TASK QUARTERLY 5 No 3 (2001), 271–300

PROGRAM PACKAGE MP-ZAVA FOR PARALLEL QUANTUM-CHEMICAL COMPUTING IN THE *SPD*-BASIS

PARVAZ K. BERZIGIYAROV[1], VALENTINE A. ZAYETS[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*
parvaz@pro.icp.ac.ru

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

(Received 29 January 2001)

Abstract: A parallel realization of the NDDO-WF technique for semi-empirical quantum-chemical calculations on large molecular systems in the *spd*-basis is described. The technological aspects of designing scalable parallel calculations on super computers (by using MPI library) are discussed. The scaling of individual algorithms and entire package was carried out for two model systems with a number of atomic orbitals of 894 and 2014, respectively. The speedup was determined in computer experiments with the RM600 E60 and Cluster Intel PIII multi-processor systems. The effect of communication rate on the package performance is discussed.

Keywords: quantum chemistry, parallel codes, semi-empirical approach, *spd*-basis

1. Introduction

Computational chemistry whose fundamentals were formulated back in the 80s [1, 2] is becoming more and more important in the view of fast-developing attack of nanotechnology on all spheres of human activity. Being quantum in its nature, nanotechnology is dealing with phenomena that take place at the atomic level and requires adequate description of these processes, their prediction, and controlling. Modern *quantum chemistry* (QC) offers a wide range of tools and techniques that suit the needs of nanotechnology, which makes it a constituent of nanotechnology processes. All this puts forward strict requirements to the structure of computation procedure. Of course, QC cannot ensure a calculation component over the entire nanotecnology cycle. It should be kept in mind that QC is only applicable to objects whose size does not exceed several nanometers. But the behavior of the atomic world within these spatial limits is a governing factor for both the microscopic and macroscopic objects whose adequate description can be only made by using the entire hierarchy

of calculation tools: from QC through classical (or semi-classical) molecular dynamics, mesoscopic dynamics to analysis of finite elements and engineering design [3].

As is known, a volume of several *nm* in size contains several thousand atoms. Besides a system size, an important feature here is an increasingly growing number of systems that has to be subjected to QC calculations when solving a given nanotechnological problems. All this requires a new approach to the basic concept of computing. The most promising for computing many-hundred atomic systems seem to be the *semi-empirical* (SE) techniques adopted in QC [4, 5]. Meanwhile, a cardinal solution to the problem can be achieved by using the multi-processor parallel codes. As an example of drastic changes in computing, we can mention one of the first successful applications of parallel codes to calculating a reconstructed Si (111) (7×7) surface [6]. These calculations (for a cluster of 700 atoms) were performed by using Thinking Machine CM-2 (16384 one-bit processors). The use was made of a combination of *ab initio* QC techniques [7], such as *ab initio* molecular dynamics, the Perdew–Zunger approximation for local electron density, and the Kleinman–Bylander pseudopotential. In this case, the total execution time was about 0.5 year.

By their concept and accuracy, most close to *ab initio* techniques are those SE QC methods based on the NDDO (*neglect of diatomic differential overlap*) approximation [5]: MNDO [8], MNDO/H (with account of hydrogen bonding) [9, 10], MNDOC (account of electron correlation) [11], AM1 [12], PM3 [13], and SAMI [14, 15]. These techniques and appropriate programs are being used in widespread one-processor packages MOPAC [16], AMPAC [17], and CLUSTER-Z1 [18] for QC calculations (for many-atomic systems in the *sp*-basis). On going to the *spd*-basis, the above techniques were modified into the NDDO-WP [19, 20] and MNDO/d [21] techniques that were used in program packages CLUSTER-Z2 [22] and MNDO94 [23], respectively.

The effectiveness of SE sequential software, and of the package CLUSTER-Z1, in particular, can be illustrated by a solution to the problem of strengthening silicon polymers upon addition (to polymer) of nanosized SiO_2 particles [24–29]. Essentially, the problem was reduced to characterization of intermolecular interaction between polymer and SiO_2 at the interface and its impact on the mechanical properties of the polymer. In two years, above 300 model systems (with the number of atoms ranging between 280 and 380) have been computed by this technique. The calculations (with full optimization of equilibrium structures and sometimes with solution of direct and inverse spectral vibrational problems) were carried out by using a two-processor Pentium-PRO personal computer.

In this work, we report on a transformation of the sequential program package CLUSTER-Z2 to parallel program MP-ZAVA (using the MPI library [30]). The latter was tested on two 16-processor computing systems — (1) SPM-system RM600 E60 Siemens Nixdorf and (2) a cluster of 16 Intel Pentium III 667 MHz processors integrated by means of 3COM switch by the 100 Mbit Fast Ethernet technology (hereinafter Cluster Intel PIII) for two many-atomic clusters. The package was found to exhibit a good scalability of algorithms and a marked speedup of execution.

2. Theoretical background of CLUSTER-Z2

MP-ZAVA is based on the MPI library and on the inner architecture of the CLUSTER-Z2 package. The latter is oriented on solution of a QC problem (*e.g.*, see [5]) for a polyatomic (and many-electron) system in the form of a molecule or model cluster. This

section describes the basics of the calculations and interprets CLUSTER-Z2 results rather than introducing QC.

Let a system under consideration contain M positively charged nuclei with the charge Z_A and N negatively charged electrons with a charge of -1 (a.u.). The position of nuclei is characterized by the vector $\mathbf{R}$ with $3M$ Cartesian coordinates. The position of electrons is given by the vector $\mathbf{r}$ with $3N$ Cartesian coordinates. The energy and wave function for the ground state of a given set of nuclei and electrons is determined from the non-relativistic Schrödinger equation:

$$H\Psi = E\Psi \tag{1}$$

The Hamiltonian for the system may be written in the form:

$$H(\mathbf{R},\mathbf{r}) = T_n(\mathbf{R}) + T_e(\mathbf{r}) + V_{ee}(\mathbf{R},\mathbf{r}) + V_{nn}(\mathbf{R},\mathbf{R}), \tag{2}$$

where $T_n(\mathbf{R})$ and $T_e(\mathbf{r})$ stand for the kinetic energy of nuclei and electrons, $V_{ee}(\mathbf{R},\mathbf{r}) = V_{ee}(\mathbf{r},\mathbf{r}) + V_{ne}(\mathbf{R},\mathbf{r})$ gives the potential energy of interaction between electrons (ee) and between electrons and nuclei (en), while $V_{nn}(\mathbf{R},\mathbf{R}) = E_{nuc}(\mathbf{R})$ gives the energy of electrostatic repulsion between nuclei. In terms of the widely adopted Born–Oppenheimer approximation, Hamiltonian (2) can be subdivided into two parts that describe the motion of light electrons and heavy nuclei:

$$H_{elec}(\mathbf{r}) = T_e(\mathbf{r}) + V_{ee}(\mathbf{r},\mathbf{r}) + V_{ne}(\mathbf{R},\mathbf{r}), \tag{3a}$$

$$H_{nuc}(\mathbf{R}) = T_n(\mathbf{R}) + E_{elec}(\mathbf{R}) + E_{nuc}(\mathbf{R}) = T_n(\mathbf{R}) + V_{PES}(\mathbf{R}), \tag{3b}$$

where

$$V_{PES}(\mathbf{R}) = E_{tot}(\mathbf{R}) = E_{elec}(\mathbf{R}) + E_{nuc}(\mathbf{R}) \tag{4}$$

is *the potential energy surface* that describes the interaction between atoms in a molecular system. Normally it is termed the 'one-point' total energy of the system, which implies a selected and fixed spatial disposition of atomic nuclei.

Accordingly, the Schrödinger equations for the electronic and nuclear subsystems have the form:

$$H_{elec}(\mathbf{r})\Psi_{elec}(\mathbf{r},\mathbf{R}) = E_{elec}(\mathbf{R})\Psi_{elec}(\mathbf{r},\mathbf{R}), \tag{5a}$$

$$H_{nuc}(\mathbf{R})\Phi_{nuc}(\mathbf{R}) = E_{nuc}\Phi_{nuc}(\mathbf{R}). \tag{5b}$$

where $\Psi_{elec}(\mathbf{r},\mathbf{R})$ and $\Phi_{nuc}(\mathbf{R})$ are the wave functions for electrons and nuclei, respectively.

The package CLUSTER-Z2 solves both equations. However, while Equation (5a) is solved for the system of electrons as it is, operator $H_{nuc}(\mathbf{R})$ in Equation (5b) is replaced by the operator of the classical vibrational problem:

$$H_{nuc}(\mathbf{R}) = T_{nuc}(\mathbf{R}) + V(\mathbf{R}). \tag{6}$$

Here $T_{nuc}(\mathbf{R})$ stands for the classical kinetic energy of nuclei, while $V(\mathbf{R}) = V_{PES}(\mathbf{R})$ is determined, according to Equation (4), as a sum of the electronic energy $E_{elec}(\mathbf{R})$ as assessed from Equation (5a) and the energy of Coulomb core-core interaction between nuclei $E_{nuc}(\mathbf{R})$ that is to be determined additionally. The vibration problem is solved in the harmonic approximation.

Generally, many-electron quantum problem (5a) cannot be solved explicitly. In order to overcome this difficulty, the so-called one-electron approximation is being used. In this

case, one-electron Hamiltonian $H_{elec}(\mathbf{r})$ is represented as a sum of one-electron operators $H_i^{eff}(\mathbf{r})$:

$$H_{elec}(\mathbf{r}) = \sum_i H_i^{eff}(\mathbf{r}), \quad i \in 1, N \tag{7}$$

Accordingly, the electronic wave function is represented as the antisymmetric product of the one-electron functions expressed in terms of the Slater determinant:

$$\Psi(1,2,\ldots,N) = \frac{1}{\sqrt{N!}} \det|\varphi_1(1), \varphi_2(2), \ldots, \varphi_N(N)|, \tag{8}$$

where φ_i is a solution to the one-electron equation

$$H_i^{eff} \varphi_i = \varepsilon_i \varphi_i, \quad i \in 1, N, \tag{9}$$

and is also known as *the molecular spin-orbital* (MO). The latter is defined by the spatial and spin coordinates of the i-th electron. The eigenvalue of Equation (9) ε_i gives the energy of a given orbital, while the MOs are orthonormalized:

$$\int \varphi_i(\mathbf{r}) \varphi_j(\mathbf{r}) d\mathbf{r} = \delta_{ij}. \tag{10}$$

Therefore, Equation (5a) is replaced by a system of N equations fo eigenvalues. Normally, φ_i is sought as a *linear combination of atomic orbitals* (LCAO) χ_k:

$$\varphi_i = \sum_{k=1}^{NO} C_{ki} \chi_k, \tag{11}$$

where NO is the number of *atomic orbitals* (AOs) and C are varied coefficients. As a result, we come to the following system of equations written in the matrix form:

$$\mathbf{HC} = \mathbf{SCE}, \tag{12}$$

where $\mathbf{E}$ is the diagonal matrix of the MO energies. Matrix $\mathbf{S}$ differs from the unitary one owing to non-ortogonality of AOs. Solution of Equation (12) with respect to the LCAO expansion coefficients $\mathbf{C}$ is performed by generalized diagonalization [31]. Matrix elements $\mathbf{H}$ are defined as:

$$H_{ij} = \int \chi_i(\mathbf{r}) H^{eff} \chi_j(\mathbf{r}) d\tau, \tag{13}$$

while the overlap integral S_{ij} for atomic orbitals χ_i and χ_j has the form:

$$S_{ij} = \int \chi_i(\mathbf{r}) \chi_j(\mathbf{r}) d\tau = \langle i \,|\, j \rangle. \tag{14}$$

Calculations are focused on determining the matrix elements for operator H^{eff}. Normally, this is attained by using the Hartree-Fock procedure [4, 5]: the one-electron Hamiltonian H^{eff} describes the motion of the i-th electron in the *self-consistent field* (SCF) induced by other electrons and nuclei fixed in space. In this case, the matrix elements of one-electron operator in Equation (9) are replaced by the Fock operator $\mathbf{F}$, so that matrix Equation (12) is transformed to the Hartree-Fock SCF equation in the Roothaan approximation:

$$\mathbf{F(C)C} = \mathbf{SCE}. \tag{15}$$

Equation (15) is written for closed electron shells in the *restricted Hartree-Fock* (RHF) approximation. Each MO is occupied by two electrons, so that MOs with opposite spins have the same spatial component. Equation (15) is third-order with respect to $\mathbf{C}$ and is

solved by successive approximation: for a given matrix **C**, we determine matrix **F**(**C**), then determine new matrix **C**, *etc.* until attaining a self-consistence with a preset accuracy.

