

Computational Chemistry Column

Column Editors:

Prof. Dr. H. Huber, University of Basel

Prof. Dr. K. Müller, F. Hoffmann-La Roche AG, Basel

Prof. Dr. H.P. Lüthi, Univ. of Geneva, ETH-Zürich

Chimia 54 (2000) 766–769

© Neue Schweizerische Chemische Gesellschaft

ISSN 0009-4293

MOLEKEL: An Interactive Molecular Graphics Tool

Stefan Portmann^{*a,b} and Hans Peter Lüthi^a

Abstract: MOLEKEL is an interactive visualization ('postprocessing') program for molecular and electronic structure data, generating high-quality graphics for use in research and education. MOLEKEL has been ported to OpenGL and is now available on a number of platforms, including PC (<http://www.cscs.ch/molekel/>). We describe details of its implementation, capabilities and the new added features.

Keywords: Computational chemistry · Molecular graphics · OpenGL · Quantum chemistry · Visualization

Visualization Tools in Computational Chemistry

The chemical building set and a pencil with a piece of paper have always been popular tools in the chemistry laboratory, but this set has recently been extended by the very powerful visualization programs on computers. These programs are not only able to fully replace the traditional tools, but they also offer many additional features.

These visualization programs allow us to interactively modify the view of a model, to overlay molecular and electronic structure information, to dynamically display molecular vibration and to prepare illustrations for presentations and publications. Today, one has the choice between many visualization programs vastly differing in price, performance and

capabilities. Among those are very complex applications, such as Cerius2, Unichem, Titan, Spartan [1–3], that lead the user through the complete process of a chemical computation (Fig. 1). Some of these, however, only support one specific application or are able to communicate with one particular server type only. If you work in a heterogeneous environment with different kinds of compute servers and applications available, these highly integrated softwares may be lacking flexibility when it comes to distributing calculations over more than one server, or when the exchange of data with other software becomes an issue. In that case, simple connectivity between the visualization tool and the hardware and software resources present in the network will be a definite advantage. Its easy integration in a distributed environment was one of the main reasons why we decided to further develop our visualization software MOLEKEL.

MOLEKEL was originally developed by Peter F. Flükiger at the University of Geneva in the context of a Ph.D. thesis [4] under the direction of Prof. J. Weber. The purpose of the molecular graphics package is the interactive visual representation of molecular structures and

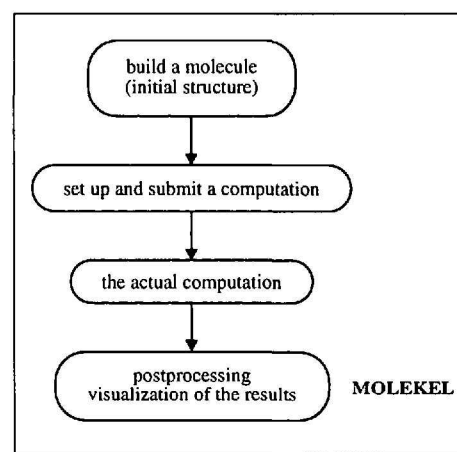


Fig. 1. A typical work flow of a chemical computation and MOLEKEL's position in the process.

properties for research and education. It is mainly designed for generating a detailed, easily interpretable and aesthetically appealing graphical output based on structural data and the results of calculations. To allow us high flexibility in a heterogeneous compute environment, the connection to the data is made *via* easily transferable text logfiles. MOLEKEL was developed further by Flükiger and Portmann at CSCS (Centro Svizzero di Calcolo Scientifico, Manno), where the

*Correspondence: S. Portmann

^aETH-Zürich

Laboratory of Physical Chemistry

Universitätsstr. 16

CH-8092 Zürich

Tel.: +41 1 632 57 82

Fax: +41 1 632 16 15

E-Mail: portmann@igc.phys.chem.ethz.ch

^bAffiliate with the Centro Svizzero di Calcolo

Scientifico (CSCS)

