

# High-Speed Pseudo-Orthogonalization for the Car-Parrinello Method

Tomoo AOYAMA<sup>a\*</sup>, Takatoshi HIGUCHI<sup>b</sup> and Umpei NAGASHIMA<sup>c</sup>

<sup>a</sup>Faculty of Engineering, University of Miyazaki  
Gakuen-Kibanadai, Miyazaki 889-2192, Japan

<sup>b</sup>Science and Technology, Mizuho Information & Research Institute, Inc.  
2-3 Kanda-Nishikicho, Chiyoda-ku, Tokyo 101-8443, Japan

<sup>c</sup>Grid Technology Research Center, National Institute for Advanced Industrial Science and Technology  
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

\*e-mail: [t0b217u@cc.miyazaki-u.ac.jp](mailto:t0b217u@cc.miyazaki-u.ac.jp)

(Received: September 11, 2006; Accepted for publication: December 14, 2006; Published on Web: April 4, 2007)

An approximate orthogonalization method called “pseudo-orthogonalization” was derived here from the Schmidt orthogonalization in a partial space. Precision of the orthogonalization is decimal 2 digits when the partial space of 1/3 of the Schmidt orthogonalization is used. The speed-up ratio is 3.3 with 1000 vectors of 1500<sup>th</sup> dimension. Although the pseudo-orthogonalization is high-speed, the precision is inadequate for general numerical calculations. Two possible approaches to improve this precision were considered: (a) introducing a band structure in the vectors, and (b) restricting the sign of the vector elements. These approaches improved the precision by a factor of 2.1 and 1.5, respectively, without increasing the CPU time.

To demonstrate the applicability of this high-speed pseudo-orthogonalization for large-scale calculations, we coded a complex-form of it into the SCF part of the Car-Parrinello method. In calculations of the total energy for bulk-silicon, the difference in total energy calculated using the pseudo-orthogonalization and that using the Schmidt orthogonalization was less than  $O(-5)$  [a.u.]. Based on this accuracy, this pseudo-orthogonalization can be used to speed-up the Car-Parrinello method when it is applied to large-scale calculations.

**Keywords:** Schmidt orthogonalization, Car-Parrinello method, Orthogonal space

## 1 Introduction

The Car-Parrinello method [1] is a well-known approach for ab-initio calculations to research properties of solids and surfaces. The method requires huge CPU power, and the dominant part is 3D-FFT [2, 3] until 1000 atoms. However, over 1000 atoms the part is the Schmidt orthogonalization [4]. Because, the number of 3D-FFT calculations is  $O(M^2 \log M)$ , and that of the Schmidt orthogonalization is  $O(M^3)$ , where  $M$  is the number of vectors and  $O(X)$  means on the order of  $X$ . In this study, our objective was to reduce the number of calculations of the Schmidt orthogonalization by considering the arithmetic operations, and data transfer (for alignment of configurable hardware such as Field Programmable Gate Array

(FPGA)).

Several approaches were considered. One approach was to improve the rate of data transfer [5]. Although this approach cannot reduce the number of calculations, it can be effective in improving data transfer among caches and main storage in parallel machines. Possible approaches to reduce the number of calculations are limited, because essential calculations of rotation and matrix-products are already algorithms of  $O(M^3)$ .

An approximate orthogonalization therefore seems to be the best option. Such an approximation is acceptable because the Self-Consistent Field (SCF) step in the Car-Parrinello method does not require decimal 13 digits at the non-convergence stages.

