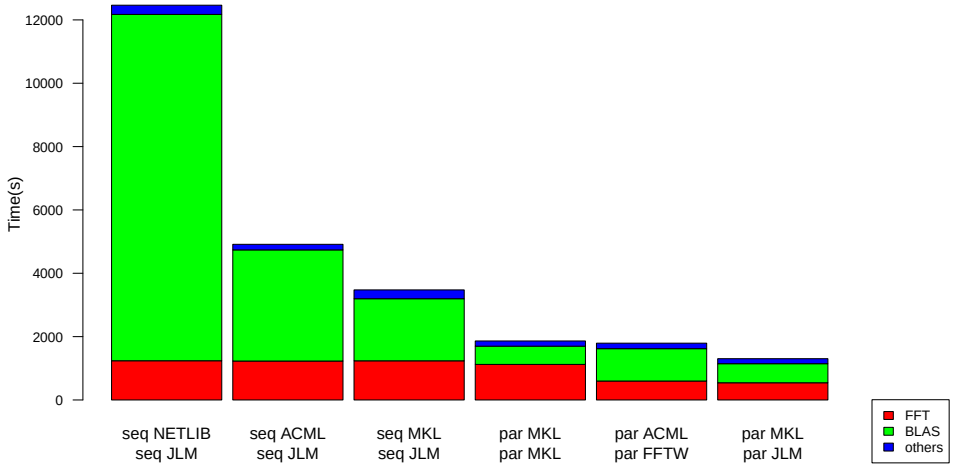


Figure 2.4: Libraries time consumption distribution. - Bar plot illustrating the time distribution of 2.1. *Seq* stands for "sequential" execution; *Par* stands for "Parallel".

The first conclusion from table 2.1 is that CPW2000 has the flexibility of using different BLAS and FFT libraries. By convention, BLAS has a standard BLAS API (28), so CPW200 can use different BLAS libraries (both free and proprietary). But it can also use different FFT libraries with little effort from the user (adding 3 files to the CPW2000 source library respecting the corresponding FFT API), which is a powerful

feature considering that FFT libraries do not have a standard API that software developers should respect. This feature is available by a special API on CPW200 which has *wrappers* for the most used FFT libraries (mainly MKL and FFTW).

Table 2.1 and Fig.2.4 show that the fastest version, a combination of the parallel version of MKL BLAS library with JLM FFT parallel version, ends all operations in 1372 seconds. But we must pay attention to the implicit corollary from Amdahl's law (29): on the first row of the column (Sequential NETLIB BLAS and JLM), BLAS and FFT consume 97.6% of the total computing time, remaining a small percentage (2.4%) to the rest of the program. As we decrease the execution time by using different libraries combinations, the percentage that was meaningless starts to have some impact and on the fastest version the weight of the rest of the program is 11.3% (155 seconds). Now that we have shown that BLAS and FFT are the most computational intensive methods, from eq.2.13 we know that the FFTs computations are only used in the convolution in $H\psi$ function:

	DGEMM + FFT	H_psi(s)	rest of the program(s)
seq.			
	Netlib + JLM	3386	9078
	ACML + JLM	1898	3016
	MKL + JLM	1632	1843
par.			
	MKL + MKL	1232	626
	ACML + FFTW3	798	975
	MKL + JLM	664	708

Table 2.2: Routines time consumption distribution. - Routines time distribution during CPW2000 execution.

We may also conclude from table 2.2 we can conclude that a single routine (*H_psi*) is consuming 49% of the total time of the program, but we also know that this function contains all the FFTs computations (table 2.1 and Fig. 2.4). Again, the corollary derived from Amdahl's law (29) states that every optimization in the *H_psi* routine (FFT's computations) will improve the overall application performance. So our work will be mainly focused on improving the efficiency of this function.

A key feature when computing FFT on CPW2000

The 3D structure that holds the complex values that will be computed by the 3D FFT algorithm several times during the program execution has a strong property: it is very

sparse (Figure 2.5), and the values that are non-zero are grouped on the edges of the cube.

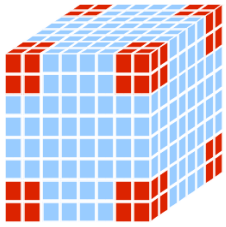


Figure 2.5: 3D FFT data structure - Conceptual view of the 3D FFT data-structure represented as a cube. Blue cubes represent memory positions holding a 0 value, while the red ones contain values different from 0.

The percentage of values that are non-zero in most 3D-FFT computations is approximately 4% of the 3D structure size, which is a very small percentage considering that the size of this structure varies from $64 \times 64 \times 64$ to $128 \times 128 \times 128$ complex points.

FFT efficiency improvements already on CPW2000

The original CPW2000 code was already optimized to reduce the computational time spent in 3D FFT computations, and the reader should be aware that the profiling results presented in the previous section were obtained with the application version that makes use of this optimization. According to the definition of multidimensional FFT which states that it can be computed by performing 1D sub-transforms along each dimension, the approach is based on the idea of skipping 1D sub-transforms that are zero. The pros and cons of this optimization will be later discussed.

2.4 Target platforms

Multi-core computing

CPW2000 in the past years has been tuned for multi-core platforms that can be found in cluster nodes or at desktop systems, mainly because a new parallel programming paradigm had to be assumed and the ability to use some of the most tuned libraries for this kind of parallel architecture (5). Current multi-core computing is characterized by a single component with several cores (each core is an independent processor), they

may be packed in a single die or packed in multiple dies in a single chip package. Each core may have its own cache memory or it may be shared (or both) among all cores, and cores may communicate trough high communication channels or among cache memory (message passing or shared memory). So far all cores in the same chip have a single view of the memory array, supporting the shared memory paradigm in parallel computing.

CUDA

Compute Unified Device Architecture (CUDA) (30) is a many-core parallel reference model architecture developed by NVIDIA, that offers software developers the massive parallelism present on GPU across some standard programming languages. It has become very popular among the scientific community because of the speedup offered by this architecture on some known algorithms (31, 32).

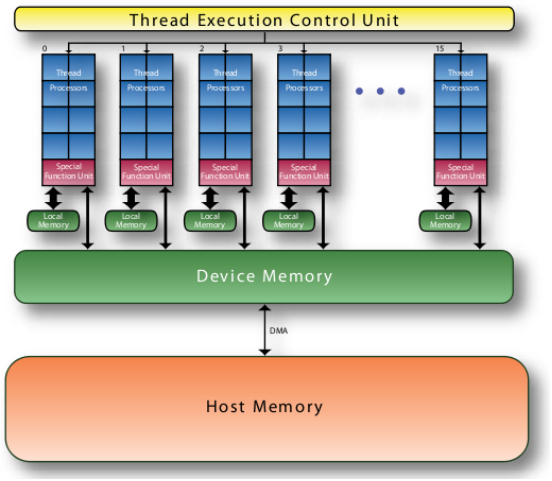


Figure 2.6: Architecure of a CUDA capable device - General CUDA architecture overview (33).

Fig. 2.6 shows a typical architecture of a CUDA capable device, organized as an array of threaded streaming processors (SM). In Fig. 2.6 there are 16 SM numbered from 0 to 15. These may be grouped to form a block, but its size can vary among CUDA GPU generations. Each SM contains several streaming processors (SP, also called CUDA cores) that share a special control unit and a local memory(8 SP per SM in Fig. 2.6). The device memory is organized in three levels: global, constant, and texture which are shared by all SMs(34).

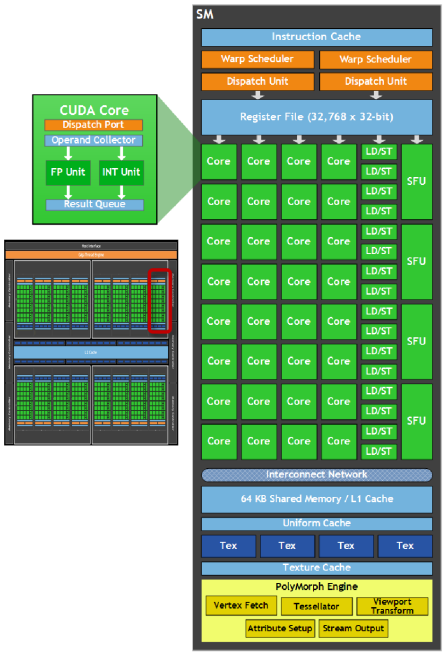


Figure 2.7: Fermi Architecture. - Fermi architecture (35).

The current CUDA GPU generation is the code-named "Fermi" (Fig. 2.7) and it has several improved aspects on its architecture when compared to earlier GPU generations (GT80 and GT200), namely: improved double precision conformance (full compliant with IEEE 754-2008 32-bit and 64-bit precision), true cache hierarchy, larger shared memory, faster context switching, dual warp scheduler and faster atomic operations. Fermi has a new ISA, and the number of CUDA cores per SM increased to 32 (up to 512 cores total) (35).

Why not focus on distributed computing

Considering the problem complexity, developing a distributed memory implementation seemed a good approach. But the gold objective of CPW2000 is to give the possibility to a researcher that may not have access to a large distributed computing platform, to run an efficient code version on a affordable CPU-GPU desktop platform. So we do not need that the most resources consuming algorithms on CPW2000 to be embarrassing parallel that it could run on 10^5 processors. Instead we opted to improve their efficiency on 10^2 processors which is currently available (e.g., a motherboard with 8 sockets for 12-core Opteron). When the time comes that a common desktop has 10^5 processors,

we may despatch 10^4 minimization programs with different starting points, different temperatures, different compositions (5).

3

Improving efficiency of CPW2000

Previous chapter identified the key computational intensive functions on CPW2000: matrix-matrix multiplication and FFT. These are mainly related/located to $H\psi$ function and more precisely with the convolution operations. The core of this work is is focused on the performance improvements of these functions on CPW2000.

3.1 Improving convolution efficiency

The CPW2000 code was optimized for Cray supercomputer available at the end of past century. It also includes the flexibility to interface to different BLAS and FFT libraries. However, CPW2000 goes further to minimize the time spent in 3D-FFT, since it skips the 1D sub-transforms that are all zero, reducing the waste of computation time/resources.

Fig.3.1 illustrates this optimization. Assuming that the dimensions are given by N_x, N_y and N_z, and assuming that K_x, K_y and K_z are the limits on each 1D sub-transform of non zero values, the size of the conceptual cube is given by $N = N_x N_y N_z$ and the size of each sub-cube is given by $K = K_x K_y K_z$, which gives us a total of $T_{1D-FFT} = N_z(N_x + N_y) + N_x N_y$ 1D sub-transforms.
Along z dimension there are no savings (due to the convolution definition), along x dimension $S_x = N_y N_z - 2K_x N_z$ 1D sub-transforms are saved , and along y dimension $S_y = N_x N_z - 2K_y N_z$ 1D sub-transforms are saved. Which gives a total of 1D sub-transforms savings of $S = T_{1D-FFT} - S_x - S_y$.

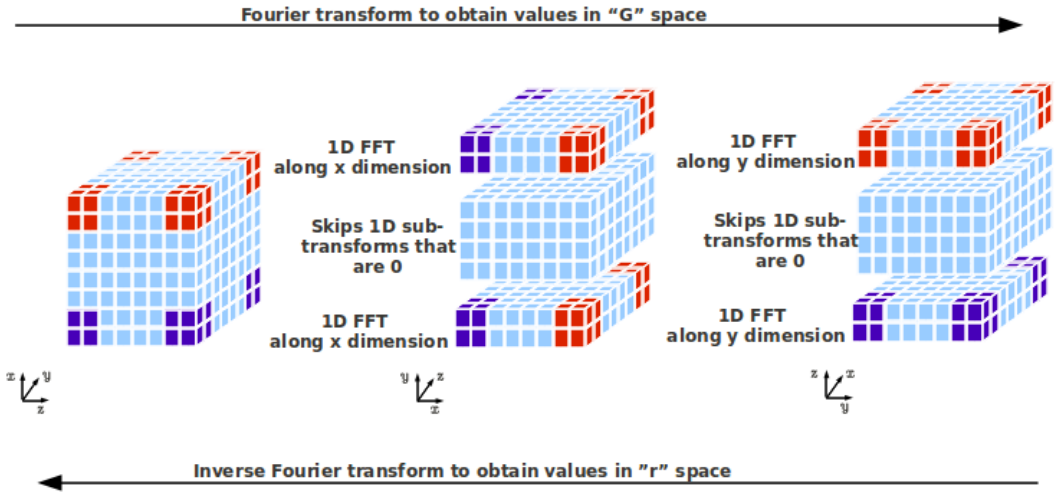


Figure 3.1: FFT optimization in CPW2000. - Blue cubes represent memory positions holding a 0 value or values that do not matter to the convolution morphology, while the red/magenta ones contain the wanted values after the transformation.

To give a sample of the total savings, assume that $N_x = N_y = N_z = 100$ complex points, and that $K_x = K_y = K_z = 25$, we have that $T_{1D-FFT} = 30000$, but only 20000 1D sub-transforms are computed ($S = 10000$ 1D sub-transforms are saved).

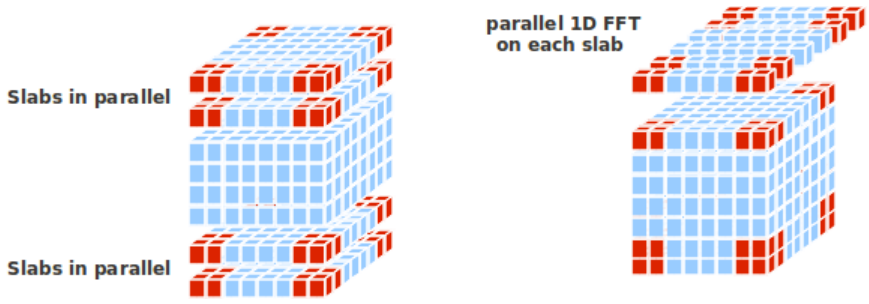


Figure 3.2: Parallel implementation problem in CPW2000. - Different parallelism implementations with different libraries.

The parallelism is implemented on slabs (each thread computes N_i 1D sub-transforms) which, considering the DFT definition (chap.2.3.2), is a good approach to minimize communications. However, there is an implementation problem already present in the CPW2000 program concerning the FFT optimization that we were not able to solve

yet: the MKL BLAS contains *wrappers* for the FFTW3 FFT routines, so when using these BLAS routines the program automatically uses the FFT routines present on this library. The problem arises because this *wrappers* of the MKL library are threaded, and since FFTW3 is a threaded library and the referred optimization also implements parallelism, this scenario makes FFT optimization not thread safe. To ensure thread safety on the FFT optimization, when used with MKL BLAS library, instead of applying parallelism on slabs the parallelism is applied on 1D sub-transforms on each slab (Fig. 3.2).

Although this approach is adequate to increase performance in FLOPS, the number of FLOPS that an algorithm can offer is not the only fact that matters in current CPU architectures. The other relevant issue is memory access time, which heavily depends on cache efficient utilization.
By definition, the FFT algorithm is a highly non-local algorithm (23), and in a multidimensional FFT this non-locality increases. Several techniques were developed (e.g., transposing data) so that the transforms are performed on contiguous data (20, 23, 27); however, this optimization does not implement any technique to minimize the non locality in multidimensional FFTs. Whether this 3D structure is stored in row-major format or in column-major format, there is one dimension where the sub-transformations are computed in contiguous data, and the other two are operated in non contiguous data (the farthest the highly non-local). Looking at Fig. 3.1 and assuming that the data is stored in row-major format, 1D sub-transforms along the z dimension are operated in contiguous data, along the x dimension the elements of each sub-transform are operated with stride N_z and along the last dimension the elements are operated with stride $N_z N_x$ (extremely non-local).

Other performance issue is the poor cache management on this approach. Take for example (in row-major format) the element stored in *structure*(0,0,0)(*structure*(N_y, N_x, N_z)). This element will be accessed on the transformations *structure*(0,0,z), *structure*(y,0,0) and *structure*(0,x,0). So, along z dimension this element is loaded to compute one transform, and along x dimension the same element is required to compute another transform. However, $N_x \times N_z$ 1D transforms afterwards we will have a cache miss (same reasoning along y dimension) since this element probably has already been cache overwritten. The reader may argue that recent CPU architectures have large cache sizes

and this would not be a problem, but older CPU architectures with smaller caches or larger problem sizes, suggest that we should try to achieve the best performance on a wide range of CPU architectures.
Summed up briefly and bottom line: this feature is a good starting point to reduce the FLOPS of the FFT computations in the convolution, and allied with an efficient use of the cache and techniques to reduce the non-locality may lead to interesting results.

Prunned FFT solution

The previous optimization lead to an interesting and promising approach, for each 1D sub-transform along each dimension, where data is 0.

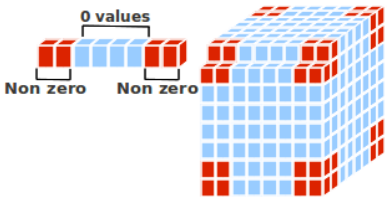


Figure 3.3: Single 1D transform. - Morphology of each 1D sub-transform. Each 1D sub-transform normally has a size of 100 complex points, where only half are non zero and are located on the edges. This morphology gives that peculiar aspect to the conceptual cube.

Recalling the FFT definition (chap. 2.3.2), the FFT algorithm breaks a DFT of composite size into a sum of smaller DFTs. Considering the morphology of each 1D sub-transform (Fig. 3.3), there are smaller DFTs where the result is zero. We could discard ("prune") some of the intermediate computations during the execution of the FFT algorithm. Two options were available:

- Compute the first K_d and the last $N_d - K_d$ ($d \in \{x, y, z\}$) outputs of a complex FFT, and, after computed, multiply each element of the output with the missing *twiddle factors*. Compute the 2 FFTs can be done with efficient libraries however, we would have to: pre-compute the twiddle factors and additional loops to multiply the twiddle factors by the output. To convert this approach into real

gains, K_d should be smaller than N_d, ideally by a factor of 100 or more(36). In our case K_d is smaller by a factor of 2 in most cases.

- Develop special "pruned" FFT algorithms in order to remove the intermediate smaller DFTs that are zero (37, 38, 39). Like the previous item, this would require K_d to be smaller than N_d by a larger factor and it would require an implementation as fast as the best ones in order to obtain real gains and overcome them, since high performance FFT libraries are specially tuned for different architectures, hence their performance.(36)

Pruning "0 computations" seemed to be an adequate approach to speed up FFT with sparse values; however, the possible resulting implementations did not produced yet the expected FFT performance improvements of our FFT problem.

Improving FFT efficiency

We concluded before that the CPW2000 FFT optimization allied with a an efficient memory management may produce interesting results. We will exploit this strategy by increasing the data reuse on cache and minimizing the operations on non-contiguous memory read/write.

