NANOPACK: PARALLEL CODES FOR SEMIEMPIRICAL QUANTUM-CHEMICAL CALCULATIONS OF LARGE SYSTEMS IN THE *SP*- AND *SPD*-BASIS

PARVAZ K. BERZIGIYAROV[1], VALENTINE A. ZAYETS[1],
ILYA YA. GINZBURG[1], VLADIMIR F. RAZUMOV[1]
AND ELENA F. SHEKA[2]

[1] *Institute of Problems of Chemical Physics,*
Russian Academy of Sciences,
Chernogolovka, Moscow 142432, Russia

[2] *Russian University of Peoples' Friendship,*
Miklukho-Maklaya 6, Moscow 117302, Russia
sheka@icp.ac.ru

(Received 10 June 2001; revised manuscript received 25 February 2002)

Abstract: A parallel implementation of the conventionally used NDDO (MNDO, AM1, PM3, CLUSTER-Z1) and modified NDDO-WF (CLUSTER-Z2) techniques for semiempirical quantum-chemical calculations of large molecular systems in the *sp*- and *spd*-basis, respectively, is described. The atom-pair distribution of data over processors forms the basis of the parallelization. The technological aspects of designing scalable parallel calculations on supercomputers (by using ScaLAPACK and MPI libraries) are discussed. The scaling of individual algorithms and entire package was carried out for model systems with 894, 1920, and 2014 atomic orbitals. The package speedup provided by different multi-processor systems involving a cluster of the Intel PIII processors, Alpha-21264-processor-built machine MBC-1000M, and CRAY-T3E, is analyzed. The effect of computer characteristics on the package performance is discussed.

Keywords: quantum chemistry, NANOPACK parallel codes, semiempirical NDDO and NDDO-WF approaches, *sp*- and *spd*-basis

1. Introduction

Modern quantum chemistry (QC) offers a wide range of tools and techniques that suit the needs of calculations of large nano-sized systems. The most promising seem to be the semiempirical (SE) techniques [1, 2], among which those based on the NDDO (*neglect of diatomic differential overlap*) approximation [2], namely, MNDO and its modifications [3–6], AM1 [7] and PM3 [8], are of the largest interest due to a close similarity with *ab initio* ones relating to reliability and accuracy of results. These techniques and appropriate programs are being used in widespread one-processor program packages MOPAC [9], AMPAC [10], and CLUSTER-Z1 [11] for QC calculations of many-atomic systems in the *sp*-basis. On going to the *spd*-basis, the above techniques were modified within the NDDO-WP [12, 13] and MNDO/d [14] approximations that led to the foundation of program packages CLUSTER-Z2 [15] and MNDO94 [16], respectively.

Meeting requirements on speeding up calculations, we have performed the transformation of two above sequential codes CLUSTER-Z1 and CLUSTER-Z2 into parallel ones. The sequential algorithms of the codes, similarly to other QC programs, represent a structure with a massive parallelism [17] that makes them promising for large system calculations. However, QC calculations of such systems face two main problems: (1) operation with and storage of big arrays of data, and (2) performance of a large bulk of calculations. The former concerns mainly accessible computer memory while the latter governs calculation speed.

Any parallelization scheme implies the performance distribution over processor arrays. As for the data, two operation schemes can be implemented related to either replication or distribution of the data following a particular algorithm. Data replication scheme implies that each processor has its own copy of all data required for calculations. The size (number of atoms, NAT, and, consequently, the number of atomic orbitals, AOs, *NORBS*) of the run problem is restricted by the RAM accessible at one processor. RAM of 256MB limits the data to $NORBS = 2200$ that approximately corresponds to NAT about 400–500 in the *sp*-basis and about 300–400 in the *spd*-one. *NORBS* of 4000–10000 will require from 2Gb to 16Gb and more memory. The parallel implementation of the CLUSTER-Z1 and CLUSTER-Z2 codes, based on the replication scheme, was done in the MP-AM1 [18] and MP-ZAVA [19] codes, respectively. As occurred, the approach offers a relatively labor-saving realization of parallel program, since only computationally intensive parts of the sequential code had to be reworked. Successful applications of the codes to the calculations of $n*\mathrm{Si}_{60}$ oligomers with n ranging from 2 to 8 [20] and of complexes of carboxylic acids with nano-sized titania substrates [21] have well exhibited a qualitatively new stage of calculations. Restrictions put by processor RAM can be avoided by the data distribution scheme. In this case, the data are partitioned over processors to provide the performance on each processor with minimum data exchange with other processors. This way, if well implemented, leads to programs that can be effectively run on distributed memory supercomputers that constitute the majority of the modern supercomputing systems. A choice of an appropriate algorithm of the data distribution is a key point of the approach. In general, this problem concerns graph-mapping and belongs to NP-complete problems. NANOPACK parallel codes, described in the current paper, present a parallel implementation of both CLUSTER-Z1 and CLUSTER-Z2 sequential codes, referred to hereinafter as NDDO and NDDO-WF modes of the package, respectively, basing on a scheme of atom-pair distribution of data.

2. Theoretical background

The inner architecture of the sequential CLUSTER-Z1 and CLUSTER-Z2 codes is oriented on the solution of a QC problem for a polyatomic (and many-electron) system in the form of a molecule or model cluster. In terms of the widely adopted Born-Oppenheimer approximation, Hamiltonian of the considered molecular system is commonly subdivided into two parts. Accordingly, the Schrodinger equation solution for the electronic subsystem is sought in the LCAO MO approximation. Further using of the Hartree-Fock procedure transforms a system of operator equations to the Hartree-Fock SCF matrix equation in the Roothaan approximation [1, 2]:

$$\mathbf{F}(\mathbf{C})\mathbf{C} = \mathbf{SCE}. \tag{1}$$

The Fock matrix ($\mathbf{F}$) elements are expressed in terms of the electron density matrix $\mathbf{P}$ (restricted Hartree-Fock closed-shell case[1]):

$$P_{ij} = 2\sum_{k}^{occ} C_{ik}C_{jk}, \tag{2}$$

where summation is performed over all occupied molecular orbitals, and have the form [2]:

$$F_{ij} = H_{ij} + \sum_{k,l=1}^{NO} P_{kl}[<ij|kl> - \frac{1}{2} <ik|jl>]. \tag{3}$$

Here H_{ij} is a matrix element of the one-electron *core Hamiltonian* and $<ij|kl>$ present *two-electron integrals*

$$<ij|kl> = \iint \chi_i(r_1 - R_A)\chi_j(r_1 - R_B)\frac{1}{r_{12}}\chi_k(r_2 - R_C)\chi_l(r_2 - R_D)dV_1dV_2, \tag{4}$$

that constitute the most serious problem of the computational process [22]. The NDDO approximation neglects three- and four-center integrals on the basis of the *zero differential overlap* of atomic orbitals. Therefore, only one-center $<AA|AA>$ and two-center $<AA|BB>$ integrals have to be calculated to obtain the Fock matrix. Using Slater functions with real spherical harmonics, the above expression is transformed into the form [23]:

$$C_{Coul}^{AB} = \sum_{l=|l_1-l_2|}^{l_1+l_2} \sum_{l'=|l_3-l_4|}^{l_3+l_4} I_{ll'}(l_1,l_2,l_3,l_4), \tag{5}$$

where

$$I_{ll'} = \sum_{m(m_1,m_2)} \sum_{m'(m_3,m_4)} \delta_{mm'} \iint \rho_{lm}(1)\frac{1}{r_{12}}\rho_{l'm'}(2)dV_1dV_2. \tag{6}$$

