

TBPAC 2007

USERS GUIDE

Mark A. Iron, Grażyna Staszewska, Tiqing Liu,
Ahren W. Jasper, and Donald G. Truhlar,
*Department of Chemistry and Supercomputing Institute,
University of Minnesota, 207 Pleasant Street S.E.,
Minneapolis, MN 55455-0431, USA.*

Computer code (version 2007) date: September 5, 2007

Manual Date: July 21, 2021

TBPAC is a complete package for tight binding calculations on molecules containing selected metal atoms (Al, Ni, Cu, Pd, Ag, Pt, and Au) plus carbon and hydrogen. It includes orthogonal and nonorthogonal Slater-Koster tight binding, penalty energies, orthogonal and nonorthogonal many-body tight binding, tight-binding with a Hubbard-like term (TB+U), and tight binding with configuration interaction (TBCI), but not self-consistent tight binding. None of the methods is supported for all possible combinations of the supported elements.

Table of Contents

1. User Agreement	1
2. Revision History	2
3. Citation.....	3
4. Overview.....	4
5. Usage.....	7
5.1. Installation and Compilation	10
5.2. List of Files.....	11
5.3. TBPAC – Stand Alone Version.....	11
6. Testing TBPAC.....	16
6.1. Test Suites	16
6.1.1. Aluminium Methods	16
6.1.2. Tight-Binding of Transition Metals and Hydrocarbons.....	17
6.2. Platforms Tested.....	17
7. References.....	18

1. User Agreement

TBPAC - version 2007 is licensed under the [Apache License, Version 2.0](#).

The manual of *TBPAC* - version 2007 is licensed under [CC-BY-4.0](#).

Publications of results obtained with the *TBPAC* - version 2007 software should cite the program and/or the article describing the program.

No guarantee is made that this software is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within. No consulting or maintenance services are guaranteed or implied.

The use of the *TBPAC* - version 2007 implies acceptance of the terms of the licenses.

2. Revision History

Versions 1.0,¹ 1.0.1,² and 2.0³ of TBPAC were called TB. Starting with version 2007, the code is renamed TBPAC.⁴ Version 1.0 contains Slater-Koster tight-binding (including gradients) for systems composed of H, C, and Ni, Pd, Pt, Cu, Ag, or Au.^{5,6} Version 1.0.1 was a minor bug fix. Version 2.0 contained many-body and Slater-Koster tight-binding for Al clusters and nanoparticles (energies only).^{7,8} Versions 1.0.1 and 2.0 both include the option for a penalty energy. TBPAC 2007 merges the capabilities of Versions 1.0.1 and 2.0 with the capabilities for tight-binding configuration interaction,⁹ nonorthogonal Slater-Koster TB,¹⁰ and nonorthogonal MBTB (MB-NOTB).¹⁰ It also includes analytical gradients for all methods. Note that at the current stage, the NOTB and MB-NOTB capabilities of TBPAC 2007 have not yet been validated.

3. Citation

Publications based on results obtained with this computer code should include the following reference:

M. A. Iron, G. Staszewska, T. Liu, A. W. Jasper, and D. G. Truhlar,
TBPAC 2007, University of Minnesota, Minneapolis, MN, 2007.

The reference(s) for the method(s) used should also be cited. These are:

- Staszewska, G.; Staszewski, P.; Schultz, N. E.; Truhlar, D. G., *Phys. Rev. B* **2005**, *71*, 045423; *erratum*, **2006**, *73*, 039903(E). **(TB and MBTB)**
- Liu, T.; A Tight Binding Model for the Energetics of Hydrocarbon Fragments on Metal Surfaces. Ph.D. Thesis, University of Minnesota, Minneapolis, MN, May 2000; Liu, A.; Truhlar, D. G., *unpublished results*. **(TB with transition metals)**
- Jasper, A. W.; Schultz, N. E.; Truhlar, D. G., *J. Chem. Theory Comput.* **2007**, *3*, 210-218. **(TB, MBTB, NOTB, and MB-NOTB)**
- Iron, M. A.; Heyden, A.; Staszewska, G.; Truhlar, D. G., *in preparation*. **(TBCI-S, TBCI-SPD, and TB+U)**