## 2 Approximation approaches

### 2.1 Direct generation of orthogonal vectors

One approach is to generate orthogonal vectors directly without the orthogonalization. The number of rotation calculations of  $M$  vectors is  $O(M^3)$ . If the rotations are limited to  $M'$  elements ( $<M$ ), this number decreases to  $O(M'M^2)$ . Consider the following vectors,

$$(Ai)_k = \{\text{sgn}(k)\delta ik; i = 1, 2, \dots, N\}, \quad (1)$$

$$\text{sgn}(k) = \text{sign}\{1, \text{random}()\}, \quad (2)$$

where  $\delta ik$  is the delta function,  $\text{sign}\{\}$  assigns the sign of the second argument to a variable of the first argument, and  $\text{random}()$  is a function to generate uniform random numbers [6] in the interval  $[-1, 1]$ . The set  $\{(Ai)_k; k=1, 2, \dots, M \leq N\}$  is orthogonal. For a set of finite rotations, we therefore can construct an orthogonal vector set, although it cannot simulate many waves. We did not adopt this approach.

### 2.2 Pseudo-orthogonalization

Another approach is approximate orthogonalization as follows. In the Schmidt orthogonalization, a normalized vector is  $(Ai)$ , and the set is  $\{(Ai)_k; i \leq N, k \leq M\}$ , where  $M$  is the number of vectors and  $N$  is the number of elements. A scalar  $P_{kl}$  based on their product is then

$$P_{kl} = \sum_i (Ai)_k^* (Ai)_l \quad (3)$$

An orthogonal set  $\{(Bi)_k\}$  can be defined as

$$(Bi)_k = (Ai)_k - \sum_{l>k} P_{kl} (Ai)_l. \quad (4)$$

The number of calculations is  $N*M*(M-1)/2$ . To reduce this number, we introduced the following pseudo-orthogonalization:

$$(Ci)_k = (Ai)_k - \sum_{K \geq l > k} P_{kl} (Ai)_l, \quad (5)$$

$$K = \min(k + \Delta, M), \Delta = \{\text{odd number} < M\}. \quad (6)$$

This orthogonalization is valid when  $K=M$  or when  $\{(Ai)_k\} = \delta ik$ . The vector set  $\{(Ci)_k\}$  is approximately orthogonal. The validity of this pseudo-orthogonalization needs to be tested for  $K$ .

## 3 Numerical tests

### 3.1 Vectors constructed using uniform random numbers

To test the validity of the pseudo-orthogonalization, first we define the following vectors,

$$\{(Ai)_k; i \leq N, k \leq M\} = \text{random numbers}. \quad (7)$$

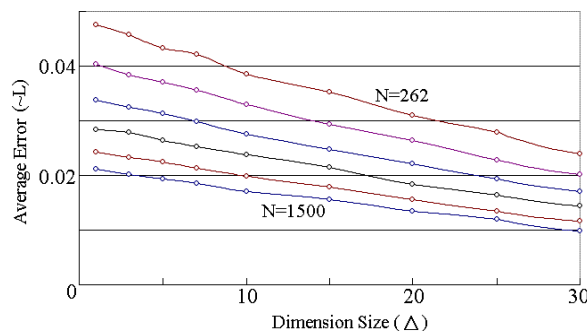


Figure 1. Average error  $\sim L$  of the pseudo-orthogonalization.

The random numbers are uniform in the interval  $[-1, 1]$ . The vectors are normalized as

$$\|(Ai)_k\| = 1. \quad (8)$$

The average error of the orthogonalization was estimated as follows.

- 1) A vector set  $\{(Ai)_k\}$  was regarded as a matrix  $A_{ik}$ ,
- 2) Off-diagonal elements,  $A_{ik}$  ( $i \neq k$ ), are zero in ideal orthogonalization. Thus,  $\sum_{i \neq k} A_{ik}^2 = 0$ .
- 3) The precision of the pseudo-orthogonalization was then represented by the error amplitude  $L_{ik}$  and the average error amplitude  $\sim L$  as follows:

$$L_{ik} = [A_{ik}^2]^{0.5}, \quad (9)$$

$$\sim L = \left[ \sum_{i \neq k} A_{ik}^2 \right]^{0.5} / (N * M - M). \quad (10)$$

Figure 1 shows  $\sim L$  for calculation using Eqs. (5) and (6) with  $\{(Ai)_k\}$  as a function of dimensionless  $\Delta = \{1, 3, 5, 7, 10, 15, 20, 25, 30\}$  and  $N = \{262, 372, 528, 748, 1060, 1500\}$ .