However, the above scheme was found to yield overestimated values of one- and two-center integrals, which results in a poor fitting with experimental data. This drawback can be released in two ways. In terms of the commonly used NDDO approximation implemented in MNDO [3], AM1 [7], PM3 [8], and MNDO/d [14] techniques, the continuous charge distribution ρ_{lm} is replaced by a set of 2^l point charges of $\pm 1/2^l$ a.u. each (zero total), on retention of the ρ_{lm} symmetry [3]. It should be noted as well that one-center integrals are not calculated therewith but are the method parameters. Linear dimensions of the point charge system depend on the quantum numbers n, l and Slater exponent ε. However, for small ε, atomic orbitals become smeared out, so that the linear dimensions of the system of point charges may

[1] What follows is totally coherent with unrestricted Hartree-Fock open-shell case [1, 2] as well.

be compared with interatomic distances. In particular, this circumstance makes the parametrization of alkali metal atoms difficult within the framework of the MNDO and AM1/PM3 techniques. Similar difficulties are encountered in the case of elements with valence *d*-electrons: in the *spd*-basis, there always exist vacant atomic orbitals either of the *p*-type (transition metals) or *d*-type (metametals) with small ε.

Another *weighting factor* technique for modifying C^{AB}_{Coul} within the *weighting-factors NDDO approximation* (NDDO-WF) was suggested in [12] and is realized in the CLUSTER-Z2 software [15]. In this case, expression (5) takes on the modified form:

$$\tilde{C}^{AB}_{Coul}=\sum_{l=|l_1-l_2|}^{l_1+l_2}\sum_{l'=|l_3-l_4|}^{l_3+l_4}F_{ll'}(l_1,l_2,l_3,l_4)*I_{ll'}(l_1,l_2,l_3,l_4), \tag{7}$$

where $I_{ll'}$, as before, are given by (6), while the weighting factors $F_{ll'}$ which reduce the values of integrals assessed from (4) are defined as follows [12]:

$$F_{ll'}(l_1,l_2,l_3,l_4,)=\frac{[C_l(l_1,l_2)*C_{l'}(l_3,l_4)]^2+A}{C_l(l_1,l_2)*C_{l'}(l_3,l_4)+A}. \tag{8}$$

Here C_l are atomic parameters (4 and 10 in the *sp*- and *spd*-basis, respectively).

In terms of the NDDO-WF approximation, expression (8) is used to determine both two-center $\tilde{C}^{AB}_{Coul}$ and one-center $\tilde{C}^{AA}_{Coul}$ integrals. In the former case, the $I_{ll'}$ values are calculated on the Slater basis by using a recurrence procedure suggested in [24]. A detailed description of the method is given elsewhere [19].

3. Schematic architecture of sequential codes

The main characteristics of the CLUSTER-Z1 and CLUSTER-Z2 codes are given in Table 1. The scheme of the CLUSTER-Z1 and CLUSTER-Z2 modules is

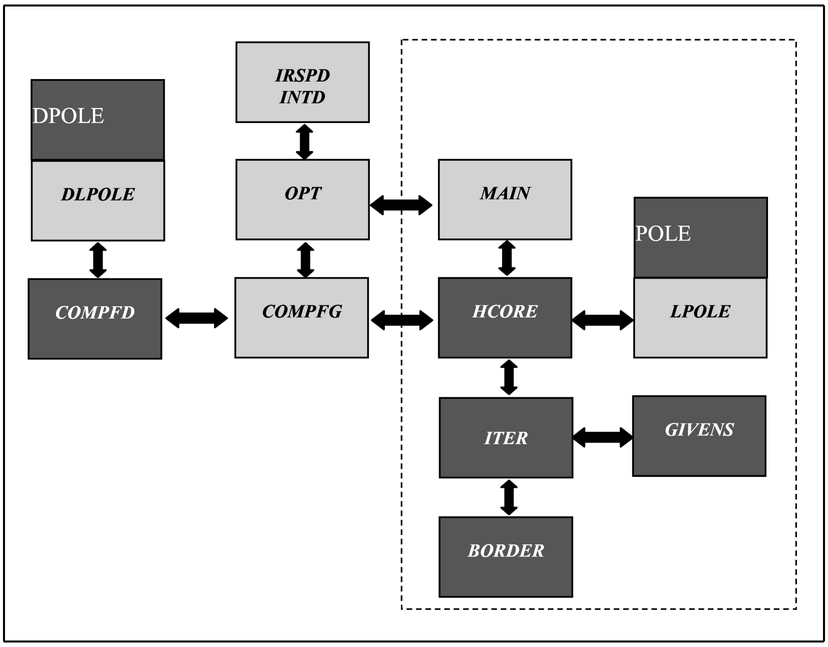


Figure 1. Scheme of the CLUSTER-Z1 and CLUSTER-Z2 sequential codes

presented in Figure 1. The modules functionality is described in details in [19]. Those inside a dashed line ensure 'one point' calculations for some fixed spatial structure of a molecule. A solid line embraces the modules that ensure the optimization of molecular geometry during a search for a minimum of the total energy. As known, SE calculations are performed through cyclization over pairs of indices i and j that enumerate either atoms or atomic orbitals. Thus calculated functions $f(i,j)$ are such that $f(i_1,j_1)$ and $f(i_2,j_2)$ are independent. This circumstance defines a general concept of QC program parallelization: ***arrays of computations (cycles) over index pairs are divided into either continuous or cyclic blocks in accordance with the number of processors.***

Table 1. Characteristics of the CLUSTER-Z1 and -Z2 packages

magnitude	values	
	CLUSTER-Z1 (NDDO mode)	CLUSTER-Z2 (NDDO-WF mode)
number of atoms	960[1]	895[1]
total number of orbitals	3840[1]	3802[1]
number of occupied orbitals	1900[1]	1900[1]

parametrized atoms	MNDO and MNDO/H[2]: H, He[3], Li, Be, B, C, N, O, F, Al, Si, P, S, Cl, Zn, Ge, Br, Sn, J, Hg, Pb;
	AM1[2]: H, He[4], C, N, O, F, Al, Si, P, S, Cl, Se, Br, Sb**, Te, Hg, Ga, Ge, As, Se;
	PM3[2]: H, He[4], C, N, O, F, Al, Si, P, S, Cl*, Zn, Ga*, Ge, As*, Se, Br, Cd*, In, Sn, Sb*, Te*, J, Pb*, Bi, Be**, Mg**, Hg**, Tl**
		H, C, N, O, F, Cl, Br, I, P, S, Li, Na, K, Rb, Cs, Ag, Ti, Al, Ga, As, Sb, W, Fe, Si

atoms under parametrization		Si, In, Ge, Cd, Sn, Te, Pb, Ni, Co, Bi, Zn, Hg, Tl

[1] There is no limit for these values; the indicated ones are optimal for use in PC Intel PIII with RAM 256MB.
[2] List of parameters repeats that of MOPAC. However, a thorough analysis shows that species marked by stars should be refrained by using both MOPAC and CLUSTER-Z1 until new parameters are obtained. One star marks atomic species for which an inequality $0 < H_{PP} < 0.1$ takes place while parameters H_{PP} are set equal to 0.1 in some places of the MOPAC calculation scheme. For atoms marked by two stars parameters H_{PP} are meaninglessly negative.
[3] Parameters are added from [25].
[4] Parameters are included by V. A. Zayets in accordance with [25].

