Manual for MOPAC, version 5.022mn

MOPAC–version 5.022mn by James J. P. Stewart, Lucas J. Fiedler, Jingjing Zheng, Ivan Rossi, Wei-Ping Hu, Gillian C. Lynch, Yi-Ping Liu, Peng Zhang, Yao-Yuan Chuang, Jingzhi Pu, Jiabo Li, Patton L. Fast, Christopher J. Cramer, Jiali Gao, and Donald G. Truhlar, based on MOPAC 5.0 by James J. P. Stewart

Program revision date:* June 16, 2015
Date of most recent manual update: July 31, 2021

This version of MOPAC is based on MOPAC 5.0, but it has been extensively modified at the University of Minnesota (i) to be portable, (ii) to contain additional capabilities, and (iii) to be suitable for interfacing with molecular dynamics programs like POLYRATE. With regard to (i) the code is fully compliant with FORTRAN 77 except that it uses the INCLUDE extension, which is widely available, runs on supercomputers, Unix workstations, and PCs running Linux.

The scope of this manual is to document the modifications and additions to the original MOPAC 5.0 code. To learn how to use MOPAC, please refer to the original MOPAC manual, which is distributed as part of the MOPAC–MN 5.022 package.

The most important additional capabilities added to the code are: inclusion of PM3 (as in MOPAC 6); PDDG/PM3; PDDG/MNDO; RM1; PM6; the dispersion-corrected methods, AM1-D, PM3-D, MNDO-D, RM1-D, and PM6-D with an alternative damping function; flexible options for external modification of parameters (specific reaction parameters); the polarized molecular orbital methods, PMOv1, PMO2, and PMO2a; class IV charges by CM2A and CM2P; the eigenvector following (EF) algorithm for geometry optimization of equilibrium structures and saddle points; improved configuration interaction options; analytic gradients for the CI method; and the capability to perform successive calculations with different input files and different sets of parameters.

* The program revision date refers to the date of final modifications to all files except the manuals. The date of the most recent change to the manual is given separately. A change in only the manual does entail a new revision number.

J. J. P. Stewart, L. Fiedler, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, P. Zhang, Y.-Y. Chuang, J. Pu, J. Li, P. L. Fast, C. J. Cramer, J. Gao, and D. G. Truhlar

TABLE OF CONTENTS

1. LICENSING

MOPAC - version 5.022mn is licensed under the Apache License, Version 2.0.
The manual of *MOPAC* - version 5.022mn is licensed under CC-BY-4.0.
Publications of results obtained with the *MOPAC* - version 5.022mn 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 *MOPAC* - version 5.022mn implies acceptance of the terms of the licenses.

2. CITATION

Citation for MOPAC–version 5.022mn:

J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, J. Gao, and D. G. Truhlar, MOPAC–version 5.022mn (2015), University of Minnesota, Minneapolis, MN 55455.

3. REVISION HISTORY

As explained in the MOPAC-version 5.0 manual, the first two digits (e.g., 5.0) of the MOPAC version number are constant for a given release of the program. Modified versions for a given release of MOPAC should have the third digit different from zero; for example, a modification of version 5.0 would result in a version with the version number 5.0x, where x is not equal to zero. Clearly this can result in multiple versions of 5.0x in different places; therefore we have added mn (Minnesota) to versions created at the University of Minnesota. Note that starting with the release of version 5.015mn, the whole program name has been given as MOPAC–version 5.015mn rather than using the old system for naming. This document provides the revision history for the University of Minnesota revisions leading up to this version.

Version 5.0
Original distributed version by James J. P. Stewart.

Version 5.01mn
Modified version of MOPAC-version 5.0. The modifications were made by Minnesota Supercomputer Center Inc. for the Cray computers.

Version 5.02mn
Modification of version 5.01mn. In this version all the subprograms were returned to their original form, which used the include extension and the include file SIZES. This was done so that dimension changes would be much easier in MORATE. (In version 5.01mn the statements which made up the include file SIZES were explicitly included in all subprograms.)

Version 5.03mn
Modification of version 5.02mn. The block data file was modified to include all the AM1 and PM3 parameters available in MOPAC-version 6.0. Save statements were added to all the subprograms so that this version of the code does not have to be compiled with either the static or the ev option on the various Crays. The include file SIZES was renamed SIZES.i and all include statements in the FORTRAN subprograms were modified accordingly; this change was made for portability purposes.

Version 5.04mn
Modification of version 5.03mn. The subprograms AM1 and MOLDAT have been modified so that the semiempirical parameters for all of the Hamiltonians (MINDO/3, MNDO, AM1, and PM3) can be modified by using the keyword EXTERNAL (In version 5.0 only the MNDO and AM1 Hamiltonian could be modified).

Version 5.05mn (July 1994)
J. J. P. Stewart, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, and D. G. Truhlar
Modification of version 5.04mn. Several new capabilities have been added, and the code has been modified to make it portable and ANSI FORTRAN-77 compliant. A bug which caused the program to crash when parameter MAXORB <= NMECI**2 has been corrected. A bug in ANALYT which caused PM3 analytical derivatives to malfunction has been corrected. The C.I.=(n,m) option and the Eigenvector Following optimization algorithm have been introduced as new capabilities. Subroutines H1ELEC, ANALYT, MULLIKEN, and AM1 (now renamed EXTPAR) have been modified to use the

new NDDO-SRP pairwise parameters BETSS, BETSP and BETPP. Relevant to this new feature is the addition to the SIZES.i file of the new user-defined parameter MXATSP.

Version 5.06mn (September 1997)
J. J. P. Stewart, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, and D. G. Truhlar
Modification of version 5.05mn. DATA statements in the EXTPAR and ROTATE subroutines have been moved so that they are the last non-executable statements in the respective subroutine to adhere to standard Fortran 77. Dummy routines, date_dum.f and second_dum.f, and the makefile make.linux have been created to run MORATE under the Linux operating system.

Version 5.07mn (December 1997)
J. J. P. Stewart, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, and D. G. Truhlar
Modification of version 5.06mn. Files date_linux.f, second_liunx.f, second1.c are added to replace the dummy routines for Linux operating system. Some routines and makefiles for Sun, DEC, and HP workstations were added without testing.

Version 5.08mn (September 1999)
Modified by: Y.-Y. Chuang, J. Li, C. J. Cramer, and D. G. Truhlar
Authors: J. J. P. Stewart, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Li, C. J. Cramer, and D. G. Truhlar
Modification of version 5.07mn. New files PARAMS.i and chgmp2.f are added to include class IV atomic charges calculated by Charge Model 2 (CM2).

The subroutines CHRGE, WRTKEY, and WRITEMO are modified, and a new subroutine SCOPY is added to fromblas.f. Two new test runs test13.dat and test14.dat, are added for evaluating partial atomic charges by the CM2A and CM2P methods, respectively.

Version 5.09mn (October 1999)
Modified by: P. L. Fast and D. G. Truhlar
Authors: J. J. P. Stewart, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar
Modification of version 5.08mn. Tested make files have been provided for Compaq and Sun computers, and the subroutine WRTKEY has been modified to correct a floating point error on Compaq and Sun machines. The value of the parameter MAXDMP, defined in SIZEZ.i, was increased from 3600 to 36000.

Version 5.010mn (December 2003)
Modified by: J. Pu and D. G. Truhlar
Authors: J. J. P. Stewart, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar
Modification of version 5.09mn. The keyword HHON is now available for adding HH repulsive Gaussian functions into AM1 and PM3. Two new test runs, test15.dat and test16.dat, are added for optimizing the saddle point geometry for *CH3+CH4 using the AM1-CHC-SRP and PM3-CHC-SRP models, respectively.
Two makefiles "make.altix" and "make.linux_g77" are added into the mopac5010/src/ directory. The former is for compiling MOPAC 5.010mn on SGI Altix systems with the Intel compiler "efc", and

the latter is for compiling MOPAC 5.010mn under the Linux operating system with the GNU Fortran compiler "g77".

Version 5.011mn (August 2006)
Modified: J. Zheng and D. G. Truhlar
Authors: J. J. P. Stewart, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar

Modification of version 5.010mn. Three new semiempirical methods are implemented, namely PDDG/PM3, PDDG/MNDO, and RM1. The corresponding keywords are PDG (PDDG/PM3), MDG (PDDG/MNDO), and RM1 (RM1). Three new test runs, test17.dat, test18.dat, and test19.dat, are added to give examples of using the PDG, MDG, and RM1 keywords, respectively. Parameters for these new methods are available for the following elements:

PDDG/PM3:	H, C, N, O, F, Cl, Br, I
PDDG/MNDO:	H, C, N, O, F, Cl, Br, I
RM1:	H, C, N, O, P, S, F, Cl, Br, I

A bug in the recognition of keywords for compiling with optimization is fixed.

The compiler "ifort" replaces "efc" in makefile "make.altix" for compiling MOPAC 5.011mn on SGI Altix systems.

Version 5.012mn (November 2006)
Modified: J. Zheng and D. G. Truhlar
Authors: J. J. P. Stewart, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar

Modification of version 5.011mn. The PDDG/PM3 parameters for silicon, phosphorus, and sulfur are implemented.

Version 5.013mn (October 2009)
Modified: L. Fiedler and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar

Implemented three new methods, AM1-D, PM3-D, and PM6. AM1-D and PM3-D add dispersion corrections (see Section 4.8) and the following elements are supported by these methods:

AM1-D:	H, C, N, O, S
PM3-D:	H, C, N, O, S

PM6 is a successor to James Stewart's earlier method, PM3, that has been much more extensively parameterized (see Section 4.9). Because elements beyond magnesium require the inclusion of d-basis functions only the elements from hydrogen to magnesium are functional is this version.

Analytical gradients are now available for the two dispersion methods, AM1-D and PM3-D.

Version 5.013mn also splits the individual Fortran code files into individual files, one for each of the subroutines, functions, data blocks, etc., to replace the increasingly obsolete organization of program segments of separating those subroutines that were modified from those that were not modified from when MOPAC–version 5.01mn branched off from version 5.0. Version 5.013mn also contains new test runs that test single-point energy calculations with AM1-D and PM3-D.

Version 5.014mn (February 2010)
Modified: L. Fiedler and D. G. Truhlar

Authors: J. J. P. Stewart, L. Fiedler, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar

Implemented analytical gradients for the PM6 method and added a keyword for PM6 ("PM6G09") to reproduce Gaussian 09-type calculations for PM6 since it differs from the PM6 author's software, MOPAC2009, for carbon-carbon triple bonds in that the Gaussian version simply adds a constant 6.0 kcal/mole for each carbon-carbon triple bond, whereas MOPAC2009 uses the correct functional form given in the PM6 paper.

Version 5.015mn (April 2010)
Modified: L. Fiedler, P. Zhang, J. Gao, and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar

Added a new atomic symbol, "Hp", to the program that acts as a hydrogen that contains p orbitals. In effect, whenever Hp atoms are specified the Hp atoms are assigned atomic number 9 (the same as fluorine) and the program then overwrites the fluorine parameters. Additionally, all fluorine atoms are considered to be Hp atoms whenever any Hp atoms have been specified in the program input.

A new keyword, "DIPG09", has been added to MOPAC in order to output both the standard MOPAC-calculated dipole moment (with origin set as the center of isotopic masses) and the Gaussian 09-type of dipole moment calculation (with origin set as the center of nuclear charges). This difference is only important for charged systems as the dipole moment of neutral systems will be invariant to the coordinate system chosen. See section 4.12 for details.