The elements of the Fock matrix are expressed in terms of the electron density matrix **P** whose elements are defined by the elements of matrix **C**:

$$P_{ij} = 2\sum_{k}^{OCC} C_{ik} C_{jk}. \tag{16}$$

Summation is performed over all occupied MOs. In terms of the density matrix, the Fock matrix elements have the form:

$$F_{ij} = H_{ij} + \sum_{k,l=1}^{NO} P_{kl}[\langle ij|kl\rangle - \frac{1}{2}\langle ik|jl\rangle] \tag{17}$$

where H_{ij} are matrix elements of the so-called core Hamiltonian and $\langle ij|kl\rangle$ present *two-electron integral*. The simplest of these, $\langle ii|jj\rangle$, describe the electrostatic repulsion of two electrons in terms of the probability densities χ_i^2 and χ_j^2, respectively. The density matrix **P** defines also the probability to find an electron in the volume $d\mathbf{r}$ in the vicinity of **r**:

$$\rho(\mathbf{r})d\mathbf{r} = \sum_{ij} P_{ij}\chi_i(\mathbf{r})\chi_j(\mathbf{r})d\mathbf{r}. \tag{18}$$

When the spatial components of MOs with opposite spins (α and β, respectively) are different, in terms of the *unrestricted Hartree-Fock* (UHF) approximation, the wave functions are written [32] in the form:

$$\begin{aligned}\Psi^{UHF} = &\left(\sqrt{(N_\alpha + N_\beta)!}\right)^{-1} \times \\ &\det\left|\varphi_1^\alpha(1), \varphi_2^\alpha(2), \ldots, \varphi_{N_\alpha+1}^\alpha(N_\alpha), \varphi_{N_\alpha}^\beta(N_\alpha+1), \ldots, \varphi_{N_\alpha+N_\beta}^\beta(N_\alpha+N_\beta)\right|,\end{aligned} \tag{19}$$

where

$$\varphi_i^\alpha = \sum_{k=1}^{NO} C_{ki}^\alpha \chi_k \qquad \text{and} \qquad \varphi_i^\beta = \sum_{k=1}^{NO} C_{ki}^\beta \chi_k. \tag{20}$$

Accordingly, Equation (15) is transformed to the coupled system of equations for *open electron shells*:

$$\mathbf{F}^\alpha(\mathbf{C}^\alpha, \mathbf{C}^\beta)\mathbf{C}^\alpha = \mathbf{S}\mathbf{C}^\alpha\mathbf{E}^\alpha \tag{21a}$$

$$\mathbf{F}^\beta(\mathbf{C}^\alpha, \mathbf{C}^\beta)\mathbf{C}^\beta = \mathbf{S}\mathbf{C}^\beta\mathbf{E}^\beta. \tag{21b}$$

In this case, Equation (16) is replaced by the following ones:

$$P_{ij}^\alpha = \sum_{k=1}^{N_\alpha} C_{ik}^\alpha C_{jk}^\alpha, \tag{22a}$$

$$P_{ij}^\beta = \sum_{k=1}^{N_\beta} C_{ik}^\beta C_{jk}^\beta, \tag{22b}$$

Equation (17) is also replaced by the system of two coupled equations:

$$F_{ij}^\alpha = H_{ij} + \sum_{k,l=1}^{NO} P_{kl}[\langle ij|kl\rangle - P_{kl}^\alpha\langle ik|jl\rangle] \tag{23a}$$

$$F_{ij}^\beta = H_{ij} + \sum_{k,l=1}^{NO} P_{kl}[\langle ij|kl\rangle - P_{kl}^\beta\langle ik|jl\rangle] \tag{23b}$$

and

$$\mathbf{P} = \mathbf{P}^{\alpha} + \mathbf{P}^{\beta}. \tag{24}$$

The equations are solved by using the same iteration scheme of self-consistence as that used in case of Equation (15).

3. NDDO-WF approximation

Computation is focused on the solution of RHF Equation (15) or UHF Equations (21). The matrix elements H_{ij} of the core Hamiltonian are given by:

$$H_{ij} = \left\langle i \left| -\frac{\nabla^2}{2} \right| j \right\rangle - \sum_{A=1}^{NAT} \left\langle i \left| \frac{Z_A}{R_A} \right| j \right\rangle, \tag{25}$$

where

$$\left\langle i \left| \frac{1}{R_A} \right| j \right\rangle = \int \chi_i(r_1 - R_i) \frac{1}{r_1 - R_A} \chi_j(r_1 - R_j) dV_1 \tag{26}$$

and *NAT* is the number of atoms. The first term in Equation (25) gives the kinetic energy of electron while the second one, the energy of electron attraction to the nuclear cores.

Two-electron integrals $\langle ij|kl\rangle$ in Equations (23) are given by:

$$\langle ij|kl\rangle = \int\int \chi_i(r_1 - R_A)\chi_j(r_1 - R_B) \frac{1}{r_{12}} \chi_k(r_2 - R_C)\chi_l(r_2 - R_D) dV_1 dV_2, \tag{27}$$

where A, B, C, D stand for atoms, $i \in A$, $j \in B$, $k \in C$, $l \in D$, $dV = dxdydz$ for the 1-st and 2-nd electrons, and $x, y, z \in (-\infty, +\infty)$.

Solving Equations (21), we find out the electronic energy E_{elec}:

$$E_{elec} = \sum_{i,j=1}^{NO} [P_{ij}^{\alpha}(F_{ij}^{\alpha} + H_{ij}) + P_{ij}^{\beta}(F_{ij}^{\beta} + H_{ij})] \tag{28}$$

or

$$E_{elec} = \mathbf{S}p\mathbf{P}(\mathbf{H} + \mathbf{F}). \tag{29}$$

Until this moment, the logic of *ab initio* and SE calculations is identical. The difference manifests itself in calculating the elements of the Fock matrix (23) and two-electron integrals (27). In *ab initio* calculations, major difficulties are encountered in finding out the two-electron integrals $\langle ij|kl\rangle$ whose number is proportional to $[NO]^4$. In advanced versions that take into account *configurational interaction* (CI), such as CI, MCI, MP2-MP4, the number of operations increases even more. According to [33], the time consumption for non-empirical calculations exhibits the following proportionality:

- MP2 $\propto N^5$
- MP3, MP4(1+2+4), CI(1+2), CC(1+2) $\propto N^6$
- MP4, RC[1+2+(3)] $\propto N^7$
- complete CI (accurate solution on a given basis) $\propto N!$

where MP denotes the Mueller-Plesset technique from the perturbation theory of the appropriate order and CC stands for the method of coupled clusters. Of key importance here is the number N of selected one-electron basis functions.

SE techniques operate by using a minimal basis set of the Slater valence orbitals. This implies that MOs (11) are formed only by the AOs of the valence electrons of a given atom. A common feature of SE methods is the *approximation of zero differential overlap*

of AOs: $\chi_i \chi_j = 0$ when i and j refer to different atoms [5]. For widely adopted valence bases (AO combinations nl, $n'l$, *e.g.*, $2s3s$ *etc.*, are absent), the matrix **S** is unitary, so that Equation (15) can be rewritten in the form:

$$\mathbf{F}(\mathbf{C})\mathbf{C} = \mathbf{C}\mathbf{E}. \tag{30}$$

The matrix elements in Equations (23) are calculated for the AOs of one atom and for the AOs of different atoms separately. Let us assume that $i,j,k,l \in A$; $m,n \in B (A \neq B)$. In this case, the matrix element of the one-electron Hamiltonian for the AOs at one atom has the form:

$$H_{ij} = U_{ij} - \sum_{B \neq A} \left\langle i \left| \frac{Z_B}{R_B} \right| j \right\rangle, \tag{31}$$

where

$$U_{ij} = \left\langle i \left| -\frac{\nabla^2}{2} \right| j \right\rangle - \left\langle i \left| \frac{Z_A}{R_A} \right| j \right\rangle, \tag{32}$$

and $\left\langle i \left| \frac{1}{R_A} \right| j \right\rangle$ is given by Equation (26). On the valence basis, $U_{ij} = 0$ for $i \neq j$. When AOs are localized at different atoms:

$$H_{im} = \left\langle i \left| -\frac{\nabla^2}{2} - \frac{Z_A}{R_A} - \frac{Z_B}{R_B} \right| m \right\rangle - \sum_{C \neq A,B} \left\langle i \left| \frac{Z_C}{R_C} \right| m \right\rangle. \tag{33}$$

In SE calculations, exact Equations (31)-(33) are replaced by the parameter-containing ones whose appearance depends on the type of computation scheme. In case of CLUSTER-Z2, the diagonal elements U_{ii} are not determined from Equation (32) but are regarded as system parameters. Practical calculations [19, 20] show that the second term in Equation (31) can be readily replaced by the expression:

$$\left\langle i \left| \frac{Z_B}{R_B} \right| j \right\rangle = -\sum_m \Theta_{mm} \langle ij|mm \rangle, \tag{34}$$

where Θ_{mm} is the density matrix for neutral atom B averaged over each orbital quantum number l. In turn, Equation (33) is approximated by:

$$H_{im} = \frac{1}{2}(\beta_i + \beta_m) S_{im}, \tag{35}$$

where β are the method parameters.

Despite the above difficulties, a more serious problem is imposed by the calculation of two-electron integrals $\langle ij|kl \rangle$ in the second term of Equations (23). Special attention was given to decreasing the number of these integrals. Schematically, these integrals can be written as $\langle AB|CD \rangle$. This implies that the constituent AOs belong to different atoms A, B, C and D. In other words, it means that one-, two-, three-, and four-center two-electron integrals have to be considered. The NDDO approximation neglects the three- and four-center integrals on the basis of the zero differential overlap of AOs. Therefore, we have to calculate only the one-center $\langle AA|AA \rangle$ and two-center $\langle AA|BB \rangle$ integrals in order to obtain the Fock matrix. This strongly decreases the number of integrals and makes feasible calculations for many-particle systems. Indeed, when $\chi_i^A \chi_j^B = 0$, Equations (23) remain unchanged if AOs are localized at one center. When AO belong to different centers, Equations (23) acquire the form:

$$F_{im}^{\alpha} = H_{im} - \sum_{j,n=1}^{NO} P_{jn}^{\alpha} \langle ij|mn \rangle \tag{36}$$

(expression for F_{im}^{β} is similar).

Let us denote the starting integrals $\langle ij\,|mn\rangle$ as:

$$C_{Coul}^{AB} = \left\langle \chi_1^A \chi_2^A \,\middle|\, \chi_3^\beta \chi_4^B \right\rangle \tag{37}$$

According to [34], the use of the Slater functions with real spherical harmonics leads to the expression:

$$C_{Coul}^{AB} = \sum_{l=|l_1-l_2|}^{l_1+l_2} \sum_{l'=|l_3-l_4|}^{l_3+l_4} I_{ll'}(l_1,l_2,l_3,l_4), \tag{38}$$

where

$$I_{ll'} = \sum_{m(m_1,m_2)} \sum_{m'(m_3,m_4)} \delta_{mm'} \iint \rho_{lm}(1) \frac{1}{r_{12}} \rho_{l'm'}(2) d V_1 d V_2, \tag{39}$$

$l+l_1+l_2$ and $l'+l_3+l_4$ are even, while and may have two values, m_+ and m_-;

$$\begin{aligned} m_\pm &= \mathrm{sgn}(m_1)\mathrm{sgn}(m_2)\big||m_1| \pm |m_2|\big|, \\ \mathrm{sgn}(x) &= x/|x|, \mathrm{sgn}(o) = 1, l \geq |m|. \end{aligned} \tag{40}$$

In Equation (39), the terms containing $m_-(m_1,m_2)$ are zero when either sign $(m_1 * m_2) = -1$ and $m_1 + m_2 = 0$ or $m_1 * m_2 = 0$ (irrespective of the value of $m_1 + m_2$). The same is valid for $m'(m_3,m_4)$. Here sign has a normal mathematical meaning, in contrast to sgn used in Equation (40). $I_{ll'}$ characterizes interaction between two continuous charge distributions ρ_{lm} and $\rho_{l'm'}$.