4. Parallel code implementation

4.1. Atom-pair distribution of data

When implementing, the data distribution algorithm was subordinated to meeting the following requirements:

- to guarantee a minimal amount of data to be replicated while a preferable amount of the data should grow linearly when *NORBS* increases;
- to minimize data exchange between different processors;

- to provide labor-saving adaptation of all used algorithms to distributed structure of data.

  Data arrays in sequential codes involve

- arrays for storing two-center integrals $<AA|BB>$ and
- triangular matrixes for storing:
  - Fock matrix $\mathbf{F}$;
  - core Hamiltonian $\mathbf{H_{core}}$;
  - density matrix $\mathbf{P}$.

Array of integrals. Data cover sets of two-center integrals and each set is determined independently from others. Let AOs of atom A be numbered from i_1 to i_2 and those of atom B from j_1 to j_2. Then the integral set with index AB will be denoted as

$$Z_{AB} = \{<AA|BB>|i_1 < A < i_2, j_1 < B < j_2\}. \tag{10}$$

If the number of AOs is N_A and N_B for atom A and B, respectively, the total number of integrals belonging to Z_{AB} set, INT_{AB}, is $INT_{AB} = N_A(N_A+1)/2 \times N_B(N_B+1)/2$. The total number of two-center integrals is $INT = \sum_{AB}^{NPAIRS} INT_{AB}$ where $NPAIRS = NAT*(NAT-1)/2$ determines the total number of atom pairs. Hereinafter all numbers related to the i^{th} processor will be denoted by a superscript i. Thus, the number of atom pairs belonging to the i^{th} process will be $NPAIRS^i$ and $NPAIRS = \sum_{i=1}^{NPROC} NPAIRS^i$, where $NPROC$ determines the number of processors. Correspondingly, $INT = \sum_{i=1}^{NPROC} \sum_{\{AB\}^i=1}^{NPAIRS^i} INT^i_{AB}$. That allows to describe a union of integral sets which joins those distributed over both atom pairs and processors, as follows

$$Z = \bigcup_i \bigcup_{\{AB\}^i} Z_{AB}. \tag{11}$$

Equation (11) presents the *atom-pair distribution of integrals* over processors.

Matrix arrays. According to Equation (3), all elements of the triangular matrixes depend on a pair of indices (i,j), when both are changed from 1 to $NORBS$ with the total number of pairs $NPAIRS = NORBS*(NORBS+1)/2$. Within the NDDO and NDDO-WF approximations matrix elements could be calculated independently. Therefore the distribution of the matrix arrays over processors could follow that of $NPAIRS$. However, the matrixes, involved in a calculation cycle, are not only interrelated but are dependent on two-center integrals as well. This dependence provokes looking for a common concept of the data distribution applied to both integrals and matrixes. Let us denote a box of one of the above-mentioned matrixes which contains elements $\mathrm{M}[i_1:i_2, j_1:j_2]$ as submatrix M_{AB} and a triangular part with elements $\mathrm{M}[i_1:i_2, i_1:i_2]$, $i_1 < i_2$ as submatrix M_A (see Figure 2). Here $i_1:i_2$ and $j_1:j_2$ numerate AOs belonging to atom A and B, respectively. Array of submatrixes M_A for all A can be denoted as M_D, so that $M_D = \bigcup\limits_A M_A$. With this notation any triangular matrix can be divided into parts corresponding to atom pairs (A,B) in a manner

$$M = (\bigcup_A M_A)\bigcup(\bigcup_{AB} M_{AB}) \tag{12}$$

that expresses the *atom-pair distribution of matrixes* similarly to Equation (11). When distributing, it is convenient to replicate the part of matrix lying near diagonal $M_D = \bigcup\limits_A M_A$, that is of the $O(N)$ order, over all processors. Non-diagonal submatrix M_{AB} will be attributed therewith to a processor to which the relevant atom pair (A,B) belongs.

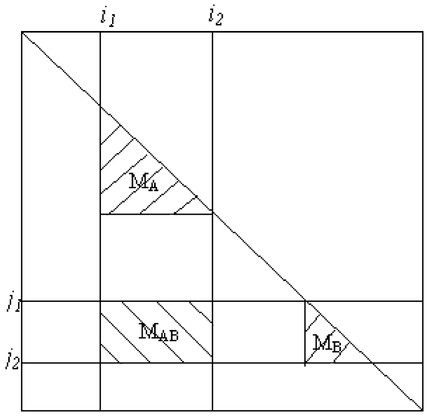


Figure 2. Scheme of a matrix division

The effectiveness of the above distribution with respect to matrixes $\mathbf{F}$, $\mathbf{H_{core}}$, and $\mathbf{P}$ depends on how much localized are the matrix calculations at a given processor. In the case of two latter matrixes, a positive answer is rather obvious. As for the Fock matrix, it is so closely interrelated with matrixes $\mathbf{H_{core}}$, and $\mathbf{P}$ as well as with two-center integrals that additional arguments are needed. Let us consider an element $F_{ij} \in F_{AB}$. As seen from Equation (3) and Figure 2, AOs i and j belong to different atoms A and B, and AOs k and l must belong to the same atoms, since the NDDO approximation neglects three- and four-center integrals. Therefore, only $P_{AB}(k \in A, l \in B \Rightarrow P_{kl} \in P_{AB})$, $P_A(k,l \in A \Rightarrow P_{kl} \in P_A)$, $P_B(k,l \in B \Rightarrow P_{kl} \Rightarrow P_B)$ and the corresponding one- and two-center integrals are involved when calculating F_{AB} providing a full localization of the calculations at one processor.

Another problem concerns F_D due to its replication over all processors. However, it is easy to show that the calculation of partial sums for F_D that involves H_{AB}, P_{AB}, and INT_{AB} located at this processor and the summation of these partial sums by using MPI [26] reducing operation MPI_REDUCE with operand MPI_SUM results in obtaining full F_D at the 0^{th} processor. Further application of MPI_BCAST copies the full F_D to other processors.

4.2. Atom-pairs distribution of integrals

When implementing the distribution practically, the main goal was to balance computational and memory load over processors. This can be achieved by equalization of the number of two-center integrals of similar kind at each processor. To meet the requirement, let us divide atom pairs into groups specified by types of atoms. Thus, in the *sp*-basis there are three groups of pairs, namely, A_sB_s, A_sB_p (A_pB_s), and A_pB_p. Respectively, the set of numbers of AOs per each atom is the following:

- $N_A = N_B = 1$ for A_sB_s pair;
- $N_A = 1$ and $N_B = 4$ for A_sB_p pair as well as $N_A = 4$ and $N_B = 1$ for A_pB_s pair;
- $N_A = N_B = 4$ for A_pB_p pair.