Fig. 3.4 represents the approach (implemented in some libraries and under research (20, 23, 27)). The number of 1D sub-transforms and the FFT computations along the z dimension are equal to the obtained in section 3.1, but the 1D FFT computations along x and y dimensions have changed to achieve better data temporal and spatial locality. Instead of sequentially compute y dimension after x dimension, on x dimension for each *slab* of size $N_x \times N_y$ we compute N_x 1D sub-transforms in parallel and store it in an auxiliary matrix transposing the output (the transpose step is implicit, e.g., each element after computed is stored with stride N_y) and then we compute the N_x sub-transforms in parallel on the auxiliary matrix saving the output to the cube. Comparing our proposed approach with the one already implemented on CPW2000 (Fig. 3.1), the number of 1D sub-transforms is the same, and our savings come on the last dimension. The 1D-sub-transforms on the last dimension are contiguous (they are allocated sequentially in the matrix), and this values are already cached since they were used in the previous dimension.

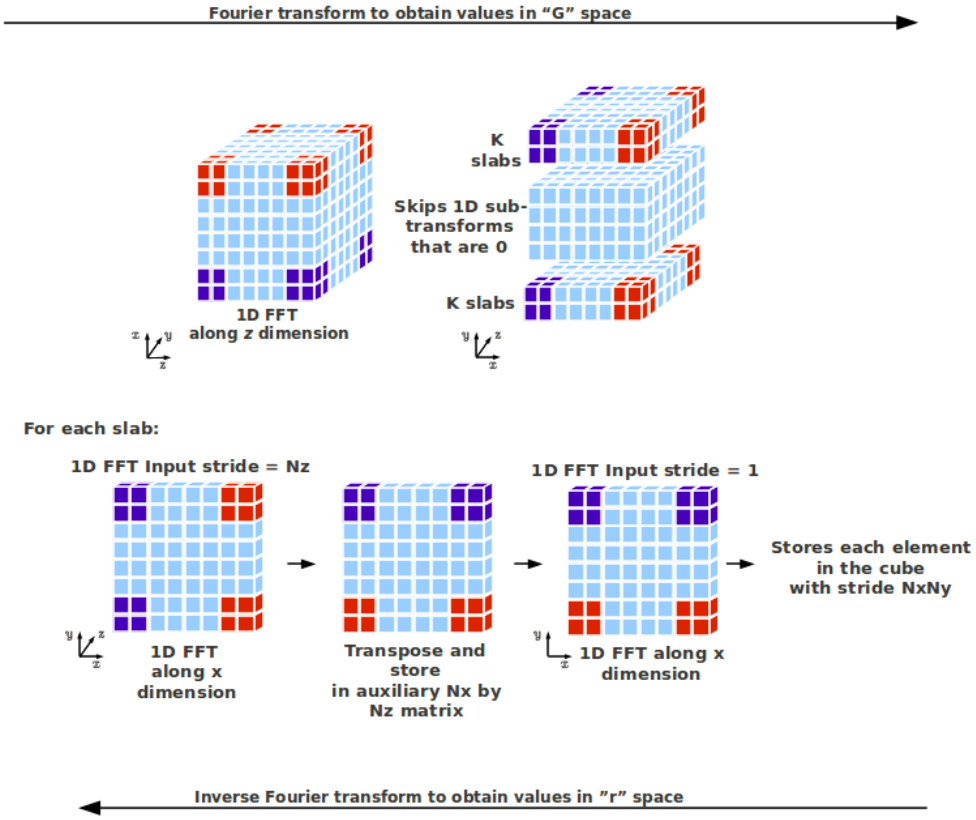


Figure 3.4: Improved FFT. - Blue cubes hold zero values or values that do not matter to the convolution morphology, while the red/magenta ones contain the wanted values after the transformation.

On Fig.3.5 we present the performance offered by the FFT libraries used on our work. The MKL library has an outstanding performance when compared with the other two

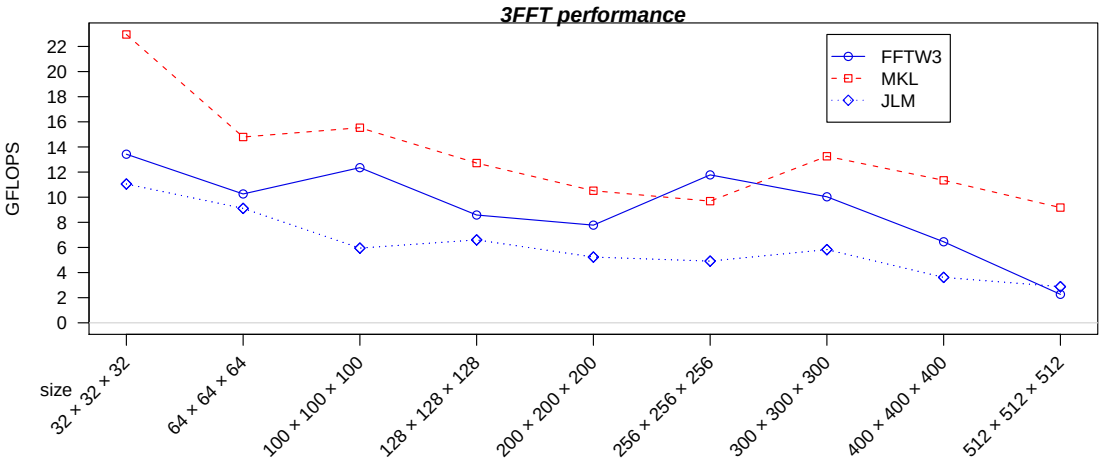


Figure 3.5: Performance of the FFT libraries. - Performance offered by the FFT libraries used in our work. the presented values are obtained when computing the 3D transform without any of the optimizations described so far.

libraries. For the most used FFT sizes during CPW2000 computations, MKL is approximately 2 GFLOPS faster than FFTW3 and JLM private routine. Which on the current CPW2000 version this performance is being slightly wasted due to thread thread safety issue described before.

Using CUDA to compute 3D FFT

The CUDA architecture has been conquering developers in different areas (32) due to the performance offered in some of the most used scientific algorithms (31). Unfortunately, the current cuFFT library (version 4.0) does not support 1D transforms in double precision with a data stride larger than one (24), which does support the implementation of any of the optimizations described so far.

Table 3.1 presents results of computing 3D FFTs in double precision on the CUDA device described in the testbed. The last array (2GB) exceeds the available device memory. The FLOPS count is the one used to benchmark FFT algorithms by reference ($5N\log_2 N$), where N is the number of complex points. This FLOPS count does

Size	CPU→GPU (ms)	GPU→CPU (ms)	Exec Kernel (ms)	Performance (GFLOPS)
$32 \times 32 \times 32$	0.18	0.17	0.07	35.73
$64 \times 64 \times 64$	1.29	1.26	0.33	70.83
$100 \times 100 \times 100$	4.85	4.78	2.05	48.51
$128 \times 128 \times 128$	10.14	10.02	2.69	81.88
$200 \times 200 \times 200$	38.63	38.17	15.73	58.32
$256 \times 256 \times 256$	80.99	80.03	32.63	61.58
$300 \times 300 \times 300$	130.32	128.80	80.63	41.61
$400 \times 400 \times 400$	308.90	308.05	191.12	43.42
$512 \times 512 \times 512$	-	-	-	-

Table 3.1: Computing 3D FFT on CUDA using cuFFT libray.

not correspond to the real FLOP count, but is based on the Cooley-Tukey algorithm asymptotic number of operations (36). The referred values used CUDA timers which have a resolution of approximately half a microsecond (30). The GPU execution of the kernels did not overlap with the the data communication to/from memory.

The performance offered by cuFFT library, without any of the FFT optimizations described before and for the CPW2000 FFT computations (sizes between $64 \times 64 \times 64$ and $128 \times 128 \times 128$ complex points), can be up to 5 times faster (Fig.3.5) which is an acceptable speedup for the application. But looking closer into table 3.1, we can conclude that using the GPU device only to compute FFTs is not very efficient, considering the time that it takes to transfer data into/from the device and the time to compute the FFT (in the 128^3 test case,computing the FFT is 7× faster than the memory transfers). In addition, we have n CPU threads completely idle, which is a waste of resources. We could solve the "idleness" of the CPU threads with a 3DFFT hybrid solution (40, 41), but our problem offers the possibility of "feeding" the GPU with more computations which could minimize the data transfers overhead and maximize the use of data on GPU memory. The strategies that we will adopt to improve CPW2000 overall performance will be based on the idea of using the CPU, GPU or both to compute the $H\psi$ (eq. 2.13).

3.2 Testbed platform

The CPW2000 software package is mainly used by physics scientists, and most of them have easy access to desktop systems but harder access to HPC platforms. However,

current computing devices can offer affordable and powerful configurations to this community. Following this trend, our work seriously considered the hypothesis of exploiting heterogeneous CPU+GPU platforms at desktop level to speed up execution times of intensive algorithms on CPW2000. The selected test platform reflects this approach:

Hardware

For our tests we used a desktop system with the following hardware configuration;

- CPU: Intel ® E5630 with Intel® Hyper-Threading Technology (4 cores, 8 threads) with 2.53 GHz clock frequency (160 GFLOPS theoretical peak). This chip has a multi-level cache: 64 KB L1 cache/core (32 KB L1 Data + 32 KB L1 Instruction), 256 KB L2 cache/core, and a 12 MB L3 cache shared by all cores. The theoretical peak bandwidth between CPU and GPU is 25.6 GB/s,

- Memory: 2 GB DDR3 of memory.

- GPU: NVIDIA® GeForce® 8400 GS used for visualization, and 2 NVIDIA® GeForce® GTX 480 used for CUDA computations. The GTX device has 480 CUDA cores (32 CUDA cores × 15 SM's) operating at 1.4 GHz clock frequency, 64 KB of RAM with a configurable partitioning of shared memory and L1 cache. The GPU board has 1536 MB GDDR5 of memory and the theoretical peak bandwidth between GPU memory and the CPU memory is 177 GB/s.

Software

This operating system is Ubuntu 10.04.3 LTS (64 bit, Linux 2.6.32-33-generic), and we used the following compilers/libraries versions:

- Compilers: Intel Fortran compiler(ifort) and Intel C Compiler(icc) v.12.0.0

- Libraries: Intel Math Kernel Library(MKL) v10.3, AMD Core Math Library (ACML) v5.0.0, FFTW v3.3 , NVIDIA CUDA Toolkit v4.0, JLM FFT private routine (available only by authorization), BLAS NETLIB, cuFFT and cuBLAS.

3.3 Improved versions

Before we present our suggestions to use the CPU to compute $H\psi$, we will discuss some issues to achieve efficient solutions on these devices.

$H\psi$ scalability and parallelism analysis

Our analysis begins with the data structures sizes since we are using a co-processor with its own device memory. Most of these structures are small, KB magnitude, which do not have an impact on performance; others have MB magnitude and have some impact in the system performance because they may have to be transferred, or part of them, to the GPU memory. The structures size may vary in function of N_{eig} and assuming the worst case, which are the sizes given in 2.3.1 and that $M_{xddim} = 50000$, the structures used during $H\psi$ computations are:

- $PSIR/PSII(M_{xddim}, N_{eig})$: *PSIR/PSII*(*i*,*j*) is the real/imaginary part of the j-*th* component of wave vector i and requires approximately 65 MB of memory each.

- $HPSIR/HPSII(M_{xddim}, N_{eig})$: *HPSII/HPSII*(*i*,*j*) is the real/imaginary part of the j-*th* component of the product of the Hamiltonian by the wave vector i and requires approximately 65 MB of memory each.

- $ANLGAR/ANLGAI(M_{xddim}, N_{anl})$ real/imaginary part of the separable pseudopotential matrix and requires approximately 110 MB of memory each.

Using the GPU to compute/assist in the $H\psi$ computation, almost 500 MB data must be transferred to/from the GPU memory, which may hide any performance improvements due to fast GPU computing. Also this can be a large value for GPU memory if we want our code improvements to execute in different GPU families. Considering that CPW2000 computations are in double precision, we are only interested on CUDA computing devices which support double precision arithmetic, and all the devices which support this arithmetic have at least 512 MB memory.

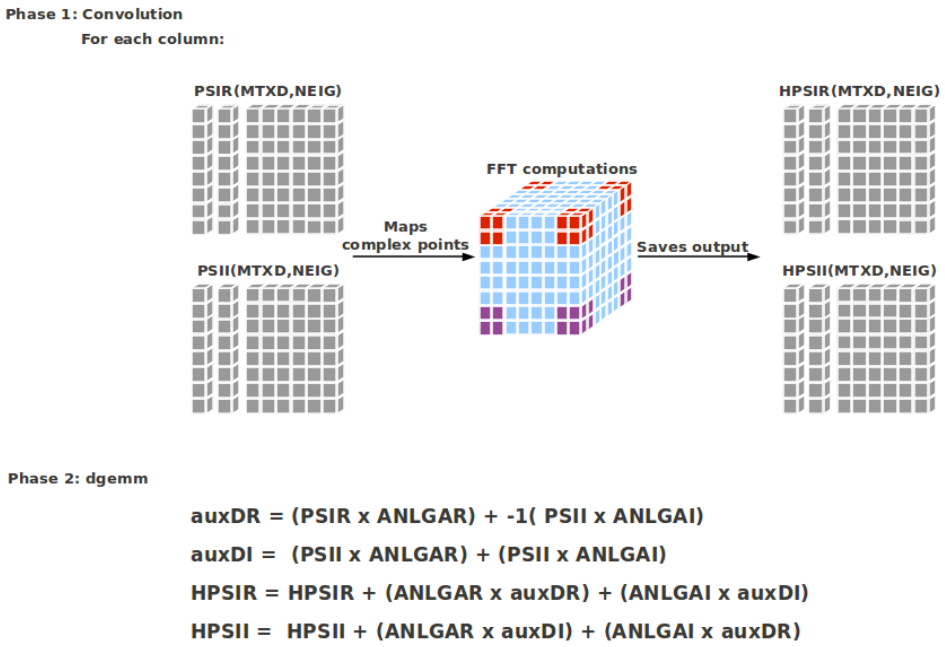


Figure 3.6: Hψ data flow diagram. - Top level overview of $H\psi$ execution.

On Chap.2 we presented the $H\psi$ formal definition. We will now analyse the computational implementation of this function.

Beyond the execution steps of the function, Fig 3.6 presents some interesting facts. The first phase is embarrassingly parallel if we apply parallelism on columns, which has two performance bottlenecks: our work was developed on the C language, so matrices are stored in row major and mapping to/into the 3D structure would be a non-coalescent access to the matrices elements. This could be easily solved by letting the first phase to be computed in the original CPW2000 version (written in Fortran, matrices stored in column-major). The other fact is that each column would have to be computed by a single thread, which is a naive scheduling policy since the workload of every thread would be extremely high and some of the available FFT libraries are specially tuned for parallelism.

The second phase Fig. 3.6, which is efficiently computed by BLAS libraries, can be divided into two phases as seen in Fig. 3.7. This approach gives two computational **independent** and **dependencies free** phases: *convolution phase* and *dgemm phase* from now on, each one with heavy algorithms on large data structures, but completely

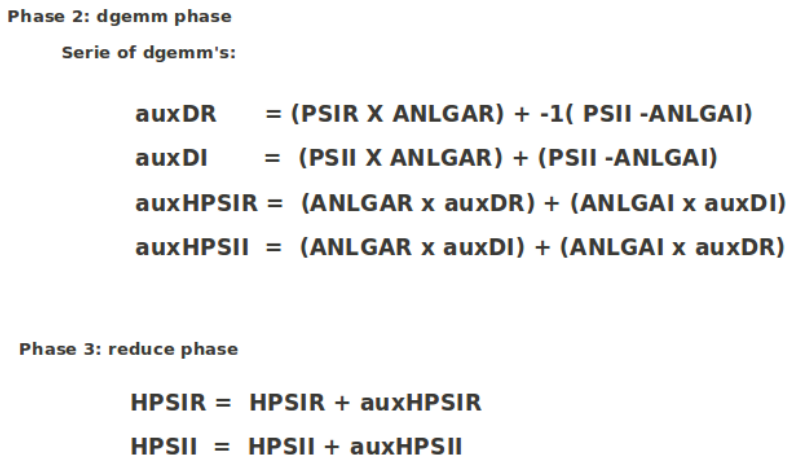


Figure 3.7: Separable computing phases - Separable computing phases, first and second phase have no dependencies between them. Third phase must be the last computed.

independent of each other. The last phase (*reduce phase*) just adds the results and must be the last computed phase. Identifying these computing stages is an opportunity to design more efficient scheduling policies between different devices.

$H\psi$ in CUDA

Our first proposed solution, uses the CUDA model to compute $H\psi$.

Fig. 3.8 illustrates the work load among devices through time but it can demonstrate an interesting fact, that this implementation wastes resources and exhibits a deficient work load policy: we have all the CPU threads idle (except for some small time fractions). To take advantage on the embarrassingly parallel convolution on the columns and the massive parallelism present on CUDA devices, our initial approach was to schedule each *complex* column (one column from *PSII* and another column from *PSIR*) of the matrices (Fig. 3.6) to a block of threads (choosing the best CUDA block dimension to achieve the best block configuration to maximize throughput). This would lead to a full independent and efficient CUDA solution (independent in a way that it only needs the CPU to initiate memory transfers and launch the kernel). Un-

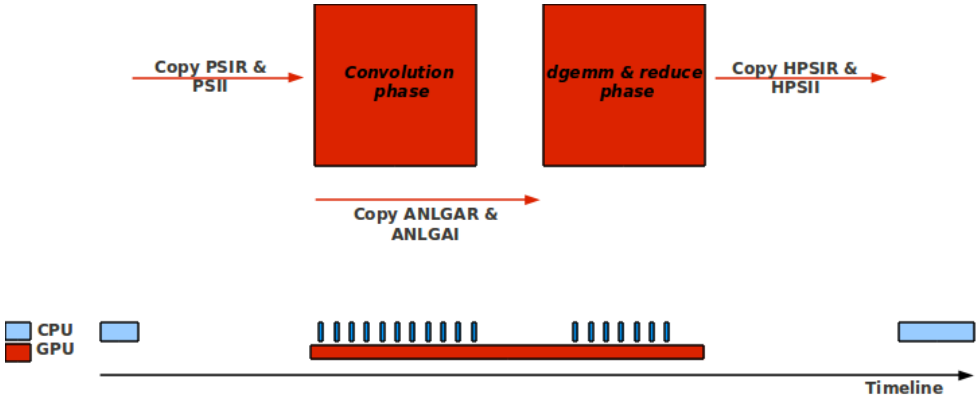


Figure 3.8: Hψ CUDA data flow diagram - Graphical representation of $H\psi$ execution, showing the work load among devices through time.

fortunately, the kernel launchers on cuFFT API and on the cuBLAS API can only be called on the host side (24, 42), which lead to this implementation where the CPU is managing the kernel launches for each column and the small work fractions of the CPU are related with these launches (these small work fractions are so small that they should not be considered at all). Considering the complexity and the length of the FFT computation, the eficiency offered by cuFFT (table 3.1), and the operations in large structures that are done among the FFT computations 2.14, we can state that the workload is well distributed among the GPU blocks.