However, the above calculation scheme suggested by Dewar and Thiel [8] was found to yield overestimated values of two-center integrals compared to exact solution of Equation (27) for $\mathbf{R}_{AB}$ values typical of chemical bonds. This drawback can be released in two ways. In terms of a commonly used NDDO approximation (basis of MNDO [8], AM1 [12], PM3 [13], and MNDO/d [21]), the continuous charge distribution ρ_{lm} is replaced by a system of 2^l point charges of $\pm 1/2^l$ a.u. each (zero total), on retention of the ρ_{lm} symmetry [8]. Charge coordinates are found from the condition for equality of multipole moment components. The latter ones are either found from the equation:

$$\left\langle \rho_{lm} \,\middle|\, x^I y^J z^K \right\rangle = \int \rho_{lm}(r_1) x_1^I y_1^J z_1^K d V_1 \tag{41}$$

(the components differ from zero when $I+J+K=l$) or are calculated directly for the system of point charges. As a result, we obtain:

$$\widetilde{C}_{Coul}^{AB} = \sum_{l,l',m,m'} \sum_{i,j} \frac{q_i q_j}{\sqrt{r_{ij}^2 + (\tau_l + \tau_{l'})^2}}, \qquad i \in 1-2^l,\ j \in 1-2^{l'}, \tag{42}$$

where q is the magnitude of point charges. Linear dimensions of the point charge system depend on the quantum numbers n, l and Slater exponent ε.

The integrals $\widetilde{C}_{Coul}^{AB}$ must meet the following limit conditions:

$$\lim_{R_{AB}\to\infty} \widetilde{C}_{Coul}^{AB} = C_{Coul}^{AB}, \tag{43a}$$

$$\lim_{R_{AB}\to 0} \widetilde{C}_{Coul}^{AB} = \widetilde{C}_{Coul}^{AA} (A \text{ and } B \text{ of the same type}). \tag{43b}$$

Equation (43a) always holds true. The conditions under which Equation (43b) is valid have to be specified. For $\widetilde{C}_{Coul}^{AB}$ in Equation (42) it can be attained by appropriate choice of τ_l and $\tau_{l'}$. In this case, the one-center integrals $\widetilde{C}_{Coul}^{AA}$ are regarded as system parameters [8, 12, 13].

The above scheme turned out to be a success, which made it possible to apply the MNDO, AM1 (PM3) techniques to large organic [12, 13] and inorganic systems containing C and Si [35–38]. But for small ε, AOs become smeared out, so the linear dimensions of the system of point charges may exceed interatomic distances that affect a dependence of $\widetilde{C}^{AB}_{Coul}$ on $\mathbf{R}_{AB}$. In particular, this circumstance makes the parametrization of alkali metal atoms difficult within the framework of the MNDO and AM1 techniques. Similar difficulties are encountered in the case of the elements that contain valence d-electrons: in the spd-basis, there always exist vacant AOs either of the p-type (transition metals) or d-type (metametals) with small ε. In addition, a procedure for determining ρ_l in the spd-basis becomes ambiguous *per se*. In order to retain the rotation invariance, we also have to average some two-electron integrals. A better solution was found in [21] to the problem of d-orbitals within the framework of the multipole approximation. The limitations imposed in the latter have been analyzed [39] in more detail.

An alternative *weighting factor technique* was suggested in [19] for calculating $\widetilde{C}^{AB}_{Coul}$. This type of the NDDO approximation is realized in CLUSTER-Z2 software (hereinafter *NDDO-WF approximation*). In this case, Equation (38) takes on the form:

$$\widetilde{C}^{AB}_{Coul} = \sum_{l=|l_1-l_2|}^{l_1+l_2} \sum_{l'=|l_3-l_4|}^{l_3+l_4} F_{ll'}(l_1,l_2,l_3,l_4) * I_{ll'}(l_1,l_2,l_3,l_4) \tag{44}$$

where $I_{ll'}$, as before, are given by Equation (39) while *weighting factors* $F_{ll'}$, which reduce the values of integrals assessed from Equation (27), are defined as follows:

$$F_{ll'}(l_1,l_2,l_3,l_4,) = \frac{[C_l(l_1,l_2) * C_{l'}(l_3,l_4)]^2 + A}{C_l(l_1,l_2) * C_{l'}(l_3,l_4) + A}. \tag{45}$$

Here $A = w_l(l_1,l_2) * w_{l'}(l_3,l_4) * R_{AB}^{l+l'+1}$. The condition $w_l > 0$ ensures the validity of the limit condition (43a). The values w_l are fixed and remain unchanged during the search of the method parameters while C_l in Equation (45) play the role. Generally, there are no limitations on quantum numbers, although the execution time increases with increasing n and l.

In terms of the NDDO-WF approximation, Equation (44) is used to determine both two-center $\widetilde{C}^{AB}_{Coul}$ and one-center $\widetilde{C}^{AA}_{Coul}$ integrals. In the former case, the $I_{ll'}$ values are calculated on the Slater basis by using a recurrence relation suggested in [40] for the double integrals in Equation (39). In the latter case, the weighed values of $C_l(l_1,l_2)C_{l'}(l_3,l_4) * I_{ll'}$ are calculated. Accordingly, the expression for $\widetilde{C}^{AA}_{Coul}$ has the form [34]:

$$\begin{aligned}
\widetilde{C}^{AA}_{Coul} &= c_1(n_1,\varepsilon_1)\cdot c_2(n_2,\varepsilon_2)\cdot c_3(n_3,\varepsilon_3)\cdot c_4(n_4,\varepsilon_4) \\
&\times \sum_{l,l'} \delta_{ll'} C_l(l_1,l_2) C_{l'}(l_3,l_4) \sum_{mm'} \delta_{mm'} \frac{4\pi}{(2l+1)(2l'+1)} \\
&\times Q_{lm}(l_1,l_2,m_1,m_1)\cdot Q_{l'm'}(l_3,m_3,l_4,m_4) \\
&\times \int_0^\infty r^{n+n_3+n_4} \exp(-r(\varepsilon_3+\varepsilon_4))\cdot [E_{n+l}(\alpha r) + A_{n-l-1}(\alpha r)]\, dr, \\
n = n_1 + n_2; \quad \alpha = \varepsilon_1 + \varepsilon_2; \quad & c(n,\varepsilon) = \left[(2\varepsilon)^{2n+1} \Big/ (2n)! \right]^{1/2}
\end{aligned} \tag{46}$$

Here

$$Q_{lm}(l_1,m_1,l_2,m_2) = \left[\frac{(2l_1+1)(2l_2+1)(2l+1)}{4\pi} \right]^{1/2} q_{lm}(l_1,m_1,l_2,m_2). \tag{47}$$

Summation over l, l', m, m' is the same as in Equations (38) and (39). The auxiliary variables $Q_n(x)$, $E_n(x)$, q_{lm} are taken from [34]. The interval $(0-\infty)$ in Equation (46) is divided into five subintervals: $(0-0.6R)$; $(0.6R-R)$; $(R-1.5R)$; $(1.5R-2.4R)$; $(2.4R-\infty)$; $R=(R_1\cdot R_2\cdot R_3\cdot R_4)^{1/4}$; where $R_i=\frac{2n_i+1}{2\varepsilon_i}$, $i\in 1-4$ is the mean size of AO, while n_i and ε_i are the quantum numbers of the AO. Within each of these intervals, the Gauss–Legendre quadrature formula [41] is applied over 26 points. The integration limits are reduced to the standard ones $(-1,+1)$:

$$\int_a^b f(x)dx=\frac{b-a}{2}\int_{-1}^{+1}\left(\frac{b-a}{2}\delta+\frac{b+a}{2}\right)d\delta, \tag{48}$$

$$\int_R^\infty f(x)dx=2\int_{-1}^{+1} f\left(\frac{1+\delta}{1-\delta}+R\right)\Big/(1-\delta)^2\ d\delta. \tag{49}$$

In order to attain better accuracy, the summation is carried out from smaller to greater terms. Up to $n=6$ and $l=3$, the calculation accuracy was of eight meaningful digits. It was checked by the $1,2\leftrightarrow 3,4$ commutation of index pairs in Equation (46), since this expression is asymmetric with respect to the operation. The values obtained for $n=2$, $l=1$ were checked according to [5]. As follows from Equation (27), for all $C_l(l_1,l_2)=1$ and $w_l(l_1,l_2)=0$, $\widetilde{C}_{Coul}^{AA(AB)}=C_{Coul}^{AA(AB)}$. Thus calculated $\widetilde{C}_{Coul}^{AB}$ and $\widetilde{C}_{Coul}^{AA}$ exhibit rotational invariance and a common appearance for any quantum number l.

The NDDO-WF parameters used in calculating in both the sp- and spd-basis are given in Table 1. The parametrization procedure is similar to that used for MNDO, AM1 and PM3 techniques [8, 12, 13]. However, in case of NDDO-WF, physically meaningful atomic charges are used at a starting stage of the parametrization. During successive geometry optimization, individual weighting coefficients for various characteristic of the molecules under consideration are used as well [20, 42]. Program package CLUSTER-Z2 is characterized in Table 2 that contains also a list of parametrized atoms.

Table 1. Parameters of the NDDO-WF method [19]

Value	Formula	Parameters	Notes
H_{ij}	(31)	$U_{ss}U_{pp}U_{dd}$	
H_{im}	(33)	$\beta_s\beta_p\beta_d$	
$I_{ll'}$	(39)	$\varepsilon_s\varepsilon_p\varepsilon_d$	Parameters ε are used in calculating ρ_{lm} [40].
S_{im}	(35)	$\varepsilon'_s\varepsilon'_p\varepsilon'_d$	For H, C, N, O, F, Cl, and Br atoms, these are identical to $\varepsilon_s\varepsilon_p\varepsilon_d$.
$\widetilde{C}_{Coul}^{AB},\widetilde{C}_{Coul}^{AA}$	(44),(46)	$C(l_1,l_2)$	4 parameters in the sp-basis and 10 parameters in the spd-basis.
$E_{nuc}^{1)}$		$\alpha K_i L_i M_i, i\in 1-4$	The same as in AM1 [12].

[1)] In the NDDO-WF approximation, E_{nuc} has the form [19]:

$$E_{nuc}^{AB}=T_{AB}(1+F_A+F_B)+\frac{Z_AZ_B}{R_{AB}}\left[\sum_{P_A}K_{P_A}\exp\left(-L_{P_A}(R_{AB}-M_{P_A})^2\right)+\sum_{P_B}K_{P_B}\exp\left(-L_{P_B}(R_{AB}-M_{P_B})^2\right)\right],$$

where $T_{AB}=\sum_{im}\theta_{ii}\theta_{mm}\langle ii\,|\,mm\rangle$, $i\in A$; $m\in B$, $F_A=\exp(\alpha_A R_{AB})$ and (for NH and OH bonds) $F_{O(N)}=R_{AB}\exp(-\alpha_{O(N)}R_{AB})$; K,L,M and α are the method parameters. The density matrix θ_{ii} for a neutral atom is averaged within each quantum number l. In AM1 [12], T_{AB} is defined by the Coulomb integral $\langle s^As^A|s^Bs^B\rangle$ for the s-orbitals of atoms A and B.

Table 2. Characteristics of the CLUSTER-Z2 package

Magnitude	Value
Number of atoms	895[2)]
Total number of orbitals	3802[2)]
Number of occupied orbitals	1900[2)]
Parametrized atoms	H, C, N, O, F, Cl, Br, I, P, S, Li, Na, K, Rb, Cs, Ag, Ti Al, Ga, As, Sb, Cr
Atoms under parametrization	Si, In, Ge, Cd, Sn, Te, Pb, Ni, Co, Bi, Zn, Hg, Tl

[2)] There is no limit for these values; the indicated ones are optimal for use in PC Intel PIII with RAM 256 Mb.

4. Program package CLUSTER-Z2 for sequential computing

The scheme of the CLUSTER-Z2 modules is presented in Figure 1. The modules 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 is known, SE calculations are performed through cyclization over pairs of indices i and j that enumarate 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 the principle of program parallelization in QC calculations: **massive of computations (cycles) over index pairs is divided into either continuos or cyclic blocks in accordance with the number of processors**.

The functionality of the modules shown in Figure 1 is as follows.

Module MAIN runs the procedure, accumulates data, and organizes the data output. During operation, it:

- reads out atomic parameters,
- calculates one-center integrals $\langle AA|AA\rangle$ and electronic energy of atoms,
- estimates the constants for calculating $\langle A|B\rangle$ and $\langle AA|BB\rangle$,
- appeals to the INPUT module that contains governing information about computing regimes and a starting molecular structure,
- appeals to the HCORE module for SCF calculations at a fixed molecular geometry or
- appeals to the OPT module that finds a minimum of the total energy and optimizes the starting structure,
- appeals to the FINAL module that initiates and performs the data output.