Three other groups are added in the *spd*-basis, namely, A_sB_d (A_dB_s), A_pB_d (A_dB_p), and A_dB_d so that the set of the AOs numbers is enlarged by adding

- $N_A = 1$ and $N_B = 9$ for A_sB_d pair as well as $N_A = 9$ and $N_B = 1$ for A_dB_s pair;
- $N_A = 4$ and $N_B = 9$ for A_pB_d pair as well as $N_A = 9$ and $N_B = 4$ for A_dB_p pair;
- $N_A = N_B = 9$ for A_dB_d pair.

By cyclic distribution of atom pairs of each group over processors one can ensure close to equal amount of pairs of the same kind at each processor that will require practically the same amount of memory and CPU time for data processing. Let L be the number of the relevant atom pairs (A,B) within a group. The corresponding pair set at the i^{th} processor will be determined as:

$$\{AB\}^i = \{(A,B) | L \bmod NPROC = i\}. \tag{13}$$

Suppose a system under study has $NAT = 5$ atoms, therewith $N_1 = N_3 = 1$; $N_2 = 4$, $N_4 = N_5 = 9$. The atom pair distribution over 4 processes, obtained according algorithm (13), is shown in Table 2. The material presented in the table shows more the philosophy of distribution rather than a valuable result. Evidently, the technique will result in practically uniform distribution of atom pairs at large *NPAIRS*. Following the distribution of atom pairs, two-center integrals are distributed over processors in accordance with the algorithm described by Equation (11).

Table 2. Distribution of atom pairs over processors

groups	$\{AB\}^0$	$\{AB\}^1$	$\{AB\}^2$	$\{AB\}^3$
1 × 1	(3,1)			
1 × 4	(2,1)	(3,2)		
1 × 9	(4,1)	(4,3)	(5,1)	(5,3)
4 × 9	(4,2)	(5,2)		
9 × 9	(5,4)			

4.3. Atom-pair distribution of matrixes

The implementation of the atom-pair distribution of matrixes has been subordinated to that one of *NPAIRS*, described in the previous Section. Figure 3 shows the Fock matrix decomposition for the above model system in accordance with the pair distribution given in Table 2. The matrix of the 24th order ($NORBS = 24$) is divided into 10 blocks of non-diagonal elements ($NPAIRS = 10$) of different size due to different numbers of AOs of the atoms. The blocks are distributed over processors similarly to two-center integrals according to the scheme of Table 2. Additionally, the matrix at each processor contains replicated array F_D. Matrixes $\mathbf{H_{core}}$ and $\mathbf{P}$ are treated

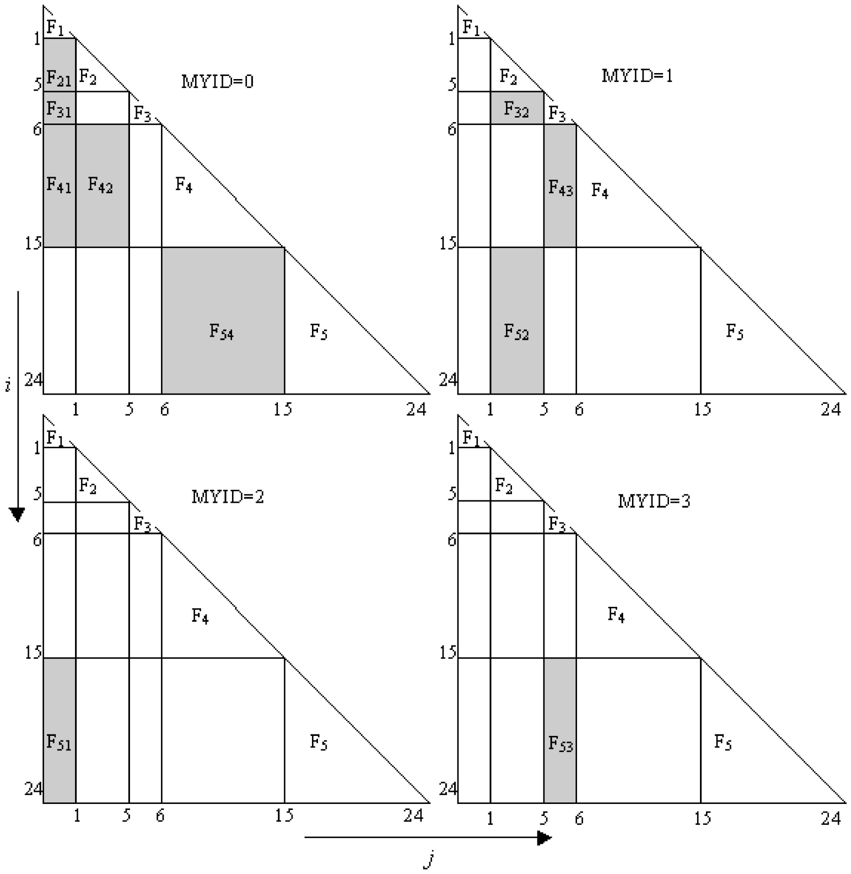


Figure 3. Atom-pair distributed matrix for a model system of 5 atoms. MYID indexes processors

analogously. Similarly to integral distribution, matrix elements are distributed over processors at large *NORBS* quite uniformly.

4.4. Adaptation of the main algorithms to the atom-pair data distribution

In the sequential codes, two-center integrals are calculated using cycle over atom pairs:

- for all $i \in 2, NAT$ (A is the i^{th} atom),
- for all $j \in 1, i-1$ (B is the j^{th} atom).

When going to parallel codes, this algorithm continues to work, however, with not all pairs but with that part of them which is located at the processor under consideration.

As for matrixes, the algorithms cover cycles over atomic orbitals:

- for all $i \in NORBS$,
- for all $j \in 1, i$.

To adapt the algorithms to the atom-pair distribution, the above cycles are substituted by nested ones, when the upper cycles cover atom pairs while the inner ones operate over atomic orbitals belonging to these pairs:

- for all $j \in 1, NPAIRS^i$ ($(A,B$ is the j^{th} atom pair on the i^{th} processor)
- for all $k \in 1, N_A$
- for all $l \in 1, N_B$.

When adapting to parallel codes, the sequential ones must be rewritten one by one that, among others, provides a perfect ground for debugging. Finally, when all data are partitioned over processors practically uniformly, the computational load is automatically parallelized and balanced.

4.5. Conversion of atom-pair distributed matrixes to two-dimensional block-cyclic ones

The eigenproblem solution (diagonalization step) and the performance of matrix multiplication (calculation of density matrix) in parallel code were performed using ScaLAPACK library [27]. As known, the ScaLAPACK package requires matrixes which are block-cyclically distributed in the following way. Suppose, we have a set of objects with a global index i_g. One-dimensional block-cyclic distribution defines mapping elements from this set to a processor array $i_g \mapsto < p, b, i_l >$, where p and b numerate processors and blocks at the processors, respectively, while i_l is a local index inside block b. If P numerates processors and m denotes the size of blocks, one dimensional block-cyclic data distribution can be given as follows:

$$i_g \mapsto < s \bmod P, \left\lfloor \frac{s}{P} \right\rfloor, i_g \bmod m >, \tag{14}$$

where $s = \left\lfloor \frac{i_g}{m} \right\rfloor$ is a quotient from integer division. Let a set of processors be viewed as logical two-dimensional grid forming $P \times Q$ matrix, with $m \times n$ blocks deposited on processors. Then two-dimensional block-cyclic distribution can be presented as a result of a combination of two one-dimensional block-cyclic distributions by rows and by columns. In this case the distribution of matrix element with global indexes (i_g, j_g) will be defined as:

$$(i_g, j_g) \mapsto < (p,q), (b_p, b_q), (i_l, j_l) > . \tag{15}$$

Finally, the matrix is divided into *generalized* blocks of $(m \cdot P) \times (n \cdot Q)$ in size, which, in their turn, are divided into $(P \cdot Q)$ the same blocks, each lying on one processor from the processor grid. Figure 4 shows the result of block-cyclic distribution of the Fock matrix of a model system with 5 atoms described in the preceding Sections. The 24×24 matrix is block-cyclically distributed using 2×2 processor grid ($P = 2$, $Q = 2$, $NPROC = 4$) with 4×4 blocks ($m = 4$, $n = 4$) at each processor from the grid. The colored parts of matrixes in Figure 4 represent elements of $\mathbf{F}$ received by a given processor. The empty rows and columns are not stored in the local processor memory but the remaining triangular matrix $\mathbf{F}$ is filled up to squared by empty elements. The correspondence between matrix elements distributed over different processors by using either atom-pair or block-cycling modes is well seen when comparing the pictures shown in Figures 3 and 4. In Figure 4, the 1×1 blocks are marked by figures which correspond to MYID in Figure 3. Elements related to the F_D array are shown by dotted fields.

When converting the atom-distributed matrixes into block-cycling ones, the former are cyclically pumped over processors to avoid collecting the whole matrix at one processing node. The data exchange steps are performed in the following way: the

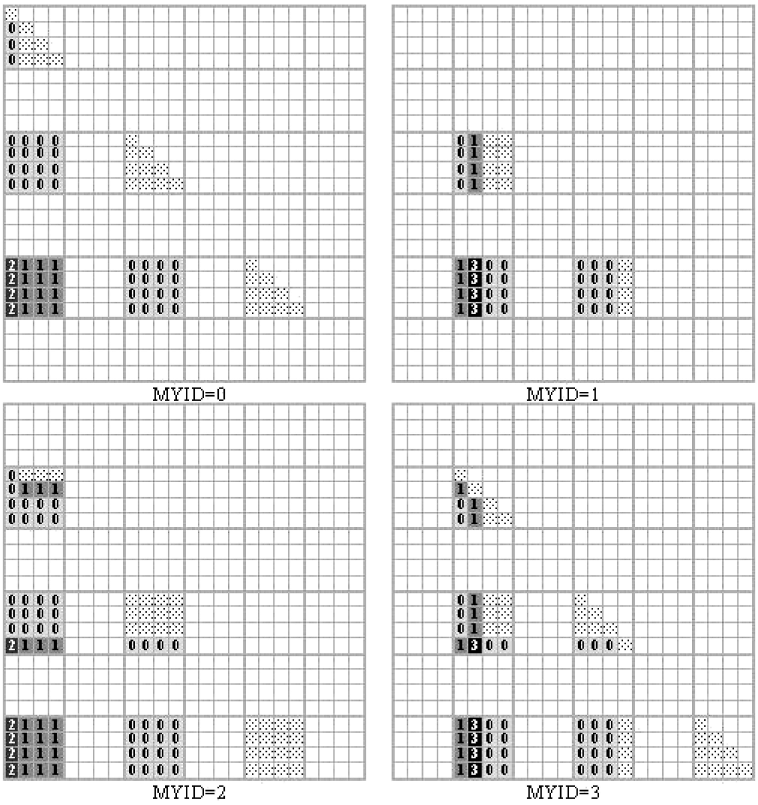


Figure 4. Conversion of an atom-distributed matrix into a block-cycling one

first run is carried out over all matrix elements which are presented at every processor initially, and the elements needed for block-cyclic distribution on this processor are copied, then processor K transmits its data to processor $K+1$ until the last processor transmits them to the 0^{th} one. Checking and copying the relevant elements is performed at each transmission step. The block-cyclically distributed matrix, which can be used when calling ScaLAPACK procedures, is obtained after repeating this procedure $NPROC-1$ times. To avoid deadlocking, every transmission step is divided into two substeps: firstly, even processors send data and odd ones receive them, then odd processors send and even receive. When implementing, blocking MPI calls (MPI_SEND, MPI_RECIEVE) are used. Reverse conversion which accumulates the calculated results in the atom-pair distribution form, is performed by using similar algorithm.

4.6. Parallel implementation of the main sequential algorithms

Practical implementation of computational codes parallelization depends on inner architecture of the relevant sequential program. Supplementing a general scheme of the codes in Figure 1, Table 3 show results of time profiling of the main algorithms of the CLUSTER-Z1 and CLUSTER-Z2 packages. The difference in the percentage distribution reflects the difference in mathematical algorithms of the NDDO and NDDO-WF approximations. Irrespective to the latter, the diagonalization of the Fock matrix is the most time-costly for both codes. Then follows the density matrix calculation which is supplemented by the core Hamiltonian and the total energy derivatives calculations in the CLUSTER-Z2 code. The diagonalization and density matrix calculation algorithms are of the $O(N^3)$ order of complexity, where $N \equiv NORBS$. The remaining parts of the programs are of the $O(N^2)$ complexity order.

Table 3. Time profiling of the CLUSTER-Z1 and CLUSTER-Z2 main algorithms: C300 molecule (MBC-1000M[1])

algorithms	%%	
	CLUSTER-Z1	CLUSTER-Z2
diagonalization	69.89	45.87
density matrix	22.08	18.15
core Hamiltonian[2]	1.29	13.71
Fock matrix	3.81	2.71
derivatives	2.93	19.56
total time	100.00	100.00

[1] Description of the test molecule and supercomputer see in Section 5.

[2] Calculations of two-centre integrals are included.

Diagonalization and calculation of the density matrix. Application of the PDSYEVX procedure from the ScaLAPACK library provides the solution of the eigenproblem $\mathbf{FC} = \mathbf{EC}$ and finding the eigenvalues E and eigenvectors C. Staying within the block-cyclically distribution mode, the density matrix $P_n = CC^T$ is calculated by using the eigenvector matrix and calling ScaLAPACK PDSYRK procedure. Thus obtained matrix P_n is converted into the atom-pair distributed and P_{n+1} matrix of the next step of SCF calculations is determined either by extrapolation or (at poor convergence) by using the level-shift technique [28, 29] to proceed SCF calculations.

The efficiency of the PDSYEVX procedure is of a key importance. Table 4 shows the procedure productivity for matrixes of different size at a supercomputer with distributed memory. A comparison of the data with those obtained for a supercomputer with shared memory [19] shows that in both cases the procedure is good enough for matrixes of the 2000th order and more. However, even in this case the speedup is sublinear. It is important to note that speedup for the same order matrixes is noticeably different when the procedure is performed on different computers.

Calculation of derivatives. Calculations of the total energy derivatives is a rather time-costly procedure of the sequential codes, particularly, of CLUSTER-Z2. The calculations are performed in the Cartesian coordinates that provide using analytical expressions which transform derivatives of energy to those of more simple constructions, such as $\frac{\partial}{\partial x}\langle AA|BB\rangle$, $\frac{\partial}{\partial x}\langle A|B\rangle$, $\frac{\partial}{\partial x}\langle A|\frac{1}{R_B}|A\rangle$, and $\frac{\partial}{\partial x}\langle B|\frac{1}{R_A}|B\rangle$ [19]. The algorithm used when calculating diagonal part F_D of the Fock matrix (see Section 4.2), occurred to be applicable to the case as well. Consequently, partial sums are calculated on each processor, and after that the obtained data are summed up by using MPI_REDUCE procedure and are broadcasted to all processors. All other quantities depending on distributed data, the core Hamiltonian among them, are calculated in the same manner. Final data before outputting are collected on the 0th processor.