Three new methods, MNDO-D, RM1-D, and PM6-D, have been added that follow the same formalism as AM1-D and PM3-D. These new methods do not occur in the literature so the dispersion parameters (R^{vdw}, C_6, S_6, and α_{disp}) and the U_{ss}, U_{pp}, β_s, β_p, and α parameters chosen for them are set to equal the corresponding parameters of AM1-D (for the new RM1-D method) and PM3-D (for the new MNDO-D and PM6-D methods). The elements for which these new methods have parameters are:

MNDO-D: H, C, N, O, S
RM1-D: H, C, N, O, S
PM6-D: H, C, N, O, S

An alternative damping function is now available for use with the dispersion energy correction of the all the "-D" dispersion methods. The keyword "SDAMP" will cause this damping function to be used. See section 4.13 for details.

Version 5.016mn (July 2010)
Modified: L. Fiedler and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, and D. G. Truhlar

This version of MOPAC has taken the previous version of MOPAC and made the entire program into a subroutine that may be called repeatedly within a larger program framework. The advantage of this modification is that version 5.016mn may be used for direct dynamics or for parameter optimization on large sets of test molecules without having to resort to repeatedly launching it as a stand-alone program. Additionally it is now possible to supress all input and output from the program by setting the common block logical variable DOPRNT to .FALSE. (by default it is set to .TRUE.).

The savings in time from avoiding repeatedly launching MOPAC and avoiding all the I/O (input/output) statements is quite substantial. This is one of the principal advantages of this version,

especially when carrying out direct dynamics, conducting parameter optimizations, or doing calculations over large sets of molecules.

The main changes to MOPAC–5.016mn are to the following program units:

File name	Details
mopac.f	Previously the main program unit, now a subroutine called by MSTART.
mstart.f	The new main program unit where the program execution begins.
doinit.f	Reinitializes all subroutine and function local variables and calls RESETC and RESETD.
resetc.f	Reinitializes all common block variables to values given in BLOCK DATA.
resetd.f	Reinitializes all other common block variables.
setntx.f	Uses a virtual input file for new input deck and geometry/symmetry specification.

It is important that if any changes are made to the block_data.f file that corresponding changes are also made to the resetc.f file.

The SETNTX subroutine can be called to use a virtual input file if the global boolean variable LINTXT is set to true. This circumvents the opening and closing of a new unit for file input/output operations and replaces this with input provided by a programmer within the SETNTX subroutine. Ideally this would be used for the purpose of reparameterization of a method or when invoking MOPAC for successive semiempirical calculations where terminating and relaunching the program would be highly inefficient. Extensive discussion of the SETNTX subroutine and the usage of its specific common block INTEXT variables for setting up a new MOPAC calculation is provided for developers within the comment section at the beginning of the SETNTX subroutine.

Without making any changes to the code MOPAC–5.016mn will continue to perform identically and give the same results as past versions. The launching point for the MOPAC subroutine (formerly the main program unit) is from the MSTART subroutine. To reinitialize all variables for a new run simply call the DOINIT subroutine. DOINIT reinitializes all common block variables and subroutine local variables that require resetting in order to call the MOPAC subroutine again.

Because STOP statements are occasionally encountered during calculations that violate common sense or represent unphysical situations, the program would terminate in older versions. In the current version alternate return statements are included to return the program to the top level (the MSTART subroutine) in order to handle these events and avoid early program termination if a set of MOPAC calculations is being carried out.

The ANALYT keyword previously would create a scratch directory for holding analytical derivative information. To avoid the creation and deletion of a new scratch file, all the file I/O it would involve, and the subsequent increase in time, a new array was created for scratch space.

During the development of this version, run-time checking was used on all of the test runs to ensure no run-time problems existed within the code. Examples of these run-time problems that were corrected include writing to arrays out-of-bounds, using invalid pointers, and using undefined variables. Many of these run-time problems existed within the code prior to version 5.016mn, and each was corrected when found during development.

Finally, the Cray workstation-specific files have been removed from this distribution of MOPAC, because we did not have an opportunity to test recent versions of MOPAC on Cray computers.

Version 5.017mn (October 2010)
Modified: L. Fiedler, P. Zhang, J. Gao, and D. G. Truhlar

Authors: J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, J. Gao, and D. G. Truhlar

This version of MOPAC has made several modifications to the semiempirical theory in order to accommodate Hp-type hydrogen atoms (hydrogens containing p-orbitals). Four modifications are made to the theory and are implemented by the respective keywords "MOD1", "MOD2", "MOD3", and "MOD4" (these MOD options are described in Section 4.15). The ability to use custom parameters is also included via the keyword "PMODS" in an external parameter file (see Sections 4.14 and 4.15).

The new semiempirical method PMO version 1 can now be used in this version of the program. It is activated by using the keyword "PMOv1" for the method name in the input file. A description of PMOv1 is in Section 4.16.

Version 5.018mn (March 2011)
Modified: L. Fiedler and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, J. Gao, and D. G. Truhlar

This version of MOPAC has small changes to the input and output files. The limitation on the number of text characters allowed in the input deck line has been expanded from 80 characters to 160 characters.

The output states for clarity that Koopmans' theorem is invoked in calculating the ionization potential that appears in the summary section.

The new keywords MOD5, MOD6, and MOD7 are implemented in this version of the program (see Section 4.15 for descriptions of each of these new modifications). They provide greater flexibility in parameterizing future semiempirical methods over previous semiempirical formalisms.

A Perl script (check.pl) is automatically invoked when running the test suite that will confirm the expected results of all test calculations and signal the user when one of the expected results is not produced. In the future, this will streamline the process of ensuring no inadvertent errors slip into the code while new features are being added to MOPAC.

Version 5.019mn (November 2011)
Modified: L. Fiedler and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, J. Gao, and D. G. Truhlar

The diatomic parameter arrays were changed from compact one-dimensional arrays to two- or four-dimensional arrays in order to simplify the program's code. These changes apply to the pairwise beta and k parameters for resonance integrals (using the MOD7 keyword; see Section 4.15) and the pairwise core-core energy parameters (using the MOD5 keyword; see Section 4.15). The external parameter file formats are not affected by this change.

Added new keyword "HYBRID=*x.xxx*" (here *x.xxx* is a value which should be between 0 and 1; it is 1 by default) which alters the total dipole moment calculation by specifying what fraction of the hybrid dipole calculation contributes to the total dipole moment reported. All three dipole calculations are still given in the output: Mulliken charge dipole moment, hybrid dipole moment, and total dipole moment.

Grimme's DFT-D3 dispersion[27] is now included for use with some of the semiempirical methods. The parameters for PBEsol were chosen as the best set of D3 parameters for use with the various semiempirical methods. New methods that include the D3 dispersion are MNDO-D3, AM1-D3, PM3-D3, RM1-D3, and PM6-D3. Two D3 dispersion parameters are also capable of being modified

with an external parameter file to select different types of DFT-D3 dispersion to use with the listed "-D3" semiempirical methods. For more information on damping, see Section 4.13.

Version 5.020mn (April 2013)
Modified: L. Fiedler and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, J. Gao, and D. G. Truhlar

The PMO2 semiemperical method[28] has been included as a built-in keyword in this version of MOPAC. It makes use of the newer MOD*x* options in MOPAC for pairwise element parameters and uses the "Hp" hydrogens like its predecessor, PMOv1.

Version 5.021mn (June 2014)
Modified: L. Fiedler and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, P. Zhang, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Pu, J. Li, C. J. Cramer, P. L. Fast, J. Gao, and D. G. Truhlar

A new atmospheric nucleation method[29] called PMO2a has been implemented as a built-in keyword "PMO2a". It makes use of the "Hp" hydrogen atoms, which contain p orbitals, and is intended for atmospheric systems containing hydrogen, carbon, nitrogen, oxygen, and sulfur (using other elements will not compute).

The programming code was also cleaned up in many places between this version and the previous version. Many debugging output statements were removed that were still in the code during various phases of development. The array structure was also improved for some parameters in order to switch from a packed array to a standard unpacked array that is more in line with the greater memory available on modern computer systems. The name for the AM-1 core-core Gaussians has been changed from GUESS to GAUSS everywhere in the code. Additional commenting was placed in specific parts of the code to make it more understandable.

Version 5.022mn (June 2015)
Modified: J. Zheng and D. G. Truhlar
Authors: J. J. P. Stewart, L. Fiedler, J. Zheng, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, P. Zhang, Y.-Y. Chuang, J. Pu, J. Li, P. L. Fast, C. J. Cramer, J. Gao, and D. G. Truhlar

Analytic gradients have for CI calculations been implemented using the method of M. J. S. Dewar and D. A. Liotard [30]. The code is mainly taken from AMSOL 7.1, and some modifications are made to accommodate the current MOPAC program. When the CI option is turned on, geometry optimization in the current version uses the CI gradient, whereas they used the SCF gradient SCF gradient in the previous versions.

4. NEW CAPABILITIES

4.1 New Configuration Interaction Keyword

C.I.=(n,m)

In addition to specifying the number of molecular orbitals in the active space, the number of electrons can also be defined. In C.I.=(n,m), n is the number of M.O.s in the active space, and m is the number of doubly filled orbitals to be included in the active space.

Examples:

Keywords	Number of M.O.s	No. of Electrons
C.I.=2	2	2 (1)
C.I.=(2,1)	2	2 (3)
C.I.=(3,1)	3	2 (3)
C.I.=(3,2)	3	4 (5)
C.I.=(3,0) OPEN(2,3)	3	2 (N/A)
C.I.=(3,1) OPEN(2,2)	3	4 (N/A)
C.I.=(3,1) OPEN(1,2)	3	N/A (3)

Values for odd-electron systems are given in parentheses, following the number for an even number of electrons. N/A denotes not applicable.

This modification is based on the MOPAC 6.0 code.

4.2 The Eigenvector Following Method for Optimization

The Eigenvector Following code included in MOPAC 5.05mn and later is based on code developed by:

Professor Frank Jensen
Department of Chemistry
Aarhus University
8000 Aarhus C
Denmark

and is taken from the MOPAC 7 code. The authors are grateful to Professor Jensen for permission to include this code in the package.

4.2.1 Description of the Algorithm

The EF optimization routine used here is a combination of the original EF algorithm of Simons et al[1] as implemented by Baker[2] and the QA algorithm of Culot et al[3], with some added features (see RMIN, RMAX, and OMIN described below) for improving stability.

The geometry optimization is based on a second order Taylor expansion of the energy around the current point. At this point the energy, the gradient, and some estimate of the Hessian are available. There are three fundamental operations in determining the next geometry based on this information:

o find the ``best'' step within or on the hypersphere with the current trust radius.
o possibly reject this step based on various criteria.
o update the trust radius.

For a minimum-energy search the correct Hessian has only 3N-6 positive eigenvalues, where N is the number of atoms. For a Transition State (TS) search the correct Hessian should have exactly one negative eigenvalue, and the corresponding eigenvector should be in the direction of the desired reaction coordinate.

The geometry step is parameterized as g/(s-H), where s is a shift factor which ensures that the step length is within or on the hypersphere. If the Hessian has the correct structure, a pure Newton-Raphson step is attempted. This corresponds to setting the shift factor to zero. If this step is longer than the trust radius, a P-RFO step is attempted. If this is also too long, then the best step on the hypersphere is made via the QA formula.