Module HCORE builds up the matrix of the core Hamiltonian H_{core} from Equations (31) and (35) and opens up the construction of the Fock matrix. During operation, it:

- appeals to the MC1 module that calculates and stores (on the disk) two-center integrals $\langle AA|BB\rangle$,
- appeals to the TAB module that calculates the integrals of electron attraction to the nucleus $\left\langle A\left|\frac{1}{R_B}\right|A\right\rangle$ and $\left\langle B\left|\frac{1}{R_A}\right|B\right\rangle$ Equation (34) (by using the data for $\langle AA|BB\rangle$),
- appeals to the SIJ1 module for calculating the overlap integrals $\langle A|B\rangle$,
- appeals to the NUCLR module for calculating the repulsion energy for atomic cores E_{nuc} (see footnotes to Table 1),

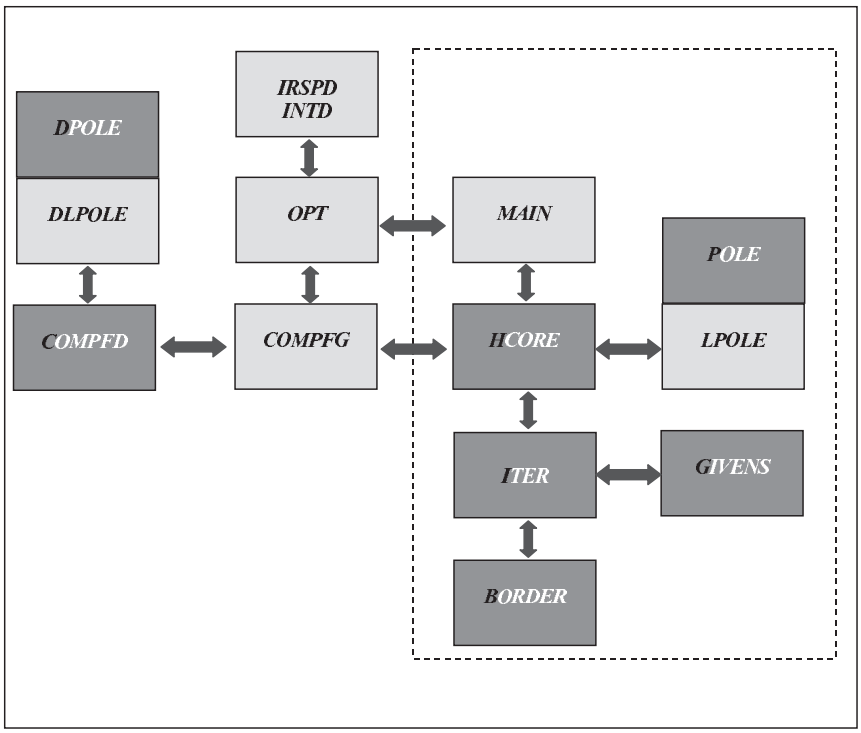

Figure 1. The scheme of the CLUSTER-Z2 package

- appeals to the ITER modulus for performing SCF calculations,
- calculates the total energy of the system $E_{tot} = E_{elec} + E_{nuc}$, Equation (4), and the heat of formation $\Delta H = E_{tot} - \sum_A E^A_{elec} + EHEAT^A$, where E^A_{elec} is the electronic energy of an isolated atom A in the NDDO-WF approximation and $EHEAT^A$ is the heat of formation for atom A.

Algorithm for constructing the core Hamiltonian matrix H_{core}:

- Contribution from two-center integrals is taken into account:
  - for all $i \in 2, NAT$ (A is the i-th atom and NAT is the number of atoms),
  - for all $j \in 1, i-1$ (B is the j-th atom).
    Calculation of $\langle A|B\rangle$ (total $N_i \times N_j$ elements, where N_i, N_j is the number of orbitals in atoms A and B).
    Calculation of $\langle AA|BB\rangle$ integrals and storage of these data on the disk for subsequent use in construction of the Fock matrix **F** (total number of $\langle AA|BB\rangle$ elements is $N_i(N_i+1)/2 \times N_j(N_j+1)/2$).
    Calculation of $\left\langle A\left|\frac{1}{R_B}\right|A\right\rangle$ and $\left\langle B\left|\frac{1}{R_A}\right|B\right\rangle$ massives (total number of elements is $N_i(N_i+1)/2$ and $N_j(N_j+1)/2$, the use is also made of $\langle AA|BB\rangle$).
  - Account of contribution from the above values into appropriate elements of the matrix **H**.
- Contribution from one-center integrals U_{ii}.

The above calculation procedure makes the basis for one-point calculations (at a fixed molecular structure). Its schematic is given in Figure 2.

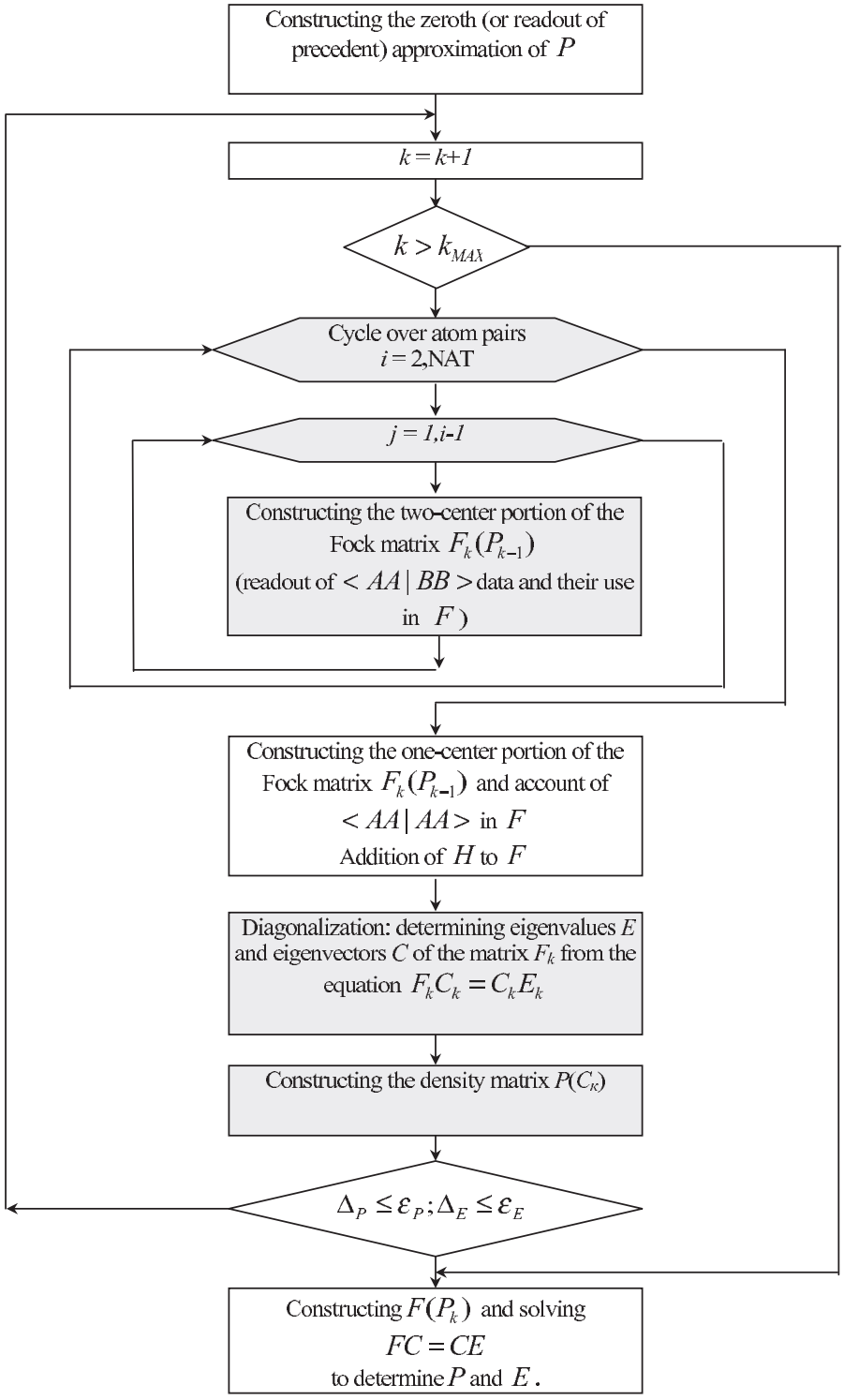

Figure 2. The scheme of SCF calculations: finding self-consistent electron density matrix **P** and electron energy E_{elec} (ITER module)

Module ITER performs SCF calculations and builds up the Fock matrix **F**. Calculations get started with zeroing the iteration counter k, *i.e.* $k = 0$, $\mathbf{E}_k = \mathbf{0}$. The maximum number of iterations, $k_{\max}$, is preset.

Constructing the zeroth approximation. The starting density matrix $\mathbf{P}_0$ is obtained by using a standard procedure or readout from the disk as a result of precedent iteration.

Iteration pitch. For all $k \leq k_{\max}$, the calculation sequence is as follows:

- constructing the Fock matrix $\mathbf{F}_k(\mathbf{P}_{k-1})$,
- diagonalization of the symmetric matrix **F**: solving the equation $\mathbf{F}_k\mathbf{C}_k = \mathbf{C}_k\mathbf{E}_k$ and finding out the eigenvectors $\mathbf{C}_k$ and eigenvalues $\mathbf{E}_k$,
- constructing the electron density matrix $\mathbf{P}_k = \mathbf{C}_k\mathbf{C}_k^{\perp}$ and calculation of $E_{elec,k} = \mathbf{S}p\mathbf{P}_k(\mathbf{H}+\mathbf{F}_k)$, Equations (28), (29),
- checking the convergence: $\Delta_P \leq \varepsilon_P$ and $\Delta_E \leq \varepsilon_E$, where $\Delta_P = |\mathbf{P}_k - \mathbf{P}_{k-1}|$, $\Delta_E = |\mathbf{E}_k - \mathbf{E}_{k-1}|$, and ε_P, ε_E are the preset calculation accuracies,
- constructing $\mathbf{F}(\mathbf{P}_k)$ and solving $\mathbf{FC} = \mathbf{CE}$ to find out resultant **P**, **E**, and E_{elec} according to Equation (28).
- When $k > k_{\max}$, the computation is stopped.

Algorithm for constructing the Fock matrix **F**:

- Two-center contributions are taken into account:
  - for all $i \in 2, NAT$ (A is the i-th atom, NAT is the number of atoms),
  - for all $j \in 1, i-1$ (B is the j-th atom).
    Readout the $\langle AA|BB\rangle$ data from the disk (total $N_i(N_i+1)/2 \times N_j(N_j+1)/2$ elements for given i and j, where N_i, N_j is the number of orbitals in the i-th and j-th atom).
    Calculation of contributions form $\langle AA|BB\rangle$ into appropriate elements of **F**.
- One-center contributions are taken into account:
  - for all $i \in 2, NAT$ (A is the i-th atom)
    Account of contributions from $\langle AA|BB\rangle$ that are stored in RAM.
- Account of **H**: $\mathbf{F}_k = \mathbf{F}_k + \mathbf{H}$
  For all this to be done, module ITER:
  - reads out (from disk) two-center integrals $\langle AA|BB\rangle$,
  - appeals to FOCKAB module to take into account integrals $\langle AA|BB\rangle$,
  - appeals to FOCINT module to take into account integrals $\langle AA|AA\rangle$,
  - constructs the resultant Fock matrix with account of the one-electron part of H_{im} (see Equations (23)),
  - appeals to module GIVENS that diagonalizes the symmetric matrix **F** and determimes the eigenvectors **C** and eigenvalues **E**,
  - appeals to module BORDER to construct the density matrix **P** and to calculate electronic energy E_{elec}, after which the iteration counter k is increased by 1,
  - checks out the degree of self-consistence relative to the density matrix **P** and energy E_{elec} until attaining the following relation: $\Delta_P \leq \varepsilon_P$ and $\Delta_E \leq \varepsilon_E$.

The scheme of SCF calculations is illustrated in Figure 3.