Galleria 2, Via Cantonale

CH-6928 Manno

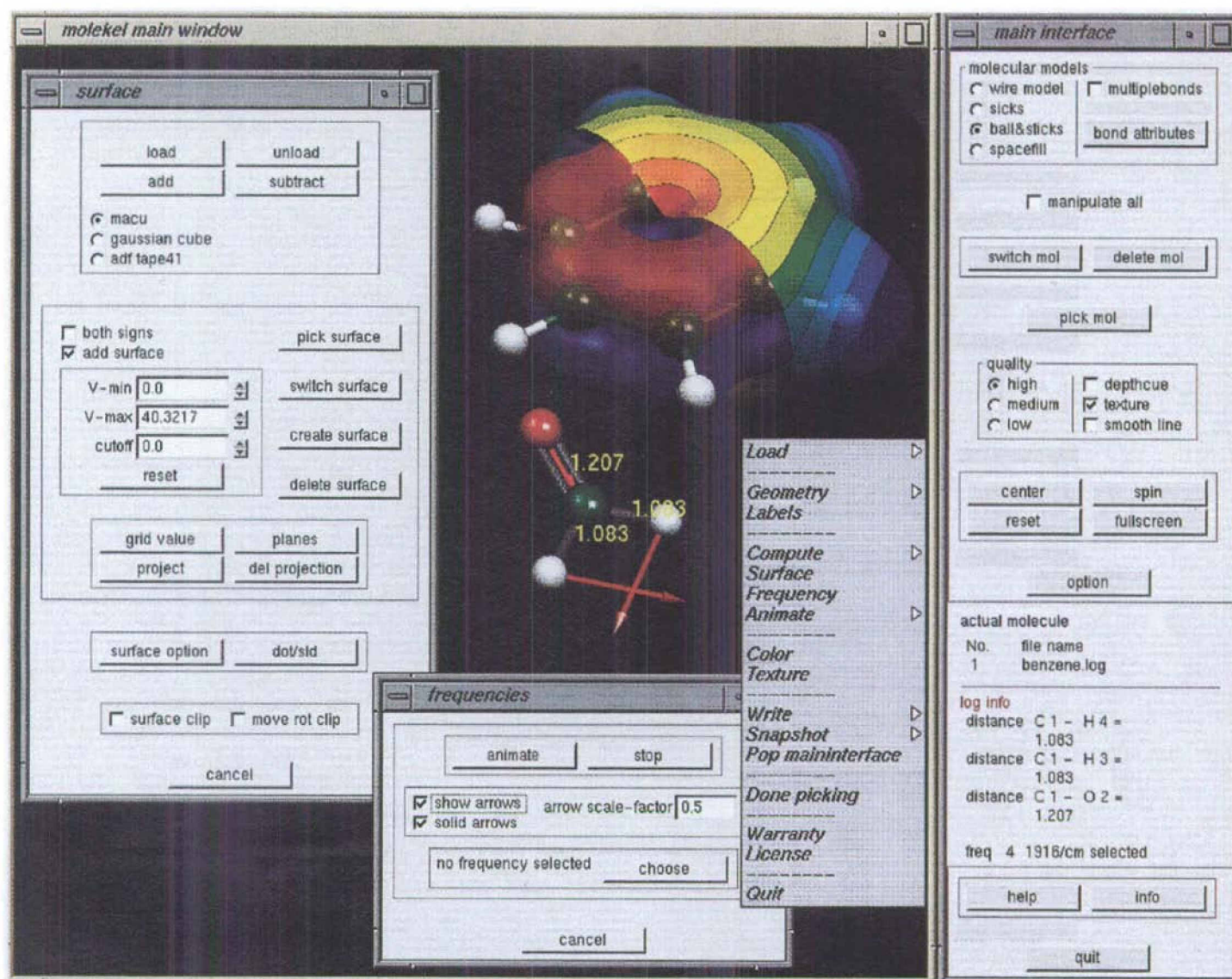


Fig. 2. Screenshot of MOLEKEL: The main graphics window containing the molecules, the main interface window on the right side, the popup menu containing submenus within the main graphics window and two of the popup windows giving control over the surfaces and the representation of the vibrational normal modes. The graphics window is resizable. The main interface window contains a scrollable info text area (log info).

program was made available to the public.