4. Overview

Tight-binding theory,¹¹ also called extended Hückel theory,¹² is a simplified form of semiempirical molecular orbital theory¹³ in which electron interactions are not treated explicitly. In the older literature, it was often considered¹⁴ to be an approximation to Hartree-Fock theory;¹⁵ in recent work it is often considered to be an approximation to Kohn-Sham¹⁶ or Harris-Foulkes^{17,18} density functional theory. The molecular orbitals, ψ_j , are linear combinations of atomic orbitals, ϕ_i , and a minimal basis set is usually employed:

$$\psi_j = \sum_i c_{ij} \phi_i \quad (1)$$

The coefficients satisfy a matrix eigenvalue equation, which may take one of two possible forms. Usually it is:

$$\mathbf{H} \mathbf{c} = \boldsymbol{\varepsilon} \mathbf{c} \quad (2)$$

where $\boldsymbol{\varepsilon}$ is a diagonal matrix of orbital eigenvalues ε_j , \mathbf{c} is a matrix whose columns are the eigenvectors of \mathbf{H} and whose meaning is defined by Eq. 1, and \mathbf{H} is called the Hamiltonian matrix; actually \mathbf{H} is a matrix representation of the approximate Fock operator or approximate Kohn-Sham or Harris-Foulkes operator that defines the tight-binding molecular orbitals.

In tight-binding theory, the matrix elements of \mathbf{H} are not calculated from the many-electron Hamiltonian; instead they are taken as simplified functional forms parameterized to reproduce experimental data or more complete calculations. In standard tight-binding theory, the diagonal matrix elements are constants, sometimes called valence-state ionization potentials (VSIPs), and the off-diagonal elements are two-center functions, that is, they depend only on the coordinates of the two nuclei on which the atomic orbitals are located. Sometimes the off-diagonal elements are proportional to overlap integrals; with proportionality constants depending on the diagonal elements; we call this the Wolfsberg-Helmholz approximation.¹⁹ The off-diagonal elements of \mathbf{H} are often called transfer or hopping integrals.

The Born-Oppenheimer energy (i.e., electronic energy plus nuclear repulsion²⁰) in standard tight-binding calculations is given by:

$$E = E_{\text{val}} + E_{\text{rep}} \quad (3)$$

where the first term is the valence energy given by

$$E_{\text{val}} = \sum_j n_j \varepsilon_j \quad (4)$$

where n_j is the occupancy of molecular orbital j ; and the second term is an effective repulsion energy and is typically written as a pairwise additive function of the internuclear distances as suggested by Foulkes and Haydock.¹⁸ E_{rep} will simply be called the repulsion energy. The valence energy is sometimes called the band energy.

Several possible extensions of the above simple formulation have been proposed, and it is useful to name some of them. The first is ***nonorthogonal tight-binding*** (NOTB) in which the eigenvalue equation (Eq. 2) is replaced by the following generalized eigenvalue equation:^{10,12,21}

$$\mathbf{H} \mathbf{c} = \varepsilon \mathbf{S} \mathbf{c} \quad (5)$$

where \mathbf{S} is the overlap matrix. The use of Eq. 2 is sometimes called orthogonal tight-binding, although we follow the common convention that “orthogonal” is understood when “nonorthogonal” is not specified. TBPAC²² includes both Eq. 2 and Eq. 5. Another program, called EHT,²³ is also available that solves Eq. 5.

Another extension is ***many-body tight binding*** (MBTB),⁷ also called environment-dependent tight binding.²⁴ In this method, the diagonal matrix elements of \mathbf{H} may depend on geometry, and the off-diagonal elements and repulsion energy may depend on the coordinates of more than two particles. TBPAC includes many-body tight-binding as well as the standard form, which we will call ***Slater-Koster tight binding***, because Slater and Koster¹¹ carried out their parameterization in a basis set assumed to have been pre-orthogonalized by the Löwdin method²⁵ (so that each orthogonal function still has a close resemblance to one of the original atomic orbitals), and they presented arguments approximately justifying the two-center approximation. One can also

combine NOTB and MBTB to obtain *many-body non-orthogonal tight-binding* (MB-NOTB). This is also available in TBPAC.