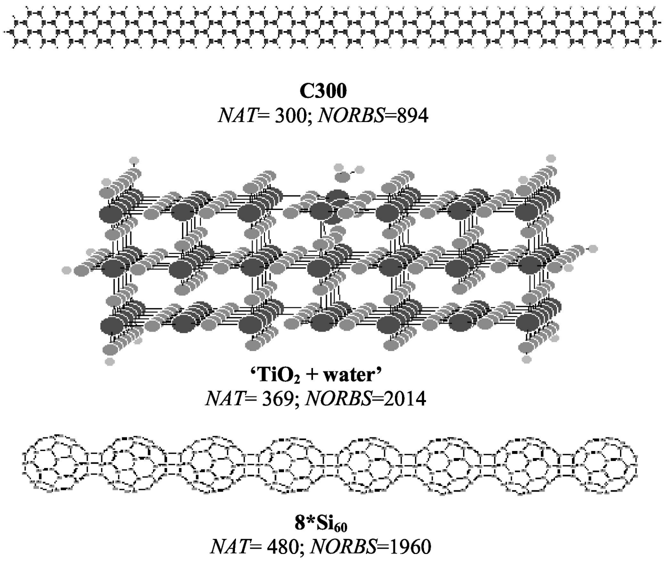
Table 4. Execution time of the PDSYEVX diagonalization (MBC-1000M)

size of matrix	*NPROC*			
	1	2	4	8
2600	126.8	95.2	48.8	31
2400	100.9	73.6	38.7	25.3
2200	79.3	58.8	31.1	21.0
2000	60.9	44.8	24.2	16.5
1800	45.9	34.1	18.5	13.1
1600	33.0	24.9	13.6	10.2
1400	23.2	18	10.1	7.6
1200	15.3	12	6.8	5.5
1000	9.4	7.7	4.5	3.8
800	5.3	4.7	2.8	2.5
600	2.5	2.4	1.5	1.5
400	1	1	0.6	0.7
200	0.3	0.3	0.3	0.5

5. Speedup and efficiency of NANOPACK

NANOPACK codes testing has been performed for a few model systems (see Figure 5) on a set of multi-processor computers. The main characteristics of computers are listed in Table 5. We will start with a carbon-graphite strip C300 with $NORBS = 894$ treated in the *sp*-basis. Table 6 shows time profiling of the NANOPACK main algorithms at one-processor performing mode. Comparing the data with those listed in Table 3, one can see the effect of the atom-pair distribution of data on percentage distribution over algorithms. As previously, diagonalization remains the most time-costly procedure, whereas the contribution of the density matrix calculations is remarkably lowered. The fact is explained by the substitution of nested cycles in the sequential codes by ScaLAPACK function call in parallel ones. A row related to matrix conversion is added additionally to Table 3. It lists time expenses required for direct and reverse conversion which connects atom-pair and block-cyclically distributed matrixes (see Section 4.4). The expenses slowly increase when *NPROC* grows.

Figure 6 presents a comparative study of the speedup of both the main algorithms of NANOPACK and the package as a whole. A 5-SCF iteration testing has been performed for the $8*\mathrm{Si}_{60}$ oligomer in the NDDO mode while 28 SCF iterations have been calculated for the '$\mathrm{TiO}_2+\mathrm{water}$' model system in the NDDO-WF mode of the package. A detailed analysis allows to highlight a similarity and difference between the two modes. The fact that the diagonalization parallel implementation is a limiting factor in both cases is well obvious. However, while for the NDDO mode it dominates absolutely, so that the speedup of the package as a whole ($S_p = 3.02$) attained for 8 processors is even less than that of the diagonalization ($S_p = 3.42$), for the NDDO-WF the total speedup ($S_p = 4.91$) prevails over diagonalization ($S_p = 4.34$). To better disclose the reasons of the observed behavior, Figure 7 shows the time profiling for both modes of the NANOPACK package when *NPROC* increases from one to eight. The figure clearly exhibits that the diagonalization procedure dominates in both cases. However, a significant growth of the matrix conversion contribution depresses the total speedup of the NDDO mode below that of diagonalization when



C300
NAT= 300; *NORBS*=894

'TiO_2 + water'
NAT= 369; *NORBS*=2014

8*Si_{60}
NAT= 480; *NORBS*=1960

Figure 5. Testing molecular systems

Table 5. Supercomputer characteristics

	RM-600[1]	Cluster PIII Intel[2]	MBC-1000M[3]	Cray-T3E
CPU type	MIPS R10000	PIII	Alpha 21264	Alpha 21164
CPU frequency, MHz	200	667	667	300
number of CPUs	16	16	128	128
point-to-point link and width, Mbit/s	3200	100	1280	4000
total memory, Gb	4	4	64	16
memory per CPU, MB	256	256	512	128
memory access time, ns	100	8	10	10
architecture	SMP	MPP, switched media	MPP, switched media	MPP, 3-D torus

[1] Siemens Nixdorf RM-600 E20/E60. http://www.citforum.ru/hardware/articles/rm600e00.shtml.
[2] Domestic manufacture of the Institute of Problems of Chemical Physics, RAS (Chernogolovka, Russia), 2000.
[3] Domestic manufacture of the Joint SuperComputer Center (Moscow, Russia), 1999.
http://www.jscc.ru/cgi-bin/show.cgi?/about.html%203#hardres.

NPROC increases. In the case of the NDDO-WF mode, the effect of the conversion contribution is compensated by rather high speedup of the core Hamiltonian and derivatives algorithms, which provides the prevailing of the total speedup over that of diagonalization.

Table 7 lists total time of testing calculations performed on different computers. The data form a good basis for a comparative study of the computer configurations on the efficiency of parallel computations. Analyzing the data presented, the following conclusions can be made.

1. MBC-1000M machine is about twice more effective for the calculations than Cluster PIII Intel with processors of the same tact frequency, and the difference increases when the number of processors involved grows.