In the present article we describe some of the most recent developments, including the port of the software from high performance graphics stations to generally available platforms.

MOLEKEL

Implementation and System Requirements

The original version of MOLEKEL was written in C and was based on the IRIS GL graphics library of Silicon Graphics, Inc. (SGI) [5]. As a consequence, the program was able to run on SGI platforms only. The IRIS GL graphics library found its successor in the OpenGL graphics system [5–7], whose availability is no longer limited to SGI systems. OpenGL is designed as a system

independent graphics library and therefore does not contain the equivalents to the event handling and window management functions of IRIS GL. This functionality needs to be covered by other libraries, such as the OpenGL Utility Toolkit GLUT [8] for event handling and window management.

GLUT is available for X-Windows systems [9] (all UNIX) and for WINDOWS PCs [10]. The library gives easy access to an OpenGL rendering context, handles mouse and keyboard events and offers easy implementation of a popup menu. To make common graphical user interface (GUI) building blocks such as buttons and checkboxes available, the GLUT-based C++ user interface library GLUI [11] was used in addition. As a consequence, MOLEKEL became a C++ code, even though major parts are still in plain C. On systems where no OpenGL is

available, it can easily be replaced by the freely available Mesa 3D Graphics Library [12] (no change in the source code needed). The new MOLEKEL version consists of two main windows, the graphical window and a main interface window, plus a popup menu and several popup windows (Fig. 2).

This choice of the libraries made the new version of MOLEKEL basically platform independent. The new version 4.0 has been successfully ported to IRIX SGI [5], SUN [13], LINUX [14] and WINDOWS PCs [10] using the systems standard C++ or the GNU gcc compilers [15]. It should also run without major difficulties on other UNIX systems. The use of a gcc compiler together with a minimalist GNU for Windows (MinGW) setup [16] on the WINDOWS PC platform made the port to that platform a rather straightforward task.

Table. Supported file formats.

quantum chemistry code	comment
GAUSSIAN [17]	logfile
GAMESS (US) [18]	logfile
ADF [19]	only coordinates read from logfile, ADF utilities used to generate grid data
ZINDO [20]	logfile
MOS [21]	logfile
HONDO [22]	log and punch file
PRDDO [23]	logfile
MKL	MOLEKEL's own format
coordinate file format	
PDB, also multiple structure [24]	protein data base file format
XYZ, also multiple structure [25]	as in XMol of MSC
grid file format	
GAUSSIAN CUBE [17]	grid file format of Gaussian
ADF TAPE41 [19]	text dump of TAPE41
MACU	MOLEKEL's own format