Module BORDER constructs the electron density matrix **P**, Equations (22) (for a given matrix **C**), and computes the electronic energy $E_{elec,k}$, Equation (28). Note that $\mathbf{P}_k$

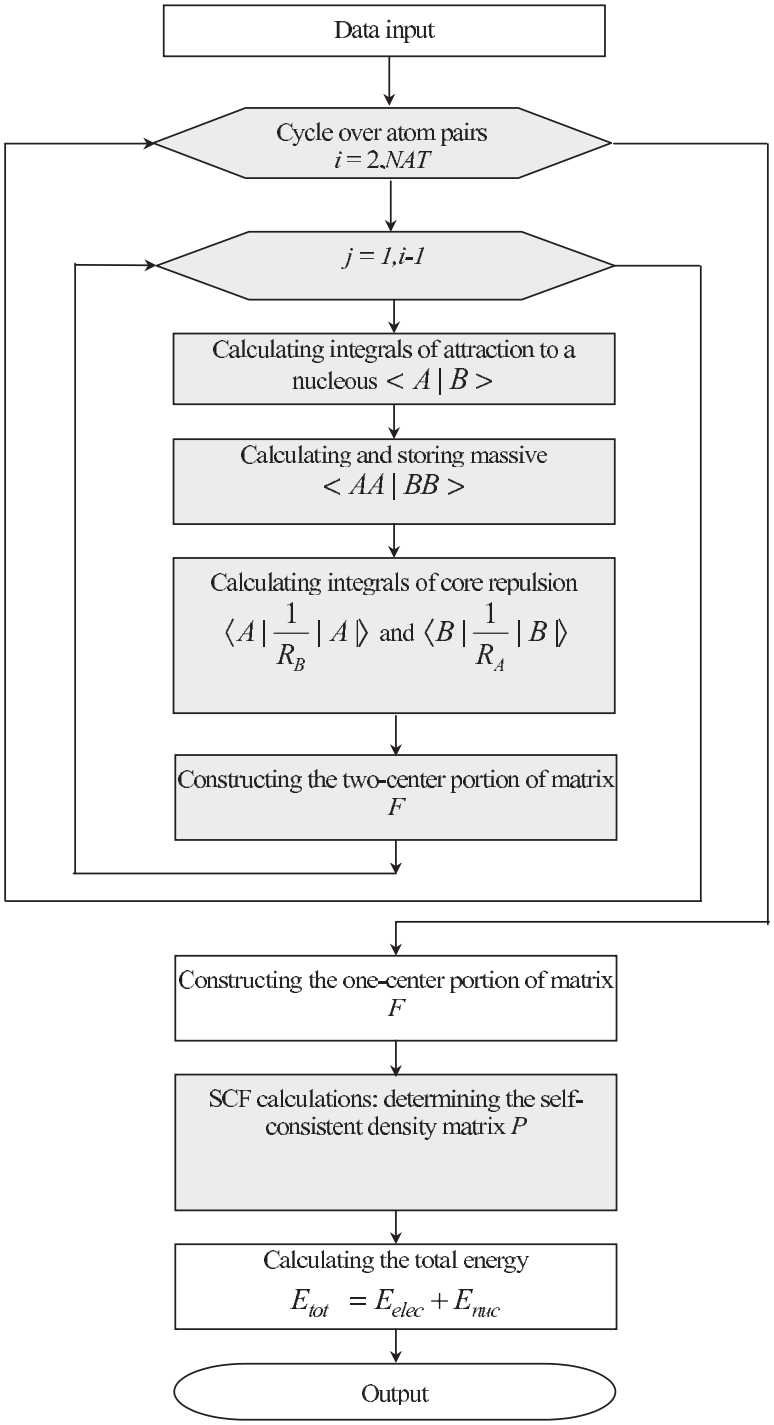

Figure 3. The scheme of one-point calculation for a fixed molecular geometry (HCORE and ITER modules)

can be constructed either by extrapolation from $\mathbf{P}_{k-2}$ or (at poor convergence) by using the level-shift technique [43, 44] (see a scheme of SCF calculations in Figure 3).

Algorithm for constructing the density matrix:

- One-center integrals are taken into account:
  - for all $i \in 1$, *NORBS* (*NORBS* is the number of orbitals).
    Calculating the one-center component of the density matrix, Equations (22).
- Two-center integrals are taken into account:
  - for all $i \in 2$, *NORBS*,
  - for all $j \in 1, i$.
    Calculating the two-center component of the density matrix.
- Computing the energy **E**: $\mathbf{E}_k = \mathbf{S}p\mathbf{P}_k(\mathbf{H}+\mathbf{F}_k)$.

Module GIVENS performs the diagonalization procedure, solves the equation $\mathbf{F}_k\mathbf{C}_k = \mathbf{C}_k\mathbf{E}_k$ and finds out $\mathbf{C}_k$ and $\mathbf{E}_k$.

The above modules are the basis in the scheme of 'one-point' calculations (Figure 1). The flow chart of these calculation is given in Figures 2 and 3.

In order to optimize the molecular geometry, the above scheme should be supplemented with modules OPT, COMFG, and COMPFD.

Module OPT:

- optimizes the molecular geometry over internal coordinates by determining a minimum of E_{tot}; during execution, appealing by choice to standard programs of unconstrained linear minimization VA09A [45], BFGS [46–50] or the Newton-Raphson technique for exact localization of a minimum or saddle point [51],
- appeals to the IRSPD and INTD modules to calculate the harmonic vibration frequencies and the relevant intensities in IR and Raman spectra.

Module COMPFD calculates the derivatives of the total energy E_{tot} with respect to the *Cartesian coordinates x, y, z* of atoms. Denoting x, y, z just as x (total $3\times$ *NAT* components), we can write Equation (4) in the form:

$$E_{tot}(x,\mathbf{P}) = E_{elec} + \sum_{A>B} E^{AB}_{nuc}. \tag{50}$$

The first term in Equation (50) gives the total electronic energy while the second one, the energy of nuclei repulsion. Because of the self-consistency of solution for **P**,

$$\frac{\partial E_{tot}}{\partial \mathbf{P}} = 0. \tag{51}$$

Therefore, we have to calculate only the derivatives:

$$\frac{\partial E_{tot}}{\partial x} = \frac{\partial}{\partial x} E_{nuc} + \mathbf{S}p\mathbf{P}\left(\frac{\partial \mathbf{H}}{\partial x} + \frac{\partial \mathbf{F}}{\partial x}\right). \tag{52}$$

that, in their turn, are determined by the following derivatives, according to Equations (23), (31), (35) for **F** and **H**:

$$\frac{\partial}{\partial x}\langle AA|BB\rangle, \quad \frac{\partial}{\partial x}\langle A|B\rangle, \quad \frac{\partial}{\partial x}\left\langle A\left|\frac{1}{R_B}\right|A\right\rangle, \quad \text{and} \quad \frac{\partial}{\partial x}\left\langle B\left|\frac{1}{R_A}\right|B\right\rangle. \tag{53}$$

In other words, only two-center integrals provide contribution to $\frac{\partial E_{tot}}{\partial x}$. During execution, the COMPFD module:

- appeals to the HCORE module to calculate E_{tot} and $\mathbf{P}$,
- appeals to the DSIJ1 module to calculate $\frac{\partial}{\partial x}\langle A|B\rangle$,
- appeals to DMC1 to calculate $\frac{\partial}{\partial x}\langle AA|BB\rangle$,
- appeals to DTAB to calculate $\frac{\partial}{\partial x}\left\langle A\left|\frac{1}{R_B}\right|A\right\rangle$ and $\frac{\partial}{\partial x}\left\langle B\left|\frac{1}{R_A}\right|B\right\rangle$,
- appeals to DFAB to calculate $\frac{\partial F_{ij}}{\partial x}$,
- appeals to NUCLR to calculate $\frac{\partial E_{nuc}^{AB}}{\partial x}$,
- appeals to DPOLE or DLPOLE to calculate field-induced corrections for $\frac{\partial E_{tot}}{\partial x}$ [52, 53],
- calculates the derivatives $\frac{\partial E_{tot}}{\partial x}$.

Algorithm for calculating $\frac{\partial \mathbf{H}}{\partial x}$ and $\frac{\partial \mathbf{F}}{\partial x}$ is in much similar to that used in constructing the matrices $\mathbf{F}$ and $\mathbf{H}$:

- finding a self-consistent solution $(\mathbf{P}, E_{tot})$ (by using HCORE) for given x_i, y_i, z_i; $i \in 1, NAT$,
- for all $i \in 2, NAT$ (A is the i-th atom, NAT is the number of atoms),
- for all $j \in 1, i-1$ (B is the j-th atom),
- calculating $\frac{\partial}{\partial x_i}\langle A|B\rangle$ (in this case, $\frac{\partial}{\partial x_j}\langle A|B\rangle = -\frac{\partial}{\partial x_i}\langle A|B\rangle$, this holds also true for other integrals),
- calculating $\frac{\partial}{\partial x_i}\langle AA|BB\rangle$,
- calculating $\frac{\partial}{\partial x_i}\left\langle B\left|\frac{1}{R_A}\right|B\right\rangle$ and $\frac{\partial}{\partial x}\left\langle A\left|\frac{1}{R_B}\right|A\right\rangle$,
- calculating $\frac{\partial E_{nuc}^{AB}}{\partial x_i}$,
- calculating $\frac{\partial E_{tot}}{\partial x_i}$ and $\frac{\partial E_{tot}}{\partial x_j}$.

Module COMPFG calculates the derivatives of the total energy E_{tot} with respect to the *internal coordinates* q (bond length, valence and dihedral angles):

$$\frac{\partial E_{tot}}{\partial q_k} = \sum_{i=1}^{3\times NAT} \frac{\partial E_{tot}}{\partial x_i} \times \frac{\partial x_i}{\partial q_k}, \tag{54}$$

where $\frac{\partial x_i}{\partial q_k}$ is determined by numerical (5-point) differentiation. During execution, COMPFG appeals to COMPFD to calculate $\frac{\partial E_{tot}}{\partial x_i}$, $i \in 1, 3\times NAT$. The COMPFD and COMPFG modules are of key importance in the optimization of molecular geometry (see Figure 4).

Modules POLE, LPOLE, DPOLE, and **DLPOLE** are used in determining the molecular structure in the electric field of any configuration [52, 53].

5. Parallel implementation of the NDDO-WF approximation in the MP-ZAVA package

The sequential algorithms of CLUSTER-Z2 similarly to other QC programs, represent a structure with a massive parallelism [54]. In Figures 1, 2, 3, and 4, these structures are allocated with color. For an estimation of the contribution of each computing modules in general time of calculations, profiling the sequential program was carried out. As show results:

- diagonalization – 40 – 60 %,
- calculation of the density matrix – up to 40 %,
- construction of the Fock matrix (calculation of intergrals of interelectronic interaction and overlap) – up to 5 %,

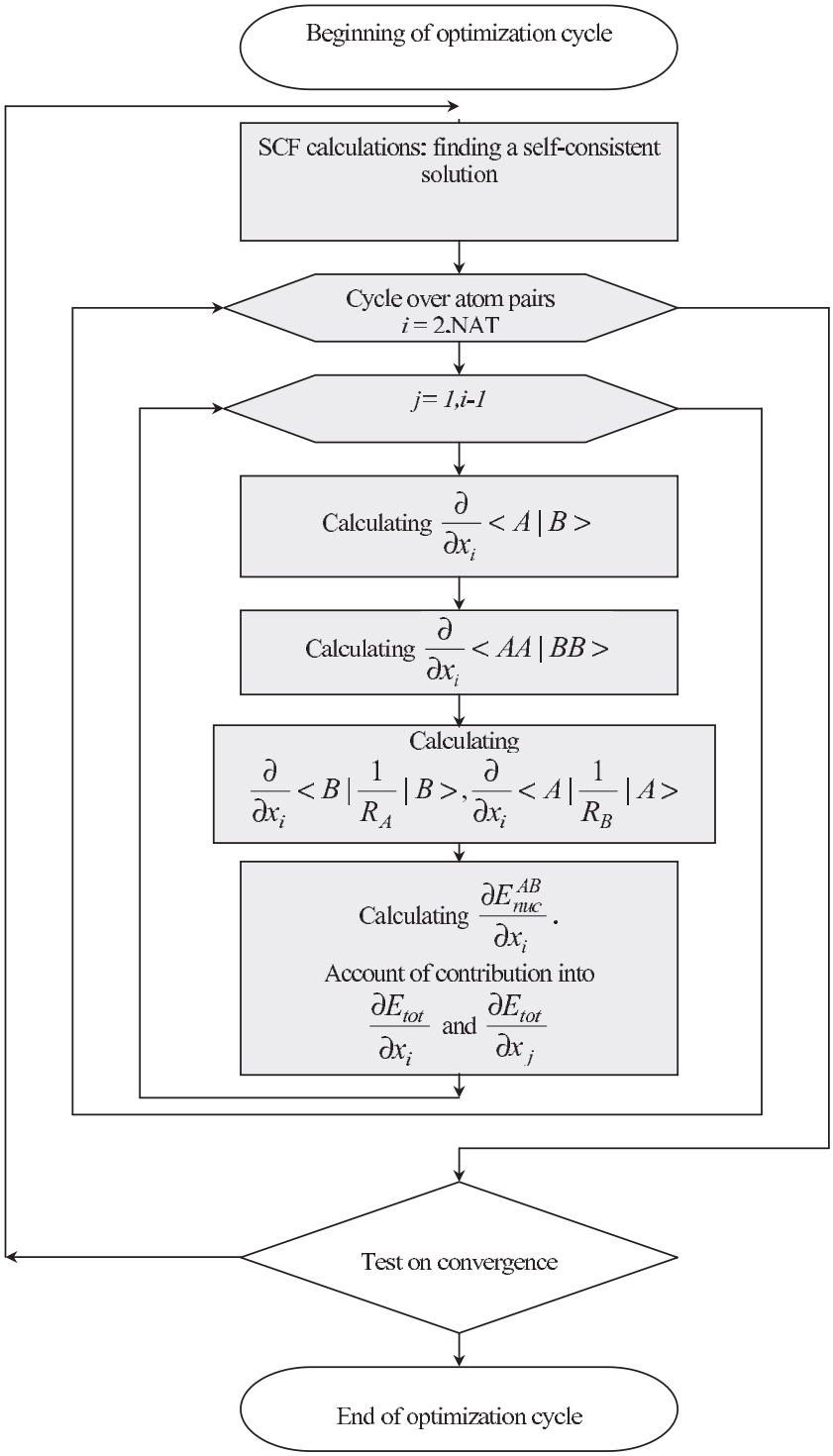

Figure 4. Optimization of molecular geometry

- construction of the core Hamiltonian – up to 10 %,
- calculation of the derivatives – up to 15 %.