Another extension to tight-binding theory is to calculate partial atomic charges from the molecular orbitals (by Mulliken analysis,²⁶ perhaps neglecting overlap, which is sometimes called Coulson^{27,28} analysis) and add Hubbard-like terms^{29,30} or inter-site charge interaction terms or both to the Hamiltonian; such terms depend on the partial charges and/or atomic orbital occupation numbers. This requires an iterative self-consistent-field-like solution to the equations and is variously called iterative extended Hückel theory,^{13,31} the ω method,³²⁻³⁴ self-consistent Hückel theory,³⁵ self-consistent tight-binding,^{36,37} or self-consistent-charge density-functional tight-binding.³⁸ TBPAC does not include this extension.

Somewhat simpler than a Hubbard-like term is a *penalty energy* based on molecular orbital occupation numbers;³⁹ this does not require an iterative formulation. In this scheme, a penalty energy is added to the sum in Eq. 4 if the orbital is doubly occupied. TBPAC does allow for using penalty energies.

A final extension we will consider is *tight binding configuration interaction* (TBCI).⁹ In the TBCI method, we do not consider the tight-binding calculation to be an approximation to Kohn-Sham theory. Rather we consider it as a way to generate approximate adiabatic orbitals for a parameterized configuration interaction calculation that includes charge balance terms depending on the partial atomic charges of each configuration. The advantage of this scheme is that the calculation includes the effects of partial atomic charges but is noniterative. TBPAC includes TBCI.

5. Usage

TBPAC is designed primarily to be an external add-on to a molecular dynamics (MD) or Monte Carlo (MC) program. The MD or MC code calls subroutine `tbpac`, passing on a geometry (Cartesian coordinates) and a number of options, and the subroutine returns the energy and the gradients (if so requested). Two such subroutines are provided, depending on the format used for storing the geometry and the resulting gradients. To use TBPAC, one uncomments the relevant subroutine, makes the relevant calls from the external code, and recompiles. One may also modify this subroutine, but one should make that certain this routine initializes certain environmental variables (see provided subroutines). The format for the calls to the two subroutine `tbpac` are:

```
call tbpac(natom,siz,x,y,z,symbol,V,dx,dy,dz,grads,method,charge)
call tbpac(natom,siz,xxx,symbol,V,dV,grads,method,charge)
```

where:

- `natom`: the number of atoms in the system
- `siz`: the dimension of the arrays containing the Cartesian coordinates and the energy gradients
- `x`, `y` and `z`: double precision arrays of dimension `siz` that contain the x , y and z Cartesian coordinates, respectively
- `symbol`: `character*2` array of dimension `siz` that holds the atomic symbols (case insensitive) or atomic numbers (either may be used) of each atom in the system
- `xxx`: double precision array of dimension $3 * \text{siz}$ containing the Cartesian coordinates in the format $\mathbf{xxx} = \{x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_{\text{natom}}, y_{\text{natom}}, z_{\text{natom}}\}$
- `v`: double precision variable that returns the energy of the system
- `dx`, `dy` and `dz`: double precision arrays of dimension `siz` that return the gradients of the energy with respect to the x , y and z Cartesian coordinates of each atom

- `dV`: double precision array of dimension $3 \times \text{siz}$ that returns the energy gradients in a similar format as `xx`
- `grads`: integer that states whether or not the energy gradients are requested (0 = no, 1 = yes)
- `method`: integer that states which method is request (*vide infra*)
- `charge`: total charge of the system

In addition, a third option (`program tbc1`) is provided to run the code as a stand-alone program (see Section 5.3). This option reads all the necessary input from an input file and writes the results to an output file.

The variable `method` determines which method is used by TBPAC. This is a two digit number where the first digit (the “tens”) determines which method and the second digit (the “ones”) determines which parameter set (if multiple sets are available). Table I lists all the methods and the relevant values. The “Elements” column of Table I indicates which elements are supported by each parameter set.

Table I Methods implemented in TBPAC and the relevant `method` values.