FFT structure size	**PSIR & PSII (ms)**	**ANLGAR & ANLGAI (ms)**	***convolution* phase (ms)**	***dgemm* & *reduce* phase (ms)**	***HPSIR* & HPSII (ms)**
$100 \times 100 \times 100$	23.72	40.57	908.91	417.49	24.92

Table 3.2: Data copies timings of single Hψ execution in CUDA - Transfer times of the major structures. Only one FFT size is presented, the most common during CPW2000 computations.

Since *ANLGAR* and *ANLGAI* are only required for *dgemm* and *reduce* phase, their copy latency is fully hidden by the *convolution* phase (table 3.2). This is possible due to the overlap execution with memory copies feature of CUDA devices (30). Latencies that can not be hidden are the ones associated with the initial and final copy, where any of the devices is performing computations. As can be seen on table 3.2 these latencies,

on our testbed, last approximately 50 ms.

Split phases Hψ

The above approach did not efficiently use the computational resources. Inspired on Figs. 3.6 and 3.7, on the analysis made on the non-dependence between *convolution* and *dgemm phase* and specially on the intensive computation work present on each *phase*, we can assign to different devices each of the independent *phases*.

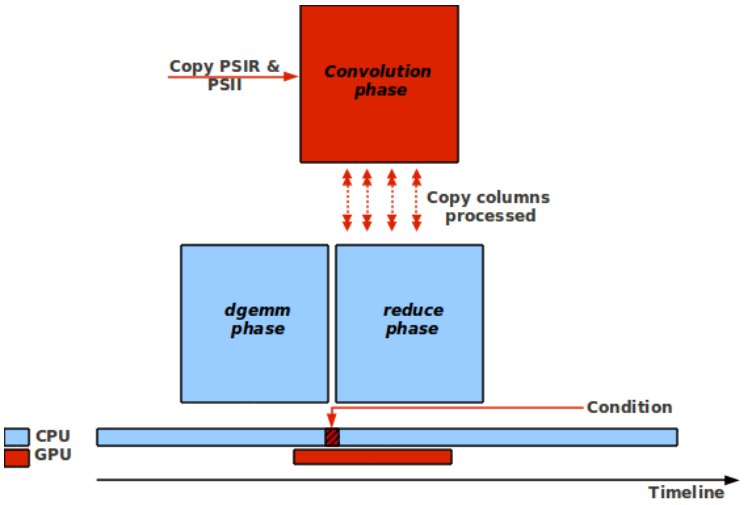


Figure 3.9: Split phases Hψ CUDA data flow diagram - Graphical representation of $H\psi$ execution, showing the work load among devices through time.

One CPU thread initiates the copy to GPU device (Fig. 3.9) and it will be responsible for managing the *convolution phase* context (similar to the previous solution - $H\psi$ in CUDA). This includes signalling columns already computed and copying back each one to the host side (overlap execution with memory transfers (30)). Meanwhile, the remaining threads start computing the *dgemm phase* and when finished we have a *reduce synchronized phase*. It starts *reducing* the columns that were already computed by the GPU and it may happen that all the columns of the *convolution phase* have been already computed and copied back to the host side, but if all the columns have not been yet computed by the GPU we need to synchronize this *reduce phase*(CPU just waits for the next column).

This solution is more efficient than the previous one: it uses a significant part of the platform computing power during the most intensive computational step of the CPW2000 applicaton (table 2.2). However, this exposed a major flaw: it is **architecture dependent**. This dependency is evidenced by the *condition* on Fig. 3.9 and it can be defined by:

$$cond: \qquad T_{CPUdgemm} \leq T_{memcpy} + T_{convolution} \tag{3.1}$$

i.e, if the time that it takes for the CPU to compute the *dgemm phase* is larger than the time it takes to copy the data to GPU and compute the *convolution phase* or if CPU computes the *dgemm phase* extremely faster, than this solution is no better than $\mathbf{H}\boldsymbol{\psi}$ in CUDA (previous solution). And we can easily identify some CPU architecture features for the to condition fail: a CPU with fewer threads or less performant (first multi-core generations for example), lower bandwidth (PCIe 1.1 for example) or even less efficient BLAS library (this one is not CPU architecture feature, but it also makes the condition fail).

Scalable $H\psi$

The two solutions presented before can be mixed to obtain a scalable version across all the devices present in the computational platform and therefore achieve better efficiency on these devices. To execute in n devices, we need to further analyse the three *phases* presented before (Figs. 3.6 and 3.7): a scalable convolution *phase* may be implemented by dividing *PSIR* and *PSII* in n equal data chunks and each device would compute n columns of the convolution. In the remaining *phases* we can use a block matrix approach in a way that each device after computing its n part of the convolution could start computing the corresponding n blocks of the *dgemm* and *reduce* phases.

Fig. 3.10 presents a graphical overview of our strategy. However, our strategy includes slight modifications to minimize communications. The computing power of each device in different computational platforms may vary. Currently, CUDA enabled devices have larger floating point operations throughput than current CPU (specially the current CUDA generation -*Fermi*) so, instead of dividing in n **equal** data chunks, we perform this block division according to the computing power of each device. Assume,

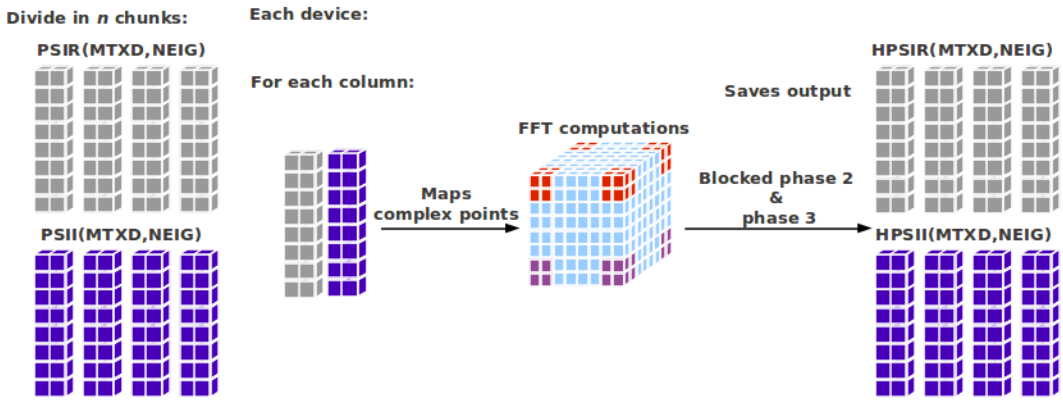


Figure 3.10: Scalable Hψ data flow diagram -

for example that the GPU is 5 times faster computing **Hψ** than other devices. Then, according with our strategy, the GPU should have a data chunk approximately 5 times larger than CPU. This strategy reduces the memory transfers latencies overhead since the GPU does not need to be constantly fetching data blocks for the next computations.

Another possible implementation is to split the data structures in a larger number of blocks, place these blocks in a pool or a queue, and each device picks a block for computation. The memory transfers between the main memory and the GPU memory can be hidden by overlapping data chunk computations with memory transfers (30). This strategy has some overheads - dividing structures, synchronization of each block already computed - that can be avoided: when compiling the library code, a embedded profiler assesses the relative performance between devices so that we may tune the sizes of the data chunks.

3.4 Overall performance assessment

We will now present the results obtained with our solutions and we will comment the obtained results. To distinguish our optimization from the libraries used in 2.1 we will call **SLABv** to our FFT optimization. We will begin our performance analyses by accessing our FFT approach results.

Performance results of the FFT optimizations

Since the FFT optimization in CPW2000 has a thread safety issue with the FFT routines of the MKL library, we made the decision of not presenting benchmark results for this library. Our comparison model will be based on the original optimization used with the FFTW3 routines. However, timings of CPW2000 executions with other libraries will be used in order to quantify our gains.

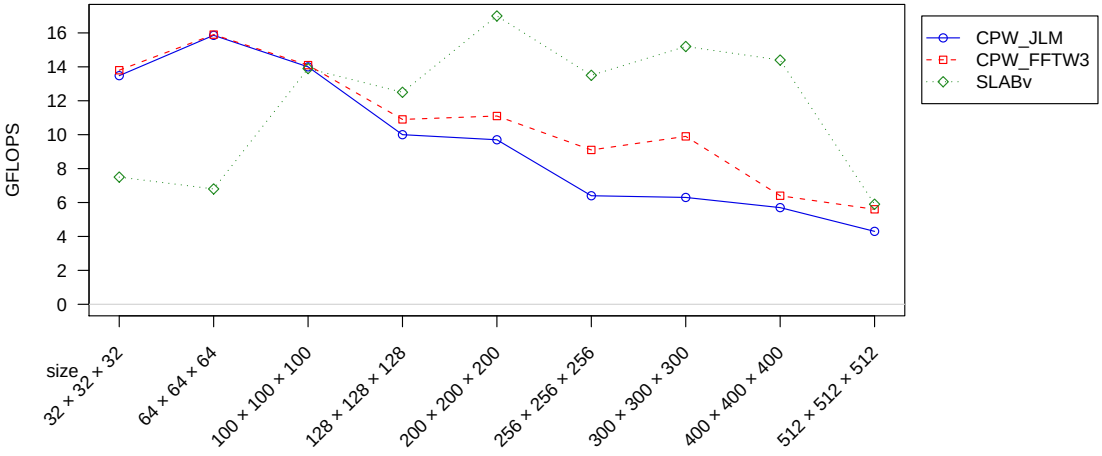


Figure 3.11: Comparison of the different FFT approaches. - *CPW_JLM* and *CPW_FFTW3* stands for the FFT optimization present in the original version of the program when used with the FFT libraries JLM and FFTW3.

Size	CPW_JLM (GFLOPS)	CPW_FFTW3 (GFLOPS)	SLABv (GFLOPS)
32 × 32 × 32	13.4	13.8	7.5
64 × 64 × 64	15.8	15.9	6.8
100 × 100 × 100	14	14.1	13.9
128 × 128 × 128	10	10.9	12.5
200 × 200 × 200	9.7	11.1	17.0
256 × 256 × 256	6.4	9.1	13.5
300 × 300 × 300	6.3	9.9	15.2
400 × 400 × 400	5.7	6.4	14.4
512 × 512 × 512	4.3	5.6	5.9

Table 3.3: Comparison of the different FFT approaches.

Our approach behaves well for large array sizes, particularly those larger than 128^3

complex points (Fig.3.11), which entirely match expectations, our evaluation of memory coalescent accesses and the reuse of data on cache. With these results we strongly believe that for CPU architectures with smaller caches size our approach may lead to real gains. Unfortunately, for the array sizes that CPW2000 more frequently uses and with the cache size of our testbed, our solution is a little disappointing considering that it does not improve the performance of the FFT computations when compared to the optimizations on CPW2000.

FFT size	**Hψ MKL + JLM (s)**	**Hψ ACML + FFTW3 (s)**	**Hψ ACML + SLABv (s)**
$100 \times 100 \times 100$	5.5	6.1	6.3

Table 3.4: Execution times of a single **Hψ** execution.

Table 3.4 presents the fact that our solution is not better that the one present in CPW2000. Our approach results, in a single **Hψ** execution, is slightly worst than the original optimization. This slight overhead is probably related with the number of calls to 1D FFT routines which, according to our analysis (chapter3.1), has a significant number of calls that may deteriorate performance.
Our solution does not beat the one present in the original version of the program; however it allows CPW2000 to use the performance gains (Fig. 3.5) that MKL clearly offers, leaving behind the strange thread safety problem that was originally present in the program that we were not able to solve yet. And this benefit can be translated into real and significant performance boost in the system overall performance.

CUDA based solutions performance results

The most significant performance gains arise from the use of the massive parallelism available on CUDA capable devices (table 3.5).

Our analysis of the scalability and functioning of **Hψ** function translated to a better distribution of the workload among the different devices of the computational platform leading to a 5x speedup on the **scalable Hψ** approach when compared with CPW2000

FFT structure size	Hψ MKL + JLM (s)	Hψ CUDA (s)	Hψ split (s)	Hψ scalable (s)
$100 \times 100 \times 100$	5.5	1.5	1.3	1.1

Table 3.5: Execution times for a single $\mathbf{H}\psi$ execution using CUDA.

original fastest version. This is not a remarkable application speedup of tens of magnitude. However, the significant achievementto be noted are reduced communication strategy and a balanced distribution of the workload, according to each device performance. This lead to an efficient version faster than a full CUDA implementation.
Note that the FFT optimizations described in this work could not be implemented in current CUDA version, since it does not support FFT computations of non contiguous data points in double precision (24). In future CUDA versions, where it is expected this feature, the application will be able to benefit from the FFT optimizations described in this work, and a extra performance boost may be achieved.

		MCr		2nd MCr		
	BLAS + FFT	**Routine**	**Time(%)**	**Routine**	**Time(%)**	**Time(s)**
seq.						
	Netlib + JLM	DGEMM	87.7	FFT	9.9	12464
	ACML + FFTW3	DGEMM	66.9	FFT	29.6	5256
	MKL + JLM	DGEMM	56.4	FFT	35.6	3475
par.						
	MKL + MKL	FFT	60.4	DGEMM	30.7	1858
	ACML + FFTW3	DGEMM	57.6	FFT	33.7	1773
	MKL + JLM	DGEMM	49.8	FFT	38.9	1372
	ACML + SLABv	**DGEMM**	**56.6**	**FFT**	35.4	**1815**
	MKL + SLABv	**DGEMM**	**48.8**	**FFT**	**38.3**	**1169**
CUDA						
	Hψ CUDA	**DGEMM**	**58.2**	**others**	**41.8**	**874**
	Split Hψ w/ MKL + JLM	**DGEMM**	**80.6**	**others**	**19.4**	**852**
	Scalable Hψ w/ MKL + SLABv(MKL)	**DGEMM**	**69.7**	**others**	**30.3**	**829**

Table 3.6: CPW2000 final profiling

The final profiling table 3.6 summarizes our initial proposed work to improve CPW2000 overall performance, and gives us some hints for future optimizations. The overall performance is still beyond our initial expectations, since we only made an attempt to improve $\mathbf{H}\psi$. Once this function has its efficiency improved, other li-

braries/routines that did not seem to have a significant performance impact, will start to have some meaning, and are candidates for further analysis.

	DGEMM + FFT	H_psi(s)	rest of the program(s)
seq.			
	Netlib + JLM	3386	9078
	ACML + JLM	1898	3016
	MKL + JLM	1632	1843
par.			
	MKL + MKL	1232	626
	ACML + FFTW3	798	975
	MKL + JLM	664	708
	ACML + SLABv	**837**	**978**
	MKL + SLABv	**551**	**618**
CUDA			
	Hψ CUDA	**166**	**708**
	Split Hψ w/ MKL + JLM	**145**	**707**
	Scalable Hψ w/ MKL + SLABv(MKL)	**121**	**708**

Table 3.7: Routines time consumption distribution. - Final Routines time distribution during CPW2000 execution.

Finall, we must re-evaluate our profiling results keeping in mind the so helpful corollaries derived from Amdahl's law (29) presented by tables 3.7 and 3.6:: when using CUDA devices, $\mathbf{H}\boldsymbol{\psi}$ is not the most resources consuming routine in CPW2000 and is probably one of the least time consuming. This allows us to extend our efficiency analysis to other software modules and attempt to improve these modules efficiency. Assuming that these modules can be optimized, the use of devices CUDA in our work may help not only to boost the application performance, but also to improve further the efficiency of the program when executed on a computational platform without CUDA enabled GPUs.

4

Conclusions

4.1 Summary

From a high performance computing point of view the computational methods used in a software that simulates a physical state are intensive and resources consuming. Though the hardly tuned libraries and the different computational platforms available, the computational work in CPW2000 continues to be extremely time and resources consuming. This initial work provided us a sensibility of the impact that an algorithm optimization can, or can not, have on computational science.

In this initial work we detected that a single routine, $H\psi$, consumes nearly 50% of the total execution time and this percentage is most spent in FFT computations on convolutions. We have identified a key feature in the FFT structure, this structure is very sparse due to the nature of the convolution, that could be an opportunity to improve the overall program efficiency. Our approach to minimize the time spent in FFT computations based on the structure key feature, and to develop an architecture independent solution did not produce the expected results. Consequently, we must conclude that the FFT issue is still present on the software. However, the attempted approaches are fully documented and detailed in this document to assist future researchers having a description of the approaches that did not produce the expected FFT efficiency improvement.

In a second phase of our work, we have identified and characterized $H\psi$ function and presented different approaches to implement parallelism on current computing devices.

We achieved an interesting application speedup when computing $H\psi$ with an efficient scalable solution using CUDA enabled devices and CPU. These results are helpful since they evidence the approximate 50% computing time of the rest of the program that did not deserve before a special attention.

4.2 Future work

This dissertation work is part of the first phase of a total of two phases related with the CPW2000 software. This first phase is only related with the computing aspects, specially, the software efficiency. Our initial work showed interesting features available in the software package when using CUDA enabled devices. Since our developed code is in a development and testing stage, we should prepare it to fit in the library to be ready for distribution.

Until the thread safety issue present in the original version is solved, we propose our FFT optimization. A user, that has the possibility, may use the FFT performance boost offered by MKL commercial library. However, respecting the CPW2000 feature of using different FFT libraries, we must develop special *wrappers* for our proposed solution.

Considering the methods that CPW2000 uses and analysing the performance results that we obtained with CUDA, we should extend our analysis to the rest of the program. A top level view of the application algorithms that we did not pay special attention, gives a hint that the most algorithms are on linear algebra. These algorithms may fit well on a SIMD architecture, well present in GPU devices, to implement parallelism and achieve an application *time to the solution* improvement. We are particularly interested to analyse these algorithms when implemented in CUDA, considering the performance boost that we achieved in our work, which have not been higher due to a library limitation (cuFFT).

The second phase of a future work, having a stable and the most possible efficient version of the software, is to give the software and its main developer a dignified place in scientific libraries. Dignified in our opinion due to two important aspects: (i) the application age, it is almost 20 years old and it includes optimizations to achieve the

best efficiency on a Cray supercomputer (available at the end of past century); (ii) it has been contributing to other scientists work for the past years.