Note that this distribution markedly depends on the type of computation procedure (one-point calculation, optimization of molecular geometry), the type of system (rigid/nonrigid systems, open/closed shells, *etc.*), and on other factors. Table 3 shows results obtained for a model C-300 molecule (referred to below as carbon-graphite strip).

Table 3. Profiling the main algorithms of CLUSTER-Z2: C-300 molecule

Algorithm	Number of calls	Time, s	%
Diagonalization	1390	64470	33.34
Construction of the electron density matrix	1387	80090	41.42
Calculation of the derivatives	188	23400	12.10
Construction of the Fock matrix	1575	6869	3.55
Construction of the core Hamiltonian	188	16480	9.75
Read/Write $\langle AA\|BB\rangle$ from the disk	79070550	5216	2.70
Total time, s		193351	100.00

5.1. Parallel implementation of the diagonalization module

In the NDDO-WF approximation, the most labor-costly is repeated diagonalization, that is, the solution of the equation $\mathbf{FC} = \mathbf{CE}$ and finding the eigenvalues $\mathbf{E}$ and eigenvectors $\mathbf{C}$. Parallel realization of this procedure has been made on the basis of libraries ScaLAPACK [55], BLACS [56], and MPI [30]. The eigenproblem was solved by applying the PDSYEVX procedure from the ScaLAPACK library. The matrix of eigenvectors were found by using the PDLACP3 procedure from the same library. The obtained data are collected in Tables 4 and 5. As show results of testing the PDSYEVX is efficient in the case of sufficiently large matrices ($N \geq 1200$). For lower N, the procedure exhibits poor scalability: for 8 processors, the speedup $S_p \leq 5$. In view of this, the procedure was used only for $N > 250$; otherwise, the use was made of the serial Givens algorithm [57]. This also implies that the use of 8 – 16 processors is inexpedient.

Table 4. Execution time for PDSYEVX diagonalization for matrices of various dimension, s

Size of matrix	Number of processors			
	1	2	4	8
2000	1094.00	609.00	251.60	144.23
1800	810.74	399.95	184.96	108.53
1600	564.71	272.55	124.74	81.09
1400	359.58	176.35	80.72	58.33
1200	213.89	105.75	51.09	43.1
1000	113.02	56.00	29.68	29.31
800	51.91	27.89	17.10	19.68
600	21.07	12.27	8.92	12.35
400	6.27	4.62	4.04	6.72
200	1.19	1.18	1.28	2.95

Table 5. Speedup of the PDSYEVX diagonalization

Size of matrix	Number of processors			
	1	2	4	8
2000	1.00	1.80	4.35	7.59
1800	1.00	2.03	4.38	7.47
1600	1.00	2.07	4.53	6.96
1400	1.00	2.04	4.45	6.16
1200	1.00	2.02	4.19	4.96
1000	1.00	2.02	3.81	3.86
800	1.00	1.86	3.04	2.64
600	1.00	1.72	2.36	1.71
400	1.00	1.36	1.55	0.93
200	1.00	1.01	0.93	0.40

5.2. Implementation of parallel calculation of the density matrix

Another time-consuming procedure is calculating the electron density matrix. These calculations are within a three-level nest of cycles (Figure 5): two outer ones successively bypass the lower triangle of the density matrix along with the diagonal elements that are stored in a linear massive $\mathbf{P}(L)$, where $L = 1, NO \times (NO+1)/2$.

Mapping the position of the (I, J) element from the lower triangular matrix onto the appropriate positions L in the data file $\mathbf{P}$ is performed by the following algorithm:

$$I \times J \to L: \quad L = (I-1) \times I/2 + J\,.$$

The reverse transform is carried out by the algorithm:

$$L \to I \times J: \begin{cases} I = \begin{cases} \sqrt{2L+\frac{1}{4}}+\frac{1}{2}, & I \neq J \\ \sqrt{2L+\frac{1}{4}}-\frac{1}{2}, & I = J \end{cases} \\ J = L-(I-1) \times I/2 \end{cases}.$$

Calculations of all elements within each cycle are independent, so that it can readily be transformed into a linear parallel cycle according to $L = 1, NO \times (NO+1)/2$, *NPROC*, where *NPROC* is the number of parallel processes. Figures 6 and 7 illustrate the cyclic and block variants of data decomposition and organization of calculations for which $MYID \in 0$ and *NPROC*-1 are the process identifiers. In a given procedure, we use the block scheme of decomposition. The above fragment of the three-level nest of cycles, but of parallel computation, is shown in Figure 8.

5.3. Parallel implementation of SCF calculations, computing the Fock matrix F, the core Hamiltonian H, and gradients of the total energy E_{tot} over the Cartesian coordinates

Sequential fragments of all procedures are lumped in the cycles of identical structure, which makes possible the use of the same scheme of parallelization. For the sake of brevity, we will consider here only the parallel realization of computing the two-center components of the E_{tot} gradients with respect to the Cartesian coordinates of atoms (see Equations (52), and (53)). The corresponding sequential algorithm is lumped in the cycle shown in Figure 9. Just as in the case of the density matrix, the algorithm performs bypassing

```
                L=0
                DO 80 I=1,NO
                DO 60 J=1,I
                L=L+1
                A=0.D0
                DO 50 K=1, Nα(Nβ)
50              A=A+C(I,K)*C(J,K)
                P(L)=A+…
                …
60              CONTINUE
                „„
80              P(L) = P(L)+„„
```

Figure 5. The structure of sequential cycles in the BORDER module

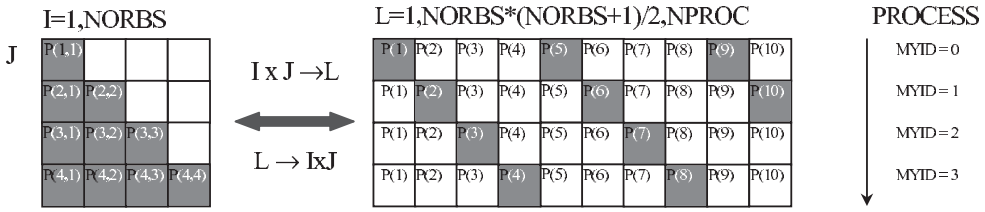

Figure 6. The cyclic scheme of data decomposition and organization of parallel computing

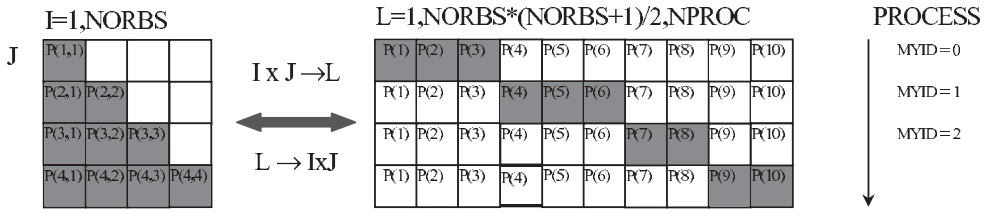

Figure 7. The block scheme of data decomposition and organization of parallel computing

```
                LENGTH= NO*(NO+1)/2
                DO 80 IE=1,LENGTH , NPROC
                L=IE+MYID
                IF(L.GT.LENGTH) GOTO 80
                I=(SQRT(2*L+0.25D0)+1.5D0
                J=L-(I-1)*I/2
                …
                A=0.D0
                DO 50 K=1, Nα(Nβ)
50              A=A+C(I,K)*C(J,K)
                P(L)=A+…
                …
80              CONTINUE
```

Figure 8. The structure of parallel cycles in the BORDER module

```
      DO 10 I=2,NAT
            DO 20 J=1,I-1
      C Некоторые вычисления,
        зависящие только
      C  от парыI  и J
            f(I,J)=...
            G(I)=G(I)+f(I,J)
            G(J)=G(J)-f(I,J)
       20   CONTINUE
       10   CONTINUE
```

Figure 9. The structure of sequential algorithm in the COMPFD module

the lower triangular matrix $\mathbf{F}(I,J)$ and stores the results in a one-dimensional massive G. As before, all calculations are independent. To perform the algorithm transform, let $NATA{=}NAT{\times}(NAT-1)/2$ denote the number of different atomic pairs. Let $NPROC$ be the number of parallel processes (preset at program start-up and defined by MPI tools [30] when running the program). As previously, we have to carry out two transformations: (a) mapping the $I\times J\to L$ positions of the (I,J) element of the lower triangular matrix onto the appropriate positions L (that assign the linear numeration of pairs) and (b) the reverse transform $L\to I\times J$.

Transform (a) is carried out by the algorithm:

$$I\times J\to L:\ L=(I-3)\times I/2+J+1.$$

The reverse transform (b) is defined as:

$$L\to I\times J:\ \begin{cases} I=\sqrt{2L+\frac{1}{4}}+\frac{1}{2} \\ J=L-(I-3)\times I/2-1 \end{cases}$$

In the interest of time, we will perform the transform only once. For this to be done, let us preliminarily load the indices of each pair I and J into arrays ISM and JSM, respectively. Thus, for $NAT=4$ we will obtain the set of indices I, J, and L shown in Table 6. Calculations of all elements within the cycle are independent, so that the latter can be easily transformed to a linear parallel cycle over the number of pairs $L=1$, $NAT{\times}(NAT-1)/2$, $NPROC$, where $NPROC$ is the number of parallel processors (Figure 10). In this case, we use the cyclic scheme of distribution (Figure 6). Each process has its own copy of array G. For $NAT=4$ and $NPROC=2$, the distribution of data is presented in Table 7. Pairs 1, 3, 5 will be processed by the zeroth processor while pairs 2, 4, 6, by the first one. Therefore, some portion of $G(1)$ terms turns out in the zeroth processor (from pair $(2,1)$), some portion in the first one (from pairs $(3,1)$ and $(4,1)$), *etc.* The final result is obtained by summation of $G(1)$ from the 0-th and 1-st processor (Table 7) by applying the MPI_REDUCE reduction procedure. Other G arrays are treated similarly. The obtained result (total NAT elements) is put into the working array GBUF of the 0-th processor, write into the G array, and then distributed by the MPI_BCAST function over other processors.

Brought together and applied to all respective sequential cycles of CLUSTER-Z2, the above parallel cycles are united by the MP-ZAVA program.

6. Speedup and efficiency of program MP-ZAVA

Let us determine the speedup (S_p) and scalability of the MP-ZAVA program for two model systems taken as some examples. We will start with a model C-300 molecule (carbon-graphite strip with 894 AOs, Figure 11) treated in the sp-basis. Table 8 shows the profiling

```
      LENGTH=NAT*(NAT-1)/2
      CALL MPI_COMM_SIZE(MPI_COMM_WORLD,NPROC,IERR)
      CALL MPI_COMM_RANK(MPI_COMM_WORLD,MYID,IERR)
      …
      DO  1 L=1,LENGTH,NPROC
      IF(L+MYID.GT.NATA)  GOTO 1
      I =ISM(L+MYID)
      J =JSM(L+MYID)
      …
      G(I)=G(I)+f(I,J)
      G(J)=G(J)-f(I,J)
 1    CONTINUE

      CALL MPI_REDUCE(G,GBUF,NAT,MPI_DOUBLE_PRECISION,..,MPI_SUM,..)

      IF(MYID.EQ.0)  THEN
      DO  12 L=1,NAT
      G(L) =GBUF(L)
12    CONTINUE
      END IF

      CALL MPI_BCAST(G,NAT, MPI_DOUBLE_PRECISION,0, …)
```

Figure 10. The structure of parallel algorithm in the COMPFD module

Table 6. Relation between I, J and L indices (direct mapping)

L	1	2	3	4	5	6
ISM	2	3	3	4	4	4
JSM	1	1	2	1	2	3

Table 7. Calculation of array G

Process	L					
	1	2	3	4	5	6
0	$G(2)$, $G(1)$		$G(3)$, $G(2)$		$G(4)$, $G(2)$	
1		$G(3)$, $G(1)$		$G(4)$, $G(1)$		$G(4)$, $G(3)$

of the main algorithms of MP-ZAVA for different number of the processors MIPS R10000 of the RM600 E60 system while Table 9 and Figure 12 present the obtained speedups.