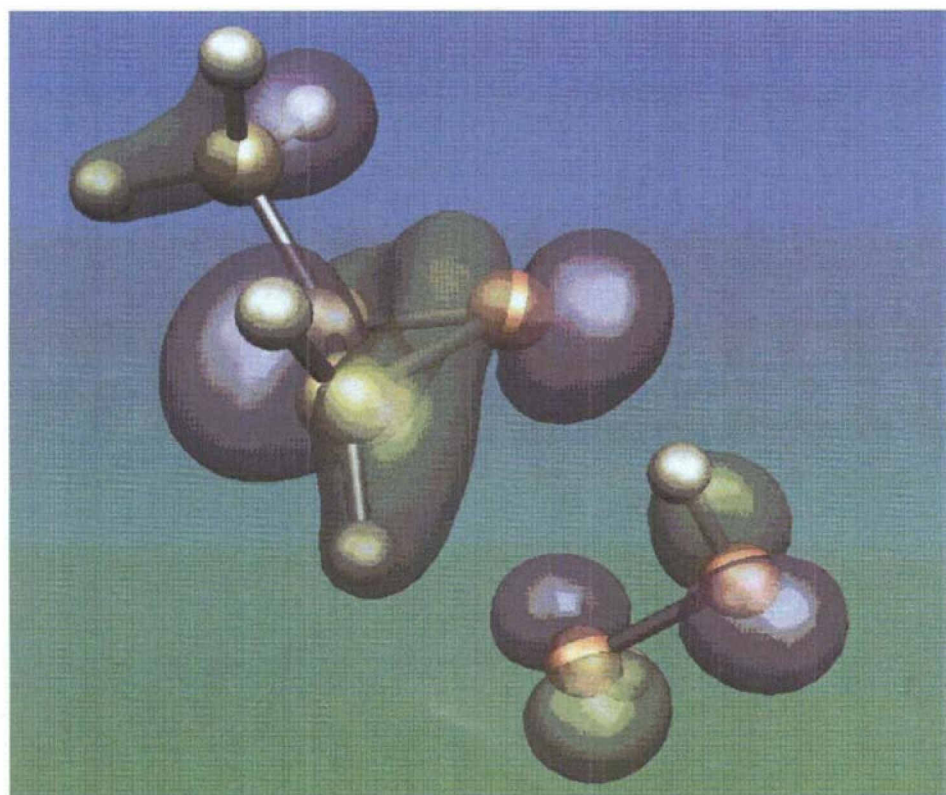


Fig. 3. An output generated by MOLEKEL illustrating its capabilities. The molecular structure is represented as a ball and stick model, the orbitals are rendered transparent using alpha blending. Non standard colors and materials are assigned to the background and the different objects. The displayed example structure, an *anti-(P)*-hydrogen peroxide *R*-2-methyl oxirane complex, is taken from a work published elsewhere [26].

Capabilities and Features

In the current version, MOLEKEL is a pure interactive postprocessing tool (Fig. 1), designed to visualize the output of a number of electronic structure programs (Table). The focus is on the analysis of computational data and on the preparation of photorealistic graphics for presentations and publications. It has no capabilities to build molecules from scratch or to generate input for external electronic structure programs. Only the text output of the electronic structure calculation program is required to visualize properties (exception ADF [19]). This makes MOLEKEL very flexible, especially in an environment of heterogeneous hardware and software resources.

In terms of features, MOLEKEL offers the common wire, stick, ball-and-stick and spacefill (CPK) representation of the atomic structure of molecules (Fig. 3). Several molecules can be loaded simultaneously and can be rotated and translated interactively and independently. In the new version it is now possible to reference the molecule to take action on with the click of the mouse, instead of stepping through the complete reference-list of molecules. Geometrical parameters like bond lengths, angles and torsional angles can be measured interactively and are displayed directly on the model. Two molecules can be superimposed by defining the matching atoms explicitly.

Based on the information parsed from an electronic structure calculation output, the application text logfile, isosurfaces of the electron density, spin density and molecular orbitals can be generated. The fact that these computations are started as independent processes allows us to run them in the background, a feature very useful when working with very large systems (no independent process is started on WINDOWS PCs). Connolly surfaces are also available [27][28]. The external original FORTRAN program of Connolly is used for this purpose. Any surface can be activated and manipulated by the click of the mouse. The active surface can be deleted independent of its position in the list of surfaces. Previously only the last surface generated could be deleted. Properties like the molecular electrostatic potential can be mapped on any isosurface using texture mapping [7][29][30]. The minimal and maximal value of the range over which the color spectrum or the texture is mapped can be set manually. In the old version the minimal and maximal value of the current data was taken, without having the possibility to intervene. This new feature is particular-