5

Appendix A

CPW2000 sample output for C60

We present a sample output to illustrate some of the results that CPW2000 computes.

```
    DENSITY-FUNCTIONAL PSEUDOPOTENTIAL PLANE-WAVE PROGRAM VERSION 4.51
    RUN ON THE 22-JUN-11 AT 03:42:12


       C60



    SINGLE GEOMETRY CALCULATION



 CRYSTAL STRUCTURE:

 LATTICE CONSTANT     27.71000 (A.U.)


 PRIMITIVE TRANSLATION VECTORS
                       IN A.U.                              IN LATTICE UNITS
 A1=  0.000000E+00  0.138550E+02  0.138550E+02         0.000    0.500    0.500
 A2=  0.138550E+02  0.000000E+00  0.138550E+02         0.500    0.000    0.500
 A3=  0.138550E+02  0.138550E+02  0.000000E+00         0.500    0.500    0.000

 NO. TYPE     POSITION(LATTICE COORD.)          POSITION(CARTESIAN COORD.)              MASS

  1   C       0.280    0.189   -0.283     -.130671E+01  -.397084E-01  0.649180E+01       12.011
  2   C      -0.029    0.189    0.222     0.568789E+01  0.267170E+01  0.220861E+01       12.011
  3   C      -0.222    0.381    0.029     0.568789E+01  -.267170E+01  0.220861E+01       12.011
  4   C       0.283    0.069    0.029     0.136499E+01  0.432290E+01  0.488031E+01       12.011
  5   C      -0.029    0.381   -0.283     0.136499E+01  -.432290E+01  0.488031E+01       12.011
  6   C       0.189    0.283   -0.189     0.130671E+01  0.000000E+00  0.653151E+01       12.011
  7   C       0.189   -0.029   -0.381     -.568789E+01  -.267170E+01  0.220861E+01       12.011
  8   C       0.381   -0.222   -0.189     -.568789E+01  0.267170E+01  0.220861E+01       12.011
  9   C       0.381   -0.029   -0.069     -.136499E+01  0.432290E+01  0.488031E+01       12.011
 10   C       0.069    0.283   -0.381     -.136499E+01  -.432290E+01  0.488031E+01       12.011
 11   C      -0.283   -0.189    0.283     0.130671E+01  0.000000E+00  -.653151E+01       12.011
 12   C       0.029   -0.189   -0.222     -.568789E+01  -.267170E+01  -.220861E+01       12.011
 13   C       0.222   -0.381   -0.029     -.568789E+01  0.267170E+01  -.220861E+01       12.011
 14   C       0.029   -0.381    0.283     -.136499E+01  0.432290E+01  -.488031E+01       12.011
 15   C      -0.283   -0.069   -0.029     -.136499E+01  -.432290E+01  -.488031E+01       12.011
```

5. APPENDIX A

```
 16   C      -0.189   -0.283    0.189     -.130671E+01  0.000000E+00  -.653151E+01       12.011
 17   C      -0.189    0.029    0.381     0.568789E+01  0.267170E+01  -.220861E+01       12.011
 18   C      -0.381    0.222    0.189     0.568789E+01  -.267170E+01  -.220861E+01       12.011
 19   C      -0.069   -0.283    0.381     0.136499E+01  0.432290E+01  -.488031E+01       12.011
 20   C      -0.381    0.029    0.069     0.136499E+01  -.432290E+01  -.488031E+01       12.011
 21   C      -0.283    0.283    0.189     0.653151E+01  -.130671E+01  0.000000E+00       12.011
 22   C       0.029    0.283    0.069     0.488031E+01  0.136499E+01  0.432290E+01       12.011
 23   C      -0.283   -0.029    0.381     0.488031E+01  0.136499E+01  -.432290E+01       12.011
 24   C       0.222   -0.029    0.189     0.220861E+01  0.568789E+01  0.267170E+01       12.011
 25   C       0.029   -0.222    0.381     0.220861E+01  0.568789E+01  -.267170E+01       12.011
 26   C      -0.189    0.189    0.283     0.653151E+01  0.130671E+01  0.000000E+00       12.011
 27   C      -0.069    0.381   -0.029     0.488031E+01  -.136499E+01  0.432290E+01       12.011
 28   C      -0.381    0.069    0.283     0.488031E+01  -.136499E+01  -.432290E+01       12.011
 29   C      -0.381    0.189   -0.029     0.220861E+01  -.568789E+01  -.267170E+01       12.011
 30   C      -0.189    0.381   -0.222     0.220861E+01  -.568789E+01  0.267170E+01       12.011
 31   C       0.283   -0.283   -0.189     -.653151E+01  0.130671E+01  0.000000E+00       12.011
 32   C       0.283    0.029   -0.381     -.488031E+01  -.136499E+01  0.432290E+01       12.011
 33   C      -0.029   -0.283   -0.069     -.488031E+01  -.136499E+01  -.432290E+01       12.011
 34   C      -0.222    0.029   -0.189     -.220861E+01  -.568789E+01  -.267170E+01       12.011
 35   C      -0.029    0.222   -0.381     -.220861E+01  -.568789E+01  0.267170E+01       12.011
 36   C       0.189   -0.189   -0.283     -.653151E+01  -.130671E+01  0.000000E+00       12.011
 37   C       0.381   -0.069   -0.283     -.488031E+01  0.136499E+01  0.432290E+01       12.011
 38   C       0.069   -0.381    0.029     -.488031E+01  0.136499E+01  -.432290E+01       12.011
 39   C       0.381   -0.189    0.029     -.220861E+01  0.568789E+01  0.267170E+01       12.011
 40   C       0.189   -0.381    0.222     -.220861E+01  0.568789E+01  -.267170E+01       12.011
 41   C       0.189   -0.283    0.283     0.000000E+00  0.653151E+01  -.130671E+01       12.011
 42   C       0.069    0.029    0.283     0.432290E+01  0.488031E+01  0.136499E+01       12.011
 43   C       0.381   -0.283   -0.029     -.432290E+01  0.488031E+01  0.136499E+01       12.011
 44   C       0.189    0.222   -0.029     0.267170E+01  0.220861E+01  0.568789E+01       12.011
 45   C       0.381    0.029   -0.222     -.267170E+01  0.220861E+01  0.568789E+01       12.011
 46   C       0.283   -0.189    0.189     0.000000E+00  0.653151E+01  0.130671E+01       12.011
 47   C      -0.029   -0.069    0.381     0.432290E+01  0.488031E+01  -.136499E+01       12.011
 48   C       0.283   -0.381    0.069     -.432290E+01  0.488031E+01  -.136499E+01       12.011
 49   C      -0.029   -0.381    0.189     -.267170E+01  0.220861E+01  -.568789E+01       12.011
 50   C      -0.222   -0.189    0.381     0.267170E+01  0.220861E+01  -.568789E+01       12.011
 51   C      -0.189    0.283   -0.283     0.000000E+00  -.653151E+01  0.130671E+01       12.011
 52   C      -0.381    0.283    0.029     0.432290E+01  -.488031E+01  -.136499E+01       12.011
 53   C      -0.069   -0.029   -0.283     -.432290E+01  -.488031E+01  -.136499E+01       12.011
 54   C      -0.189   -0.222    0.029     -.267170E+01  -.220861E+01  -.568789E+01       12.011
 55   C      -0.381   -0.029    0.222     0.267170E+01  -.220861E+01  -.568789E+01       12.011
 56   C      -0.283    0.189   -0.189     0.000000E+00  -.653151E+01  -.130671E+01       12.011
 57   C      -0.283    0.381   -0.069     0.432290E+01  -.488031E+01  0.136499E+01       12.011
 58   C       0.029    0.069   -0.381     -.432290E+01  -.488031E+01  0.136499E+01       12.011
 59   C       0.029    0.381   -0.189     0.267170E+01  -.220861E+01  0.568789E+01       12.011
 60   C       0.222    0.189   -0.381     -.267170E+01  -.220861E+01  0.568789E+01       12.011


POTENTIALS :
------------

C   ca  nrl  nc
atom 5.69 29-JAN-08   Improved Troullier - Martinskb-loc= 0
2s(  2.00)  rc= 1.292p(  2.00)  rc= 1.29
NQL=1600 DELQL=0.015


    LOCAL DENSITY APROXIMATION USING CEPERLEY AND ALDER AS PARAMETRIZED BY PERDEW AND ZUNGER

    ENERGY CUTOFF FOR WAVE-FUNCTION KINETIC ENERGY IS     32.00 HARTREE

    SCF IS CONVERGED IF DIFFERENCE IN POTENTIALS IS LESS THEN   0.00050000
    FOR ATOMIC ORBITALS SCF PARAMETER IS   0.00050000
    ITERATIVE DIAGONALIZATION IS CONVERGED IF ERROR IN |H PSI - E PSI| IS LESSTHEN   0.00100000

    THE TEMPERATURE FOR ELECTRON FERMI DISTRIBUTION IS      0.00 KELVIN



    PLANE-WAVE BASIS CALCULATION
```

```
     5319.24000275      volume

REAL-SPACE METRIC

       383.92205000     191.96102500     191.96102500      metric  g11,g12,g13
                        383.92205000     191.96102500      metric      g22,g23
                                         383.92205000      metric          g33

    19.59392891      19.59392891      19.59392891   length 1,2,3  (A.U.)

    60.00000000      60.00000000      60.00000000   angle 12,13,23 (DEGREES)

       POSITION (LATTICE COORD.)         POSITION (CARTESIAN COORD. A.U.)     NO. TYPE

     0.28000    0.18855   -0.28287     -.13067E+01  -.39708E-01  0.64918E+01   1   C     position
    -0.02914    0.18855    0.22198     0.56879E+01  0.26717E+01  0.22086E+01   2   C     position
    -0.22198    0.38139    0.02914     0.56879E+01  -.26717E+01  0.22086E+01   3   C     position
     0.28287    0.06938    0.02914     0.13650E+01  0.43229E+01  0.48803E+01   4   C     position
    -0.02914    0.38139   -0.28287     0.13650E+01  -.43229E+01  0.48803E+01   5   C     position
     0.18855    0.28287   -0.18855     0.13067E+01  0.00000E+00  0.65315E+01   6   C     position
     0.18855   -0.02914   -0.38139     -.56879E+01  -.26717E+01  0.22086E+01   7   C     position
     0.38139   -0.22198   -0.18855     -.56879E+01  0.26717E+01  0.22086E+01   8   C     position
     0.38139   -0.02914   -0.06938     -.13650E+01  0.43229E+01  0.48803E+01   9   C     position
     0.06938    0.28287   -0.38139     -.13650E+01  -.43229E+01  0.48803E+01  10   C     position
    -0.28287   -0.18855    0.28287     0.13067E+01  0.00000E+00  -.65315E+01  11   C     position
     0.02914   -0.18855   -0.22198     -.56879E+01  -.26717E+01  -.22086E+01  12   C     position
     0.22198   -0.38139   -0.02914     -.56879E+01  0.26717E+01  -.22086E+01  13   C     position
     0.02914   -0.38139    0.28287     -.13650E+01  0.43229E+01  -.48803E+01  14   C     position
    -0.28287   -0.06938   -0.02914     -.13650E+01  -.43229E+01  -.48803E+01  15   C     position
    -0.18855   -0.28287    0.18855     -.13067E+01  0.00000E+00  -.65315E+01  16   C     position
    -0.18855    0.02914    0.38139     0.56879E+01  0.26717E+01  -.22086E+01  17   C     position
    -0.38139    0.22198    0.18855     0.56879E+01  -.26717E+01  -.22086E+01  18   C     position
    -0.06938   -0.28287    0.38139     0.13650E+01  0.43229E+01  -.48803E+01  19   C     position
    -0.38139    0.02914    0.06938     0.13650E+01  -.43229E+01  -.48803E+01  20   C     position
    -0.28287    0.28287    0.18855     0.65315E+01  -.13067E+01  0.00000E+00  21   C     position
     0.02914    0.28287    0.06938     0.48803E+01  0.13650E+01  0.43229E+01  22   C     position
    -0.28287   -0.02914    0.38139     0.48803E+01  0.13650E+01  -.43229E+01  23   C     position
     0.22198   -0.02914    0.18855     0.22086E+01  0.56879E+01  0.26717E+01  24   C     position
     0.02914   -0.22198    0.38139     0.22086E+01  0.56879E+01  -.26717E+01  25   C     position
    -0.18855    0.18855    0.28287     0.65315E+01  0.13067E+01  0.00000E+00  26   C     position
    -0.06938    0.38139   -0.02914     0.48803E+01  -.13650E+01  0.43229E+01  27   C     position
    -0.38139    0.06938    0.28287     0.48803E+01  -.13650E+01  -.43229E+01  28   C     position
    -0.38139    0.18855   -0.02914     0.22086E+01  -.56879E+01  -.26717E+01  29   C     position
    -0.18855    0.38139   -0.22198     0.22086E+01  -.56879E+01  0.26717E+01  30   C     position
     0.28287   -0.28287   -0.18855     -.65315E+01  0.13067E+01  0.00000E+00  31   C     position
     0.28287    0.02914   -0.38139     -.48803E+01  -.13650E+01  0.43229E+01  32   C     position
    -0.02914   -0.28287   -0.06938     -.48803E+01  -.13650E+01  -.43229E+01  33   C     position
    -0.22198    0.02914   -0.18855     -.22086E+01  -.56879E+01  -.26717E+01  34   C     position
    -0.02914    0.22198   -0.38139     -.22086E+01  -.56879E+01  0.26717E+01  35   C     position
     0.18855   -0.18855   -0.28287     -.65315E+01  -.13067E+01  0.00000E+00  36   C     position
     0.38139   -0.06938   -0.28287     -.48803E+01  0.13650E+01  0.43229E+01  37   C     position
     0.06938   -0.38139    0.02914     -.48803E+01  0.13650E+01  -.43229E+01  38   C     position
     0.38139   -0.18855    0.02914     -.22086E+01  0.56879E+01  0.26717E+01  39   C     position
     0.18855   -0.38139    0.22198     -.22086E+01  0.56879E+01  -.26717E+01  40   C     position
     0.18855   -0.28287    0.28287     0.00000E+00  0.65315E+01  -.13067E+01  41   C     position
     0.06938    0.02914    0.28287     0.43229E+01  0.48803E+01  0.13650E+01  42   C     position
     0.38139   -0.28287   -0.02914     -.43229E+01  0.48803E+01  0.13650E+01  43   C     position
     0.18855    0.22198   -0.02914     0.26717E+01  0.22086E+01  0.56879E+01  44   C     position
     0.38139    0.02914   -0.22198     -.26717E+01  0.22086E+01  0.56879E+01  45   C     position
     0.28287   -0.18855    0.18855     0.00000E+00  0.65315E+01  0.13067E+01  46   C     position
    -0.02914   -0.06938    0.38139     0.43229E+01  0.48803E+01  -.13650E+01  47   C     position
     0.28287   -0.38139    0.06938     -.43229E+01  0.48803E+01  -.13650E+01  48   C     position
    -0.02914   -0.38139    0.18855     -.26717E+01  0.22086E+01  -.56879E+01  49   C     position
    -0.22198   -0.18855    0.38139     0.26717E+01  0.22086E+01  -.56879E+01  50   C     position
    -0.18855    0.28287   -0.28287     0.00000E+00  -.65315E+01  0.13067E+01  51   C     position
    -0.38139    0.28287    0.02914     0.43229E+01  -.48803E+01  -.13650E+01  52   C     position
    -0.06938   -0.02914   -0.28287     -.43229E+01  -.48803E+01  -.13650E+01  53   C     position
    -0.18855   -0.22198    0.02914     -.26717E+01  -.22086E+01  -.56879E+01  54   C     position
    -0.38139   -0.02914    0.22198     0.26717E+01  -.22086E+01  -.56879E+01  55   C     position
```