The dissociative adsorption of water on the (101) surface of TiO_2 (rutile) was modeled [58] in the *spd*-basis for a cluster of 369 atoms (2014 AOs), hereinafter termed the 'TiO_2 + water' cluster (see Figure 13). In computations, we used both the SMP-RM600 E60B system and the Cluster Intel PIII system with distributed memory. The data obtained for the former system are collected in Tables 10, 11, and Figure 14, while those for the latter, in Tables 12, 13 and Figure 15. Let us analyze these data with regard to the main algorithms of MP-ZAVA.

Diagonalization PDSYEVX. As follows from Tables 8, 9, and Figure 12, when calculating the 300-atoms carbon-graphite strip in the *sp*-basis, a maximum value of $S_P = 2.51$ is attained in case of 4 processors. This is in line with the data of test calculations based on the algorithm *PDSYEVX* for matrices with $N \leq 1000$. In this case, the diagonalization algorithm is obviously a rate-limiting factor that makes efficient use of a higher number of processors impossible.

Figure 11. A model C-300 molecule (carbon-graphite strip): large balls stand for carbon atoms, while small ones – for hydrogen atoms

Table 8. Execution time for the main algorithms of MP-ZAVA for the SMP-system RM600 E60: C-300 molecule, s

Algorithm	Number of processors			
	1	2	4	8
Diagonalization	64470	40140	25650	37820
Construction of the electron density matrix	80090	47660	25930	18760
Construction of the core Hamiltonian	16480	9703	4392	2373
Calculation of the derivatives	23400	13380	5884	2984
Construction of the Fock matrix	6869	4424	2356	1669
Read/Write $\langle AA\|BB\rangle$ from the disk	5216	3484	1793	993
Total time	193351	119068	69388	70573

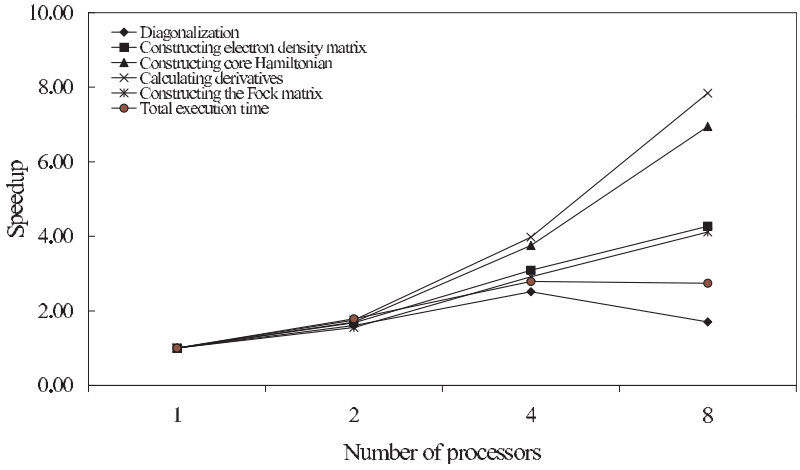

Figure 12. The speedup of the main algorithms of MP-ZAVA in the SMP-system RM600 E60: C-300 molecule

Table 9. Speedup for the main algorithms of MP-ZAVA for the SMP-system RM600 E60: C-300 molecule

Algorithm	Number of processors			
	1	2	4	8
Diagonalization	1.00	1.61	2.51	1.70
Construction of the electron density matrix	1.00	1.68	3.09	4.27
Construction of the core Hamiltonian	1.00	1.70	3.75	6.94
Calculation of the derivatives	1.00	1.75	3.98	7.84
Construction of the Fock matrix	1.00	1.55	2.92	4.12
Read/Write integrals	1.00	1.50	2.91	5.25
Overall speedup	1.00	1.79	2.79	2.74

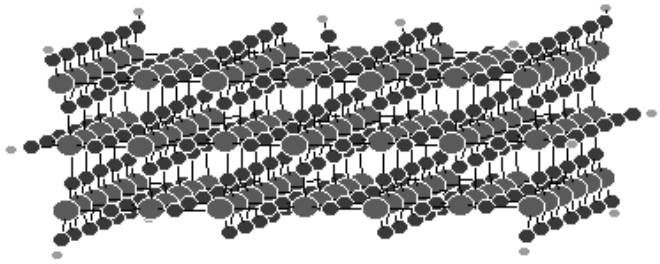

Figure 13. A model 'TiO_2 + water' cluster: the largest balls stand for Ti, medium (dark) ones – for O, and the smallest (light) balls correspond to H atoms [58]

As established earlier, *PDSYEVX* becomes efficient for matrices with $N \geq 1200$. This is well supported by the data obtained for the 'TiO_2 + water' cluster with 2014 AOs. As follows from Table 11, on going from 2 to 8 processors in the SMP-system RM600 E60, the speedup is 2.91. Large speedup relative to 1 processor is a result of exceedingly low productivity of the 1-processor scheme (execution time 12 days). With regard to this circumstance, a real value can be expected to be around 5. For the Cluster Intel PIII system, the speedup for 8 processors attains a value of 2.09 (Table 13). Such a relatively low value can be attributed to a low carrying capacity of the commutation compared to the communicating subsystem SMP in the RM600 E60 system that utilizes scalable coherent interface [59].

Calculation of the electron density matrix. Generally, our data are indicative of a fairly good speedup attained in case of 8 processors: $S_P = 4.27-7.14$ (see Tables 9, 11, 13). This algorithm is scalable, at least for a relatively low number of processors ($P \leq 16$). As for massively parallel systems, a major rate-limiting factor here is the mounting of all matrix segments by executing the MPI_ALLGATHER procedure whose efficiency depends on the realization quality for any system.

Construction of the core Hamiltonian. In case of 8 processors, $S_P = 6.94-7.36$, which is indicative of good scalability. This is important for calculations with a relatively large contribution of the integrals to be calculated.

Calculation of the derivatives. For the derivatives $\frac{\partial}{\partial x}\langle AA|BB\rangle$, $\frac{\partial}{\partial x}\langle A|B\rangle$, $\frac{\partial}{\partial x}\left\langle A\left|\frac{1}{R_B}\right|A\right\rangle$ and $\frac{\partial}{\partial x}\left\langle B\left|\frac{1}{R_A}\right|B\right\rangle$), the process is characterized by a high speedup close to limiting values found for 8 processors: $S_P = 7.84-7.97$. It is necessary to note high efficiency and scalability the parallelization procedure.

Construction of the Fock matrix **F**. The algorithm is characterized by a relatively high value of $S_P = 4.12$.

READ and WRITE $\langle AA|BB\rangle$ from/on the disk. Just as in the most of QC programs, such as MOPAC [16], CLUSTER-Z1 [18], CLUSTER-Z2 [22], GAMESS [60, 61] *etc.*, repeated computing of integrals (whose number may be large) is avoided by their recording on the disk followed by their multiple readout as required. Ineffective realization of this procedure may strongly decrease the program's performance. In order to improve the program's efficiency, this procedure was also parallelized. Each processor performs the recording of calculated integrals into its own file. Accordingly, each processor is using only the integrals that have been obtained in a given processor. This is attained upon coordination

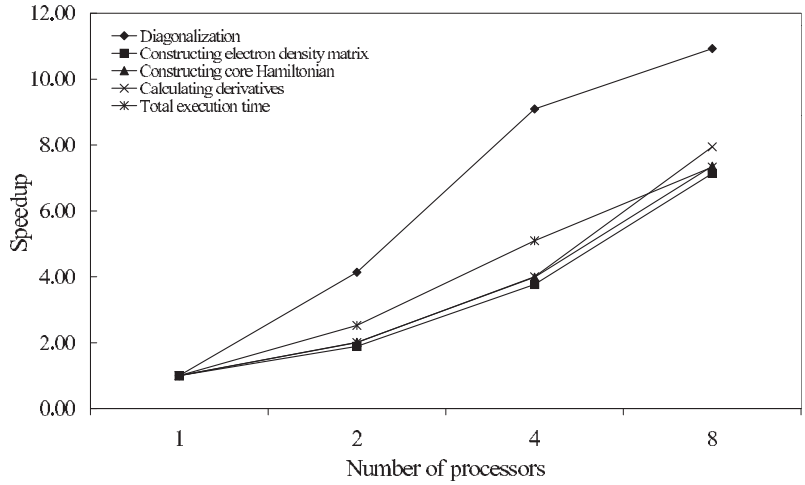

Figure 14. The speedup of the main algorithms of MP-ZAVA in the SMP-system RM600 E60: 'TiO_2 + water' cluster

Table 10. Execution time for the main algorithms of MP-ZAVA in the SMP system RM600 E60: 'TiO_2 + water' cluster, s

Algorithm	Number of processors			
	1	2	4	8
Diagonalization	657280	158900	72250	60160
Construction of the electron density matrix	333824	177100	88620	46780
Construction of the core Hamiltonian	73920	36810	18530	10050
Calculation of the derivatives	117780	58860	29430	14820
Total time	1145030	453557	224870	156273

Table 11. Speedup for the main algorithms of MP-ZAVA in the SMP system RM600 E60: 'TiO_2 + water' cluster

Algorithm	Number of processors			
	1	2	4	8
Diagonalization	1.00	4.14	9.10	10.93
Construction of the electron density matrix	1.00	1.88	3.77	7.14
Construction of the core Hamiltonian	1.00	2.01	3.99	7.36
Calculation of the derivatives	1.00	2.00	4.00	7.95
Overall speedup	1.00	2.52	5.09	7.33

of the parallelization schemes for appropriate procedures: calculating and writing, on the one hand, as well as reading and use in further calculations, on the other. As a result, the procedures of computing, writing, reading, and subsequent using turn out independent in the framework of individual processes. In case of 8 processors, we found that $S_P = 5.25$.

Overall speedup. For 8 processors, the speedup of the entire program $S_P = 2.74-5.20$. This implies that high efficiency of parallel calculation can be attained even for the systems with relatively slow commutation. Thus, for the 'TiO_2 + water' cluster (one point

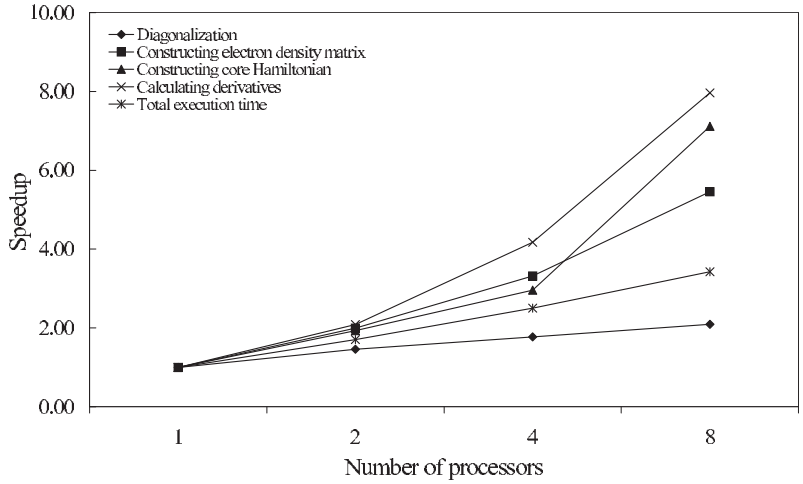

Figure 15. The speedup of the main algorithms of MP-ZAVA in the Cluster Intel PIII system: 'TiO_2 + water' cluster

Table 12. Execution time for the main algorithms of MP-ZAVA for the system Cluster Intel PIII: 'TiO_2 + water' cluster, s

Algorithm	Number of processors			
	1	2	4	8
Diagonalization	49290	33750	27760	23530
Construction of the electron density matrix	49940	25040	15060	9156
Construction of the core Hamiltonian	23430	12110	7917	3292
Calculation of the derivatives	33660	16120	8061	4225
Total time	173246	101756	69209	50554

Table 13. Speedup of the main algorithms of MP-ZAVA in the system Cluster Intel PIII: 'TiO_2 + water' cluster

Algorithm	Number of processors			
	1	2	4	8
Diagonalization	1.00	1.46	1.78	2.09
Construction of the matrix of electron density	1.00	1.99	3.32	5.45
Construction of the core Hamiltonian	1.00	1.93	2.96	7.12
Calculation of the derivatives	1.00	2.09	4.18	7.97
Overall speedup	1.00	1.70	2.50	3.43

calculations, 8 processors of the Cluster Intel PIII system), the execution time is only 14 h (compared to 4–10 days in the case of the sequential program). For small systems, relatively low efficiency can be attributed to the slowness of the diagonalization algorithm. On going to larger systems, this drawback gradually disappears. However, extensive computation with matrices of medium size ($N \leq 1200$) will require a new (compared to *PDSYEVX*) and more effective algorithm.