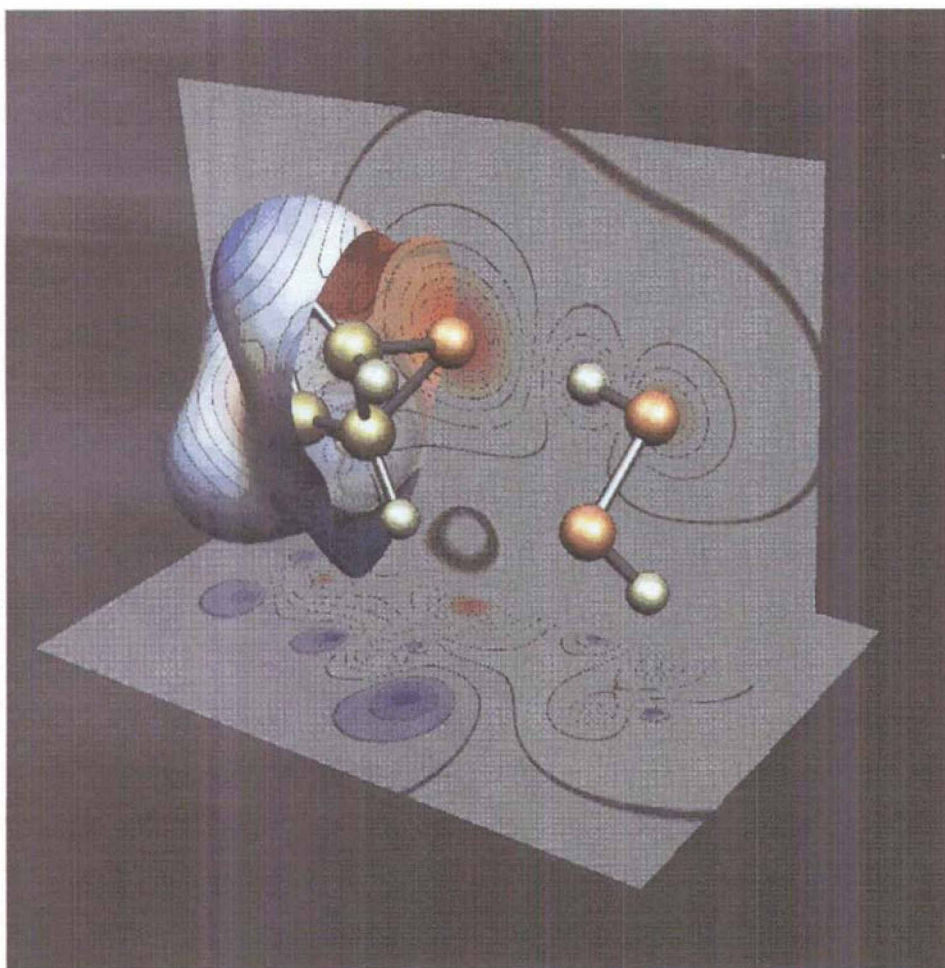


Fig. 4. The same example as for Fig. 3 is taken to illustrate the possibilities of displaying grid data. The molecular electrostatic potential is mapped on the clipplane cut isosurface of the electron density, projected along an axis and cut with a plane. The potential is color coded using a predefined contour texture.

ly useful when comparing a series of molecules. To view areas behind molecule covering surfaces, surfaces can be opened using movable and rotatable clipplanes (one clipplane per surface; only one clipplane per molecule in the old version) and also be rendered transparent, either using polygon stippling or alpha blending [7]. The computed three-dimensional grid data used for the isosurface generation is saved in an external file in MOLEKEL's own 'macu' format for later use. As a new feature, grid data of Gaussian (cube files) [17] and ADF (TAPE41 files) [19] can also be read into MOLEKEL. Besides rendering isosurfaces from grid data, the data can also either be projected on a normal plane along any one of the principal axes or on a movable plane along a principal axes cutting through the object. Using texture mapping, the planes are color coded according to the projected values of the grid points or according to the values of the grid at the planes actual position, respectively (Fig. 4).