The results reveal that  $\sim L$  decreases with increasing  $\Delta$ . The orthogonalization is complete at  $K=M$  (where  $K$  is in Eq. (6)). The result,  $\sim L < O(-2)$  when  $\Delta \geq 30$ ,  $N=1500$ , is acceptable for the initial SCF steps of the Car-Parrinello method. When  $N$  is a sufficiently large number,  $\sim L \rightarrow 0$  due to the normalization of Eq. (8). The CPU speed-up ratio is  $M/K$ .

Figure 2 shows the maximum  $L_{ik}$ ,  $\max(L_{ik})$ , of the pseudo-orthogonalization.

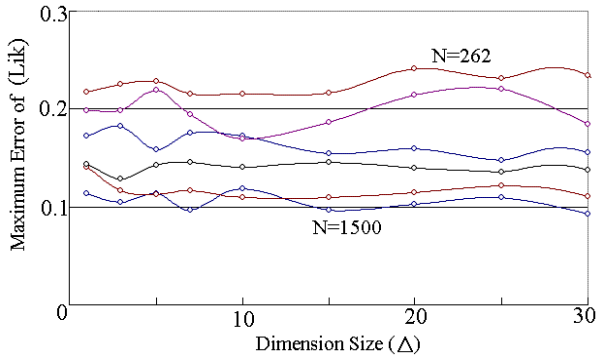


Figure 2. Maximum error  $\max(Lik)$  of the pseudo-orthogonalization.

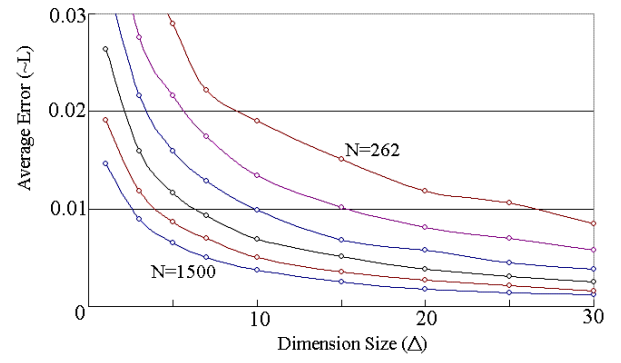


Figure 4. Average error  $\sim L$  of the off-diagonal elements in a band vector orthogonalization.

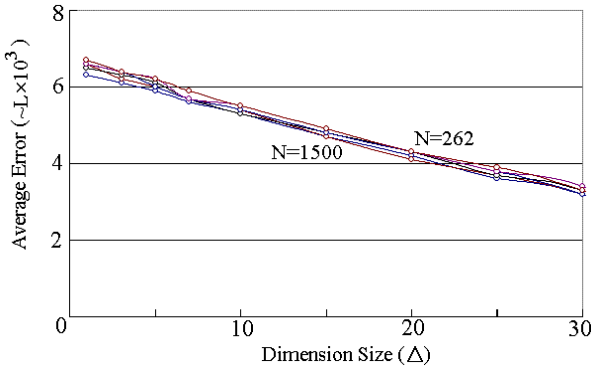


Figure 3. Average error  $\sim L$  for the off-diagonal elements of diagonal dominance vectors.

The only deficiency in this pseudo-orthogonalization is that  $\max(Lik)$  does not approach zero when  $N \rightarrow \infty$ . The non-zeros are in the space between  $K$  and  $M$  of Eq. (6). That is  $\{i \leq K \text{ and } k \leq K: (Ai)_k = \delta ik; \text{ otherwise: } (Ai)_k = Lik \neq 0\}$ . If the waves, vectors in the Car-Parrinello method, are distributed in the space, the pseudo-orthogonalization is not valid.