5. APPENDIX A

```
 -0.28287    0.18855   -0.18855     0.00000E+00  -.65315E+01  -.13067E+01  56   C     position
 -0.28287    0.38139   -0.06938     0.43229E+01  -.48803E+01  0.13650E+01  57   C     position
  0.02914    0.06938   -0.38139     -.43229E+01  -.48803E+01  0.13650E+01  58   C     position
  0.02914    0.38139   -0.18855     0.26717E+01  -.22086E+01  0.56879E+01  59   C     position
  0.22198    0.18855   -0.38139     -.26717E+01  -.22086E+01  0.56879E+01  60   C     position


 ROTATION MATRICES AND FRACTIONAL TRANSLATIONS IN LATTICE COORDINATES

1     1  0  0     0  1  0     0  0  1     0.0000 0.0000 0.0000

368139 G-VECTORS ARE SET UP IN  184070 STARS - KMAX =  49 49 49

 1 K POINTS GENERATED BY PROGRAM FROM PARAMETERS :
N =    1    1    1      S =  0.62  0.29  0.00      NB =  140

 EWALD

     -30.90713624      ewaldpot energy


     CONTRAVARIANT STRESS TENSOR  (A.U.)              CARTESIAN STRESS (GPA)

     -0.040774    0.013128    0.014087        -0.565270E+02    0.687382E+00    0.577960E+00   ewaldstr 1
      0.013128   -0.040377    0.013706         0.687382E+00   -0.561397E+02    0.155681E+00   ewaldstr 2
      0.014087    0.013706   -0.040274         0.577960E+00    0.155681E+00   -0.582856E+02   ewaldstr 3

      -0.00193681        -56.98412952      ewaldpress (au and GPa)


       FORCE (LATTICE COORD.)            FORCE (CARTESIAN COORD. A.U)       NO. TYPE

  0.19015    0.15042   -0.18534     -.48378E+00  0.66715E-01  0.47186E+01   1   C     ewaldfrc
 -0.03372    0.13556    0.13950     0.38109E+01  0.14656E+01  0.14110E+01   2   C     ewaldfrc
 -0.13955    0.24136    0.03380     0.38123E+01  -.14652E+01  0.14105E+01   3   C     ewaldfrc
  0.16117    0.05769    0.01306     0.98029E+00  0.24140E+01  0.30324E+01   4   C     ewaldfrc
 -0.01398    0.23240   -0.16090     0.99061E+00  -.24230E+01  0.30262E+01   5   C     ewaldfrc
  0.15696    0.19174   -0.15443     0.51689E+00  0.35056E-01  0.48313E+01   6   C     ewaldfrc
  0.13558   -0.03392   -0.24140     -.38146E+01  -.14661E+01  0.14086E+01   7   C     ewaldfrc
  0.24144   -0.13955   -0.13554     -.38113E+01  0.14673E+01  0.14117E+01   8   C     ewaldfrc
  0.23208   -0.01291   -0.05807     -.98336E+00  0.24109E+01  0.30366E+01   9   C     ewaldfrc
  0.05683    0.16142   -0.23243     -.98382E+00  -.24329E+01  0.30238E+01  10   C     ewaldfrc
 -0.19183   -0.15437    0.19180     0.51863E+00  -.31238E-03  -.47966E+01  11   C     ewaldfrc
  0.03367   -0.13561   -0.13945     -.38110E+01  -.14655E+01  -.14124E+01  12   C     ewaldfrc
  0.13948   -0.24137   -0.03371     -.38113E+01  0.14655E+01  -.14117E+01  13   C     ewaldfrc
  0.01342   -0.23186    0.16087     -.98352E+00  0.24147E+01  -.30265E+01  14   C     ewaldfrc
 -0.16101   -0.05752   -0.01346     -.98349E+00  -.24172E+01  -.30277E+01  15   C     ewaldfrc
 -0.15441   -0.19180    0.15438     -.51851E+00  -.48303E-03  -.47968E+01  16   C     ewaldfrc
 -0.13565    0.03374    0.24140     0.38121E+01  0.14651E+01  -.14120E+01  17   C     ewaldfrc
 -0.24134    0.13945    0.13562     0.38111E+01  -.14647E+01  -.14116E+01  18   C     ewaldfrc
 -0.05759   -0.16085    0.23189     0.98416E+00  0.24148E+01  -.30266E+01  19   C     ewaldfrc
 -0.23195    0.01344    0.05752     0.98311E+00  -.24168E+01  -.30275E+01  20   C     ewaldfrc
 -0.19187    0.19180    0.15448     0.47978E+01  -.51799E+00  -.91245E-03  21   C     ewaldfrc
  0.01350    0.16092    0.05760     0.30275E+01  0.98516E+00  0.24166E+01  22   C     ewaldfrc
 -0.16096   -0.01338    0.23197     0.30286E+01  0.98385E+00  -.24155E+01  23   C     ewaldfrc
  0.13953   -0.03368    0.13549     0.14106E+01  0.38104E+01  0.14665E+01  24   C     ewaldfrc
  0.03370   -0.13942    0.24131     0.14117E+01  0.38102E+01  -.14647E+01  25   C     ewaldfrc
 -0.15446    0.15438    0.19191     0.47978E+01  0.51883E+00  -.11140E-02  26   C     ewaldfrc
 -0.05762    0.23199   -0.01330     0.30300E+01  -.98258E+00  0.24158E+01  27   C     ewaldfrc
 -0.23190    0.05757    0.16095     0.30275E+01  -.98296E+00  -.24153E+01  28   C     ewaldfrc
 -0.24140    0.13555   -0.03373     0.14108E+01  -.38119E+01  -.14665E+01  29   C     ewaldfrc
 -0.13580    0.24142   -0.13941     0.14134E+01  -.38131E+01  0.14634E+01  30   C     ewaldfrc
  0.19186   -0.19192   -0.15437     -.47979E+01  0.51940E+00  -.96077E-03  31   C     ewaldfrc
  0.16141    0.01278   -0.23229     -.30414E+01  -.98205E+00  0.24134E+01  32   C     ewaldfrc
 -0.01345   -0.16093   -0.05756     -.30272E+01  -.98374E+00  -.24161E+01  33   C     ewaldfrc
 -0.13962    0.03373   -0.13557     -.14110E+01  -.38127E+01  -.14672E+01  34   C     ewaldfrc
 -0.03397    0.13949   -0.24141     -.14121E+01  -.38154E+01  0.14619E+01  35   C     ewaldfrc
  0.15440   -0.15452   -0.19180     -.47983E+01  -.51823E+00  -.15844E-02  36   C     ewaldfrc
  0.23248   -0.05779   -0.16091     -.30301E+01  0.99163E+00  0.24202E+01  37   C     ewaldfrc
  0.05757   -0.23193    0.01342     -.30275E+01  0.98350E+00  -.24158E+01  38   C     ewaldfrc
```

```
    0.24139   -0.13548    0.03364     -.14109E+01  0.38106E+01  0.14674E+01  39   C     ewaldfrc
    0.13559   -0.24129    0.13943     -.14113E+01  0.38104E+01  -.14645E+01  40   C     ewaldfrc
    0.15442   -0.19179    0.19180     0.93957E-04  0.47968E+01  -.51783E+00  41   C     ewaldfrc
    0.05757    0.01343    0.16091     0.24154E+01  0.30270E+01  0.98364E+00  42   C     ewaldfrc
    0.23196   -0.16091   -0.01342     -.24154E+01  0.30279E+01  0.98440E+00  43   C     ewaldfrc
    0.13610    0.13941   -0.03402     0.14601E+01  0.14142E+01  0.38171E+01  44   C     ewaldfrc
    0.24176    0.03647   -0.14050     -.14415E+01  0.14029E+01  0.38549E+01  45   C     ewaldfrc
    0.19186   -0.15436    0.15435     -.12175E-03  0.47968E+01  0.51953E+00  46   C     ewaldfrc
   -0.01347   -0.05754    0.23193     0.24162E+01  0.30268E+01  -.98376E+00  47   C     ewaldfrc
    0.16094   -0.23192    0.05755     -.24158E+01  0.30273E+01  -.98336E+00  48   C     ewaldfrc
   -0.03372   -0.24130    0.13554     -.14654E+01  0.14107E+01  -.38105E+01  49   C     ewaldfrc
   -0.13949   -0.13552    0.24133     0.14659E+01  0.14109E+01  -.38103E+01  50   C     ewaldfrc
   -0.15460    0.19183   -0.19180     0.50691E-03  -.47994E+01  0.51588E+00  51   C     ewaldfrc
   -0.23196    0.16091    0.01345     0.24158E+01  -.30275E+01  -.98443E+00  52   C     ewaldfrc
   -0.05768   -0.01345   -0.16091     -.24158E+01  -.30284E+01  -.98551E+00  53   C     ewaldfrc
   -0.13561   -0.13944    0.03370     -.14651E+01  -.14119E+01  -.38109E+01  54   C     ewaldfrc
   -0.24132   -0.03370    0.13944     0.14650E+01  -.14115E+01  -.38105E+01  55   C     ewaldfrc
   -0.19198    0.15437   -0.15438     -.10236E-03  -.47988E+01  -.52101E+00  56   C     ewaldfrc
   -0.16104    0.23195   -0.05750     0.24170E+01  -.30278E+01  0.98240E+00  57   C     ewaldfrc
    0.01330    0.05748   -0.23195     -.24173E+01  -.30295E+01  0.98067E+00  58   C     ewaldfrc
    0.03344    0.24183   -0.13529     0.14761E+01  -.14112E+01  0.38139E+01  59   C     ewaldfrc
    0.13938    0.13555   -0.24519     -.15190E+01  -.14660E+01  0.38092E+01  60   C     ewaldfrc

IN FFT FOR HARTREE-XC  N =   100  100  100
MAX AND MIN VALUES OF CHARGE DENSITY:   1387.32339      0.13768    (.74E-12)


COMPUTING TIME FOR STARTING          1.60


IN FFT FOR LOCAL POTENTIAL N =   100  100  100

MAX AND MIN OF POTENTIAL     1.7110   -1.9057    0.0000


  K  MTXD      En(K)
  1 46036    -14.74752-14.25602-14.24178-14.23632-13.29134-13.27455-13.26853-13.26705       -0.07   0.07   0.21
            -13.26120-12.27644-12.26286-12.25615-11.45078-11.43093-11.42866-11.42336
            -10.18691-10.16875-10.16287-10.15977-10.15597 -9.71798 -9.70471 -9.70314
             -9.69174 -8.24188 -8.22495 -8.22055 -8.21872 -8.21451 -8.05770 -8.05485
             -8.04454 -7.07676 -7.07628 -7.06948 -6.13911 -6.13241 -6.12837 -6.12684
             -6.12464 -5.78619 -5.72794 -5.71189 -5.69720 -4.27131 -4.25150 -4.24641
             -4.23577 -3.64087 -3.61342 -3.61162 -3.60709 -3.60065 -3.52200 -3.51575
             -3.50676 -2.87861 -2.86839 -2.86388 -2.03403 -2.01400 -2.01037 -2.00387
             -1.83194 -1.81908 -1.81442 -1.81076 -1.80078 -1.42996 -1.41680 -1.41161
             -1.40351 -1.30787 -0.92425 -0.90538 -0.89938 -0.86824 -0.82174 -0.77918
             -0.76184 -0.75840 -0.74781 -0.73130  0.18651  0.30199  0.31311  0.34318
              0.34743  0.43545  0.45381  0.46567  0.47332  0.48005  0.68876  0.71113
              0.71749  0.71874  0.73740  1.48743  1.49648  1.60406  2.03109  2.06513
              2.09618  2.11553  3.50906  3.53662  3.57774  3.59110  3.63220  3.66161
              3.69965  3.72508  3.75519  4.71345  4.76916  4.80808  4.82672  4.85645
              6.53547  6.55656  6.65074  7.57683  7.64193  7.68927  8.65615  8.68252
              8.70949  8.73655  8.78517  8.87604  8.91851  8.94432 11.25818 11.46828
             11.54344 12.08625 12.15253 12.64772

THE FERMI LEVEL IS AT     4.8565 [eV]


IN FFT FOR HARTREE-XC  N =   100  100  100
MAX AND MIN VALUES OF CHARGE DENSITY:   1946.70871      0.01148    (.95E-12)


 ITERATION NUMBER  1

  I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

  1   0  0  0     0.00000 240.00000  -0.31509
  2  -1 -1 -1     0.07712  44.20728   0.64597   0.61526   0.03071  -0.68444
                          -0.01982  -0.00027  -0.00036   0.00009   0.00041
  3   0 -1  0     0.07712  44.12648   0.64477   0.61436   0.03041  -0.68345
```

5. APPENDIX A

```
                            0.00751   0.00015   0.00002   0.00012   0.00000
  4   0  0  1     0.07712  44.12604   0.64473   0.61435   0.03038  -0.68345
                            0.00477   0.00009   0.00002   0.00007   0.00000
  5   1  0  0     0.07712  44.18581   0.64566   0.61531   0.03035  -0.68450
                           -0.02117  -0.00034  -0.00020  -0.00013   0.00021
  6  -1 -1  0     0.10283   9.77488   0.11040   0.09694   0.01346  -0.10661
                            0.06340   0.00068   0.00067   0.00000  -0.00078
  7   0 -1 -1     0.10283   9.77973   0.11050   0.09684   0.01366  -0.10646
                           -0.00524  -0.00008  -0.00001  -0.00007   0.00000
  8   1  0  1     0.10283   9.81925   0.11093   0.09683   0.01410  -0.10645
                            0.06933   0.00077   0.00070   0.00007  -0.00079
  9  -2 -1 -1     0.20566 -43.32371  -0.21804  -0.20638  -0.01166   0.27201
                            0.13402   0.00070   0.00058   0.00013  -0.00074
 10  -1 -1 -2     0.20566 -43.36038  -0.21819  -0.20666  -0.01153   0.27238
                           -0.06955  -0.00039  -0.00026  -0.00013   0.00031
 11  -1  1  0     0.20566 -43.28349  -0.21784  -0.20632  -0.01152   0.27195
                           -0.04826  -0.00024  -0.00024   0.00000   0.00032
 12   0  1 -1     0.20566 -43.35395  -0.21819  -0.20651  -0.01168   0.27216
                            0.00922   0.00008   0.00002   0.00006   0.00000
 13   1  0 -1     0.20566 -43.36100  -0.21823  -0.20661  -0.01162   0.27231
                           -0.04782  -0.00023  -0.00026   0.00003   0.00035
 14   1  2  1     0.20566 -43.29443  -0.21787  -0.20630  -0.01157   0.27189
                           -0.05120  -0.00027  -0.00021  -0.00006   0.00027
 15  -2 -2 -1     0.28278 -36.62420  -0.13312  -0.12166  -0.01146   0.16917
                            0.06215   0.00023   0.00022   0.00002  -0.00029
 16  -2 -1  0     0.28278 -36.64526  -0.13320  -0.12173  -0.01147   0.16926
                           -0.00787  -0.00002   0.00001  -0.00003   0.00000
 17  -1 -2 -2     0.28278 -36.47699  -0.13259  -0.12123  -0.01136   0.16858
                           -0.05317  -0.00019  -0.00016  -0.00004   0.00022
 18  -1 -1  1     0.28278 -36.57380  -0.13292  -0.12153  -0.01139   0.16899
                           -0.00953  -0.00003   0.00000  -0.00003   0.00000
 19  -1  1 -1     0.28278 -36.57291  -0.13292  -0.12153  -0.01139   0.16897
                           -0.03659  -0.00016  -0.00009  -0.00007   0.00010
 20   0 -2 -1     0.28278 -36.52690  -0.13272  -0.12134  -0.01138   0.16873
                            0.00899   0.00005   0.00001   0.00004   0.00000


        ITERATION NUMBER  1 :
        ----------

        BAND ENERGY          =     -32.7587247774
        HXC  CORRECTION      =     511.8737912644
        -------------------------------------
        KINETIC  ENERGY      =     269.6332452973
        IONIC    ENERGY      =    -688.4289750217
        NONLOCAL ENERGY      =    -125.8367863174
        HARTREE  ENERGY      =     312.6052058594
        EXCHANGE CORRELATION =    -106.7400596044
        -------------------------------------
        ALPHA TERM           =      28.7276840051
        EWALD TERM           =     -30.9071362392
        -------------------------------------
        TOTAL ENERGY         =    -340.9468220208

 COMPUTING TIME FOR ITERATION     1      1726.14

 IN FFT FOR LOCAL POTENTIAL N =   100  100  100

 MAX AND MIN OF POTENTIAL     1.8911   -1.7555    0.0000


  K  MTXD      En(K)
  1 46036    -13.22764-12.72314-12.71650-12.71090-11.74810-11.73558-11.72699-11.71721       -0.07   0.07   0.21
             -11.71317-10.73474-10.72771-10.72103 -9.83233 -9.82140 -9.81753 -9.81121
              -8.59694 -8.58593 -8.57753 -8.52916 -8.51805 -8.05616 -8.05017 -8.04348
              -8.01551 -6.50018 -6.48095 -6.47641 -6.47487 -6.46807 -6.44768 -6.44531
              -6.43608 -5.46927 -5.46709 -5.45981 -4.33729 -4.33320 -4.33000 -4.31144
              -4.30797 -4.11514 -4.06632 -4.05657 -4.03963 -2.72730 -2.71391 -2.70108
              -2.62965 -1.86932 -1.85074 -1.82678 -1.82111 -1.81759 -1.73457 -1.72837
```

```
-1.72347 -1.12715 -1.11933 -1.11429 -0.69381 -0.36478 -0.34743 -0.34314
-0.33297 -0.24628 -0.19442 -0.09381  0.06415  0.06841  0.07488  0.09768
 0.10699  0.50950  0.51441  0.52453  0.54265  0.83376  0.84772  0.85488
 0.86610  1.02184  1.04042  1.07811  1.09229  1.21611  1.24609  1.25655
 1.26543  2.22109  2.22913  2.30056  2.31126  2.32032  2.33763  2.35674
 2.37448  2.60480  2.61447  2.65260  2.65771  2.66776  2.81408  2.85820
 2.88526  2.90575  4.32668  4.35831  4.40961  4.43223  4.48290  4.52037
 4.57198  4.63183  4.64918  5.63118  5.71974  5.75449  5.78009  5.81203
 7.38626  7.40833  7.53256  8.50740  8.58628  8.64507  9.40673  9.42581
 9.49141  9.55270  9.59738  9.71181  9.72226  9.74936 11.26161 11.52889
11.71086 11.85136 11.89718 12.33130

THE FERMI LEVEL IS AT     5.8120 [eV]


IN FFT FOR HARTREE-XC  N =   100  100  100
MAX AND MIN VALUES OF CHARGE DENSITY:   1805.79043      0.01913    (.82E-12)


 ITERATION NUMBER  2

  I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

  1   0  0  0     0.00000 240.00000  -0.31509
  2  -1 -1 -1     0.07712  43.30033   0.63259   0.61844   0.01415  -0.68444
                           -0.02393  -0.00033  -0.00035   0.00002   0.00041
  3   0 -1  0     0.07712  43.23533   0.63164   0.61751   0.01413  -0.68345
                            0.00052   0.00004   0.00004   0.00000   0.00000
  4   0  0  1     0.07712  43.24055   0.63167   0.61749   0.01418  -0.68345
                            0.00382   0.00007   0.00002   0.00005   0.00000
  5   1  0  0     0.07712  43.30172   0.63263   0.61845   0.01419  -0.68450
                           -0.01117  -0.00018  -0.00022   0.00003   0.00021
  6  -1 -1  0     0.10283   9.01553   0.10226   0.09876   0.00350  -0.10661
                            0.06731   0.00072   0.00067   0.00005  -0.00078
  7   0 -1 -1     0.10283   8.99832   0.10210   0.09868   0.00342  -0.10646
                            0.00439   0.00003  -0.00002   0.00005   0.00000
  8   1  0  1     0.10283   9.02138   0.10235   0.09874   0.00361  -0.10645
                            0.06523   0.00073   0.00071   0.00001  -0.00079
  9  -2 -1 -1     0.20566 -42.84423  -0.21555  -0.20930  -0.00624   0.27201
                            0.11332   0.00059   0.00061  -0.00002  -0.00074
 10  -1 -1 -2     0.20566 -42.89071  -0.21574  -0.20955  -0.00619   0.27238
                           -0.04693  -0.00026  -0.00029   0.00003   0.00031
 11  -1  1  0     0.20566 -42.82467  -0.21547  -0.20921  -0.00627   0.27195
                           -0.05072  -0.00025  -0.00024  -0.00001   0.00032
 12   0  1 -1     0.20566 -42.87329  -0.21568  -0.20944  -0.00624   0.27216
                           -0.00173   0.00001   0.00003  -0.00002   0.00000
 13   1  0 -1     0.20566 -42.89119  -0.21578  -0.20952  -0.00626   0.27231
                           -0.05822  -0.00028  -0.00025  -0.00003   0.00035
 14   1  2  1     0.20566 -42.82987  -0.21545  -0.20920  -0.00625   0.27189
                           -0.03944  -0.00020  -0.00022   0.00002   0.00027
 15  -2 -2 -1     0.28278 -35.69470  -0.12980  -0.12510  -0.00470   0.16917
                            0.05958   0.00022   0.00022   0.00000  -0.00029
 16  -2 -1  0     0.28278 -35.72249  -0.12990  -0.12517  -0.00473   0.16926
                            0.00123   0.00001   0.00000   0.00002   0.00000
 17  -1 -2 -2     0.28278 -35.57397  -0.12938  -0.12464  -0.00474   0.16858
                           -0.04327  -0.00015  -0.00017   0.00001   0.00022
 18  -1 -1  1     0.28278 -35.66562  -0.12968  -0.12494  -0.00474   0.16899
                           -0.00298  -0.00001  -0.00001   0.00001   0.00000
 19  -1  1 -1     0.28278 -35.65858  -0.12965  -0.12495  -0.00471   0.16897
                           -0.01630  -0.00008  -0.00011   0.00003   0.00010
 20   0 -2 -1     0.28278 -35.62205  -0.12949  -0.12475  -0.00474   0.16873
                           -0.00054   0.00001   0.00002  -0.00001   0.00000


        ITERATION NUMBER  2 :
        ---------------------

        BAND ENERGY          =     -19.5716336083   13.18709117
        HXC  CORRECTION      =     512.5978159390    0.72402467
```