Vibrational normal modes can be viewed statically using arrows indicating the direction of the motion of a mode or dynamically through animation. The arrows can now also be rendered as solid objects not just as lines (Fig. 2). Sequences of coordinates, e.g. from a geometry optimization, can also be animated. Coordinates are exportable in the original orientation or the current orientation in PDB and now also in XYZ format [24][25]. The colors and materials of the objects are flexible (Fig. 3), textures using any picture, available in SGI's RGB file format [31], can be assigned to various objects. As a new feature, four, instead of only one, predefined contour textures are already compiled into the executable. It is possible to save the content of the graphics window in SGI's RGB [31] or in TIFF format [32].

Conclusion, Program Availability

MOLEKEL is already an established interactive molecular graphics program, a fact illustrated by the large number of

downloads of the old version. Its wealth of features and high-quality graphics, its platform independence and easy integration into almost any compute infrastructure make the new version of MOLEKEL an attractive tool for researchers and educators performing or teaching chemical computation. The new, self-explanatory GUI along with manuals available in pdf, postscript or html format, MOLEKEL is even more simple to use. It is based on a up-to-date software technology, which makes the code also an interesting starting point for further development. Among the options considered are the extension of supported file formats, the addition of a molecular builder, and the upgrade of MOLEKEL to a collaborative engineering tool [33].

MOLEKEL can be downloaded from <http://www.cscs.ch/molekel/> free of charge (precompiled executables). The copyright for the software is owned by CSCS. It is possible to obtain the source code in the framework of a collaborative agreement with CSCS (please contact Dr. D. Maric; E-Mail: dmaric@cscs.ch)

Acknowledgment

We would like to thank CSCS for their support and the opportunity to work on the MOLEKEL project.

Received: November 7, 2000

- [1] <http://www.msi.com/life/products/cerius2/>
- [2] <http://www.oxmol.com/software/unichem/>
- [3] <http://www.wavefun.com/software/software.html>
- [4] Peter F. Flükiger, 'Development of the molecular graphics package MOLEKEL and its application to selected problems in organic and organometallic chemistry', Thèse No 2561, Département de chimie physique, Université de Genève, Genève, 1992.
- [5] <http://www.sgi.com>
- [6] <http://www.opengl.org>
- [7] OpenGL Programming Guide, 'The Official Guide to Learning OpenGL', M. Woo, J. Neider, T. Davis, D. Shreiner (OpenGL Architecture Review Board), Addison-Wesley, 1999.
- [8] <http://reality.sgi.com/mjk/glut3/glut3.html>
- [9] <http://www.x.org>
- [10] <http://www.microsoft.com/windows/>
- [11] <http://www.cs.unc.edu/~rademach/gluil/>
- [12] <http://www.mesa3d.org>
- [13] <http://www.sun.com>
- [14] <http://www.linux.org>
- [15] <http://www.gnu.org/software/gcc/gcc.html>
- [16] <http://www.mingw.org/>; the mingw32 distribution available from <ftp://ftp.xraylith.wisc.edu/pub/khan/gnu-win32/mingw32/gcc-2.95.2/> was used.
- [17] Gaussian 98, M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, V.G. Zakrzewski, J.A.