Using the step determined, the new energy and gradient are evaluated. If it is a TS search, two criteria are used in determining whether the step is "appropriate". The ratio between the actual and predicted energy change should ideally be 1. If it deviates substantially from this value, the second order Taylor expansion is no longer accurate. If the ratio is outside the interval defined by the RMIN and RMAX limits, the step is rejected, the trust radius reduced by a factor of two and a new step is determined. The second criteria is that the eigenvector along which the energy is being maximized should not change substantially between iterations. The minimum overlap of the TS eigenvector with that of the previous iteration should be larger than OMIN, otherwise the step is rejected.

4.2.2 New Eigenvector Following Keywords

EF
The Eigenvector Following routine is an alternative to the default BFGS algorithm. This keyword invokes the Eigenvector Following routine to optimize to a minimum-energy structure. EF is particularly efficient for refining structures, when the gradient is already small.

TS
With the TS keyword, the Eigenvector Following routine is invoked to optimize to a transition state. The TS method is much faster and more reliable than either SADDLE or NLLSQ.
TS appears to work well even with Cartesian coordinates.

MODE=n
MODE=n specifies that the nth Hessian eigenvector will be followed in the first step of an Eigenvector Following optimization. MODE=1 means the eigenvector with the lowest eigenvalue, MODE=2 the second lower, and so on. Note that the eigenvectors corresponding to translational and rotational motion, which have zero eigenvalue, are projected out of the Hessian and automatically renumbered as the last six eigenvectors. The next steps will be selected on the basis of the overlap between two following steps (see OMIN). If MODE=0 the eigenvector with the lowest eigenvalue will be followed, regardless of the overlap with the previous optimization step.

The defaults are MODE=1 (TS keyword) and MODE=0 (EF keyword).

HESS=n
    HESS=n specifies how the Hessian matrix will be calculated.

HESS=0 The initial Hessian will be approximated as diagonal
HESS=1 Calculate the Hessian using forward finite differences
HESS=2 Read Hessian from disk
HESS=3 Calculate the Hessian using central finite differences

The default is HESS=0 for minimum optimization (EF keyword) and HESS=1 for transition state optimization (TS keyword).

RECALC=n
    In an Eigenvector Following optimization, the RECALC=n keyword requests the program to recalculate the Hessian every n iterations. This is very effective but CPU-intensive.
    The Hessian will be recalculated using the method specified by HESS keyword.

DMAX=n
    DMAX=n changes the value of the starting trust radius (in Angstroms) in Eigenvector Following optimizations. The default is DMAX=0.2

DDMIN=n
DDMAX=m
   These keywords set the limits for trust radius (in Angstroms) in Eigenvector Following optimizations. The defaults are DDMIN=0.001 and DDMAX=0.3 (TS keyword) or DDMAX=0.5 (EF keyword).

RMIN=n
RMAX=m
    For an Eigenvector Following step to be accepted, the value of the ratio of the calculated energy to the predicted energy must be bracketed by the values of RMIN and RMAX. Default values are RMIN=0 and RMAX=4.

OMIN=n
    During transition state optimizations, the algorithm calculates the dot product between the previously followed direction and the Hessian eigenvectors. The new step will be along the direction defined by the eigenvector for which this dot product is maximum, if this value is greater than OMIN. The default is OMIN=0.8 (TS keyword) and OMIN=0.0 (EF keyword). See also keyword MODE.

IUPD=n
    IUPD=n selects the Hessian updating scheme in Eigenvector Following optimizations.

 IUPD=0   No updating
 IUPD=1   Powell updating scheme[4]
 IUPD=2   BFGS updating scheme[5]

The defaults are IUPD=1 for transition state search (TS keyword) and IUPD=2 for a minimum-energy search (EF keyword).

RSCAL
    RSCAL scales the Eigenvector Following step to trust radius instead of using QA formula. The default is to use QA formula for scaling.

NONR
    Skip Newton-Raphson step in Eigenvector Following optimizations. See Section 2.2.1.

4.2.3 How to Fine Tune Optimizations

RMIN and RMAX:
        The acceptance criterion for the optimization step is that the ratio of the calculated energy to the predicted energy should be larger than RMIN and lower than RMAX. If this ratio is outside this interval, the step is rejected, the trust radius reduced by a factor of two and a new step is predicted.

        Setting RMIN and RMAX close to one will give a very stable, but also very slow, optimization. Wide limits on RMIN and RMAX may in some cases give a faster convergence, but there is always the risk that very poor steps are accepted, causing the optimization to diverge. The use of the default values of 0 and 4 lead to a rare rejection of steps. This leads to a faster convergence, but occasionally leads to the acceptance of poor steps. If TS searches are found to cause problems, the first try should be to change the limits of RMIN and RMAX, narrowing the interval to 0.5 and 2, for example. Tighter limits like 0.8 and 1.2, or even 0.9 and 1.1, will almost always slow the optimization down significantly, but they may be necessary in some cases.

OMIN:
        OMIN has been designed for ensuring that the nature of the TS mode only changes gradually, specifically the overlap between two successive geometrical displacement should be higher than OMIN. While this technique at first appears very promising, it may cause problems when the Hessian is updated. As the updated Hessian in each step is only approximately correct, there is an upper limit on how large the TS mode overlap between steps can be. To understand this, consider a series of steps made from the same geometry (e.g. at some point in the optimization), but with steadily smaller step sizes. The update adds corrections to the Hessian to make it a better approximation to the exact Hessian. As the step size becomes small, the updated Hessian converges toward the exact Hessian, at least in the direction of the step. The overlap between TS modes does not converge toward 1, but rather to a constant value which indicate how good a guess the first approximate Hessian was to the exact Hessian. It appears that an updated Hessian in general is not of sufficient accuracy for reliably rejecting steps with TS overlaps much greater than 0.80. The default OMIN of 0.80 reflects the typical use of an updated Hessian and allows fairly large changes to occur, and should be suitable for most uncomplicated systems.
    If problems are encountered with many step rejections due to small TS mode overlaps, try reducing OMIN, maybe all the way down to 0. This most likely will work if the TS mode is the lowest Hessian eigenvector, but it is doubtful that it will produce any useful results if a high lying mode is followed.
        Note that the only way to turn off the step rejection criteria is to give suitable values to RMIN, RMAX, and OMIN, e.g. the choice of RMIN=-100 and RMAX=100 effectively inhibits step rejection.

Similarly setting OMIN=0 disables step rejection based on large changes in the structure of the TS mode.

MODE:

The algorithm has the capability of following Hessian eigenvectors other than the one with the lowest eigenvalue (see MODE keyword explanation) toward a TS, using the keyword MODE. It is always more difficult to make such higher-mode following. Ideally, as the optimization progresses, the TS mode should at some point become the lowest eigenvector. Care must be taken during the optimization, however, that the nature of the mode does not change suddenly, leading to optimization to a different TS than the one desired.

Note that during TS optimizations, the default value MODE=1 means that mode following is active (See OMIN and MODE keyword explanation). This means that the TS MODE=1 will be followed, and in some cases this may eventually change to some higher mode, causing the optimization to fail. To turn off mode following, and thus following at every step the mode with lowest eigenvalue, set MODE=0.

Remember that following modes other than the one with the lowest eigenvalue toward a transition state indicates that the starting geometry is not a good guess of the transition state one. In most cases it is better to further refine the starting geometry, than to try following high-lying modes. There are cases, however, where it is very difficult to locate a starting geometry which has the desired Hessian, and higher-mode following may be useful. Otherwise, if RECALC=1 the TS mode overlap does converge toward 1 as the step size goes toward zero, and in this cases there is no problems following high lying modes.

HESS and RECALC:

In certain very rigid systems or in some transition state optimizations, the initial default Hessian may result to be too approximate. In this case the algorithm cannot find an acceptable step larger than DDMIN, so the optimization terminates after only a few cycles with the "TRUST RADIUS BELOW (DDMIN value)" warning long before the stationary point is reached. In such cases RMIN could be set to some negative value, thereby allowing steps which increase the energy. But in general it is more useful to increase the precision of the Hessian calculation using the HESS keyword.

In some very difficult cases it is even necessary to recalculate the Hessian every few iterations, using the RECALC keyword. Unfortunately setting RECALC to low values is very expensive in terms of computer time, but if used in conjecture with OMIN=0.90 (or possibly an even higher value) and maybe with tighter values of RMIN and RMAX, it represents an option for locating transition structures that otherwise might not be possible.

4.3 NDDO-SRP Calculations

Neglect-of-diatomic-differential-overlap (NDDO) molecular orbital theory with specific reaction parameters (SRP)[18,19] can be invoked in this program in the following way. An NDDO-SRP calculation, which can be used in conjunction with all methods except MINDO, can be invoked by using the MOPAC keyword EXTERNAL=filename. The parameters adjusted specifically for the given reaction should be defined in the file given in EXTERNAL=filename according to the discussion in the MOPAC 5.0 manual.

In order to allow the wavefunction to be more flexible, three new parameters, BETSS, BETSP, and BETPP, have been defined and are recognized in an NDDO-SRP calculation. In standard calculations the so called one-electron resonance integral of NDDO theory is calculated as the overlap integral of the two atomic orbitals times a scaling factor, which is defined as the mean of two atomic BETA parameters. In MOPAC–version 5.022mn, however, every scaling factor can be specified separately for every pair of atomic species[6], specifying the BETxy parameters in the EXTERNAL file, as:

BETxy *atom1 atom2 value*

where:

x = type of orbital of atom type atom1 (only S-type and P-type orbitals available)
y = type of orbital of atom type atom2 (only S-type and P-type orbitals available)
atom1 = chemical symbol
atom2 = chemical symbol

Example:

```
BETSS    C    Br      -17.557832
BETPP    C    Br       -8.338239
BETSP    Br   C       -13.559582
BETSP    C    Br      -12.336489
```

Note that BETSP C Br is a different parameter from BETSP Br C. The value of BETxy should be in eV units. Avoid tab characters in the external parameter file. Also be certain that the external parameter file name is saved with all upper-case characters and the file itself terminates with a blank line or the word "END".

These parameters must be specified in the EXTERNAL file after all the other parameters. See test run 10 for an example, and in particular see file TEST10.SRP in the mopac5022/test directory.

Setting BETSS for C and Br to the average of the standard BETAS for C and Br is equivalent to doing nothing.

4.4 Charge Model 2 (CM2)

The keyword CM2 specifies that Charge Model 2 (CM2) [7] should be used. CM2 parameterizations are available in MOPAC for either the AM1 or PM3 Hamiltonian. CM2 with AM1 is called CM2A or CM2/AM1; CM2 with PM3 is called either CM2P or CM2/PM3. When the CM2 keyword is chosen in addition to the AM1 or PM3 Hamiltonian, the Charge Model 2 charges and the dipole moment calculated from the CM2 charges will be be printed. CM2 results will replace the Mulliken charges in the FOR12 file with MOPAC input file format.

CM2 is parameterized for H, C, N, O, F, Si, P, S, Cl, Br, and I.

4.5 HH repulsive Gaussians for AM1 and PM3

The keyword HHON specifies that additional Gaussian repulsion terms are added for all hydrogen pairs[8]. Analytical gradients are available with HHON, and Hessians can be calculated by numerical differentiation of the analytical gradients.

It is well known that the PM3 Gaussian functions introduce an unphysical stabilization for HH pairs at a distance of ~1.8 Å. The performance of PM3 can be improved by adding repulsive HH Gaussian terms of the form:

$$E(HH) = A*Exp\{-[(rHH-r0)/lambda]**2\}$$

where A is the amplitude of the Gaussian funciton; rHH is the distance between a pair of hydrogens; and r0 and lambda denote the location and the width of the Gaussian function. A similar idea can be used for improving AM1. This option is especially useful for calculating reaction barrier heights and geometries of hydrogen atom transfer reactions between two hydrocarbon fragments, where a set of Gaussian parameters for AM1 and PM3, called "AM1-CHC-SRP" and "PM3-CHC-SRP" respectively, are available.