Method	Parameter Set		Elements	method
TBCI-S			Al	11
TBCI-SPD			Al	21
TB+U			Al	31
TB (Slater-Koster)	WH	PRB ^a (0.07 eV ^c)	Al	41
		JCTC ^b (0.00 eV ^c)	Al	42
		JCTC ^b (0.04 eV ^c)	Al	43
	EWH	PRB ^a (0.07 eV ^c)	Al	44
	OWH	PRB ^a (0.07 eV ^c)	Al	45
	TM [‡]	Liu ^d	H, C, M	46
MBTB	S	PRB ^a (0.07 eV ^c)	Al	51
	SIP	JCTC ^b (0.00 eV ^c)	Al	52
	SR	JCTC ^b (0.00 eV ^c)	Al	53
	SOD	JCTC ^b (0.00 eV ^c)	Al	54
	CN	PRB ^a (0.07 eV ^c)	Al	55
	BA	PRB ^a (0.07 eV ^c)	Al	56
NOTB	WH	JCTC ^b (0.00 eV ^c)	Al	61
	WH	JCTC ^b (0.20 eV ^c)	Al	62
MB-NOTB	SIP	JCTC ^b (0.00 eV ^c)	Al	63
	SR	JCTC ^b (0.00 eV ^c)	Al	64
	SOD	JCTC ^b (0.00 eV ^c)	Al	65

^a From Reference 7.

^b From Reference 10.

^c Value of the penalty energy.

^d From References 5,6. M denotes a transition metal (specifically Ni, Cu, Pd, Ag, Pt and Au). At the present, only one type of metal atom may be present in a single calculation.

5.1. Installation and Compilation

TBPAC is provided as a compressed tar file called `tbpac2007.tar.gz`. One copies this file to the directory where one wants TBPAC to reside and runs the following command:

```
gunzip -v tbpac2007.tar.gz ; tar cvf tbpac.tar
```

This will result in the files and directories listed in Section 5.2 to be created.

When compiling TBPAC, there are two important issues to remember. First, although TBPAC is written in fixed (i.e., FORTRAN77) format, it includes a number of FORTRAN90 constructs, including modules and `select case`. Therefore, the compiler used must be able to handle these features. The Intel (`ifc` or `ifort`), the Portland Group (`pgf77` and `pgf90`), and the IBM (`xf`) compilers can all handle these features, although the gnu compiler (`g77` or `gfortran`) may not. Second, TBPAC uses external libraries (either `scs` or the `lapack/blas` combination, depending on the computer system) and must be linked to these libraries. (It is likely that MD or MC code that calls TBPAC already uses these libraries, amongst others.)

To install TBPAC, one simply adds to one's external MD, MC, or other code the appropriate calls to `subroutine tbpac` (see Section 5) and compiles the entire package. See the appropriate external code's manual for full details on how to handle external energy routines within that application. In order to compile, one will require two of the FORTRAN source files provided (Section 5.2): `tbpac.f` and, depending on the format of the storage of the Cartesian coordinates, either `tbpac-sub_xyz.f` or `tbpac-sub_xxx.f`. One should make sure that the options set at the beginning of `subroutine tbpac` (in one of the latter two files) are set correctly. Details on the options are provided in Section 5 and in the comments in the FORTRAN code.

The stand alone version (Section 5.3), which comprises two files – `tbpac.f` and `tbpac-program.f` – may be compiled using the provided script `comp.x`. This file contains the compilation command for the different compilers with which this code was tested (see Section

6.2). To compile, one simply selects the desired command (by deleting the # symbol at the beginning of the line) and then runs the compilation script. The executable file `tbpac` will result.

5.2. List of Files

TBPAC is provided as a compressed tar file (`tbpac2007.tar.gz`), which contains the following files:

- `tbpac.f`: the main part of TBPAC containing all of the subroutines
- `tbpac-program.f`: along with `tbpac.f`, this comprises the stand alone version of TBPAC (see Section 5.3)
- `tbpac-sub_xyz.f`: contains the version of subroutine `tbpac` to be used with an external code when the Cartesian coordinates are stored in three separate arrays `x`, `y` and `z`
- `tbpac-sub_xyz.f`: contains the version of subroutine `tbpac` to be used with an external code when the Cartesian coordinates are stored in three separate arrays `xxx`
- `comp.x`: the compilation script (see Section 5.1)
- `testtbpac.x`: a script that will run the test suite and compare the resulting output files to those provided (see Section 6)
- `testsuite`: a directory containing the test suite (see Section 6.1)
- `testoutput`: a directory containing the provided output files for the test suite for comparison (see Section 6.1)

5.3. TBPAC – Stand Alone Version

While TBPAC is primarily intended to be interfaced to another code, such as an MD or MC program, a stand-alone (SA) version can be compiled. In the SA version, all the options are

passed to the code via an input file. Note that this version also requires the first subroutine `tbpac` described in Sections 5 and 5.1. The syntax for running the SA version of TBPAC is:

```
tbpac input_file output_file
```

The input file contains all the necessary information to run TBPAC and the output is written to the output file. If no output file is give, `.out` will be affixed to the name of the input file. Note that the file names should be given as arguments to the `tbpac` command and **not** as redirection of the standard input and output FORTRAN units (i.e., do not use the `<` and `>` symbols in the command line as is common in many codes).