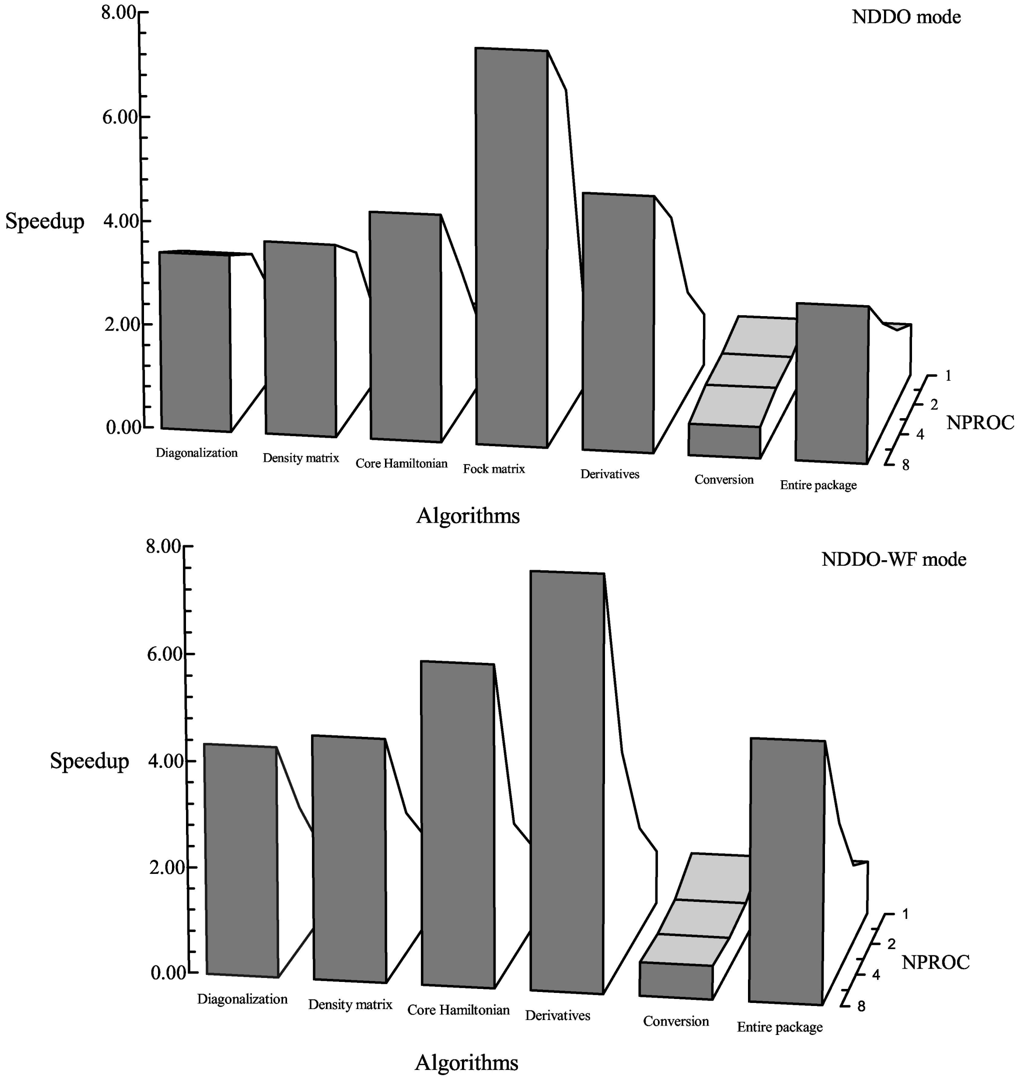


Figure 6. Speedup of the NANOPACK main algorithms

Table 6. Time profiling of the NANOPACK main algorithms for one-processor operation: C300 molecule (MBC-1000M)

algorithms	%%	
	NDDO mode	NDDO-WF mode
diagonalization	76.22	46.54
density matrix	5.84	3.83
core Hamiltonian	0.97	13.75
Fock matrix	5.74	9.04
derivatives	3.25	20.54
conversion	3.66	1.43
others	4.32	4.87
total time (entire package)	100.00	100.00

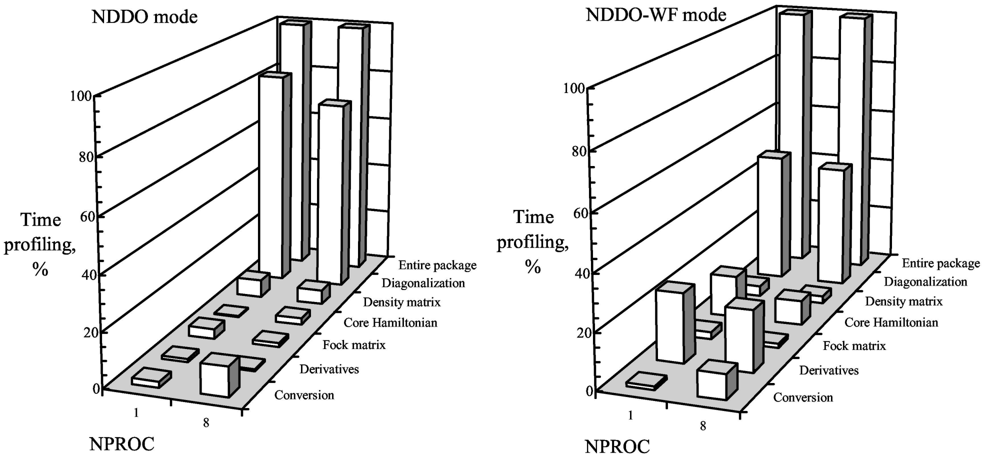


Figure 7. Time profiling of the NANOPACK main algorithms on processors array of different *NPROC*

Table 7. Calculation total time in s

computer	*NPROC*			
	1	2	4	8
NDDO mode, $8{*}\mathrm{Si}_{60}$, *NORBS*=1920, 5 SCF iterations				
Cluster PIII Intel	6270	4559	4559	3982
MBC-1000M	3649	2528	1710	1209
Cray-T3E			3733	2356
NDDO-WF mode, ‘TiO_2+water’, *NORBS*=2014, 28 SCF iterations				
Cluster PIII Intel	125069	68712	52772	40105
MBC-1000M	51763	34654	18153	10532
Cray-T3E				34495

Table 8. Calculation scalability of NANOPACK (Cray T3E), time in s

algorithms	*NPROC*				
	4	8	16	32	64
NDDO mode, $8{*}\mathrm{Si}_{60}$, *NORBS*=1920, 5 SCF iterations					
diagonalization	2826	1699	983	705	576
core Hamiltonian	15	7	3	2	1
Fock matrix	91	45	24	11	6
derivatives	76	38	19	9	5
conversion	257	236	246	237	239
total time (entire package)	3733	2356	1536	1198	1043
NDDO-WF mode, ‘TiO_2+water’ system, *NORBS*=2014, 28 SCF iterations					
diagonalization		15121	8915	6235	4969
core Hamiltonian		1378	694	355	190
Fock matrix		8845	4423	2215	1112
derivatives		5409	2705	1355	680
conversion		2197	2231	2168	2237
total time (entire package)		34495	19839	12852	9557

2. When comparing MBC-1000M and CRAY-T3E, more than two times lowering of productivity of the latter due to two-times difference in the tact frequency takes place. The difference becomes bigger when the number of processors increases.
3. CPU RAM of 256MB provides the performance of both testing problems with *NORBS* ~ 2000 in a one-processor mode that always operates with a total bulk of data. However, CPU RAM of 128MB was not enough for the task and 4-processor mode was needed to run the problem of the $8*\mathrm{Si}_{60}$ molecule in the *sp*-basis at CRAY-T3E, while already 8-CPU mode of the computer was required to perform calculations for the 'TiO_2+water' system under condition of over-processor-distributed data.

The last test on the scalability of both NANOPACK package as a whole and its algorithms has been performed in the same comparative manner as above, but on one computer. Table 8 lists the data of calculations for both package modes. As seen from the table, speedup is close to saturation when *NPROC* exceeds 16. Non-systematic testing on the MBC-1000M machine proves this tendency as well. Therefore, a processor arrangement in a (8-16)-CPU set seems to be optimal for massive parallel computations.