To activate the HHON option, the user needs to specify the keyword "HHON" in the MOPAC 5.022mn input file. With "HHON" turned on, the AM1-CHC-SRP and PM3-CHC-SRP HH Gaussian parameters will be used as defaults for AM1 and PM3, respectively. Besides using the default parameters, user-defined HH repulsive Gaussians can also be read from an external parameter file, where the keyword "HHON=filename" should be specified in the MOPAC–5.022mn input file. The string "filename" following the "HHON=" specifies the name of the parameter file containing Gaussian function parameters for hydrogen pairs.

The user needs to create the parameter file in the working directory for MOPAC–5.022mn. Up to three Gaussian functions can be specified in the HH Gaussian parameter file with the following syntax:

 ngauss
 A1 r0_1 lambda1
 [A2 r0_2 lambda2]
 [A3 r0_3 lambda3]

where ngauss is an integer between 1 and 3 that specifies the number of Gaussian functions that are used; and A, r, and lambda denote the amplitude (in kcal/mol), the location (in Å), and the width (in Å) of each Gaussian function added for all hydrogen pairs. The second and third Gaussians in the bracket are optional. All Gaussian parameters should be floating point numbers.

4.6 PDDG/PM3 and PDDG/MNDO

The PDDG/PM3 and PDDG/MNDO semiempirical methods[9, 10, 11] use a Pairwise Distance Directed Gaussian (PDDG) modification of the PM3 and MNDO methods, respectively. These methods were implemented in MOPAC 5.012mn. The PDDG modifications consist of a single function, which is added to the existing pairwise core repulsion functions within PM3 and MNDO, a reparameterized semiempirical parameter set, and modified computation of the energy of formation of a gaseous atom. The keywords for the PDDG/PM3 and PDDG/MNDO methods are PDG and MDG, respectively. Both PDDG methods are parameterized for H, C, N, O, F, Cl, Br, and I.

4.7 RM1

RM1[12] (Recife Model 1) is a reparameterization of AM1 and was implemented in MOPAC 5.012mn. Unlike AM1, and similarly to PM3, all RM1 parameters have been optimized. Therefore, RM1 has been designed with the goal of improving over AM1 and PM3. RM1 has the same analytic form and the same number of parameters for each atom as AM1. The keyword RM1 is used for this method. RM1 is parameterized for H, C, N, O, P, S, F, Cl, Br, and I.

4.8 AM1-D, PM3-D, MNDO-D, RM1-D, and PM6-D

The AM1-D and PM3-D semiempirical methods developed by Hillier with his parameterization[13] with the dispersion functional form taken from Grimme[14] are modifications of the respective AM1 and PM3 methods. The "-D" dispersion methods described in this section use Grimme's version 1 dispersion functional form that also uses the suffix "-D1" to distinguish it from later versions (note that "-D" and not "-D1" are used for method names in MOPAC). Some of the original semiempirical method parameters have been modified[13] (for instance, USS and UPP) but most importantly additional core repulsion functions with a long-range R^{-6} form are included to give the method better dispersion energies for noncovalently bonded complexes.

The AM1-D and PM3-D methods currently are parameterized for the following elements: H, C, N, O, F, Ne, and S.

The MNDO-D, RM1-D, and PM6-D methods were not developed by Hillier and coworkers at the time that AM1-D and PM3-D were developed; however, they are implemented in this version using the same dispersion parameters (R^{vdw}, C_6, S_6, and α_{disp}) and U_{ss}, U_{pp}, β_s, β_p, and α as AM1-D (for the RM1-D method) and PM3-D (for the MNDO-D and the PM6-D methods). Thus the same elements as listed above are available for use with MNDO-D, RM1-D, and PM6-D.

The dispersion method parameters can be set using the EXTERNAL keyword and including the following lines in the external parameter file for the van der Waals radius parameters (RVDW), the C_6 parameters (C6), the S_6 parameter (S6), and the α parameter (ALPD).

```
RVDW    H      1.110
C6      H      0.160
S6             1.400
ALPD          23.000
```

For using the EXTERNAL keyword, see Section 2.3 of the MOPAC 5th Edition user manual as well as section 4.14 of this manual for more details.

4.9 PM6

The PM6 semiempirical method[15] is a successor to PM3. It is reparameterized for much of the periodic table, although in MOPAC–version 5.022mn only hydrogen through magnesium are currently implemented due to the requirement in PM6 of d basis functions for those elements beyond magnesium. This method contains new functional forms not included in the PM3 formalism that include parameters for pairs of elements in contrast to the other semiempirical methods.

The two diatomic parameters, α_{ij} and x_{ij}, can be set using the EXTERNAL keyword and including the following lines in the external parameter file for the α_{ij} parameters (ALPM6) and the x_{ij} parameters (XPM6).

```
ALPM6   H    C    1.028
XPM6    H    C    0.217
```

For using the EXTERNAL keyword, see Section 2.13 of the MOPAC 5th Edition user manual.

4.10 Gradients

Not every method has analytic gradients available in MOPAC-MN. The following methods have implemented analytic gradients as of MOPAC–version 5.022mn: MINDO/3, MNDO, AM1, PM3, RM1, AM1-D, PM3-D, and PM6; these methods also have analytic gradients available for CI. Methods for which only numerical gradients are available in version 5.022mn are the following: PDDG/MNDO and PDDG/PM3.

4.11 Hydrogen atoms with p orbitals

Hydrogen atoms containing 2p orbitals have been added to the list of available atom types. The new atom symbol for such hydrogens is "Hp". Whenever the Hp atom symbol is used the Hp atoms are assigned the atomic number 9 (fluorine), and all fluorine atoms and all Hp atoms are assigned new parameters consistent with hydrogen atoms with p orbitals.

The core charge on these atoms is 1 in order to be consistent with the way hydrogen atoms in a valence basis set are treated. The exponential parameter for the 2p Slater orbitals is 0.3000 bohr^{-1}. It can be set to other values by defining the ZP parameter for element Hp in an external parameter file (see section 2.3 in the MOPAC 5.0 manual).

The inclusion of p orbitals on hydrogens is necessary for the new PMO method[26].

4.12 Origin for dipole moments

The origin for dipole moments calculated by MOPAC is chosen to be the center of isotopic-averaged masses of all atoms. In the Gaussian programs the origin is chosen as the center of nuclear charge. For neutral systems the results are identical, however for charged systems the dipole moment depends on the choice of origin. The keyword, "DIPG09", is used to specify that, in addition to the standard MOPAC-calculated dipole moment, the Gaussian 09-type dipole moment is also to be reported in the output.

The correction necessary to convert the dipole moment of a charged system from one choice of origin, $\vec{x}$, to another is:

$$\vec{m}_{\text{Gaussian}} = \vec{m}_{\text{MOPAC}} + q_{\text{total}} \cdot \left(\vec{x}_{\text{Gaussian}} - \vec{x}_{\text{MOPAC}} \right) \tag{1}$$

The origins are calculated as a weighted-average of nuclear positions, $\vec{R}_A$, as given in equation (2). The weights, w_A, in the case of MOPAC are chosen as the isotopically-averaged atomic masses; the weights in the case of Gaussian 09 are the nuclear charges of the atoms.

$$\vec{x} = \frac{\sum_{A}^{\text{atoms}} w_A \cdot \vec{R}_A}{\sum_{A}^{\text{atoms}} w_A} \tag{2}$$

When comparing the output files of MOPAC and Gaussian 09 be aware that the coordinate systems may differ, however the dipole moment vectors will be equivalent.

4.13 The D3 dispersion and alternative damping functions for dispersion

Grimme's version 3 DFT dispersion[27] using the suffix "-D3" is available with some of the standard semiempirical methods (MNDO, AM1, PM3, RM1, and PM6) and all elements. The dispersion includes two new parameters, $s_{r,6}$ and s_8, that can be modified with an external parameter file (see Section 4.14). The choice of default values for these two parameters (those for PBEsol-D3) was determined by testing various DFT-D3 parameters with the "-D3" semiempirical methods described here. The damping function used with the dispersion function is the "zero" damping function of Grimme's (goes to zero as the internuclear distance vanishes) instead of the alternative Becke-Johnson damping function.

Unlike the "-D" dispersion methods (described in Section 4.8) using the "-D3" dispersion does not alter the original semiempirical parameters.

Within the dispersion methods, e.g. MNDO-D, AM1-D, and PM3-D, the dispersion energy correction uses a damping function to cause the R^{-6} contributions to vanish as interatomic distances go to zero. An alternative damping function used by Misquitta and Stone[16] and developed by Tang and Toennies[17] is offered as an option with the dispersion methods listed above by specifying the keyword "SDAMP". If this keyword is specified without using one of the "-D" methods above an error will be generated. Currently the SDAMP option is available for all of the same elements as the underlying dispersion method will allow.

The default damping function used in AM1-D and PM3-D by Hillier[13] and Grimme[14] is given in equation (3).

$$f_{damp}\left(R_{ij}\right) = \left[1 + \exp\left(-\alpha\left(\frac{R_{ij}}{R_i^{vdw} + R_j^{vdw}} - 1\right)\right)\right]^{-1} \tag{3}$$

Here R_{ij} is the internuclear separation between atoms i and j. The R^{vdw} are Hillier's parameters for each element's van der Waals radius[17]. The parameter, α, equals 23.0 and is the same for all elements in the dispersion methods.

The alternative damping function (specified by the "SDAMP" keyword) used by Misquitta and Stone[16] is given in equation (4).

$$f_{damp}\left(R_{ij}\right) = 1 - \sum_{k=0}^{6} \frac{\left(b_{ij}R_{ij}\right)^k}{k!} e^{-b_{ij}R_{ij}} \tag{4}$$

Here β depends on the ionization potentials of atoms i (I_i) and j (I_j) as given in equation (5). They are taken from the CRC Handbook of Chemistry and Physics[25] and are given for all elements in the periodic table except for astatine and those beyond atomic number 104.

$$b_{ij} = \sqrt{2I_A} + \sqrt{2I_B} \tag{5}$$

In equations (4) and (5), I_A, I_B, β_{ij}, and R_{ij} are all in atomic units. The formula for the dispersion correction used by Hillier[17] is given by equations (6) and (7). (When using Stone's alternative damping function with the SDAMP keyword, equations (6) and (7) still are used for calculating the dispersion energy with the only difference being the alternative f_{damp} function as given in equation (4) above.)

$$E_{\text{disp}}\left(R_{ij}\right) = -\left(S_6\right) f_{\text{damp}}\left(R_{ij}\right) \frac{C_6^{ij}}{R_{ij}^6} \tag{6}$$

$$C_6^{ij} = \frac{2C_6^i C_6^j}{C_6^i + C_6^j} \tag{7}$$

Here the C_6^i values are specified for individual elements by Hillier for his dispersion methods. Because C_6^i and R_i^{vdw} values for fluorine and neon were not given by Hillier, values for C_6^i of 0.57 and 0.45 J nm^6 mol^{-1} are used and values for R_i^{vdw} of 1.43 and 1.38 Å are used, respectively. The S_6 parameter is chosen as 1.4 and is common to all elements.

The elements H, C, N, O, F, Ne, and S are supported for the dispersion methods MNDO-D, AM1-D, PM3-D, RM1-D, and PM6-D.