5. APPENDIX A

```
         ------------------------------------------------------
         KINETIC  ENERGY      =     256.0999353859  -13.53330991
         IONIC    ENERGY      =    -671.3587119282   17.07026309
         NONLOCAL ENERGY      =    -116.9106730051    8.92611331
         HARTREE  ENERGY      =     297.1941074018  -15.41109846
         EXCHANGE CORRELATION =    -104.7107937823    2.02926582
         ------------------------------------------------------
         ALPHA TERM           =      28.7276840051    0.00000000
         EWALD TERM           =     -30.9071362392    0.00000000
         ------------------------------------------------------
         TOTAL ENERGY         =    -341.8655881619   -0.91876614

 COMPUTING TIME FOR ITERATION     2      1741.47

 IN FFT FOR LOCAL POTENTIAL N =   100  100  100

 MAX AND MIN OF POTENTIAL     1.9012   -1.7364    0.0000


  K  MTXD      En(K)
  1 46036    -12.47075-11.96489-11.95967-11.95110-10.97904-10.97213-10.95851-10.95459       -0.07   0.07   0.21
            -10.95054 -9.93380 -9.92546 -9.91629 -9.12564 -9.11262 -9.10911 -9.10357
             -7.82248 -7.81572 -7.80553 -7.80385 -7.79753 -7.35602 -7.34888 -7.34023
             -7.31572 -5.79503 -5.78186 -5.77700 -5.77315 -5.76811 -5.68909 -5.68596
             -5.67491 -4.73479 -4.73118 -4.72507 -3.63483 -3.63021 -3.62567 -3.61692
             -3.61051 -3.33633 -3.32260 -3.30530 -3.30190 -1.97895 -1.96855 -1.95443
             -1.93988 -1.14373 -1.12527 -1.12440 -1.11624 -1.11232 -0.97889 -0.97473
             -0.96684 -0.41310 -0.40525 -0.40016 -0.22710  0.20251  0.26562  0.35557
              0.35992  0.36847  0.38467  0.39160  0.80411  0.81087  0.81621  0.81931
              0.83157  1.21722  1.22521  1.23551  1.25482  1.32567  1.50757  1.53147
              1.54612  1.55892  1.56475  1.57812  1.58895  1.96081  1.97132  1.98105
              1.98993  2.70527  2.71635  2.86454  3.01655  3.03700  3.04216  3.05337
              3.06239  3.29663  3.32727  3.33448  3.34045  3.34192  3.36336  3.37416
              3.38695  3.40375  4.80281  4.84779  4.90212  4.92329  4.97043  5.01077
              5.07456  5.13850  5.14672  6.11286  6.21482  6.26015  6.28560  6.32780
              7.87267  7.89534  8.03819  8.99617  9.08304  9.14940  9.85520  9.89198
              9.97474 10.01558 10.06953 10.20047 10.22168 10.24994 11.52733 11.60160
             11.68445 11.82066 11.92031 12.24410

 THE FERMI LEVEL IS AT     6.3278 [eV]


 IN FFT FOR HARTREE-XC  N =   100  100  100
 MAX AND MIN VALUES OF CHARGE DENSITY:   1812.14682      0.02835    (.89E-12)


  ITERATION NUMBER  3

   I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

   1   0  0  0     0.00000 240.00000  -0.31509
   2  -1 -1 -1     0.07712  42.95249   0.62782   0.62109   0.00673  -0.68444
                            -0.02498  -0.00035  -0.00035   0.00000   0.00041
   3   0 -1  0     0.07712  42.88645   0.62686   0.62016   0.00670  -0.68345
                             0.00235   0.00007   0.00004   0.00003   0.00000
   4   0  0  1     0.07712  42.88732   0.62682   0.62015   0.00667  -0.68345
                             0.00171   0.00004   0.00003   0.00001   0.00000
   5   1  0  0     0.07712  42.94180   0.62769   0.62111   0.00659  -0.68450
                            -0.01397  -0.00023  -0.00021  -0.00001   0.00021
   6  -1 -1  0     0.10283   8.74903   0.09951   0.09964  -0.00012  -0.10661
                             0.06352   0.00068   0.00068  -0.00001  -0.00078
   7   0 -1 -1     0.10283   8.73515   0.09939   0.09954  -0.00015  -0.10646
                            -0.00106  -0.00003  -0.00001  -0.00002   0.00000
   8   1  0  1     0.10283   8.75508   0.09960   0.09964  -0.00004  -0.10645
                             0.06272   0.00070   0.00072  -0.00002  -0.00079
   9  -2 -1 -1     0.20566 -42.65825  -0.21487  -0.21213  -0.00274   0.27201
                             0.12202   0.00064   0.00060   0.00004  -0.00074
  10  -1 -1 -2     0.20566 -42.70621  -0.21506  -0.21235  -0.00271   0.27238
                            -0.05740  -0.00032  -0.00028  -0.00004   0.00031
  11  -1  1  0     0.20566 -42.63085  -0.21476  -0.21205  -0.00272   0.27195
```

```
                          -0.04760  -0.00024  -0.00025   0.00001   0.00032
 12   0  1 -1     0.20566 -42.68552  -0.21499  -0.21227  -0.00272   0.27216
                           0.00209   0.00004   0.00002   0.00001   0.00000
 13   1  0 -1     0.20566 -42.69904  -0.21507  -0.21236  -0.00271   0.27231
                          -0.04937  -0.00024  -0.00027   0.00003   0.00035
 14   1  2  1     0.20566 -42.63456  -0.21472  -0.21203  -0.00268   0.27189
                          -0.04500  -0.00024  -0.00022  -0.00002   0.00027
 15  -2 -2 -1     0.28278 -35.37866  -0.12892  -0.12766  -0.00126   0.16917
                           0.06263   0.00024   0.00022   0.00001  -0.00029
 16  -2 -1  0     0.28278 -35.39695  -0.12899  -0.12775  -0.00124   0.16926
                          -0.00408  -0.00001   0.00001  -0.00001   0.00000
 17  -1 -2 -2     0.28278 -35.23961  -0.12844  -0.12723  -0.00121   0.16858
                          -0.05009  -0.00018  -0.00016  -0.00002   0.00022
 18  -1 -1  1     0.28278 -35.33666  -0.12875  -0.12753  -0.00123   0.16899
                          -0.00743  -0.00002  -0.00001  -0.00001   0.00000
 19  -1  1 -1     0.28278 -35.33529  -0.12874  -0.12751  -0.00123   0.16897
                          -0.02511  -0.00012  -0.00010  -0.00002   0.00010
 20   0 -2 -1     0.28278 -35.29177  -0.12855  -0.12734  -0.00122   0.16873
                           0.00358   0.00003   0.00002   0.00001   0.00000


        ITERATION NUMBER  3 :
        ---------------------

        BAND ENERGY          =     -13.6714667194    5.90016689
        HXC  CORRECTION      =     514.3466786833    1.74886274
        ------------------------------------------------------
        KINETIC  ENERGY      =     254.1231904772   -1.97674491
        IONIC    ENERGY      =    -665.9837432756    5.37496865
        NONLOCAL ENERGY      =    -116.1575926043    0.75308040
        HARTREE  ENERGY      =     292.4872724516   -4.70683495
        EXCHANGE CORRELATION =    -104.2522452982    0.45854848
        ------------------------------------------------------
        ALPHA TERM           =      28.7276840051    0.00000000
        EWALD TERM           =     -30.9071362392    0.00000000
        ------------------------------------------------------
        TOTAL ENERGY         =    -341.9625704834   -0.09698232

COMPUTING TIME FOR ITERATION     3      1444.68

IN FFT FOR LOCAL POTENTIAL N =   100  100  100

MAX AND MIN OF POTENTIAL     1.9229   -1.7153    0.0000


  K  MTXD      En(K)
  1 46036    -12.03042-11.52209-11.51815-11.50958-10.53441-10.52383-10.51275-10.50844       -0.07   0.07   0.21
            -10.50657 -9.48043 -9.47378 -9.46501 -8.68210 -8.67048 -8.66651 -8.65948
             -7.37233 -7.36673 -7.35576 -7.35130 -7.34783 -6.90552 -6.89991 -6.89194
             -6.87945 -5.34713 -5.33303 -5.32760 -5.32410 -5.31836 -5.23485 -5.23081
             -5.21918 -4.28149 -4.27680 -4.27062 -3.16823 -3.16158 -3.15993 -3.15628
             -3.15044 -2.89554 -2.88346 -2.86308 -2.82459 -1.53243 -1.52568 -1.51887
             -1.50463 -0.69003 -0.67826 -0.67418 -0.66519 -0.65808 -0.51078 -0.50634
             -0.49855  0.03338  0.05100  0.05779  0.06268  0.46535  0.53091  0.65249
              0.80400  0.80682  0.81486  0.82514  1.27498  1.28296  1.28676  1.29080
              1.30177  1.59336  1.69428  1.69944  1.71083  1.71906  1.78781  1.81247
              1.86911  1.87236  1.99472  2.00623  2.01561  2.44078  2.44878  2.45795
              2.46794  2.99383  3.00682  3.15664  3.47956  3.49877  3.50218  3.50884
              3.52379  3.58670  3.62840  3.67694  3.69848  3.79520  3.80904  3.81271
              3.81729  3.83652  5.08972  5.13326  5.18772  5.21318  5.26613  5.31372
              5.39499  5.43761  5.46341  6.41005  6.52776  6.57273  6.60237  6.64448
              8.16094  8.18440  8.34067  9.29474  9.38963  9.45976 10.11752 10.16553
             10.26670 10.29656 10.37080 10.50160 10.52863 10.55742 11.37148 11.84862
             11.92391 12.01495 12.08678 12.10781

THE FERMI LEVEL IS AT     6.6445 [eV]


IN FFT FOR HARTREE-XC  N =   100  100  100
```

5. APPENDIX A

```
 MAX AND MIN VALUES OF CHARGE DENSITY:   1799.74836      0.03814    (.86E-12)


  ITERATION NUMBER  4

   I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

   1   0  0  0     0.00000 240.00000  -0.31509
   2  -1 -1 -1     0.07712  42.71974   0.62460   0.62289   0.00171  -0.68444
                            -0.02528  -0.00035  -0.00035  -0.00001   0.00041
   3   0 -1  0     0.07712  42.65547   0.62367   0.62195   0.00172  -0.68345
                            -0.00145   0.00001   0.00004  -0.00003   0.00000
   4   0  0  1     0.07712  42.66098   0.62370   0.62194   0.00177  -0.68345
                             0.00142   0.00004   0.00004   0.00000   0.00000
   5   1  0  0     0.07712  42.71963   0.62464   0.62288   0.00176  -0.68450
                            -0.01143  -0.00019  -0.00021   0.00003   0.00021
   6  -1 -1  0     0.10283   8.63526   0.09834   0.09986  -0.00152  -0.10661
                             0.06497   0.00069   0.00069   0.00001  -0.00078
   7   0 -1 -1     0.10283   8.61706   0.09817   0.09975  -0.00158  -0.10646
                             0.00319   0.00001  -0.00001   0.00003   0.00000
   8   1  0  1     0.10283   8.63250   0.09833   0.09989  -0.00156  -0.10645
                             0.06385   0.00071   0.00071   0.00000  -0.00079
   9  -2 -1 -1     0.20566 -42.50768  -0.21428  -0.21395  -0.00033   0.27201
                             0.11331   0.00059   0.00061  -0.00002  -0.00074
  10  -1 -1 -2     0.20566 -42.55765  -0.21447  -0.21416  -0.00032   0.27238
                            -0.04636  -0.00026  -0.00029   0.00003   0.00031
  11  -1  1  0     0.20566 -42.48644  -0.21421  -0.21386  -0.00035   0.27195
                            -0.04955  -0.00025  -0.00024   0.00000   0.00032
  12   0  1 -1     0.20566 -42.53769  -0.21441  -0.21408  -0.00033   0.27216
                            -0.00120   0.00002   0.00003  -0.00001   0.00000
  13   1  0 -1     0.20566 -42.55393  -0.21450  -0.21417  -0.00033   0.27231
                            -0.05673  -0.00028  -0.00026  -0.00002   0.00035
  14   1  2  1     0.20566 -42.49391  -0.21417  -0.21383  -0.00034   0.27189
                            -0.04000  -0.00021  -0.00022   0.00001   0.00027
  15  -2 -2 -1     0.28278 -35.12815  -0.12813  -0.12894   0.00081   0.16917
                             0.05977   0.00023   0.00023   0.00000  -0.00029
  16  -2 -1  0     0.28278 -35.15345  -0.12823  -0.12902   0.00080   0.16926
                            -0.00005   0.00001   0.00000   0.00001   0.00000
  17  -1 -2 -2     0.28278 -35.00281  -0.12770  -0.12849   0.00078   0.16858
                            -0.04260  -0.00015  -0.00017   0.00002   0.00022
  18  -1 -1  1     0.28278 -35.09698  -0.12800  -0.12879   0.00079   0.16899
                            -0.00315  -0.00001  -0.00002   0.00001   0.00000
  19  -1  1 -1     0.28278 -35.09254  -0.12798  -0.12878   0.00080   0.16897
                            -0.01812  -0.00009  -0.00010   0.00001   0.00010
  20   0 -2 -1     0.28278 -35.05354  -0.12781  -0.12860   0.00079   0.16873
                            -0.00064   0.00001   0.00002  -0.00001   0.00000


         ITERATION NUMBER  4 :
         ---------------------

         BAND ENERGY          =     -10.0180570443    3.65340968
         HXC  CORRECTION      =     515.0649041257    0.71822544
         ------------------------------------------------------
         KINETIC  ENERGY      =     252.1056268647   -2.01756361
         IONIC    ENERGY      =    -662.1703220272    3.81342125
         NONLOCAL ENERGY      =    -115.0182660074    1.13932660
         HARTREE  ENERGY      =     289.1647993966   -3.32247306
         EXCHANGE CORRELATION =    -103.8772653802    0.37497992
         ------------------------------------------------------
         ALPHA TERM           =      28.7276840051    0.00000000
         EWALD TERM           =     -30.9071362392    0.00000000
         ------------------------------------------------------
         TOTAL ENERGY         =    -341.9748793877   -0.01230890

 COMPUTING TIME FOR ITERATION     4      1542.50

 IN FFT FOR LOCAL POTENTIAL N =   100  100  100
```

```
MAX AND MIN OF POTENTIAL     1.9169   -1.7186    0.0000


  K  MTXD      En(K)
  1 46036    -12.06995-11.56201-11.55750-11.54878-10.57403-10.56458-10.55154-10.54932       -0.07    0.07    0.21
             -10.54565 -9.52296 -9.51534 -9.50668 -8.71807 -8.70817 -8.70301 -8.69524
              -7.41195 -7.40553 -7.39467 -7.39057 -7.38709 -6.94331 -6.93698 -6.92869
              -6.91559 -5.38636 -5.37233 -5.36698 -5.36303 -5.35728 -5.27424 -5.27045
              -5.25894 -4.31683 -4.31227 -4.30619 -3.20910 -3.20319 -3.19908 -3.19591
              -3.18786 -2.93541 -2.92337 -2.90317 -2.86662 -1.56728 -1.56090 -1.55337
              -1.53890 -0.73185 -0.71766 -0.71495 -0.70608 -0.70058 -0.55488 -0.55035
              -0.54244  0.00960  0.01759  0.02378  0.03854  0.46506  0.53089  0.65178
               0.76683  0.76957  0.77815  0.78902  1.22888  1.23678  1.24122  1.24487
               1.25619  1.58802  1.64872  1.65324  1.66407  1.67537  1.78400  1.80849
               1.86497  1.86819  1.95391  1.96656  1.97504  2.38942  2.39872  2.40730
               2.41724  2.98757  3.00020  3.14964  3.43338  3.45449  3.45575  3.46585
               3.47838  3.57755  3.61967  3.66930  3.69099  3.74830  3.76055  3.76460
               3.77375  3.79097  5.08055  5.12487  5.18012  5.20407  5.25711  5.30312
               5.38152  5.42219  5.45235  6.39781  6.51284  6.55974  6.58784  6.63186
               8.15057  8.17392  8.32841  9.28056  9.37396  9.44394 10.11479 10.16275
              10.26335 10.29061 10.36255 10.49555 10.52114 10.55137 11.40852 11.83020
              11.90188 11.95534 12.06036 12.07023

THE FERMI LEVEL IS AT     6.6319 [eV]


IN FFT FOR HARTREE-XC  N =   100  100  100
MAX AND MIN VALUES OF CHARGE DENSITY:   1805.80057      0.03984    (.89E-12)


 ITERATION NUMBER  5

  I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

  1   0  0  0     0.00000 240.00000  -0.31509
  2  -1 -1 -1     0.07712  42.77611   0.62547   0.62310   0.00237  -0.68444
                           -0.02547  -0.00036  -0.00035  -0.00001   0.00041
  3   0 -1  0     0.07712  42.71199   0.62454   0.62216   0.00238  -0.68345
                            0.00070   0.00004   0.00004   0.00000   0.00000
  4   0  0  1     0.07712  42.71430   0.62453   0.62216   0.00237  -0.68345
                            0.00121   0.00003   0.00004   0.00000   0.00000
  5   1  0  0     0.07712  42.77232   0.62546   0.62310   0.00236  -0.68450
                           -0.01301  -0.00021  -0.00021   0.00000   0.00021
  6  -1 -1  0     0.10283   8.68019   0.09881   0.09966  -0.00085  -0.10661
                            0.06419   0.00069   0.00069   0.00000  -0.00078
  7   0 -1 -1     0.10283   8.66477   0.09867   0.09955  -0.00088  -0.10646
                            0.00099  -0.00001  -0.00001   0.00000   0.00000
  8   1  0  1     0.10283   8.68025   0.09883   0.09969  -0.00086  -0.10645
                            0.06300   0.00070   0.00071  -0.00001  -0.00079
  9  -2 -1 -1     0.20566 -42.52139  -0.21437  -0.21407  -0.00030   0.27201
                            0.11772   0.00061   0.00061   0.00001  -0.00074
 10  -1 -1 -2     0.20566 -42.57204  -0.21457  -0.21427  -0.00030   0.27238
                           -0.05209  -0.00029  -0.00029  -0.00001   0.00031
 11  -1  1  0     0.20566 -42.49535  -0.21427  -0.21399  -0.00029   0.27195
                           -0.04855  -0.00024  -0.00025   0.00000   0.00032
 12   0  1 -1     0.20566 -42.55096  -0.21450  -0.21420  -0.00029   0.27216
                            0.00021   0.00003   0.00002   0.00000   0.00000
 13   1  0 -1     0.20566 -42.56541  -0.21459  -0.21429  -0.00029   0.27231
                           -0.05290  -0.00026  -0.00026   0.00001   0.00035
 14   1  2  1     0.20566 -42.50388  -0.21424  -0.21395  -0.00029   0.27189
                           -0.04259  -0.00022  -0.00022   0.00000   0.00027
 15  -2 -2 -1     0.28278 -35.18050  -0.12832  -0.12874   0.00041   0.16917
                            0.06067   0.00023   0.00023   0.00000  -0.00029
 16  -2 -1  0     0.28278 -35.20165  -0.12840  -0.12882   0.00042   0.16926
                           -0.00213   0.00000   0.00000   0.00000   0.00000
 17  -1 -2 -2     0.28278 -35.04774  -0.12786  -0.12829   0.00042   0.16858
                           -0.04694  -0.00017  -0.00016   0.00000   0.00022
 18  -1 -1  1     0.28278 -35.14386  -0.12817  -0.12859   0.00042   0.16899
                           -0.00549  -0.00002  -0.00001   0.00000   0.00000
 19  -1  1 -1     0.28278 -35.14165  -0.12815  -0.12857   0.00042   0.16897
```