The syntax of each line in the input file (except `geometry`, *vide infra*) is:

```
option value
```

where the allowed options (many of which are the same as in Section 5) and their default values are:

- `title`: string containing a description of the job (default: blank field, maximum length 60 characters)
- `method`: which computational model to use (see Table I, default: 11)
- `grads`: whether (1) or not (0) the energy gradients are desired (default: 0)
- `print`: level of desired amount of information (see Table II) printed to output file (0=none, 1=minimal, 2=standard, 3=high, 4=debug, default: 2) (**Caution:** `print 0` results in absolutely NO output and is not recommended for use with the SA version. It is intended for use when TBPAC is used in conjunction with an external code.)
- `unitd`: which units of length to use (1=Ångstroms, 2=Bohrs (Å), default: 1)
- `unite`: which units of energy to use (1=eV, 2=kcal/mol, 3=kJ/mol, 4=cm⁻¹, 5=Hartree, default=1)
- `charge`: total charge of the system (default: 0)
- `cutMO`: cutoff for printing AO contributions to MOs; relevant only if `print` requires printing of the MO descriptions (default: 0.6).

- `cutCi`: cutoff for printing the CSFs in a TBCI calculation; relevant only if `print` requires printing of the CSF descriptions (default: 0.8).
- `cutEj`: cutoff for including E_j in the evaluation of the overall TBCI energy, where only those CSFs with weights $\left(|C_j|^2\right)$ greater than `cutEj` are included; for more details see TBCI paper⁹ (default: 10^{-8})
- `natom`: the number of atoms in the system (default: none, **mandatory option**)
- `geometry`: This directive (**mandatory**), which must be the last directive in the input file, has a slightly different construction. The keyword `geometry` indicates that on the following lines the Cartesian coordinates of each atom are provided. The `geometry` section terminates the input file. Each line in the geometry block contains, in order, the atomic symbol of the atom (case insensitive, NOT the atomic number!!) and the x -, y - and z -Cartesian coordinates of the atom in the units defined by `unitd`. If there are more than `natom` lines in the geometry section, only the first `natom` lines will be read.

An example input file for Al_3 with the default options specified explicitly is:

```

title Al3 sample file
method 11
grads 0
print 2
unitd 1
unite 1
charge 0
cutMO 0.6
cutCi 0.8
cutEj 1.0d-8
natom 3
geometry
Al      0.00000      0.00000      0.00000
Al      2.00000      0.00000      0.00000
Al      1.41421      1.41421      0.00000

```

Note that the order of the options is irrelevant (except that the `geometry` directive must be last) and that the last instance of an option will be used if the same option appears more than once. Any input the `geometry` directive and the geometry lines will be ignored. Note also that all keywords are case-insensitive and that the maximum length of any line is 80 characters. There should **not** be any spaces at the beginning of any line. Any line that cannot be understood by the program is simply ignored and the program writes a warning message that a line has been ignored.

Table II List of what is printed at each PRINT level

What?	PRINT	What?	PRINT
Parameters	1	Partial Atomic Charges	2
Input Geometry	2	MO Energies and Occupations	2
Method Description	2	MO Descriptions in AO Basis	3
Energy	1	Hamiltonian Matrix	4
Energy Components	2	Overlap Matrix	4
Energy Gradients	1	Reference & Dominant CSFs (TBCI)	3

6. Testing TBPAC

In order to test whether one's installation of TBPAC is functioning properly, a suite of test jobs and the resulting output files are provided. After installing TBPAC, the prospective user can run these jobs and compare the resultant output files with those provided. If all goes as expected, one should not find any differences between the two sets. For convenience, a C Shell script, `testtbpac.x`, is provided. This script will run all the test jobs, compare the obtained output files to those provided and report any differences to the file `testresults.txt`.