4.14 Additional custom parameter specifications

In addition to the parameters that can be specified with a parameter file using the EXTERNAL keyword (see MOPAC 5.0 manual, section 2.3), there are extra parameters whose values can be specified in an external parameter file. These extra parameters belong to new keywords that have been implemented in MOPAC after version 5.0 and they are described here. The following list gives the parameters and the methods for which they apply:

ALPM6	α_{PM6}	The α diatomic parameter for the PM6 method.
XPM6	χ_{PM6}	The χ diatomic parameter for the PM6 method.
RVDW	R^{vdw}	The van der Waals radius for the dispersion methods (e.g. AM1-D).
C6	C_6	The C_6 parameter for the dispersion methods (e.g. AM1-D).
S6	S_6	The S_6 parameter for the dispersion methods (e.g. AM1-D).
ALPD	α_{disp}	The α parameter for the dispersion methods (e.g. AM1-D).
ATMIP	IP	The atomic ionization energy for Stone's alternative damping function.
PMODS	P_i	Specify custom parameter #i for use with the "MODx" modification options.
C0AB	$C_{AB}^{(0)}$	Coefficient used in equation (14).
C1AB	$C_{AB}^{(1)}$	Coefficient used in equation (14).
C2AB	$C_{AB}^{(2)}$	Coefficient used in equation (14).
C3AB	$C_{AB}^{(3)}$	Coefficient used in equation (14).
ALPPR	α_{AB}	Pairwise alpha factor used in equation (14).
C3PWR	P_{AB}	Pairwise power of R_{AB} used in equation (14).
ALPP3	α'_{AB}	Pairwise alpha factor used in equation (14).
KPRSS	$k_{\mu\nu}$	Pairwise k factor used in equation (16); s orbital on atom A and B.
KPRSP	$k_{\mu\nu}$	Pairwise k factor used in equation (16); s orbital on A and p orbital on B.
KPRPP	$k_{\mu\nu}$	Pairwise k factor used in equation (16); p orbital on atom A and B.
SR6	$s_{r,6}$	Coefficient used in the damping function for the "-D3" methods.

S8 s_8 Coefficient for the r^{-8} terms in the "-D3" methods.

For the diatomic parameters (used in PM6, for example) the external file should specify the replacement parameters as follows:

```
Line 1:   ALPM6   O      H      1.300
Line 2:   XPM6    O      H      0.200
Line 3:   XPM6    O      O      0.540
Line 4:   PMODS   1             2.250
Line 5:   PMODS   2             3.100
Line 6:   END
```

The PMODS keyword changes specific parameters used by modifications to the usual semiempirical theory as discussed in Section 4.15. Using lines 4 and 5 in the example above would have the effect of setting the α_{HH} variable to 2.250 and the α_{OO} variable to 3.100.

Further examples of specifying the above parameters can be found in test run files test32.dat, test33.dat, and test34.dat. Also avoid tab characters in the external parameter file. Be certain that the external parameter file name is saved with all upper-case characters and the file itself terminates with a blank line or the word "END".

4.15 Modified functional forms that can be used with PMO

As mentioned in Section 3, version 5.022mn includes the ability to activate specific modifications to make to the functional forms of the standard semiempirical methods.

The first modification (MOD1) is a change to the resonance integrals that involve two different Hp hydrogen atoms or an Hp hydrogen with an oxygen atom. This modification is activated by including the keyword "MOD1" in a line of the input deck. The "MOD1" keyword activates all of the modifications given in equations (8a)-(10). The formulas for the modified resonance integrals are included below:

$$b_{p_H p_{H'}} = 0 \tag{8a}$$

$$b_{p_H s_{H'}} = 0 \tag{8b}$$

$$b_{p_O s_O} = (0.03)\frac{b_{p_H} + b_{s_O}}{2} \tag{9}$$

$$b_{p_O p_O} = (0.15)\frac{b_{p_H} + b_{p_O}}{2} \tag{10}$$

The constant coefficients 0.03 and 0.15 that appear in equations (9) and (10) above can be modified using an external parameter file (see Section 4.14). Within the external parameter file the PMODS #3 value corresponds to the coefficient of equation (9) and the PMODS #4 value corresponds to the coefficient of equation (10).

The "MOD1" keyword also assigns special exponents to the orbitals used in calculating the overlap integral for the resonance integral calculation of Hp-Hp atom pairs (PMODS #5 value) and O-O atom pairs (PMODS #6 value).

The second modification (MOD2) is a change to the 2-electron integrals ($s_H s_H | p_{H'} p_{H'}$) as follows in equation (11). It is activated by including the keyword "MOD2" in the input deck.

$$(s_H s_H \mid p_{H'} p_{H'})_{\text{modified}} = (s_H s_H \mid p_{H'} p_{H'})_{\text{standard}} \left[1 - 0.75 e^{-\left(1.1\,\text{Å}^{-2}\right) R_{HH'}^2} \right] \tag{11}$$

The third modification (MOD3) alters the pairwise core-core interactions that involve orbitals on two different Hp hydrogen atoms or two different oxygen atoms. It is activated by including the keyword "MOD3" in the input deck:

$$E_N \left(H, H'\right)_{\text{modified}} = Z_H Z_{H'} \left(s_H s_H \mid s_{H'} s_{H'}\right)\left(1 + 2e^{-a_{HH} R_{HH'}}\right) \tag{12}$$

$$E_N \left(O, O'\right)_{\text{modified}} = Z_O Z_{O'} \left(s_O s_O \mid s_{O'} s_{O'}\right)\left(1 + 2e^{-a_{OO} R_{OO'}}\right) \tag{13}$$

By default, the parameters $\alpha_{HH'}$ and $\alpha_{OO'}$ are equal to 2.466 Å^{-1} and 3.304 Å^{-1}, respectively. These two parameters can be defined differently by using the PMODS keyword in an external parameter file (see Section 4.14). The PMODS value #1 corresponds to α_{HH}, and the PMODS value #2 corresponds to α_{OO}.

A fourth modification (MOD4) concerns the inclusion of quadrupole moment contributions to the 2-electron integrals of a pair of Hp atoms, ($s_H s_H | p_{H'} p_{H'}$), calculated in the REPP subroutine. By using the MOD4 option (by including the "MOD4" keyword in the input deck), these quadrupole contributions are neglected.

Any combination of the four MODx (x = 1, 2, 3, or 4) options may be specified. These modifications were intended to be used in an MNDO-D calculation on a system containing Hp-type hydrogen atoms.

Several new modifications were implemented in version 5.022mn using the keywords "MOD5", "MOD6", and "MOD7".

The MOD5 keyword introduces a more general core-core energy formalism to replace the previous semiempirical formalisms. Its form is given in equation (14). This expression provides more flexibility for developing new core-core energy expressions and it can also be made consistent with the older semiempirical formalism by correct choice of the constants appearing in equation (14). The constants can be defined by user using an external parameter file (details are in Section 4.14). (Notice that the final term allows the user to specify a power for the internuclear distance factor.)

$$E_N \left(A, B\right) = Z_A Z_B \left(s_A s_A \mid s_B s_B\right)$$
$$\left(1 + C_{AB}^{(0)} R_{AB} e^{-a_{AB} R_{AB}} + C_{AB}^{(1)} e^{-a_A R_{AB}} + C_{AB}^{(2)} e^{-a_B R_{AB}} + C_{AB}^{(3)} R_{AB}^{r_{AB}} e^{-a_{AB} R_{AB}}\right) \tag{14}$$

The MOD6 keyword only applies to systems with Hp hydrogen atoms and comprises two modifications. The first modification supercedes the older MOD2 keyword by replacing equation (11) with equation (15). The only difference is the removal of the unitless factor of 0.75 in equation (11). If both keywords are specified the MOD6 keyword will override the MOD2 keyword.

$$\left(s_H s_H \mid p_{H'} p_{H'}\right)_{\text{modified}} = \left(s_H s_H \mid p_{H'} p_{H'}\right)_{\text{standard}} \left[1 - e^{-\left(1.1\,\text{Å}^{-2}\right)R_{HH'}^2}\right] \tag{15}$$

The second modification implied by using the MOD6 keyword affects the core-core energy calculation for Hp-O atom pairs. The modified core-core energy expression is given in equation (16). The default constant k_{HpO} is 0.24325 a_0^{-1}, however, it can be overridden by specifying the parameter "PMODS 7" when using an external parameter file (see Section 4.14).

$$E_N\left(Hp, O\right)_{\text{modified}} = Z_{Hp} Z_O \left(s_{Hp} s_{Hp} \mid s_O s_O\right) \left[e^{k_{HpO}R_{HpO}}\left(1 + e^{-a_{Hp}R_{HpO}} + e^{-a_O R_{HpO}}\right)\right]$$

$$\tag{16}$$

The MOD7 keyword implements pairwise beta parameters used in the calculation of the resonance integral terms. The new formula is given in equation (17). Like the new core-core formalism, this new pairwise beta formalism allows much greater flexibility than previous semiempirical methods would allow. The constant $k\mu\nu$ has units of inverse bohrs and the constant $B\mu\nu$ is unitless. The user may specifically define the $k\mu\nu$ constants appearing in equation (17) by using an external parameter file (details are given in Section 4.14). The $B_{\mu\nu}$ constants are modified using the BETSS, BETSP, and BETPP keywords as described in Section 4.3.

$$b_{mn} = B_{mn} S_{mn} e^{\left(k_{mn} R_{AB}\right)} \tag{17}$$

4.16 PMO version 1 (PMOv1)

A new semiempirical method called "PMO"[] is included as of version 5.018mn. Calculations using PMO version 1 can be specified by using the keyword "PMOv1" in the input deck. Version 1 of PMO is only parameterized for systems containing Hp (hydrogens with p-orbitals) and O atoms. Future versions will extended the PMO method to carbon, nitrogen, and other elements.

If a PMOv1 calculation is run on a system with ordinary hydrogen atoms (those without p-orbitals that are entered as just "H" in the input file) these ordinary hydrogen atoms are automatically converted to Hp atoms. This is in contrast to previous methods in MOPAC where the distinction between H and Hp atoms is preserved and are treated as two discrete types of atoms.

Running the PMOv1 method will run an MNDO-D calculation with the MOD1, MOD2, MOD3, and MOD4 options automatically activated (see Section 4.15 for details on the MODx options). Additionally, the parameter set has been optimized for molecules, ions, and clusters containing Hp and O atoms for PMOv1. If the PMOv1 keyword is used on a calculation containing any atoms other than hydrogen and oxygen the calculation will not run because the PMOv1 method is only parameterized for systems with hydrogen and oxygen atoms.

Currently PMOv1 and the MODx keywords do not have analytical gradients available. Any calculation attempting to use the PMOv1 or the MODx keywords with the ANALYT keyword will cause the program to terminate prematurely and print an error message warning the user that this combination of keywords is not available.

4.17 PMO version 2 (PMO2)

The PMO2 semiempirical method[28] is included in MOPAC-5.022mn. PMO2 is capable of calculations on systems containing not only hydrogen and oxygen atoms (like the PMOv1 method does) but carbon atoms as well. It is invoked by using the keyword "PMO2" in the input deck. The calculation sets the proper semiempirical parameters, the dispersion settings, and the MODx options (specifically MOD4, MOD5, MOD6, and MOD7) in order to perform PMO2 calculations. Examples of PMO2 calculations, both with the PMO2 keyword and without the PMO2 keyword (using the MODx options and external parameters explicitly given in the input deck), are provided in the runs in the MOPAC-5.022mn test suite.