6. Concluding remarks

Appearance of powerful and inexpensive parallel computers with massive parallelism opens up access to highly productive calculations for a wide range of users. Meeting the request on QC software which can provide the computational consideration of nano-sized many-atomic systems, a practically useful NANOPACK package based on semiempirical NDDO and modified NDDO-WF approximations has been suggested. Detailed testing of the tool has highlighted the package's strong and weak sides. As for obvious merits of the tool, large size of the systems considered opens the list. Alongside with a large set of parametrized chemical species, this provides an extended field of possible applications. A possible choice between *sp*-QC algorithms of the conventional NDDO approximation (mode) and *spd*-QC algorithms involved in the NDDO-WF mode further enlarges the package facilities. The described package implements calculations on the basis of the atom-pair distribution scheme to use supercomputers with distributed memory. A rather small cluster of efficient CPUs, equipped with NANOPACK and similar packages, can provide a large front for practical application of computational chemistry.

Sublinear scaling of the calculation and, as a consequence, moderate and even low speedup should be attributed to weak points. The fact is not characteristic to the NANOPACK package itself but should be addressed to all QC programs where algorithms of the eigenproblem solution either dominate or are rather valuable. The problem is more mathematical than conceptual. The only available ScaLAPACK library offering the corresponding tool for the parallelization implementation of the problem greatly restricts the procedure. New approaches to be elaborated will undoubtedly speed the calculations.

Characterizing the efficiency of the package by and large, a brief report on its applications should be added. For less than half a year of the package in operation, three bulks of computational chemistry applications have been performed at the MBC-1000M machine:

- a full cycle of *sp*-QC calculations in the NDDO mode of a set of 13 systems related to silicon fullerene Si_{60} and its oligomers (up to $8*Si_{60}$) with *NORBS* from 720 to 1920; for 7 oligomers in both singlet and triplet spin states [20];
- a full cycle of *spd*-QC calculations in the NDDO-WF mode of 24 systems related to the adsorption of water and carboxylic acids on the rutile and anatase nanoparticles with *NORBS* of 2000–2200 [21];
- a full cycle of *spd*-QC calculations in the NDDO-WF mode of 38 systems related to Ag^+-dipyridil complex oligomers with *NORBS* from 200 to 2200 [30].

The package behaves steadily and is quite easy to operate.

Another NANOVIBR version has been elaborated by the authors that is based on the replication data scheme [31]. The tool is well efficient with not too large systems ($NAT \leq 200$) but a big bulk of calculations (say, harmonic vibrational problems with $NMODES \leq 600$) has to be solved. Data replication, in this case, promotes the facilitation of both IR and Raman spectra intensities calculations to a great extent.

Acknowledgements

This work was supported by the Russian Foundation for Basic Research (project no. 01-07-900-56). The authors greatly acknowledge the Joint SuperComputer Center for making available computational resources on MBC-1000M. One of the authors (IYaG) is deeply thankful to Prof. S. deLeeuw for providing the testing of the NANOPACK package on CRAY-T3E of the Delft Technical University.

References

[1] Dewar M J S 1969 *The Molecular Orbital Theory of Organic Chemistry* McGraw-Hill, New York

[2] Pople J A and Beveridge D L 1970 *Approximate Molecular Orbital Theory* McGraw-Hill, New York

[3] Dewar M J S and Thiel W 1977 *J. Am. Chem. Soc.* **99** 4899

[4] Burstein K Y and Isaev A N 1984 *Theor. Chim. Acta* **64** 397

[5] Goldblum A J 1987 *Comp. Chem.* **8** 835

[6] Thiel W 1981 *J. Am. Chem. Soc.* **103** 1413

[7] Dewar M J S, Zoebisch E G, Healey E F and Stewart J J P 1985 *J. Am. Chem. Soc.* **107** 3902

[8] Stewart J J P 1989 *J. Comp. Chem.* **10** 209

[9] Stewart J J P 1990 *J. Comp. Aided Mol. Design* **4** 1; 1983 *MOPAC: A General Molecular Orbital Package* QCPE program 455; 1983 *QCPE Bull.* **3** 43; 1995 *Program MOPAC93* Release 2

[10] Dewar Research Group and Stewart J J P *AMPAC: Austin Method I Package* QCPE program 506; 1986 *QCPE Bull.* **6** 24

[11] Zayets V A 1990 *CLUSTER-Z1 Quantum-Chemical Software (sp-Basis)*, Institute of Surface Chemistry, Kiev

[12] Grebenjuk A G, Zayets V A and Gorlov Yu I 1993 *React. Kinet. Catal. Lett.* **50** 257

[13] Zayets V A, Gerda V I, Gorlov Yu I and Klimenko V E 1998 *Zh. Strukt. Khim.* **39** 163

[14] Thiel W and Vojtyuk A A 1996 *J. Phys. Chem.* **100** 616

[15] Zayets V A 1990 *CLUSTER-Z2 Quantum-Chemical Software (spd-Basis)*, Institute of Surface Chemistry, Kiev

[16] Thiel W 1994 *Program MNDO94, Version 4.1*, Zurich

[17] Berzigiyarov P K 2000 *Proc. II All-Russia School on Supercomputing Systems and Information Technologies for Physics and Chemistry*, 1–2 November, Chernogolovka, p. 85

[18] Berzigiyarov P K, Zayets V A, Gunzburg I Ya, Razumov V F and Sheka E F 2000 *Program package MP-AM1 for parallel quantum-chemical computing in the sp-basis*, Institute of Problems of Chemical Physics, Chernogolovka

[19] Berzigiyarov P K, Zayets V A, Razumov V F and Sheka E F 2001 *TASK Quarterly* **5** (3) 271

[20] Sheka E F and Nikitina E A 2001 *Bulletin Russ. Ac. Sci.* **378** 1

[21] Sheka E, Nikitina E, Zayatz V, Schoonman J and Goossens A 2000 *Third Joint ESP-NSF Symposium "Nanoparticles: Applications in Material Science and Enviromental Science and Engineering"*, 6 September, Dublin, Ireland, p. 70

[22] Pople J A and Nesbet R K 1954 *J. Chem. Phys.* **22** 571

[23] Sielver D M 1971 *J. Math. Phys.* **12** 1937

[24] Harris F E 1969 *J. Chem. Phys.* **51** 4770

[25] Kolb M and Thiel W 1993 *J. Compt. Chem.* **14** 37

[26] Snir M, Otto S W, Huss-Lederman S, Walker D and Dongarra J 1996 *MPI: The Complete Reference*, MIT Press, Boston

[27] Blackford L S, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, Dongarra J, Hammarling S, Henry G, Petitet A, Stanley K, Walker D and Whaley R C 1997 *ScaLAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia

[28] Saunders V R and Hillier I H 1973 *Int. J. Quant. Chem.* **4** 699

[29] Maslov V G 1979 *Zh. Strukt. Khim.* **20** 761

[30] Nikitina E A, Zayets V A and Razumov V F (private communication)

[31] Berzigiyarov P K, Zayets V A, Ginzburg I Ya, Razumov V F and Sheka E F 2001 *NANOVIBR: parallel codes for semiempirical quantum-chemical and harmonic-vibration calculations of large systems*, Institute of Problems of Chemical Physics, Chernogolovka