6.1. Test Suites

6.1.1. Aluminium Methods

There are 5 test jobs for each method. These are listed in Table III. In general, a number of different, small complexes are evaluated using a variety options. This set is repeated for each of the available methods in TBPAC (see Table I). The test suite, including the `testtbpac.x` script, is located in the `testsuite` directory. The provided output files are located in the `testoutput` directory. In both cases, the files are located in subdirectories `method11`, `method12`, etc.

Table III Contents of the test suite for aluminium.

File Name	Structure	Options			
		unite	unitd	print	Other
<code>al2re.in</code>	Al_2 at r_e	1	1	4	<code>grads=1 cutCi=0.1 cutMO=0.2</code>
<code>al3d3h.in</code>	Al_3 D_{3h}	2	2	4	<code>grads=1</code>
<code>al5.in</code>	Al_5	3	2	3	<code>grads=1</code>
<code>al7.in</code>	Al_7 (stretched O_h)	4	1	2	<code>grads=1</code>
<code>al177.in</code>	Al_{177}	5	1	2	<code>grads=1</code>

6.1.2. Tight-Binding of Transition Metals and Hydrocarbons

For the test suite for evaluating `method 46`, the tight-binding model for transition metals and hydrocarbons, the test suite from TB 1.0.1² was used, to which ethane was added. This is a set of nine jobs and is described in Table IV. The input files and example output files are located in `testsuite/method46` and `testoutput/method46` directories, respectively. The test suite is evaluated using the `testtbpac.x` script along with the tests in Section 6.1.1.

Table IV Contents of the Transition Metal-Hydrocarbon Test Suite.

File Name	System	File Name	System
<code>agh.in</code>	AgH	<code>au3plus.in</code>	Au ₃ ⁺
<code>auau.in</code>	Au ₂	<code>cuc.in</code>	CCu
<code>ni13minu.in</code>	Ni ₁₃ ⁻	<code>nicch.in</code>	NiC ₂ H
<code>nini.in</code>	Ni ₂	<code>pd2ch3.in</code>	Pd ₂ CH ₃
<code>ptch3.in</code>	PtCH ₃	<code>ethane.in</code>	C ₂ H ₈

6.2. Platforms Tested

TBPAC (stand alone version) has been tested on the following computer platforms (with the compilers and operating systems used in parentheses):

- IBM Power4 Regatta (XLF compiler, AIX 5.2)
- SGI Altix Linux Clusters
 - Intel Itanium 2 processors (ifort compiler, SGI Linux 3 with SGI Propack 3.4)
 - Intel Clovertown processors (ifort compiler, SuSE Enterprise Linux Server 10)
- IBM BladeCenter H*
 - AMD Opteron processors (ifort compiler, SuSE Enterprise Linux 9, Service Pack 3)
- IBM Netfinity Linux Cluster
 - Intel Pentium III and 4 processors (pgf90 compiler, Redhat Linux Enterprise 3)