4.18 PMO2a

The PMO2a semiempirical method[29] is implemented in MOPAC-5.022mn and is invoked by the method keyword "PMO2a" in the input deck. This method allows the use of the elements hydrogen, carbon, nitrogen, oxygen, and sulfur. It is primarily intended for use on atmospheric clusters, several examples of important atmospheric molecules that were used in the development and validation of the method are included in the test runs for the "PMO2a" keyword.

Like the previous polarized molecular orbital methods, PMO2a automatically uses the MODn options MOD4, MOD5, MOD6, and MOD7 as well as specific parameters for the PMO2a method. The method also must use p-orbital-containing hydrogens, "Hp" atoms. Examples of the PMO2a method being implemented in MOPAC with the MODn options explicitly placed in the input deck along with an external parameter file for the PMO2a parameters are given in the test runs. By contrast, examples of the PMO2a method being run on the same systems by only including the "PMO2a" keyword in the input deck are also included in the test runs for verification. An atmospheric cluster binding energy calculation is performed by the testsuite.sh (or alternatively the testsuite.bat script for PCs) and the check.pl scripts.

4.19 Analytical CI gradient

Analytic gradients have been implemented for CI calculations by using the method of M. J. S. Dewar and D. A. Liotard [30]. No new keyword is added for this implementation. When a CI calculation is performed, the gradients of Cartesian coordinates are calculated using this method for the state specified in the input file. When geometry optimization is required, geometry is optimized using the CI gradients for the specified state. See test runs 6, 8, and 11. The implementation of the CI gradients has been checked against the ones calculated by AMSOL 7.1.

Note that the implemented CI gradient algorithm scales as $O(N^4)$. A more efficient algorithm by Patchkovskii and Thiel (S. Patchkovskii and W. Thiel, Theor. Chim. Acta, 93 (1996), 87.) has scaling as $O(N^3)$, and this is under consideration for future implementation.

5. INSTALLATION

5.1 Compiling

The MOPAC–version 5.022mn distribution package can be obtained from the University of Minnesota in one form: tarred and compressed (mopac5022mn.tgz). The mopac5022mn.tgz file can be uncompressed and unarchived by typing the following:

    tar -zxvf mopac5022mn.tgz

This creates the directory mopac5022 in the current working directory. The tar file remains in the current working directory. This file should be saved and backed up for future reference or re-installation.

Building MOPAC–version 5.022mn involves two commands. First type the following command:

    cd mopac5022mn/src

to change the working directory to the directory that contains all the source code. Then issue the command:

    make -f make.xxx mopac5022mn

or to have the executable named automatically use:

    make –f make.xxx

where xxx is machine dependent. The makefiles, located in the mopac5022mn/src directory, distributed with MOPAC–version 5.022mn are:

    make.compaq   (to compile on Compaq workstations)
    make.dec   (to compile on DEC workstations
    make.hp   (to compile on Hewlett-Packard workstations)
    make.ibm   (to compile on IBM RS/6000, SP, and Regatta systems)
    make.linux   (to compile on LINUX systems)
    make.linux_g77   (to compile on LINUX systems)
    make.sgi   (to compile on Silicon Graphics Origin systems)
    make.sgi_gfortran (to compile on Silicon Graphics Origin systems with GNU Fortran)
    make.sun   (to compile on Sun workstations)
    make.pc   (to compile on IBM-compatible personal computers with GNU Fortran)

The following makefiles are specific to the type of compiler rather than the type of workstation and may be used instead based upon which compilers are available to the end-user:

    make.f77   (to use the f77 compiler)
    make.g77   (to use the GNU Fortran g77 compiler)

make.ifort   (to use the Intel ifort compiler)
make.ifort.check   (used during development to check the program)
make.ifort.debug   (used during development to check the program)
make.xlf   (to use the IBM XL Fortran xlf compiler)

This will create the executable file mopac5022mn. After successful compilation, to save disk space, type:

make -f make.xxx clean

(where xxx is replaced by ibm, sgi, etc.) in order to remove all the object files.

5.2 Files included in distribution

Main directory "mopac5022mn"

doc   directory containing the manuals
src   directory containing the source code
test   directory containing the test input files
testo   directory containing the test output files (for comparison)

Directory "doc"

mopac.doc   the original MOPAC 5.0 manual (ASCII file)
mopac5022mn.doc   this document (Microsoft Word document)

Directory "src"

Main program file:
        mstart.f

Main program, subroutine, and function files:
        aababc.f aabacd.f aabbcd.f anad3.f analyt.f anavib.f axis.f babbbc.f babbcd.f bangle.f
        bfn.f block_data.f bonds.f calpar.f capcor.f chgmp2.f chrge.f ciparm.f cnvg.f coe.f
        compfg.f corgen.f dang.f daxpy.f dcart.f ddot.f debug.f delml2.f delmol.f delri.f denrot.f
        densit.f depvar.f deri0.f deri1.f deri21.f deri22.f deri23.f deri2.f deriv.f ders.f dfock2.f
        dfpsav.f dgedi.f dgefa.f dgemm.f dhc.f dhcore.f diag.f diagi.f diagiv.f diat2.f diat.f digit.f
        dihed.f dijkl1.f dijkl2.f dipind.f dipole.f doinit.f dpqtkl.f drc.f drcout.f dscal.f dswap.f ef.f
        efovlp.f efsav.f efstr.f enpart.f extpar.f ffhpol.f flepo.f fmat.f fock1.f fock2d.f fock2.f
        fock.f force.f formd.f forsav.f frame.f freqcy.f geout.f getbet.f getgeo.f gethes.f getsym.f
        gmetry.f gover.f grid.f h1elec.f haddon.f hcore.f helect.f hqrii.f idamax.f ijkld.f ijkl.f
        inibet.f interp.f iter.f jab.f jcarin.f kab.f linmin.f local.f locmin.f lsame.f mamult.f
        matout.f mecid.f meci.f mecih.f mecip.f method.f moldat.f molval.f mopac.f mstart.f
        mtxm.f mullik.f mxm.f nextwd.f nllsq.f nuchar.f osinv.f parsav.f paths.f perm.f pm6de2.f
        pm6der.f pm6dpy.f pm6pyr.f pmodsb.f pmsix.f polar.f powsav.f powsq.f pqtkl.f prjfc.f

prtdrc.f prthes.f pulay.f quadr.f react1.f reada.f readmo.f refer.f repp.f resetc.f resetd.f result.f rotate1.f rotate.amsol rotate.f rotate.mopac rotat.f rsp.f sample.f schmib.f schmit.f scopy.f search.f setbet.f setd3.f set.f setntx.f setupg.f solrot.f spcg.f spline.f ss.f swap.f symtry.f thermo.f timout.f tql2.f tqlrat.f trbak3.f tred3.f update.f updhes.f util.f vecprt.f word.f writmo.f wrtkey.f xerbla.f xyzgeo.f xyzint.f

Interface subroutines between subroutines MSTART and MOPAC:
doinit.f resetc.f resetd.f setntx.f

BETxy auxiliary subroutines:
getbet.f inibet.f setbet.f

Machine-dependent time routines:
date_compaq.f date_dec.f date_hp.f date_ibm.f date_linux.f date_sgi.f date_sun.f
dateclock.c second_ibm.f second_linux.f second_sgi.f

Source code for BLAS subroutines used:
daxpy.f ddot.f dgemm.f dscal.f dswap.f idamax.f lsame.f scopy.f xerbla.f

General machine-dependent subroutines:
dgedi.f dgefa.f diag.f drc.f flepo.f fmat.f force.f locmin.f mullik.f nllsq.f powsav.f
powsq.f prtdrc.f pulay.f react1.f search.f tql2.f tqlrat.f trbak3.f tred3.f writmo.f ef.f
efovlp.f efsav.f efstr.f formd.f gethes.f prjfc.f prthes.f updhes.f

Makefiles:

make.dec	makefile for DEC workstations
make.hp	makefile for HP workstations
make.ibm	makefile for IBM RS/6000 and SP systems
make.sgi	makefile for Silicon Graphics systems
make.sun	makefile for Sun workstations
make.linux	makefile for LINUX systems using f77 compiler
make.linux_g77	makefile for LINUX systems using g77 compiler
make.pc	makefile for personal computers (IBM-compatible)
make.f77	makefile for f77 compilers
make.g77	makefile for GNU g77 compilers
make.ifort	makefile for Intel ifort compilers
make.xlf	makefile for IBM xlf compilers
make.ifort.debug	makefile for Intel ifort compilers with full debugging options
make.ifort.check	makefile for Intel ifort compilers with run-time checking enabled

Remaining files:

SIZES.i	include file for calculation and array size limits
PARAMS.i	include file for CM2 mapping parameters

<u>Directory "test"</u>

testsuite.sh	the Bourne shell script to run the full test suite
testsuite.bat	the MS-DOS shell batch script to run full test suite
test01.dat – test*xx*.dat	MOPAC test input files (see Section 7.2 for list)
xxx.PAR and *xxx*.SRP files	external parameter and external specific reaction parameter files
PM3_CHC_SRP	external HH Gaussian parameter file for test run #16
check.pl	Perl script to verify test file output results (called by testsuite.sh)

<u>Directory "testo"</u>

test01.out – test*xx*.out	MOPAC test output files (see Section 7.2 for list)

6. COMPUTERS AND OPERATING SYSTEMS ON WHICH CURRENT VERSION HAS BEEN TESTED

MOPAC–version 5.022mn has been tested on the following computers and operating systems as given in Table 1.

Computer/Processor	Operating System	Compiler
HP Proliant BL280c Intel "Nehalem"	CentOS 6.6 Linux	ifort Intel 13.1.3
HP Linux Cluster with Haswell E5-2680v3 processors	CentOS 6.6 Linux	ifort Intel 13.1.3
IBM-compatible PC AMD Athlon	MS-DOS (Windows XP)	gfortran 4.4.0 (GNU Fortran)
Table 1. System information for platforms on which MOPAC has been tested.		

The compiler and loader commands are presented in the corresponding make.xxx files as mentioned in Section 5.2. Some numerical problems have arisen using the Intel ifort compiler version 8.1.

If you have difficulty running MOPAC–version 5.022mn on your computer system, such as SCF nonconvergence or inaccurate results and so on, it is suggested to try turning off the compiler optimization during compilation. On personal computers it may be necessary to increase the paging file size (under virtual memory options) in order to meet the memory requirements for MOPAC–version 5.022mn.

7. TEST RUNS

7.1 Running test runs

To run the complete test suite for MOPAC–version 5.022mn, change the current working directory to mopac5022mn/test and execute the shell script testsuite.sh as follows:

./testsuite.sh

If using a personal computer the test suite can be launched as follows:

testsuite.bat

After some minutes, a group of files called test##.out will be created in the mopac5022mn/test directory. You can check the results with the reference test suite outputs, named test##.out that are in the mopac5022mn/testo directory. Additionally you may run the Perl script, check.pl, located in the mopac5022mn/test directory on the test##.out files to verify that correct calculation results were obtained.

The reference output files were created on an SGI Altix 3700 with Intel Itanium2 processors running the Linux 2.6.5 operating system. The testsuite.sh script also creates a text file called test.timing containing timing information about the test runs. You can compare them with the reference timings given in the table below.