### 3.2 Diagonal dominance vectors

The orthogonalization is complete when  $\{(Ai)_k\} = \delta ik$ . When the vectors are arranged into a matrix that has diagonal dominance, and thus are called “diagonal dominance vectors,” the pseudo-orthogonalization approaches complete transformation. Here, these dominance vectors were constructed as

$$\{(Ai)_k; i \leq N, k \leq M\} = \delta ik + R * \text{random}(), \quad (11)$$

$$0 < R,$$

where  $\text{random}()$  is a function to generate uniform random numbers, and  $R$  is a small positive coefficient.

When  $R=0.1$ , the amplitude of the off-diagonal elements is  $O(-1)$ . When  $\{R=O(-n), n \geq 3\}$ , however, the amplitude is smaller, namely,  $O(-n-1)$ , and depends on  $\Delta$  and  $N$ . Figure 3 shows  $\sim L$  for the off-diagonal elements. When the boundary condition is  $R=0.01$ , there is little difference between  $\sim L$  for  $N=262$  and that for  $N=1500$ . In conclusion, the pseudo-orthogonalization is effective for small  $N$  (i.e.,  $N \approx 300$ ).

## 4 Precision of pseudo-orthogonalization for initial vectors

### 4.1 Band vector

The ordering of non-orthogonal vectors, namely, initial vectors, can be considered a set. Here, we consider this set as a matrix, and define a “band vector” as a band-structured matrix. In the Car-Parrinello method, the band vector is a localized wave on the atoms related to the band and is defined as

$$\{(Ai)_k; i \leq N, k \leq M, |i - k| \leq W\} = \text{random}(), \quad (12)$$

$$\{(Ai)_k; i \leq N, k \leq M, |i - k| > W\} = R * \text{random}(), \quad (13)$$

where  $W$  is the band width and  $R$  is the same coefficient as in Eq. (11). Eq. (12) is for the band part. The band vector is normalized also. Figure 4 shows the precision of the off-diagonal elements when  $R=0.1$  and  $W=3$ ; i.e., the band width is 7. The initial sharp decrease in  $\sim L$  reveals the band structure. When  $N=1500$ ,  $\sim L$  is 2.1 lower, indicating an improvement in precision. In conclusion, the use of band vectors in the pseudo-orthogonalization is effective for localized waves.

Table 1. Average precision of off-diagonal elements of an orthogonal set.

$\Delta$	Original	Sign-restricted	Improvement
10	0.0171	0.0114	1.5
15	0.0155	0.0101	1.5
20	0.0135	0.0090	1.5
25	0.0119	0.0080	1.5
30	0.0099	0.0070	1.4

## 4.2 Alternative sign vectors

The band vector improves the precision of the pseudo-orthogonalization certainly; however, it is restricted within localized waves. We wish the approach to general waves. We consider orthogonal vectors,

$$\{\text{Re}[(Ai)]_k; i \leq N, k \leq M\} = \{\sin(x)\}, \quad (14)$$

$$\{\text{Im}[(Ai)]_k; i \leq N, k \leq M\} = \{\sin(x + \pi)\}, \quad (15)$$

$$x = 2\pi(M + k)(i/N), (Ai) *_k (Ai)_l = \delta_{kl}, \quad (16)$$

where  $\text{Re}[]$  and  $\text{Im}[]$  are extraction functions of the real and imaginary parts of the complex. Although the vectors themselves have not been adopted in the Car-Parrinello method because of the small degree-of-freedom, the sign of the vectors can be used to improve the precision of non-localized waves. We wish to gain precision from cost of the degree-of-freedom. First, we consider a set of synthesized random numbers,

$$\text{S-random}(i, k) = \text{Sgn}(i, k) * \text{random}(), \quad (17)$$

$$\text{Sgn}(i, k) = \text{sign}\{1, \sin(2\pi(M + k)(i/N))\}, \quad (18)$$

where  $\text{Sgn}()$  is a function to obtain the sign of the vector and  $\text{sign}\{\}$  is a function in Fortran language. Here,  $\text{S-random}()$  is called "sign-restricted (random) numbers." Using these sign-restricted numbers and the concept of band vector, the real parts of the initial vectors can be defined as,

$$\{(Ai)_k; i \leq N, k \leq M, |i - k| \leq W\} = \text{random}(), \quad (19)$$

$$\{(Ai)_k; i \leq N, k \leq M, |i - k| > W\} = \text{S-random}(i, k), \quad (20)$$