5. APPENDIX A

```
                          -0.02169  -0.00010  -0.00010   0.00000   0.00010
 20   0 -2 -1     0.28278 -35.09978  -0.12797  -0.12840   0.00043   0.16873
                           0.00185   0.00002   0.00002   0.00000   0.00000


         ITERATION NUMBER  5 :
         --------------------

         BAND ENERGY          =     -10.3063534914   -0.28829645
         HXC  CORRECTION      =     515.4078942119    0.34299009
         ------------------------------------------------------
         KINETIC  ENERGY      =     252.6190647292    0.51343786
         IONIC    ENERGY      =    -662.9653549096   -0.79503288
         NONLOCAL ENERGY      =    -115.3679575228   -0.34969152
         HARTREE  ENERGY      =     289.8714771793    0.70667778
         EXCHANGE CORRELATION =    -103.9540605333   -0.07679515
         ------------------------------------------------------
         ALPHA TERM           =      28.7276840051    0.00000000
         EWALD TERM           =     -30.9071362392    0.00000000
         ------------------------------------------------------
         TOTAL ENERGY         =    -341.9762832913   -0.00140390

 COMPUTING TIME FOR ITERATION     5      1332.33

 IN FFT FOR LOCAL POTENTIAL N =   100  100  100

 MAX AND MIN OF POTENTIAL     1.9242   -1.7131    0.0000


   K  MTXD      En(K)
   1 46036    -12.04767-11.53887-11.53178-11.52273-10.54681-10.53489-10.52816-10.51956       -0.07   0.07   0.21
              -10.51850 -9.51405 -9.50312 -9.49535 -8.64642 -8.64434 -8.62768 -8.62017
               -7.37543 -7.36253 -7.35313 -7.32792 -7.31705 -6.86075 -6.85294 -6.84651
               -6.83607 -5.30181 -5.27780 -5.27683 -5.27141 -5.26470 -5.22806 -5.22528
               -5.21265 -4.23487 -4.23066 -4.22387 -3.10208 -3.09718 -3.09122 -3.08641
               -3.07894 -2.87949 -2.86706 -2.84827 -2.80764 -1.50891 -1.49430 -1.47825
               -1.45473 -0.64772 -0.62890 -0.62794 -0.61987 -0.61609 -0.48484 -0.47933
               -0.47223  0.11275  0.12131  0.12720  0.14871  0.57319  0.63934  0.75772
                0.84366  0.85919  0.86431  0.87288  1.31372  1.32218  1.32727  1.33199
                1.34157  1.69513  1.74826  1.75414  1.76430  1.77280  1.87849  1.90227
                1.96852  1.97534  2.03653  2.05250  2.05971  2.47353  2.48377  2.49463
                2.50193  3.08527  3.09651  3.23435  3.51690  3.53156  3.54288  3.55281
                3.56795  3.67567  3.71477  3.76802  3.79230  3.83116  3.84607  3.86297
                3.86686  3.88105  5.17274  5.21500  5.26831  5.29886  5.35667  5.40805
                5.48551  5.49610  5.55805  6.49305  6.60724  6.65654  6.68476  6.72956
                8.24387  8.26698  8.42070  9.37472  9.46681  9.53706 10.23252 10.28483
               10.39408 10.41044 10.49732 10.61637 10.65468 10.68167 11.66891 11.82940
               11.96036 12.01316 12.05720 12.16311

 THE FERMI LEVEL IS AT     6.7296 [eV]


 IN FFT FOR HARTREE-XC  N =   100  100  100
 MAX AND MIN VALUES OF CHARGE DENSITY:   1802.81642      0.06683    (.85E-12)


  ITERATION NUMBER  6

   I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

   1   0  0  0     0.00000 240.00000  -0.31509
   2  -1 -1 -1     0.07712  42.89649   0.62758   0.62570   0.00188  -0.68444
                            -0.02364  -0.00033  -0.00035   0.00002   0.00041
   3   0 -1  0     0.07712  42.83283   0.62666   0.62476   0.00190  -0.68345
                             0.00395   0.00009   0.00003   0.00006   0.00000
   4   0  0  1     0.07712  42.82700   0.62652   0.62477   0.00175  -0.68345
                             0.00169   0.00004   0.00004   0.00001   0.00000
   5   1  0  0     0.07712  42.88923   0.62751   0.62570   0.00181  -0.68450
                            -0.01515  -0.00024  -0.00020  -0.00004   0.00021
```

```
  6  -1 -1  0     0.10283   8.86866   0.10077   0.09842   0.00235  -0.10661
                             0.06309   0.00067   0.00069  -0.00002  -0.00078
  7   0 -1 -1     0.10283   8.86817   0.10079   0.09825   0.00254  -0.10646
                            -0.00344  -0.00006   0.00001  -0.00007   0.00000
  8   1  0  1     0.10283   8.87871   0.10090   0.09843   0.00247  -0.10645
                             0.06310   0.00070   0.00071   0.00000  -0.00079
  9  -2 -1 -1     0.20566 -42.47242  -0.21428  -0.21553   0.00125   0.27201
                             0.12718   0.00066   0.00059   0.00007  -0.00074
 10  -1 -1 -2     0.20566 -42.52426  -0.21448  -0.21572   0.00124   0.27238
                            -0.06325  -0.00035  -0.00026  -0.00009   0.00031
 11  -1  1  0     0.20566 -42.43861  -0.21414  -0.21546   0.00132   0.27195
                            -0.04756  -0.00024  -0.00025   0.00001   0.00032
 12   0  1 -1     0.20566 -42.50273  -0.21441  -0.21566   0.00125   0.27216
                             0.00305   0.00004   0.00002   0.00003   0.00000
 13   1  0 -1     0.20566 -42.51104  -0.21447  -0.21575   0.00129   0.27231
                            -0.04549  -0.00022  -0.00028   0.00006   0.00035
 14   1  2  1     0.20566 -42.44757  -0.21411  -0.21541   0.00130   0.27189
                            -0.04754  -0.00025  -0.00021  -0.00004   0.00027
 15  -2 -2 -1     0.28278 -35.24238  -0.12849  -0.12774  -0.00075   0.16917
                             0.06396   0.00024   0.00023   0.00002  -0.00029
 16  -2 -1  0     0.28278 -35.25812  -0.12854  -0.12783  -0.00071   0.16926
                            -0.00698  -0.00002   0.00001  -0.00003   0.00000
 17  -1 -2 -2     0.28278 -35.09696  -0.12798  -0.12731  -0.00067   0.16858
                            -0.05431  -0.00020  -0.00015  -0.00005   0.00022
 18  -1 -1  1     0.28278 -35.19633  -0.12830  -0.12761  -0.00069   0.16899
                            -0.01086  -0.00004  -0.00001  -0.00003   0.00000
 19  -1  1 -1     0.28278 -35.19507  -0.12828  -0.12758  -0.00070   0.16897
                            -0.02568  -0.00012  -0.00009  -0.00003   0.00010
 20   0 -2 -1     0.28278 -35.14886  -0.12808  -0.12741  -0.00068   0.16873
                             0.00404   0.00003   0.00001   0.00002   0.00000


        ITERATION NUMBER  6 :
        ---------------------

        BAND ENERGY          =      -9.6043856223    0.70196787
        HXC  CORRECTION      =     516.9826117849    1.57471757
        ------------------------------------------------------
        KINETIC  ENERGY      =     252.6714509855    0.05238626
        IONIC    ENERGY      =    -663.9003384441   -0.93498353
        NONLOCAL ENERGY      =    -115.3581099486    0.00984757
        HARTREE  ENERGY      =     290.7700693280    0.89859215
        EXCHANGE CORRELATION =    -103.9720424045   -0.01798187
        ------------------------------------------------------
        ALPHA TERM           =      28.7276840051    0.00000000
        EWALD TERM           =     -30.9071362392    0.00000000
        ------------------------------------------------------
        TOTAL ENERGY         =    -341.9684227178    0.00786057

COMPUTING TIME FOR ITERATION     6      1578.82

IN FFT FOR LOCAL POTENTIAL N =   100  100  100

MAX AND MIN OF POTENTIAL     1.9199   -1.7156    0.0000


  K  MTXD      En(K)
  1 46036    -12.03948-11.53132-11.52606-11.51723-10.54196-10.53227-10.52149-10.51682       -0.07   0.07   0.21
             -10.51447 -9.49975 -9.49111 -9.48290 -8.66825 -8.66086 -8.65287 -8.64480
              -7.37660 -7.36679 -7.35512 -7.35141 -7.34123 -6.88995 -6.88319 -6.87531
              -6.86397 -5.33118 -5.31470 -5.30979 -5.30697 -5.30084 -5.23766 -5.23437
              -5.22251 -4.26645 -4.26200 -4.25550 -3.14846 -3.14324 -3.13844 -3.13381
              -3.12618 -2.89343 -2.88115 -2.86143 -2.82745 -1.52498 -1.51121 -1.50159
              -1.49551 -0.67952 -0.66341 -0.66145 -0.65279 -0.64847 -0.50947 -0.50447
              -0.49703  0.06810  0.07627  0.08219  0.10712  0.53258  0.59834  0.71824
               0.81772  0.82302  0.83093  0.83884  1.28025  1.28853  1.29286  1.29713
               1.30759  1.65456  1.70661  1.71125  1.72207  1.73175  1.84503  1.86838
               1.93055  1.93363  2.00548  2.01906  2.02736  2.44132  2.45079  2.45966
               2.46920  3.04977  3.06169  3.20567  3.48561  3.50420  3.51061  3.51649
```

5. APPENDIX A

```
              3.53165  3.64005  3.68034  3.73190  3.75587  3.80150  3.81824  3.82222
              3.82280  3.84418  5.14281  5.18499  5.24082  5.26550  5.32184  5.36821
              5.44318  5.46919  5.51890  6.46041  6.57266  6.62235  6.64944  6.69663
              8.21059  8.23365  8.38704  9.34028  9.43263  9.50294 10.18118 10.22972
             10.33574 10.35915 10.43599 10.56448 10.59336 10.62274 11.61883 11.87800
             11.93132 11.96533 11.98268 12.08947

 THE FERMI LEVEL IS AT     6.6966 [eV]


 IN FFT FOR HARTREE-XC  N =   100  100  100
 MAX AND MIN VALUES OF CHARGE DENSITY:   1805.67061      0.05619    (.81E-12)


  ITERATION NUMBER  7

   I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

   1   0  0  0     0.00000 240.00000  -0.31509
   2  -1 -1 -1     0.07712  42.80825   0.62617   0.62503   0.00114  -0.68444
                            -0.02490  -0.00035  -0.00035   0.00000   0.00041
   3   0 -1  0     0.07712  42.74424   0.62524   0.62410   0.00115  -0.68345
                             0.00165   0.00005   0.00003   0.00002   0.00000
   4   0  0  1     0.07712  42.74441   0.62519   0.62410   0.00110  -0.68345
                             0.00138   0.00004   0.00004   0.00000   0.00000
   5   1  0  0     0.07712  42.80485   0.62616   0.62503   0.00113  -0.68450
                            -0.01400  -0.00022  -0.00020  -0.00002   0.00021
   6  -1 -1  0     0.10283   8.75474   0.09955   0.09891   0.00064  -0.10661
                             0.06419   0.00069   0.00069   0.00000  -0.00078
   7   0 -1 -1     0.10283   8.74302   0.09945   0.09877   0.00068  -0.10646
                             0.00001  -0.00002   0.00000  -0.00002   0.00000
   8   1  0  1     0.10283   8.75634   0.09959   0.09894   0.00065  -0.10645
                             0.06368   0.00071   0.00071   0.00000  -0.00079
   9  -2 -1 -1     0.20566 -42.50545  -0.21438  -0.21492   0.00054   0.27201
                             0.11960   0.00062   0.00060   0.00002  -0.00074
  10  -1 -1 -2     0.20566 -42.55748  -0.21458  -0.21511   0.00053   0.27238
                            -0.05428  -0.00030  -0.00028  -0.00003   0.00031
  11  -1  1  0     0.20566 -42.47786  -0.21428  -0.21484   0.00056   0.27195
                            -0.04840  -0.00024  -0.00025   0.00000   0.00032
  12   0  1 -1     0.20566 -42.53544  -0.21451  -0.21505   0.00054   0.27216
                             0.00099   0.00003   0.00002   0.00001   0.00000
  13   1  0 -1     0.20566 -42.54948  -0.21459  -0.21514   0.00055   0.27231
                            -0.05198  -0.00025  -0.00027   0.00002   0.00035
  14   1  2  1     0.20566 -42.48756  -0.21425  -0.21480   0.00055   0.27189
                            -0.04379  -0.00023  -0.00021  -0.00001   0.00027
  15  -2 -2 -1     0.28278 -35.20828  -0.12838  -0.12808  -0.00029   0.16917
                             0.06065   0.00023   0.00023   0.00000  -0.00029
  16  -2 -1  0     0.28278 -35.22824  -0.12845  -0.12817  -0.00028   0.16926
                            -0.00291   0.00000   0.00000  -0.00001   0.00000
  17  -1 -2 -2     0.28278 -35.07302  -0.12791  -0.12764  -0.00027   0.16858
                            -0.04851  -0.00017  -0.00016  -0.00002   0.00022
  18  -1 -1  1     0.28278 -35.17025  -0.12822  -0.12794  -0.00028   0.16899
                            -0.00641  -0.00002  -0.00001  -0.00001   0.00000
  19  -1  1 -1     0.28278 -35.16870  -0.12820  -0.12792  -0.00029   0.16897
                            -0.02281  -0.00011  -0.00010  -0.00001   0.00010
  20   0 -2 -1     0.28278 -35.12633  -0.12802  -0.12774  -0.00028   0.16873
                             0.00238   0.00002   0.00002   0.00001   0.00000


         ITERATION NUMBER  7 :
         ---------------------

         BAND ENERGY          =      -9.8498093112   -0.24542369
         HXC  CORRECTION      =     516.1734065250   -0.80920526
         ------------------------------------------------------
         KINETIC  ENERGY      =     252.6591076897   -0.01234330
         IONIC    ENERGY      =    -663.3081606057    0.59217784
         NONLOCAL ENERGY      =    -115.3741629202   -0.01605297
         HARTREE  ENERGY      =     290.1912042338   -0.57886509
```

```
        EXCHANGE CORRELATION =    -103.9642345932    0.00780781
        ------------------------------------------------------
        ALPHA TERM           =      28.7276840051    0.00000000
        EWALD TERM           =     -30.9071362392    0.00000000
        ------------------------------------------------------
        TOTAL ENERGY         =    -341.9756984297   -0.00727571

COMPUTING TIME FOR ITERATION     7      1466.82

IN FFT FOR LOCAL POTENTIAL N =   100  100  100

MAX AND MIN OF POTENTIAL     1.9199   -1.7165    0.0000


  K  MTXD      En(K)
  1 46036    -12.03597-11.52845-11.52355-11.51457-10.54037-10.52948-10.52171-10.51394       -0.07   0.07   0.21
            -10.51348 -9.49606 -9.48785 -9.47926 -8.67596 -8.66333 -8.65951 -8.65277
              -7.37794 -7.36933 -7.35776 -7.35588 -7.34823 -6.89748 -6.89056 -6.88454
              -6.87405 -5.34002 -5.32656 -5.32124 -5.31646 -5.31071 -5.24148 -5.23796
              -5.22620 -4.27594 -4.27147 -4.26521 -3.16068 -3.15378 -3.15223 -3.14839
              -3.14225 -2.89927 -2.88724 -2.86722 -2.83371 -1.53253 -1.51910 -1.51273
              -1.50358 -0.68964 -0.67594 -0.67256 -0.66387 -0.65759 -0.51786 -0.51290
              -0.50557  0.05471  0.06270  0.06863  0.10108  0.52608  0.59196  0.71230
               0.80710  0.81203  0.81993  0.82777  1.26949  1.27802  1.28148  1.28638
               1.29629  1.64693  1.69476  1.69965  1.71016  1.71587  1.83817  1.86166
               1.92434  1.92757  1.99466  2.00787  2.01624  2.43038  2.44016  2.44843
               2.45837  3.04433  3.05646  3.20081  3.47481  3.49347  3.49966  3.50548
               3.52050  3.63225  3.67156  3.72383  3.74880  3.78979  3.80635  3.81023
               3.81099  3.83217  5.13580  5.17785  5.23312  5.25731  5.31383  5.36007
               5.43565  5.46028  5.51151  6.45163  6.56354  6.61311  6.64038  6.68757
               8.20207  8.22524  8.37911  9.33136  9.42412  9.49441 10.16555 10.21443
              10.31937 10.34322 10.42104 10.54808 10.57784 10.60629 11.61946 11.86841
              11.92156 11.95266 11.96796 12.06163

THE FERMI LEVEL IS AT     6.6876 [eV]


IN FFT FOR HARTREE-XC  N =   100  100  100
MAX AND MIN VALUES OF CHARGE DENSITY:   1805.00237      0.05428    (.83E-12)


 ITERATION NUMBER  8

  I   K-PROT         EK       DEN     V(OUT)    V(IN)     DELTA V   VIONIC

  1   0  0  0     0.00000 240.00000  -0.31509
  2  -1 -1 -1     0.07712  42.76301   0.62550   0.62501   0.00049  -0.68444
                           -0.02493  -0.00035  -0.00035   0.00000   0.00041
  3   0 -1  0     0.07712  42.69852   0.62456   0.62407   0.00049  -0.68345
                            0.00106   0.00005   0.00003   0.00001   0.00000
  4   0  0  1     0.07712  42.69996   0.62453   0.62407   0.00046  -0.68345
                            0.00135   0.00004   0.00004   0.00000   0.00000
  5   1  0  0     0.07712  42.75879   0.62548   0.62501   0.00046  -0.68450
                           -0.01328  -0.00021  -0.00020  -0.00001   0.00021
  6  -1 -1  0     0.10283   8.71818   0.09916   0.09903   0.00013  -0.10661
                            0.06426   0.00069   0.00069   0.00000  -0.00078
  7   0 -1 -1     0.10283   8.70669   0.09906   0.09889   0.00017  -0.10646
                            0.00032  -0.00002   0.00000  -0.00001   0.00000
  8   1  0  1     0.10283   8.72021   0.09920   0.09905   0.00015  -0.10645
                            0.06353   0.00071   0.00071   0.00000  -0.00079
  9  -2 -1 -1     0.20566 -42.52380  -0.21445  -0.21468   0.00024   0.27201
                            0.11887   0.00062   0.00061   0.00001  -0.00074
 10  -1 -1 -2     0.20566 -42.57510  -0.21465  -0.21488   0.00023   0.27238
                           -0.05320  -0.00030  -0.00028  -0.00002   0.00031
 11  -1  1  0     0.20566 -42.49763  -0.21435  -0.21460   0.00025   0.27195
                           -0.04871  -0.00024  -0.00025   0.00000   0.00032
 12   0  1 -1     0.20566 -42.55339  -0.21458  -0.21481   0.00024   0.27216
                            0.00057   0.00003   0.00002   0.00000   0.00000
 13   1  0 -1     0.20566 -42.56725  -0.21466  -0.21490   0.00024   0.27231
                           -0.05234  -0.00025  -0.00027   0.00001   0.00035
```