7. References

1. Liu, T.; Truhlar, D. G. TB, Version 1.0; University of Minnesota: Minneapolis, MN, 2000.
2. Liu, T.; Truhlar, D. G. TB, Version 1.0.1; University of Minnesota: Minneapolis, MN, 2004.
3. Staszewska, G.; Liu, T.; Truhlar, D. G. TB, Version 2.0; University of Minnesota: Minneapolis, MN, 2003.
4. Iron, M. A.; Staszewska, G.; Liu, T.; Jasper, A. W.; Truhlar, D. G. TBPAC 2007, University of Minnesota: Minneapolis, MN, 2007.
5. Liu, T. A Tight Binding Model for the Energetics of Hydrocarbon Fragments on Metal Surfaces. Ph.D. Thesis, University of Minnesota, Minneapolis, MN, May 2000.
6. Liu, T.; Truhlar, D. G., *unpublished results*.
7. Staszewska, G.; Staszewski, P.; Schultz, N. E.; Truhlar, D. G., *Phys. Rev. B* **2005**, *71*, 045423.
8. Staszewska, G.; Staszewski, P.; Schultz, N. E.; Truhlar, D. G., *Phys. Rev. B* **2006**, *73*, 039903(E).
9. Iron, M. A.; Heyden, A.; Staszewska, G.; Truhlar, D. G., *J. Chem. Theory Comput.*, in preparation.
10. Jasper, A. W.; Schultz, N. E.; Truhlar, D. G., *J. Chem. Theory Comput.* **2007**, *3*, 210-218.
11. Slater, J. C.; Koster, G. F., *Phys. Rev.* **1954**, *94*, 1498-1524.
12. Hoffmann, R., *J. Chem. Phys.* **1963**, *39*, 1397-1412.
13. Zerner, M. C., *Rev. Comput. Chem.* **1991**, *2*, 313-365.
14. Blyholder, G.; Coulson, C. A., *Theor. Chim. Acta* **1968**, *10*, 316-324.
15. Roothaan, C. C., J., *Rev. Mod. Phys.* **1951**, *23*, 69-89.
16. Kohn, W.; Sham, L. J., *Phys. Rev.* **1965**, *140*, A1133-A1138.
17. Harris, J., *Phys. Rev. B* **1985**, *31*, 1770-1779.
18. Foulkes, W. M. C.; Haydock, R., *Phys. Rev. B* **1989**, *39*, 12520-12546.
19. Wolfsberg, M.; Helmholz, L., *J. Chem. Phys.* **1952**, *20*, 837-843.
20. Atkins, P.; Friedman, R., In *Molecular Quantum Mechanics*, 4 ed.; Oxford University Press: Oxford, 2005; pp 249-251.
21. Pettifor, D. G., *Phys. Rev. B* **1970**, *2*, 3031-3034.
22. Iron, M. A.; Staszewska, G.; Liu, T.; Jasper, A. W.; Truhlar, D. G. TBPAC, Version 3.0; University of Minnesota: Minneapolis, MN, 2006.
23. Liu, T.; Truhlar, D. G. ETH, Version 1.0; University of Minnesota: Minneapolis, MN, 2000.
24. Tang, M. S.; Wang, C. Z.; Chan, C. T.; Ho, K. M., *Phys. Rev. B* **1996**, *53*, 979-982.
25. Lödwin, P.-O., *J. Chem. Phys.* **1950**, *18*, 365-375.
26. Mulliken, R. S., *J. Chem. Phys.* **1955**, *23*, 1833-1840.
27. Coulson, C. A.; Longuet-Higgins, H. C., *Proc. R. Soc. London, Ser. A* **1947**, *191*, 39-60.
28. Maslen, V. W.; Coulson, C. A., *J. Chem. Soc.* **1957**, 4041-4049.
29. Hubbard, J., *Proc. R. Soc. London, Ser. A* **1963**, *276*, 238-257.

30. Lomba, E.; Molina, D.; Alvarez, M., *Phys. Rev. B* **2000**, *61*, 9314-9321.
31. Rein, R.; Fukuda, N.; Win, H.; Clarke, G. A., *J. Chem. Phys.* **1966**, *45*, 4743-4744.
32. Wheland, G. W.; Mann, D. E., *J. Chem. Phys.* **1949**, *17*, 264-268.
33. Streitwieser, A., *J. Am. Chem. Soc.* **1960**, *82*, 4123-4135.
34. Gupta, S. P.; Krishna, B., *J. Am. Chem. Soc.* **1970**, *92*, 7247-7248.
35. Harris, F. E., *J. Chem. Phys.* **1968**, *48*, 4027-4028.
36. Kohyama, M.; Yamamoto, R.; Ebata, Y.; Kinoshita, M., *Phys. Stat. Sol. (B)* **1989**, *152*, 533-541.
37. Andriotis, A. N.; Menon, M., *Phys. Rev. B* **1999**, *59*, 15924-45949.
38. Elstner, M.; Porezag, D.; Jungnickel, G.; Elsner, J.; Haugk, M.; Frauenheim, T.; Suhai, S.; Seifert, G., *Phys. Rev. B* **1998**, *58*, 7260-7268.
39. Wang, Y.; Mak, C. H., *Chem. Phys. Lett.* **1995**, *235*, 37-46.
40. Jones, H. W., *Int. J. Quant. Chem.* **1980**, *18*, 709-713.
41. Jones, H. W., *Int. J. Quant. Chem.* **1981**, *19*, 567-574.
42. Jones, H. W.; Weatherford, C. A., *Int. J. Quant. Chem. Symp.* **1978**, *12*, 483-488.
43. Lathiotakis, N. N.; Andriotis, A. N.; Menon, M.; Connolly, J., *J. Chem. Phys.* **1996**, *104*, 992-1003.
44. Dykstra, C. E.; Jasien, P. G., *Chem. Phys. Lett.* **1984**, *109*, 388-393.