Be aware that not all of the makefiles have been tested as of version 5.022mn, but the test suite of earlier versions successfully ran the test suites as listed below. The full test suite has been run successfully on the following machines in the past:

Computer	Processors	Operating system	Makefile used
Compaq ES40	Alpha 500 MHz	Tru64 4.0F	make.compaq
IBM SP	Power3	AIX 4.3	make.ibm
IBM Regatta	Power4	AIX 5.1	make.ibm
Pentium	Intel Pentium III	Linux kernel 2.4.18	make.linux
SGI Origin 2000	R12000	IRIX 6.5	make.sgi
Sun Blade 2000	UltraSparc-III	SunOS 5.8	make.sun
SGI Altix 3700	Intel Itanium2	Linux kernel 2.6.5	make.altix
IBM-compatible PC	AMD Athlon	Windows XP	make.pc
HP	Intel Xeon X5560	Linux kernel 2.6.32	make.ifort

The timings have been done using the UNIX time command (/bin/time), however timings are not yet available for the personal computer build of MOPAC–version 5.022mn.

User + System CPU time (s)

	Compaq	IBM SP	Regatta	Pentium	Origin	Sun	Altix
Test 1	0.5	0.8	0.3	0.7	1.1	0.7	0.2
Test 2	0.3	0.6	0.2	0.5	0.7	0.4	0.2
Test 3	0.3	0.5	0.2	0.5	0.7	0.4	0.2
Test 4	0.1	0.1	0.0	0.1	0.1	0.0	0.0
Test 5	3.4	5.9	2.1	5.8	6.7	4.9	1.5
Test 6	0.5	0.6	0.3	0.6	1.6	0.7	0.2
Test 7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Test 8	0.1	0.0	0.1	0.1	0.3	0.1	0.0
Test 9	0.4	0.2	0.2	0.6	0.8	0.5	0.2
Test 10	0.2	0.6	0.1	0.1	0.1	0.1	0.1
Test 11	0.1	0.1	0.0	0.1	0.2	0.0	0.0
Test 12	0.2	0.3	0.2	0.3	0.5	0.3	0.1
Test 13	0.4	0.1	0.1	0.1	0.2	0.1	0.0
Test 14	0.8	0.2	0.1	0.2	0.3	0.2	0.0
Test 15	0.1	0.1	0.1	0.1	0.2	0.1	0.0
Test 16	0.1	0.1	0.1	0.1	0.2	0.0	0.0
Test 17	---	---	0.0	0.0	0.0	---	0.0
Test 18	---	---	0.0	0.1	0.0	---	0.0
Test 19	---	---	0.0	0.1	0.1	---	0.0
Test 20	---	---	---	---	---	---	0.0
Test 21	---	---	---	---	---	---	0.2
Test 22	---	---	---	---	---	---	0.0
Test 23	---	---	---	---	---	---	0.0
Test 24	---	---	---	---	---	---	0.0
Test 25	---	---	---	---	---	---	0.0
Test 26	---	---	---	---	---	---	0.2
Test 27	---	---	---	---	---	---	0.1
Test 28	---	---	---	---	---	---	0.0
Test 29	---	---	---	---	---	---	0.0
Test 30	---	---	---	---	---	---	0.0
Test 31	---	---	---	---	---	---	0.0
Test 32	---	---	---	---	---	---	0.0
Test 33	---	---	---	---	---	---	0.0
Test 34	---	---	---	---	---	---	0.0
Test 35	---	---	---	---	---	---	0.0
Test 36	---	---	---	---	---	---	0.0
Test 37	---	---	---	---	---	---	0.0
Test 38	---	---	---	---	---	---	0.0
Test 39	---	---	---	---	---	---	0.0
Test 40	---	---	---	---	---	---	0.0
Test 41	---	---	---	---	---	---	0.0
Test 42	---	---	---	---	---	---	0.0
Test 43	---	---	---	---	---	---	0.0
Test 44	---	---	---	---	---	---	0.0

Test 45	---	---	---	---	---	---	0.0
Test 46	---	---	---	---	---	---	0.0
Test 47	---	---	---	---	---	---	0.0
Test 48	---	---	---	---	---	---	0.0
Test 49	---	---	---	---	---	---	0.0
Test 50	---	---	---	---	---	---	0.0
Test 51	---	---	---	---	---	---	0.0
Test 52	---	---	---	---	---	---	0.0
Test 53	---	---	---	---	---	---	0.0
Test 54	---	---	---	---	---	---	0.0
Test 55	---	---	---	---	---	---	0.0
Test 56	---	---	---	---	---	---	0.0
Test 57	---	---	---	---	---	---	0.0
Test 58	---	---	---	---	---	---	0.0
Test 59	---	---	---	---	---	---	0.0
Test 60	---	---	---	---	---	---	0.0
Test 61	---	---	---	---	---	---	0.0
Test 62	---	---	---	---	---	---	0.0
Test 63	---	---	---	---	---	---	0.0
Test 64	---	---	---	---	---	---	0.0
Test 65	---	---	---	---	---	---	0.0
Test 66	---	---	---	---	---	---	0.0
Test 67	---	---	---	---	---	---	0.0
Test 68	---	---	---	---	---	---	0.0
Test 69	---	---	---	---	---	---	0.0
Test 70	---	---	---	---	---	---	0.0
Test 71	---	---	---	---	---	---	0.0
Test 72	---	---	---	---	---	---	0.0
Test 73	---	---	---	---	---	---	0.0
Test 74	---	---	---	---	---	---	0.0
Test 75	---	---	---	---	---	---	0.0
Test 76	---	---	---	---	---	---	0.0
Test 77	---	---	---	---	---	---	0.0
Test 78	---	---	---	---	---	---	0.0
Test 79	---	---	---	---	---	---	0.0

Note: 0.0 means less than 0.05.

7.2 Description of test runs

The input files and reference output files for the set of MOPAC–version 5.022mn test runs are included in the mopac5022mn/test and mopac5022/testo directories, respectively. Below is a brief description of each of the test runs.

Test run 1. The EF, GRADIENTS, NOINTER, PRECISE, and SYMMETRY keywords are specified for calculating the energy and optimized geometry of methylcyclohexane.

Test run 2. The AM1, EF, GNORM, GRAD, SCFCRT, and SYMMETRY keywords are specified for calculating the energy and optimized geometry of morpholine. Note: GRAD is a short form of GRADIENTS and means the same thing.

Test run 3. The GRADIENTS, MNDO, PRECISE, and SYMMETRY keywords are specified for calcualting the energy and optimized geometry of methyl butanoate.

Test run 4. The AM1, CHARGE, PRECISE, SYMMETRY, and TS keywords are specified for calculating the energy and optimized geometry for the transition state of the SN2 reaction of bromide methyl iodide.

Test run 5. The PM3, PRECISE, SYMMETRY, and TS keywords are specified for calculating the energy and optimized geometry of the transition state for the Diels-Alder reaction of cyclopentadiene and MVK.

Test run 6. The C.I., GRADIENTS, MECI, PM3, ROOT, SINGLET, and VECTORS keywords are specified for calculating the energy and optimized geometry for the first singlet excited state of acroleine using analytical CI gradient.

Test run 7. The 1SCF, ANALYT, GRADIENTS, PM3, PRECISE, and VECTORS keywords are specified for calculating the energy of formaldehyde with the C-O bond stretched.

Test run 8. The AM1, CHARGE, C.I., EF, GRADIENTS, PRECISE, and VECTORS keywords are specified for calculating the energy and optimized geometry of the allyl cation using its ground state gradients.

Test run 9. The AM1, GNORM, MODE, PRNT, RECALC, SCFCRT, and TS keywords are specified for calculating the energy and optimized geometry for the MeNC to MeCN transition state.

Test run 10. The EXTERNAL, GNORM, LET, PM3, PRECISE, TS, and UHF keywords are specified for calculating the NDDO-SRP (in particular, PM3-SRP) energy and optimized geometry of the transition state of the NH3 + OH reaction. The input and NDDO-SRP files are in mopac5022mn/test/test10.dat and mopac5022mn/test/TEST10.SRP, respectively.

Test run 11. The 1SCF, AM1, CHARGE, C.I., GRADIENTS, MECI, PRECISE, and VECTORS keywords are specified for calculating the energy and CI gradients of the allyl cation.

Test run 12. The GRAD, MINDO3, and PRECISE keywords are specified for calculating the energy and optimized geometry of thiophene. Note: GRAD is a short form of GRADIENTS and has the same meaning.

Test run 13. The AM1, CM2, and SYMMETRY keywords are specified for calculating the CM2 charges, the energy, and the optimized geometry of 1,2-ethanediol.

Test run 14. The CM2, PM3, and SYMMETRY keywords are specified for calculating the CM2 charges, the energy, and the optimized geometry of 1,2-ethanediol.

Test run 15. The HHON keyword is specified for calculating the energy and optimized geometry for the transition state of the *CH3+CH4 reaction using AM1-CHC-SRP. In this test run, no external HH Gaussian parameter file is specified. Therefore the AM1-CHC-SRP HH Gaussian is used as a default.

Test run 16. The HHON keyword is specified for calculating the energy and optimized geometry for the transition state of the $*CH_3 + CH_4$ reaction using PM3-CHC-SRP. In this test run, the HH Gaussian parameters are specified in an external file explicitly. The input files are mopac 5022mn/test/test16.dat and mopac5022mn/test/PM3_CHC_SRP.

Test run 17. The PDG and UHF keyword is specified for calculating the energy and optimized geometry of the methyl radical.

Test run 18. The MDG keyword is specified for calculating the energy and optimized geometry of the ethyl cation.

Test run 19. The RM1 and UHF keyword is specified for calculating the energy and optimized geometry for the transition state of the $H + FCH_3 \rightarrow HF + CH_3$ reaction.

Test run 20. The AM1-D keyword is specified for calculating the optimized geometry and ground-state energy of the water dimer.

Test run 21. The PM3-D keyword is specified for calculating the optimized geometry and ground-state energy of the water...hydroxide ion complex.

Test run 22. The AM1-D keyword is specified for calculating the optimized geometry and ground-state energy of methanethiol.

Test run 23. The PM3-D keyword is specified for calculating the single-point energy of the ammonium ion.

Test run 24. The PM3-D keyword is specified for calculating the single-point energy of the methane...benzene van der Waals complex.

Test run 25. The PM6 keyword is specified for calculating the single-point energy of a water molecule.

Test run 26. The PM6 keyword is specified for calculating the optimum geometry of a diethylamine molecule. The first analytic gradient is calculated.

Test run 27. The EXTERNAL, LET, PM3, TS, and UHF keywords are specified for calculating the NDDO-SRP (in particular, PM6-SRP) energy and optimized geometry of the transition state of the NH3 + OH reaction. The input and NDDO-SRP files are in mopac5022mn/test/test27.dat and mopac5022mn/test/TEST27.SRP, respectively.

Test run 28. The PM6G09 keyword is specified for the calculation of the single-point energy of acetylene.

Test run 29. The DIPG09 keyword is specified for the reporting of the dipole moment of hydroxide ion via the usual MOPAC coordinate system and also in the Gaussian 09 coordinate system.

Test run 30. The SDAMP keyword is specified for the use of Stone's alternative damping function with the dispersion energy correction in AM1-D on the geometry optimization of a water molecule.

Test run 31. The MNDO-D keyword is specified for calculating the single-point energy of the ammonium ion.

Test run 32. The AM1-D keyword is specified with external parameters for optimizing the geometry of the water dimer. The external parameter file is TEST32.PAR.

Test run 33. The PM6 keyword is specified with external parameters for calculating the single-point energy of the water molecule. The external parameter file is TEST33.PAR.