Note the absence of the parameter  $R$  as in Eq. (13). Table 1 shows the precision of the off-diagonal elements calculated using the sign-restricted numbers for  $N=1500$ . This approach improved the precision by a factor of about 1.5, without requiring an  $R$ -parameter and without increasing the CPU time.

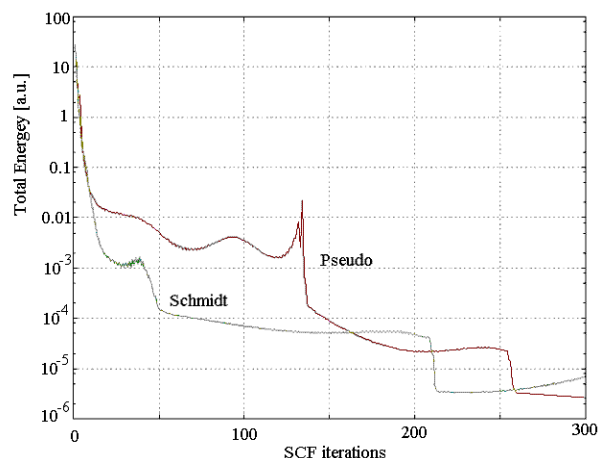


Figure 5. Convergence of SCF energies for bulk-silicon calculated using pseudo-orthogonalization and Schmidt orthogonalization.

## 5 Application of pseudo-orthogonalization to the Car-Parrinello method

We coded a complex form of the pseudo-orthogonalization into the SCF part of the Car-Parrinello method, where  $M=16$ ,  $N=147$ ,  $\Delta=5$ , and the molecule was bulk-silicon. Compared with metals, bulk-silicon is a semiconductor and the wave function is localized on atoms. Figure 5 shows the calculated total energy (in units of a.u.) as a function of SCF iterations.

The difference in energy calculated using the pseudo-orthogonalization and that using the Schmidt-orthogonalization was less than  $10^{-5}$  [a.u.], which is sufficient precision to discuss the energy status. Both calculation methods converged relatively rapidly.

## 6 Hardware problems using FPGA

When pseudo-orthogonalization is performed using a FPGA with  $10^6$  gates [7], the Input/Output (I/O) interface is the key component. Despite the vast arithmetic resources currently available, the number of I/O pins is inadequate to take advantage of pseudo-orthogonalization on decimal 13 digits. This problem of a narrow interface needs to be solved by hardware designers today. The problem is found generally on 64 bits floating point numerical calculations.

### 6.1 Data compression

Several data compression methods are currently available for general applications. One method involves a simple

algorithm specialized for an I/O interface for an 8-byte Floating Point expression (FP8), which has 13 significant digits. Pseudo-orthogonalization has about 2-3 significant digits. The element values of orthogonal vectors are in the  $[-\xi, +\xi]$  interval, where  $\xi$  is a positive finite number less than 1. On the condition, the FP8 should be converted to the fixed point format based on the complementary expression of 2. The fixed point format has multiple expressions of 1-8 bytes, which express numbers in  $[-2^n, +2^n-1]$ ,  $n=\{7, 15, 31, 63\}$ . The conversion and the reverse algorithms are

$$T = X(2^n - 1)/\xi, \quad (21)$$

$$X = T\xi/(2^n - 1), \quad (22)$$

where  $X$  is a number in FP8, and  $T$  is a number in the fixed point format of  $n$ -bits. These can be expressed as FTn, which can be transferred through  $n$ -bits interface pins. Thus, such conversion improves the data transfer rate in pseudo-orthogonalization by  $64/n$ .