5. APPENDIX A

```
 14   1  2  1     0.20566 -42.50558  -0.21432  -0.21456   0.00025   0.27189
                            -0.04313  -0.00022  -0.00022  -0.00001   0.00027
 15  -2 -2 -1     0.28278 -35.19100  -0.12832  -0.12818  -0.00013   0.16917
                             0.06107   0.00023   0.00023   0.00000  -0.00029
 16  -2 -1  0     0.28278 -35.21182  -0.12839  -0.12827  -0.00013   0.16926
                            -0.00270   0.00000   0.00000  -0.00001   0.00000
 17  -1 -2 -2     0.28278 -35.05696  -0.12785  -0.12774  -0.00012   0.16858
                            -0.04773  -0.00017  -0.00016  -0.00001   0.00022
 18  -1 -1  1     0.28278 -35.15353  -0.12816  -0.12804  -0.00012   0.16899
                            -0.00608  -0.00002  -0.00001  -0.00001   0.00000
 19  -1  1 -1     0.28278 -35.15134  -0.12814  -0.12802  -0.00013   0.16897
                            -0.02194  -0.00011  -0.00010  -0.00001   0.00010
 20   0 -2 -1     0.28278 -35.10885  -0.12796  -0.12784  -0.00012   0.16873
                             0.00170   0.00002   0.00002   0.00000   0.00000


         ITERATION NUMBER  8 :
         ---------------------

         BAND ENERGY          =      -9.9192726190   -0.06946331
         HXC  CORRECTION      =     515.8605449641   -0.31286156
         -------------------------------------------------------
         KINETIC  ENERGY      =     252.6636532995    0.00454561
         IONIC    ENERGY      =    -663.0582848848    0.24987572
         NONLOCAL ENERGY      =    -115.3851859978   -0.01102308
         HARTREE  ENERGY      =     289.9448358987   -0.24636834
         EXCHANGE CORRELATION =    -103.9622529834    0.00198161
         -------------------------------------------------------
         ALPHA TERM           =      28.7276840051    0.00000000
         EWALD TERM           =     -30.9071362392    0.00000000
         -------------------------------------------------------
         TOTAL ENERGY         =    -341.9766869019   -0.00098847

     -341.9766869019    total kohn-sham energy


 COMPUTING TIME FOR ITERATION     8      1097.79

      -341.97668690      potential energy


       CONTRAVARIANT STRESS TENSOR  (A.U.)              CARTESIAN STRESS (GPA)

       -0.001401    0.000434    0.000439        -0.186714E+01   -0.256806E-01    0.121296E-01   stress 1
        0.000434   -0.001494    0.000632        -0.256806E-01   -0.217820E+01    0.110999E+00   stress 2
        0.000439    0.000632   -0.001529         0.121296E-01    0.110999E+00   -0.215180E+01   stress 3

        -0.00007021         -2.06571254      pressure (au and GPa)


         FORCE (LATTICE COORD.)            FORCE (CARTESIAN COORD. A.U)       NO. TYPE

     0.00087   -0.00083    0.00051     -.44048E-02  0.19151E-01  0.68215E-03   1   C     force
    -0.00006   -0.00048    0.00014     -.46519E-02  0.11266E-02  -.73871E-02   2   C     force
    -0.00015   -0.00039    0.00011     -.39001E-02  -.50903E-03  -.74238E-02   3   C     force
    -0.00006   -0.00062    0.00024     -.53391E-02  0.23625E-02  -.94983E-02   4   C     force
    -0.00015   -0.00038    0.00002     -.50005E-02  -.18518E-02  -.73599E-02   5   C     force
    -0.00068    0.00001    0.00050     0.71756E-02  -.24968E-02  -.92897E-02   6   C     force
    -0.00042   -0.00015    0.00041     0.35953E-02  -.24484E-03  -.79183E-02   7   C     force
    -0.00040   -0.00013    0.00040     0.37321E-02  -.13443E-04  -.72534E-02   8   C     force
    -0.00041   -0.00006    0.00044     0.52500E-02  0.37043E-03  -.66093E-02   9   C     force
    -0.00066   -0.00002    0.00039     0.50566E-02  -.37620E-02  -.94963E-02  10   C     force
     0.00002    0.00053    0.00005     0.80125E-02  0.99411E-03  0.75397E-02  11   C     force
     0.00005    0.00046   -0.00012     0.47718E-02  -.99383E-03  0.70994E-02  12   C     force
     0.00015    0.00039   -0.00011     0.39125E-02  0.60760E-03  0.75615E-02  13   C     force
     0.00016    0.00037   -0.00005     0.43540E-02  0.15355E-02  0.73301E-02  14   C     force
     0.00004    0.00064   -0.00022     0.58931E-02  -.25126E-02  0.93763E-02  15   C     force
     0.00049    0.00004   -0.00063     -.80697E-02  -.18620E-02  0.74393E-02  16   C     force
     0.00047    0.00015   -0.00041     -.36859E-02  0.76860E-03  0.85747E-02  17   C     force
```

```
  0.00039    0.00012   -0.00039     -.37530E-02  0.69613E-04  0.71113E-02  18   C     force
  0.00057    0.00007   -0.00041     -.47409E-02  0.22405E-02  0.87861E-02  19   C     force
  0.00040    0.00012   -0.00050     -.53376E-02  -.13281E-02  0.72052E-02  20   C     force
  0.00001   -0.00002   -0.00053     -.75946E-02  -.71669E-02  -.13739E-03  21   C     force
  0.00012   -0.00004   -0.00047     -.70955E-02  -.48540E-02  0.11074E-02  22   C     force
  0.00002   -0.00012   -0.00045     -.79005E-02  -.60817E-02  -.13932E-02  23   C     force
  0.00015    0.00003   -0.00054     -.71473E-02  -.53929E-02  0.25129E-02  24   C     force
 -0.00006   -0.00010   -0.00036     -.64045E-02  -.58754E-02  -.23236E-02  25   C     force
  0.00053   -0.00059    0.00003     -.76568E-02  0.77544E-02  -.85478E-03  26   C     force
  0.00042   -0.00037   -0.00008     -.62055E-02  0.47590E-02  0.73886E-03  27   C     force
  0.00041   -0.00046    0.00002     -.62326E-02  0.58787E-02  -.76960E-03  28   C     force
  0.00039   -0.00046   -0.00010     -.78233E-02  0.39825E-02  -.96288E-03  29   C     force
  0.00049   -0.00040   -0.00022     -.86864E-02  0.36628E-02  0.11821E-02  30   C     force
 -0.00001    0.00000    0.00055     0.76350E-02  0.74105E-02  -.87839E-04  31   C     force
  0.00000    0.00006    0.00044     0.68780E-02  0.61127E-02  0.86637E-03  32   C     force
 -0.00012    0.00004    0.00047     0.70666E-02  0.48616E-02  -.10296E-02  33   C     force
 -0.00016   -0.00002    0.00055     0.73117E-02  0.53354E-02  -.25597E-02  34   C     force
  0.00004    0.00013    0.00035     0.67186E-02  0.53660E-02  0.23775E-02  35   C     force
 -0.00050    0.00058   -0.00004     0.75530E-02  -.74384E-02  0.11222E-02  36   C     force
 -0.00040    0.00046   -0.00001     0.62384E-02  -.56116E-02  0.88185E-03  37   C     force
 -0.00044    0.00038    0.00010     0.66437E-02  -.47970E-02  -.85216E-03  38   C     force
 -0.00037    0.00045    0.00008     0.73980E-02  -.40006E-02  0.11353E-02  39   C     force
 -0.00049    0.00040    0.00022     0.85757E-02  -.36982E-02  -.12860E-02  40   C     force
 -0.00038    0.00000   -0.00014     -.19845E-02  -.71984E-02  -.52162E-02  41   C     force
 -0.00041    0.00019   -0.00007     0.16515E-02  -.66589E-02  -.30875E-02  42   C     force
 -0.00042   -0.00003   -0.00006     -.12155E-02  -.66916E-02  -.62857E-02  43   C     force
 -0.00047    0.00026   -0.00027     -.74974E-04  -.10219E-01  -.28789E-02  44   C     force
 -0.00073    0.00035   -0.00028     0.92851E-03  -.14070E-01  -.53308E-02  45   C     force
 -0.00004    0.00041   -0.00048     -.90431E-03  -.71999E-02  0.51496E-02  46   C     force
 -0.00017    0.00038   -0.00029     0.13008E-02  -.63224E-02  0.29481E-02  47   C     force
  0.00001    0.00045   -0.00056     -.14701E-02  -.76140E-02  0.64464E-02  48   C     force
 -0.00010    0.00044   -0.00048     -.53406E-03  -.79570E-02  0.47538E-02  49   C     force
 -0.00029    0.00053   -0.00046     0.97445E-03  -.10287E-01  0.33270E-02  50   C     force
  0.00038    0.00000    0.00014     0.18406E-02  0.71556E-02  0.51994E-02  51   C     force
  0.00043    0.00002    0.00005     0.88620E-03  0.65423E-02  0.61685E-02  52   C     force
  0.00038   -0.00019    0.00008     -.14001E-02  0.64586E-02  0.27140E-02  53   C     force
  0.00051   -0.00030    0.00026     -.57051E-03  0.10753E-01  0.29328E-02  54   C     force
  0.00044   -0.00014    0.00014     -.67989E-04  0.80059E-02  0.41565E-02  55   C     force
  0.00003   -0.00041    0.00049     0.11497E-02  0.72687E-02  -.52506E-02  56   C     force
 -0.00001   -0.00045    0.00054     0.13522E-02  0.73576E-02  -.63452E-02  57   C     force
  0.00016   -0.00036    0.00029     -.10010E-02  0.61646E-02  -.28364E-02  58   C     force
  0.00008   -0.00043    0.00053     0.14389E-02  0.84283E-02  -.48122E-02  59   C     force
 -0.00004   -0.00052    0.00019     -.45576E-02  0.21583E-02  -.77213E-02  60   C     force


    1    -341.976687  iteration, kohnsham energy = totalenergy

TOTAL COMPUTING TIME    12062.01      ELAPSED TIME      12076.65
```

5. APPENDIX A

Bibliography

[1] K.P. Driver. *Establishing Quantum Monte Carlo and Hybrid Density Functional Theory as Benchmarking Tools for Complex Solids*. PhD thesis, 2011. 1, 2

[2] Enrique R. Batista, Jochen Heyd, Richard G. Hennig, Blas P. Uberuaga, Richard L. Martin, Gustavo E. Scuseria, C. J. Umrigar, and John W. Wilkins. **Comparison of screened hybrid density functional theory to diffusion Monte Carlo in calculations of total energies of silicon phases and defects**. *Phys. Rev. B*, **74**:121102, Sept. 2006. 1

[3] E. Schrödinger. **An Undulatory Theory of the Mechanics of Atoms and Molecules**. *Phys. Rev.*, **28**:1049–1070, Dec. 1926. 2

[4] Andrew Leach. *Molecular Modelling: Principles and Applications (2nd Edition)*. Prentice Hall, 2 edition, April 2001. 2, 6, 7

[5] José Luís Martins. Private communications, 2010-2011. 2, 6, 7, 9, 15, 20, 23

[6] Shekhar Borkar. **Thousand core chips: a technology perspective**. In *Proceedings of the 44th annual Design Automation Conference*, DAC '07, pages 746–749, New York, NY, USA, 2007. ACM. 3

[7] Dominik Marx, J Grotendorst Ed, and John Von Neumann. *Ab initio molecular dynamics : Theory and Implementation*. 2000. 6, 7

[8] M. Born and K. Huang. *Dynamical theory of crystal lattices*. Oxford classic texts in the physical sciences. Clarendon Press, 1998. 7

[9] J.R. Shewchuk. **An introduction to the conjugate gradient method without the agonizing pain**. Technical report, 1994. 7

[10] Richard H. Byrd, Richard H. Byrd, Peihuang Lu, Peihuang Lu, Jorge Nocedal, Jorge Nocedal, Ciyou Zhu, and Ciyou Zhu. **A Limited Memory Algorithm for Bound Constrained Optimization**. *SIAM Journal on Scientific Computing*, **16**:1190–1208, 1994. 7

[11] M C Payne, M P Teter, J D Joannopoulos, T A Arias, and J D Joannopoulos. **Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients**. *Reviews of Modern Physics*, **64**[4]:1045–1097, 1992. 13

[12] Ernest R. and Davidson. **The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices**. *Journal of Computational Physics*, **17**[1]:87 – 94, 1975. 14

[13] H. Rutishauser. **Simultaneous iteration method for symmetric matrices**. *Numerische Mathematik*, **16**:205–223, 1970. 10.1007/BF02219773. 14

[14] D M Wood and A Zunger. **A new method for diagonalising large matrices**. *Journal of Physics A: Mathematical and General*, **18**[9]:1343, 1985. 14

[15] José Luís Martins and Marvin L. Cohen. **Diagonalization of large matrices in pseudopotential band-structure calculations: Dual-space formalism**. *Phys. Rev. B*, **37**:6134–6138, April 1988. 14, 15

[16] A.F. Breitzman. *Automatic Derivation and Implementation of Fast Convolution Algorithms*. PhD thesis, Drexel University, 2003. 15

[17] D. Donnelly and B. Rust. **The fast Fourier transform for experimentalists. Part II. convolutions**. *Computing in Science Engineering*, **7**[4]:92 – 95, July-Aug. 2005. 15

[18] James W. Cooley and John W. Tukey. **An Algorithm for the Machine Calculation of Complex Fourier Series**. *Mathematics of Computation*, **19**[90]:297, April 1965. 16

BIBLIOGRAPHY

[19] E.C. Chu and A. George. *Inside the FFT black box: serial and parallel fast Fourier transform algorithms*. Computational mathematics series. CRC Press, 2000. 16, 17

[20] B.P. Flannery W.H. Press, S.A. Teukolsky, W.T. Vetterling. *Numerical recipes: the art of scientific computing*. Cambridge University Press, 2007. 16, 17, 27, 29

[21] Pierre Duhamel and Martin Vetterli. **Fast fourier-transforms - A tutorial review and a state-of the art**. *Signal Processing*, **4**[19]:259–299, 1990. 16, 17

[22] Michael T. Heideman, Don H. Johnson, and C. Sidney Burrus. **Gauss and the history of the fast Fourier transform**. *Archive for History of Exact Sciences*, **34**[3]:265–277, 1985. 16

[23] M. Frigo and S.G. Johnson. **The Design and Implementation of FFTW3**. *Proceedings of the IEEE*, **93**[2]:216 –231, Feb. 2005. 16, 17, 27, 29

[24] NVIDIA Corporation. **CUDA CUFFT Library-Documentation**, 2011. 17, 31, 37, 43

[25] Intel Corporation. **Intel® Math Kernel Library – Documentation**, 2011. 17

[26] Advanced Micro Devices (AMD) Corporation. **AMD Core Math Library - Documentation**, 2011. 17

[27] Ian Kirker. *Demanding Parallel FFTs : Slabs & Rods*. Master's thesis, 2008. 17, 27, 29

[28] Michael Susan L. Blackford, James W. Demmel, Jack Dongarra, Iain S. Duff, Sven Hammarling, Greg Henry and And and Whaley., R. Clint Heroux, Linda Kaufman, Andrew Lumsdaine, Antoine Petitet, Roldan Pozo, Karin Remington. **An updated set of basic linear algebra subprograms (BLAS)**. *ACM Trans. Math. Softw.*, **28**:135–151, June 2002. 18

[29] Mark D. Hill and Michael R. Marty. **Amdahl's Law in the Multicore Era**. *Computer*, **41**[7]:33–38, July 2008. 19, 44

[30] NVIDIA Corporation. **Compute Unified Device Architecture - CUDA**, 2011. 21, 32, 37, 38, 40

[31] G. Cummins, R. Adams, and T. Newell. **Scientific computation through a GPU**. In *Southeastcon, 2008. IEEE*, pages 244 –246, April 2008. 21, 31

[32] D. Luebke. **CUDA: Scalable parallel programming for high-performance scientific computing**. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 836 –838, May 2008. 21, 31

[33] Michael Wolfe. **Programming GPUs Today**. "Compilers and More" series. Oct. 2008. Accessed on June 2011, available: http://www.pgroup.com/resources/articles.htm. 21

[34] David Kirk and Wen mei Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 1 edition, 2010. 21

[35] NVIDIA. **NVIDIA's Next Generation CUDA Compute Architecture: Fermi**. NVIDIA whitepaper. Accessed on September 2011, available:http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf. 22

[36] M. Frigo and S.G. Johnson. **Pruned FFTs**. `http://www.fftw.org/pruned.html`. Accessed December 4, 2010. 29, 32

[37] S. Singh and S. Srinivasan. **Architecturally efficient FFT pruning algorithm**. *Electronics Letters*, **41**[23]:1305 – 1306, Nov. 2005. 29

[38] Yihu Xu, DaeHwan Kim, SangHak Lee, and Myong-Seob Lim. **Split-radix FFT pruning for OFDM based Cognitive Radio system**. In *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*, pages 421 –424, Sept. 2009. 29

[39] Franz Franchetti and Markus Püschel. **Generating High-Performance Pruned FFT Implementations**. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 549–552, 2009. 29

[40] Y. Ogata, T. Endo, N. Maruyama, and S. Matsuoka. **An efficient, model-based CPU-GPU heterogeneous FFT library**. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1 –10, April 2008. 32

[41] C.P. da Silva, L.F. Cupertino, D. Chevitarese, M.A.C. Pacheco, and C. Bentes. **Exploring Data Streaming to Improve 3D FFT Implementation on Multiple GPUs**. In *Computer Architecture and High Performance Computing Workshops (SBAC-PADW), 2010 22nd International Symposium on*, pages 13 –18, oct. 2010. 32

[42] NVIDIA Corporation. **CUDA CUBLAS Library - Documentation**, 2011. 37