Test run 34. The AM1-D and SDAMP keywords are specified with external parameters for optimizing the water molecule geometry. The external parameter file is TEST34.PAR.

Test run 35. The RM1-D keyword is specified for calculating the single-point energy of the ammonium ion.

Test run 36. The PM6G09-D keyword is specified for calculating the single-point energy of the acetylene molecule.

Test run 37. The PM6-D keyword is specified for calculating the single-point energy of the ammonium ion.

Test run 38. The AM1 keyword is specified for calculating the optimized bond length of the H_2 molecule using the "Hp" atom type.

Test run 39. Tight optimization of the sulfur-containing compound cysteine with the AM1-D method.

Test run 40. Tight optimization of the sulfur-containing compound cysteine with the PM3-D method.

Test run 41. Test of MNDO-D with MOD1, MOD2, MOD3, and MOD4 keywords on dihydrogen molecule. The external parameter file is TEST41.PAR.

Test run 42. Test of MOD1 keyword with water molecule containing Hp-type atoms. The external parameter file is TEST42.PAR.

Test run 43. Test of MOD2 keyword with water molecule containing Hp-type atoms. The external parameter file is again TEST42.PAR.

Test run 44. Test of MOD3 keyword with water molecule containing Hp-type atoms. Also uses the PMODS keyword in the external parameter file with the MOD3 option. The external parameter file is again TEST42.PAR.

Test run 45. Test of MOD2 and MOD4 keywords with water molecule containing Hp-type atoms. The external parameter file is again TEST42.PAR.

Test run 46. Test of MOD1, MOD2, MOD3, and MOD4 keywords with water molecule containing Hp-type atoms. The external parameter file is again TEST42.PAR.

Test run 47. Test of MOD1 keyword with water molecule containing Hp-type atoms using custom PMODS #3 and #4 parameters. The external parameter file is again TEST47.PAR.

Test run 48. Test of PMOv1 geometry optimization of a water molecule.

Test run 49. Test of PMOv1 single-point energy and polarizability calculation of a water dimer.

Test run 50. Test of PMOv1 single-point energy calculation on a hydronium ion (H_3O^+).

Test run 51. Test of PMOv1 geometry optimization of the water dimer.

Test run 52. Test of PM6-D3 method (PM6 with D3 dispersion) on acetic acid-imidazole dimer with the default D3 dispersion parameters.

Test run 53. Test of MNDO-D3 method (MNDO with D3 dispersion) on sandwich benzene dimer with the default dispersion parameters.

Test run 54. Test of AM1-D3 method (AM1 with D3 dispersion) on acetic acid-imidazole dimer with external parameter file TEST54.PAR for dispersion parameters coming from PBE0-D3.

Test run 55. Test of MNDO-D3 method's dispersion calculation on the system of one chlorine molecule approaching a carbon dioxide molecule.

Test run 56. Test of PMOv1 calculation on the geometry optimization of a water dimer using explicit keywords (MOD1, MOD2, MOD3, MOD4, and MNDO-D) and an external parameter file for a comparison to the same calculation done using only the "PMOv1" keyword.

Test run 57. Test of MNDO-D with PMOv1 external parameters on a water molecule for calculation of the method's complexation energy.

Test run 58. Test of MNDO-D with PMOv1 external parameters on a water dimer for calculation of the method's complexation energy for the water dimer.

Test run 59. Test of MNDO-D with PMOv1 external parameters on a water hexamer for calculation of the method's complexation energy for the water hexamer.

Test run 60. Polarizability calculation on the dihydrogen molecule using the "POLAR" keyword built into MOPAC for comparison to the numerical derivative calculation of the polarizability using charged sparkles.

Test runs 61-69. Calculations of the dihydrogen molecule with charged sparkles at different positions around the system for the numerical calculation of the polarizability. This is compared to the calculation of the polarizability done in test run 60 using the built-in "POLAR" keyword.

Test run 70. Calculation of acetic acid using the "PMO2" keyword.

Test run 71. Calculation of acetic acid using the equivalent set of keyword options and external parameter file to reproduce the results of the PMO2 method. The results are compared to those of test run 70.

Test run 72. Test of the PMO2 keyword on pyruvic acid.

Test run 73. Test of the PMO2a method using the MODx options and external parameters on the methionine dipeptide.

Test run 74. Test of the PMO2a method using the "PMO2A" keyword on the methionine dipeptide.

Test run 75. Test of the PMO2a method using the MODx options and external parameters on sulfuric acid.

Test run 76. Test of the PMO2a method using the "PMO2A" keyword on sulfuric acid.

Test run 77. Test of the PMO2a method using the MODx options and external parameters on ammonia.

Test run 78. Test of the PMO2a method using the "PMO2A" keyword on ammonia.

Test run 79. Test of the PMO2a method using the "PMO2A" keyword on a cluster comprising 3 sulfuric acid molecules and 2 ammonia molecules. Verification of the cluster binding energy is made with this test run and those of test runs 76 and 78.

8. ADDITIONAL NOTES

8.1 Evaluation of Slater Overlap Integrals in Resonance Terms

Within the Neglect of Diatomic Differential Overlap (NDDO) formalism the two-center, one-electron terms are not ignored contrary to a straightforward interpretation of the meaning of NDDO. Instead, these terms are approximated by a "resonance term", $\beta_{\mu\nu}=S_{\mu\nu}\cdot(\beta_\mu+\beta_\nu)/2$, where β_μ and β_ν are given parameters (beta parameters) for the elements and orbital angular momenta corresponding to the basis functions, μ and ν, respectively.

The evaluation of the term $S_{\mu\nu}$ for MNDO (and all later NDDO methods derived from MNDO) is done using the exact overlap integral of Slater ns or np basis functions (where n is valence shell's principal quantum number of the atom on which the basis functions lie). However, when the ANALYT keyword is specified an STO-6G basis function is used in place of the corresponding Slater basis function in order to calculate the overlap integral and the partial derivatives of the overlap integral with respect to the nuclear coordinates.

This is summarized in Table 2 for both MOPAC–version 5.018mn and compared to the Gaussian 03[20] and Gaussian 09[21] programs:

Software mode	MOPAC 5 and MOPAC–version 5.022	Gaussian 03	Gaussian 09	
	Default	Default	AM1=OLD	Default
Default	Slater orbitals with numerical derivatives	STO-3G with analytical derivatives	STO-3G with analytical derivatives	STO-6G with analytical derivatives
ANALYT keyword	STO-6G with analytical derivatives			
PMOv1 or MOD1 keyword	STO-6G with numerical derivatives			

Table 2. Methods of calculating energy and gradients of Slater overlap integral in the NDDO resonance term.

The method of analytical evaluation of Slater orbital overlap integrals can be found in Appendix B.4 of the book, "*Approximate Molecular Orbital Theory*"[22]. The method of analytical evaluation of STO-*n*G overlap integrals (which are simply linear combinations of gaussian functions on two different atom centers) can be found in Appendix A of "*Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*"[23].

The STO-6G gaussian expansion coefficients used by MOPAC and the STO-3G gaussian expansion coefficients used by Gaussian 03 and Gaussian 09 (with the OLD keyword) can be found in "*Small Gaussian Expansions of Slater-Type Orbitals*"[24]. Conversions for the exponential coefficients (values of alpha) for the particular Slater exponents used in semiempirical methods follows equation (4) of reference 22.

Using the PMOv1 or the MOD1 keyword will invoke the STO-6G calculation of overlap integrals just as the ANALYT keyword will, however the program will calculate gradients numerically (unlike the ANALYT keyword which will use analytical derivatives).

9. REFERENCES

[1] A. Banerjee, N. Adams, J. Simons, and R. Shepard, J. Phys. Chem. 89 (1985), 52.

[2] J. Baker, J. Comp. Chem. 7 (1985), 385.

[3] P. Culot, G. Dive, V. H. Nguyen, J. M. Ghuysen, Theor. Chim. Acta 82 (1992), 189.

[4] M. J. D. Powell, Math. Prog., 1 (1971), 26.

[5] R. Fletcher, Practical Methods of Optimization: Unconstrained Optimization, Vol. 1, Wiley, New York (1980).

[6] J. C. Corchado, J. Espinosa-Garcia, W.-P. Hu, I. Rossi, D. G. Truhlar, J. Phys. Chem. 99 (1995), 687-694.

[7] J. Li, T. Zhu, C. J. Cramer, and D. G. Truhlar, J. Phys. Chem. A 102, 1820-1831 (1998).

[8] A. Dybala-Defratyka, P. Paneth, J. Pu, D. G. Truhlar, J. Phys. Chem. A, 108 (2004) 2475-2486.

[9] M. P. Repasky, J. Chandrasekhar, W. L. Jorgensen, J. Comput. Chem. 23 (2002) 1601-1622.

[10] I. Tubert-Brohman, C. R. W. Guimaraes, M. P. Repasky, W. L. Jorgensen, J. Comput. Chem. 25 (2004) 138-150.

[11] I. Tubert-Brohman, C. R. W. Guimaraes, W. L. Jorgensen J. Comput. Theory Chem. 1 (2005) 817-823.

[12] G. B. Rocha, R. O. Freire, A. M. Simas, J. J. P. Stewart, J. Comput. Chem. 27 (2006) 1101-1111.

[13] J. McNamara, I. Hillier, Phys. Chem. Chem. Phys. 9 (2007) 2362-2370.

[14] S. Grimme, J. Comp. Chem. 25 (2004) 1463-1473.

[15] J. J. P. Stewart, J. Mol. Model. 13 (2007) 1173-1213.

[16] A. Misquitta, A. Stone, Mol. Phys. 106 (2008) 1631-1643.

[17] K. T. Tang, J. P. Toennies, J. Chem. Phys. 80 (1984) 3726-3741.

[18] A. Gonzalez-Lafont, T.N. Truong, D. G. Truhlar, J. Phys. Chem. 95 (1991) 4618.

[19] I. Rossi, D. G. Truhlar, Chem. Phys. Lett. 233 (1995) 231.

[20] Gaussian 03, Revision E.01, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Wallingford CT, 2004.

[21] Gaussian 09, Revision A.02, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts,

R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, Gaussian, Inc., Wallingford CT, 2009.

[22] J. Pople; D. Beveridge. *Approximate Molecular Orbital Theory*, 1st ed.; McGraw-Hill: New York, 1970; pp 199-200.

[23] A. Szabo; N. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, 1st ed., rev.; McGraw-Hill: New York, 1989; pp 410-412.

[24] R. F. Stewart, J. Chem. Phys. 52 (1970) 431-438.

[25] D. Lide, Ed., "CRC Handbook of Chemistry and Physics", 90th ed.; CRC Press: Boca Raton, FL. 2009.

[26] P. Zhang, L. Fiedler, H. Leverentz, D. G. Truhlar, J. Gao, J. Chem. Theory Comp. 7 (2011), 857.

[27] S. Grimme, J. Antony, S. Ehrlich, H. Krieg, J. Chem. Phys. 132 (2010), 154104.

[28] M. Isegawa, L. Fiedler, H. Leverentz, Y. Wang, S. Nachimuthu, J. Gao, D. Truhlar, J. Chem. Theory Comp. 9 (2013), 33.

[29] L. Fiedler, H. Leverentz, S. Nachimuthu, J. Friedrich, D. Truhlar, J. Chem. Theory Comp. In print (2014); DOI: 10.1021/ct5003169. Publication date: June 16, 2104.

[30] M. J. S. Dewar; D. A. Liotard, J. Mol. Struct. Theochem. 206 (1990), 123.