7. Concluding remarks

Appearance of powerful systems with massive parallelism and inexpensive parallel computers [62, 63] opens up access to highly productive calculations for a large range of users. However, wide application of these systems in science and technology is still restricted not only by high cost but also (and largely) by the complexity and labor cost of parallel programming.

Computational quantum chemistry which requires bulky calculations is a natural field for application of parallel computing. Parallel calculations have been reportedly involved in such well-known complexes as MOPACK [64], MNDO94 [65, 66], and GAMESS [60, 61, 67]. Nevertheless, there still remains a current need for new parallel programs for use in QC calculations. In this paper, we considered some aspects of parallel realization of the semi-empirical QC NDDO-WF method [19, 20] that allows calculations in the *spd*-basis. During this work, we managed to parallelize the following labor-consuming calculation procedures:

- diagonalization,
- calculation of the electron density matrix,
- construction of the Fock matrix,
- construction of the core Hamiltonian,
- calculation of the gradients of the total energy with respect to the Cartesian coordinates of atoms.

Our numerical calculations demonstrated the acceptable (from the practical point of view) effectiveness of the new program MP-ZAVA. This program makes possible the 'interactive' real-time calculations for the systems of practical interest. The experience assimilated in parallelization of CLUSTER-Z2 program allows us to formulate a general approach to designing the algorithmic systems with massive parallelism. It is based on the identification and repeated use of typical (frequently used) blocks. The effectiveness of this approach can be confirmed by the parallelization of codes in the CLUSTER-Z1 program that is now close to its successful completion. Repeated use of algorithmic structures with massive parallelizm (first utilized in MP-ZAVA) allowed us to significantly shorten the lead time for a new program package NANOPACK [68] that involves (besides NDDO-WF) also MNDO (AM1 and PM3) computational scheme.

Meanwhile, testing the MP-ZAVA program revealed some problems whose resolution will further extend the range of applicability for this program. One is related to the algorithm of diagonalization. As shown previously, the PDSYEVX procedure from the ScaLAPACK library is inefficient for the matrices of medium and low size. Moreover, this procedure exhibits poor scalability. Another problem is the need for a lower memory capacity and extension of the program to larger molecular systems. Its resolution is related to a refusal from the mechanism of replication (dubbing) of arrays and going to the parallezation scheme based on data decomposition. It will also need the replacement of the static memory distribution with the dynamic one.

Acknowledgements

The authors acknowledge greatly the assistance of Yu. B. Scheck in preparing the English version of the manuscript.

This work was supported by the Russian Foundation for Basic Research (project no. 99-07-90370).

References

[1] Clark T A 1985 *Handbook of Computational Chemistry*, Wiley, New York
[2] 1994 *Computational Chemistry: Molecules, Visualization, and Simulation*, Hypercube Inc., Waterloo, Canada
[3] Cagin T, Che J, Qi Y, Zhou Y, Demiralp E, Gao G and Goddard III W A 2000 *J. Nanopart. Res.* **1** 51
[4] Dewar M J S 1969 *The Molecular Orbital Theory of Organic Chemistry*, McGraw-Hill, New York
[5] Pople J A and Beveridge D L 1970 *Approximate Molecular Orbital Theory*, McGraw-Hill, New York
[6] Bromer K D, Needels M, Larson B E and Joannopoulos J D 1992 *Phys. Rev. Lett.* **68** 1355
[7] Hehre W J, Radom L, Schleyer P v R and Pople J A 1986 *Ab Initio Molecular Orbital Theory*, Wiley, New York
[8] Dewar M J S and Thiel W 1977 *J. Am. Chem. Soc.* **99** 4899
[9] Burstein K Y and Isaev A N 1984 *Theor. Chim. Acta* **64** 397
[10] Goldblum A 1987 *J. Comp. Chem.* **8** 835
[11] Thiel W 1981 *J. Am. Chem. Soc.* **103** 1413
[12] Dewar M J S, Zoebisch E G, Healey E F and Stewart J J P 1985 *J. Am. Chem. Soc.* **107** 3902
[13] Stewart J J P 1989 *J. Comp. Chem.* **10** 209, 221
[14] Dewar M J S, Jie C and Yu J 1993 *Tetrahedron* **49** 5003
[15] Holder A J, Dennington R D and Jie C 1984 *Tetrahedron* **50** 627
[16] 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
[17] Dewar Research Group and Stewart J J P 1986 *AMPAC: Austin Method I Package*, QCPE program 506; see also 1986 *QCPE Bull.* **6** 24
[18] Zayets V A 1990 *CLUSTER-Z1 Quantum-Chemical Software*, Institute of Surface Chemistry, Kiev
[19] Grebenjuk A G, Zayets V A and Gorlov Yu I 1993 *React. Kinet. Catal. Lett.* **50** 257
[20] Zayets V A, Gerda V I, Gorlov Yu I and Klimenko V E 1998 *Zh. Strukt. Khim.* **39** 163
[21] Thiel W and Vojtyuk A A 1996 *J. Phys. Chem.* **100** 616
[22] Zayets V A 1990 *CLUSTER-Z2 Quantum-Chemical Software (spd-Basis)*, Institute of Surface Chemistry, Kiev
[23] Thiel W 1994 *Program MNDO94*, Version 4.1, Zürich
[24] Khavryutchenko V D, Nikitina E A, Sheka E F, Barthel H and Weis J 1998 *Phys. Low-Dim. Struct.* **5/6** 1; 1999 **9/10** 1
[25] Nikitina E, Khavryutchenko V, Sheka E, Barthel H and Weis J 1997 *Surf. Rev. Lett.* **4** 879
[26] Nikitina E, Khavryutchenko V, Sheka E, Barthel H and Weis J 1999 *Composite Interf.* **6** 3
[27] Nikitina E A, Khavryutchenko V D, Sheka E F, Barthel H and Weis J 1999 *J. Phys. Chem.* **103A** 11355
[28] Sheka E, Natkaniec I, Khavryutchenko V, Nikitina E, Barthel H and Weis J 2000 *Physica* **276/278** 244
[29] Sheka E, Barthel H, Khavryutchenko V, Natkaniec I, Nikitina E and Weis J 2000 *Phys. Low-Dim. Struct.* **7/8** 127
[30] Snir M, Otto S W, Huss-Lederman S, Walker D and Dongarra J 1996 *MPI: The Complete Reference*, MIT Press, Boston
[31] Wilkinson J H and Reinsch C 1972 *Handbook for Atomic Computations: Linear Algebra,* Springer, Heidelberg
[32] Pople J A and Nesbet R K 1954 *J. Chem. Phys.* **22** 571
[33] Nemukhin A V and Granovslii A A 1999 *Proc. I All-Russia School on Supercomputing Systems and Information Technologies for Physics and Chemistry*, Nov. 1–2, Chernogolovka, p. 71
[34] Sielver D M 1971 *J. Math. Phys.* **12** 1937
[35] Sheka E, Khavrytchenko V and Nikitina E 1996 *J. Chem. Vap. Dep.* **5** 1; *Phys. Low-Dim. Struct.* **11/12** 65
[36] Khavryutchenko V, Sheka E, Huang D and Aono M 1965 *Phys. Low-Dim. Struct.* **1** 1

[37] Khavryutchenko V, Sheka E, Aono M and Huang D 1996 *Phys. Low-Dim. Struct.* **9/10** 15; 1996 **11/12** 1; 1998 **3/4** 81
[38] Sheka E F, Khavryutchenko V D and Markichev I 1995 *Russ. Chem. Rev.* **64** 389
[39] Zayets V A and Sheka E F to be published
[40] Harris F E 1969 *J. Chem. Phys.* **51** 4770
[41] Abramowitz M, Stegun I A Eds. 1964 *Handbook of Mathematical Functions with Formulae, Graphs and Mathematical Tables*, National Bureau of Standards Applied Mathematics Series, No. 55
[42] Zayets V A 1990 *Parametrization of Atomic Species in the NDDO-WF Approximation*, Kiev
[43] Saunders V R and Hillier I H 1973 *Int. J. Quant. Chem.* **4** 699
[44] Maslov V G 1979 *Zh. Strukt. Khim.* **20** 761
[45] Fletcher R 1972 *FORTRAN Subroutines for Minimization by Quasi-Newton Methods*, U.K. Atom Energy Auth. Res. Group, NAERE R7125, p. 29
[46] Broyden C G 1970 *J. Inst. Mathem. Appl.* **6** 222
[47] Fletcher R 1970 *Comput. J.* **13** 317
[48] Goldfarb D 1970 *Mathem. Computns.* **24** 23
[49] Shanno D F 1970 *Mathem. Computns.* **24** 647
[50] Shanno D F 1985 *J. Opt. Theor. Applns.* **46** 87
[51] Pilipenko A T, Zayets V A, Khavryuchenko V D and Falendysh E R 1987 *Zh. Strukt. Khim.* **28** 155
[52] Sheka E F, Khavryutchenko V D and Zayets V A 1995 *Phys. Low-Dim. Struct.* **2–3** 59
[53] Sheka E F, Khavryutchenko V D and Zayets V A 1996 *Int. J. Quant. Chem.* **57** 741
[54] Berzigiyarov P K 2000 *Proc. II All-Russia School on Supercomputing Systems and Information Technologies for Physics and Chemistry*, Nov. 1–2, Chernogolovka, p. 85
[55] Blackford L S *et al. ScaLAPACK Users' Guide*, available from: http//www.netlib.org/scalapack/slug/scalapack_slug.html
[56] Dongarra J J and Whaley R C *The BLACS v1.0 User's Guide*, available from: http//www.cs.utk.edu/~rwhaley/BLACS/Papers.html
[57] Ortega J M 1967 in *Articles in Mathematics for Digital Computers*, Ralston S, Wilf S Eds., Wiley, London, Vol. 2, p. 94
[58] Sheka E, Nikitina E, Zayets V, Schoonman J and Goossens A 2000 *Nanoparticles: Applications in Materials Science and Environmental Science & Engineering*, Proc. Third Joint ESF-NSF Symposium, Dublin, September 6, 2000, Fissan H, Ed., ESF, p. 35
[59] James D V, Laundrie A T, Gjessing S and Sohi G S 1990 *IEEE Computer* **23** 74; see also at: http://sunrise.scu.edu/
[60] Windus T L, Schmidt M W and Gordon M S 1995 *Toward Teraflop Computing and New Grand Challenge Applications*, Kalia R J, Vashishta P Eds., Nova Science Publishers, New York, p. 9.189
[61] Windus T L, Schmidt M W and Gordon M S 1995 *ACS Symposium Series on Parallel Computing in Chemistry* (# 592), Mattson T Ed.
[62] Strohmaier E, Dongara J J, Meuer H W and Simon H D 1999 *Parallel Compt.* **25** 1517
[63] Thole C-E and Stuben K 1999 *Parallel Compt.* **25** 2015
[64] Lin T-H, Haupt T and Fox G 1995 *Technical Report SCCS-744*, Syracuse University, NPAC, Syracuse NY
[65] Thiel W and Green D G 1995 in *Methods and Techniques in Computational Chemistry METECC-95,* Clementi E, Corongiu G, Eds., STEF, Cagliari, p. 141
[66] Thiel W 2000 in *Modern Methods and Algorithms of Quantum Chemistry,* Grotendorst J, Ed., NIC Series **1** 233
[67] Fletcher G D, Guest M F, Rendell A P and Sherwood P in *Molecular Modelling through Parallelism*, EUROPORT 2 Consortium, IMMP
[68] Berzigiyarov P K, Zayets V A, Razumov V F and Sheka E F 2000 *NANOPACK Quantum-Chemical Software for Parallel Computing*, Institute of Chemical Physics Problems, Russ. Acad. Sci.