## 6.2 Precision of the FP8 to FT8/FT16 conversion

Table 2 shows the precision of conversions between FP8 and FT8/FT16 formats using uniform random numbers in the interval  $[-1,1]$  for  $N=1500$ .

The precision for the off-diagonal elements and band parts was  $O(-3)$  and  $O(-5)$ , respectively, both of which are sufficient to transfer pseudo-orthogonalization vectors. In conclusion, because the data transfer in pseudo-orthogonalization predominantly involves off-diagonal elements, the conversion can increase the transfer rate through the interface by a factor of 8.

## 7 Conclusions

A high-speed orthogonalization method derived from the Schmidt orthogonalization in partial space was developed. Because this is an approximate transformation, it is called a pseudo-orthogonalization. The speed-up ratio is about 3.3, and the precision is  $O(-2)$ , which was estimated based on the amplitude of the off-diagonal elements in  $1500^{\text{th}}$  dimensional vectors. Both the speed-up ratio and precision are affected by the dimensions of the partial space.

Although this pseudo-orthogonalization is high-speed, its precision is insufficient for general applications. We therefore considered two approaches to improve the precision: introduction of a band structure for vectors, and use of alternative sign random numbers. The precision improved by a factor of 2.1 and 1.5, respectively, without increasing the CPU time. The first approach can be used only for localized waves, whereas the latter can be used for general applications.

Table 2. Precision of the off-diagonal elements ( $n=8$ ) and band parts ( $n=16$ , or diagonal elements) in the conversion from 8 or 16-bit fixed-point format to 8-byte floating point format.

	Speed-up ratio	Stand.Dev.	Maximum error
$n=8$	8	0.0046	0.0079
$n=16$	4	0.000018	0.000031

To demonstrate the applicability of this high-speed pseudo-orthogonalization for large-scale calculations, we coded a complex form of the pseudo-orthogonalization into the SCF part of the Car-Parrinello method. The difference in total energy for bulk-silicon calculated using the pseudo-orthogonalization and the Schmidt orthogonalization was less than  $O(-5)$ . Although this is acceptable, the orthogonalization is an approximation with  $O(-2)$ ; and therefore should be replaced by the original Schmidt-orthogonalization at the convergence limit of the SCF.

In further research, we will execute ab-initio calculations using configurable devices such as FPGA. Recent advances in FGPA include advances in various arithmetic resources, cache, FIFO (First In First Out memory), and programmable mega gates. The "bottleneck" seems to be I/O interfaces. To overcome this, in this study we considered a data compression method for the 8-byte floating point format. The compression ratio is 4 or 8, and the precision is  $O(-5)$  or  $O(-3)$ , respectively, both of which are sufficient to rapidly transfer pseudo-orthogonalization vectors.

## References

- [1] Car, R. and Parrinello, M., *Phys. Rev. Lett.*, **55**, 2471-2474 (1985).
- [2] Tohru Sasaki and Kazuaki Murakami, Reconfigurable 3D-FFT Processor for the Car-Parrinello Method (in Japanese), *J. Comput. Chem. Jpn*, **4**, 147-154 (2005).
- [3] A representative mathematical library for 3D\_FFT is Intel Math Kernel Library (version 8.1). The introduction is <http://www.xlsoft.com/jp/product/intel/perflib/mkl/whatsnew.html>
- [4] <http://www.cc.tsukuba.ac.jp/mimosa/workshop/2004/>

- [5] Hiroshi Murakami, An orthogonalization to realize massive data-transfer effectively for huge dimensional vectors (in Japanese), *Soc. of Compt. Chem. of Japan, annual conference*, 2P14 (2004).
- [6] D. E. Knuth, *The Art of Computer Programming*, Stanford Univ. Press.
- [7] Introduction for Virtex-4 FPGA family, *Xcelljournal*, **52**, XILINX Co. Ltd., San Jose, CA, 2005.