- Montgomery, Jr., R.E. Stratmann, J.C. Burant, S. Dapprich, J. M. Millam, A.D. Daniels, K.N. Kudin, M.C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G.A. Petersson, P.Y. Ayala, Q. Cui, K. Morokuma, D.K. Malick, A.D. Rabuck, K. Raghavachari, J.B. Foresman, J. Cioslowski, J.V. Ortiz, B.B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R.L. Martin, D.J. Fox, T. Keith, M.A. Al-Laham, C.Y. Peng, A. Nanayakkara, C. Gonzalez, M. Challacombe, P.M.W. Gill, B. Johnson, W. Chen, M.W. Wong, J.L. Andres, C. Gonzalez, M. Head-Gordon, E.S. Replogle, and J.A. Pople, Gaussian, Inc., Pittsburgh PA, 1998. <http://www.gaussian.com>
- [18] M.W. Schmidt, K.K. Baldridge, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, K.A. Nguyen, S.J. Su, T.L. Windus, together with M. Dupuis, J.A. Montgomery, *J. Comput. Chem.* **1993**, *14*, 1347. <http://www.msg.ameslab.gov/GAMESS/GAMESS.html>
- [19] E.J. Baerends, D.E. Ellis, P. Ros, *Chem. Phys.* **1973**, *2*, 41; L. Versluis, T. Ziegler, *J. Chem. Phys.* **1988**, *322*, 88; G. te Velde, E.J. Baerends, *J. Comput. Phys.* **1992**, *99*(1), 84; C. Fonseca Guerra, J.G. Snijders, G. te Velde, E.J. Baerends, *Theor. Chem. Acc.* **1998**, *99*, 391, <http://www.scm.com/>
- [20] J. Ridley, M.C. Zerner, *Theoret. Chim. Acta* **1976**, *42*, 223.
- [21] by H. Baumann ETHZ.
- [22] M. Dupuis, J.D. Watts, H.O. Villar, G.J.B. Hurst, *Comput. Phys. Comm.*, **1989**, *52*, 415.
- [23] A. Derecskei-Kovacs, D. S. Marynick, *Internat. J. Quantum Chem.*, **1996**, *58*, 193.
- [24] <http://www.pdb.org>
- [25] Xmol of Minnesota Supercomputer Center, Inc., <http://www.msc.edu/msc/docs/xmol/XMol.html>
- [26] S. Portmann, A. Inauen, H.P. Lüthi, S. Leutwyler, *J. Chem. Phys.* **2000**, *21*.
- [27] M.L. Connolly, *J. Appl. Cryst.* **1983**, *16*, 548.
- [28] M.L. Connolly, *Quantum Chemistry Program Exchange Bull.* **1981**, *1*, 74.
- [29] P.S. Heckbert, *IEEE Computer Graphics and Applications* **1986**, *56*.
- [30] P.S. Heckbert, 'Fundamentals of Texture Mapping and Image Warping', M.Sc. Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, **1989**.
- [31] <http://reality.sgi.com/grafica/sgiimage.html>
- [32] <http://www.libtiff.org>
- [33] P.F. Flükiger, S. Portmann, H.P. Lüthi, **1999** Proceedings, Lecture Notes in Computer Science 1615, Springer, 11.

Chimia 54 (2000) 770

© Neue Schweizerische Chemische Gesellschaft
ISSN 0009-4293

Verleihung des Hellmann-Preises für Theoretische Chemie



Aus Anlass des 95. Geburtstages und des 60. Todestages des Quantenchemie-Pioniers Hans G.A. Hellmann hatten 1998 die deutsche, österreichische und schweizerische Trägerorganisation des 'Symposiums für Theoretische Chemie' be-

schlossen, jährlich einen Hellmann-Preis für hervorragende Leistungen in diesem Gebiet an Nachwuchswissenschaftler und -wissenschaftlerinnen zu vergeben. Nach der ersten Verleihung an Wim Klopper 1999 wurde der Preis dieses Jahr an Andreas Görling vergeben. Wir publizieren nachstehend die Laudation verbunden mit unserer Gratulation an den Preisträger.

Andreas Görling, Jahrgang 1960, studierte Chemie an der Technischen Universität München. Ende der 80er Jahre rankten sich seine Dissertationsarbeiten bei Prof. Rösch um verschiedene anwendungsorientierte Themenstellungen der Theoretischen Chemie. Dabei machte er erste Bekanntschaft mit der damals auch bei Chemikern in Mode gekommenen Dichtefunktionalmethode (deren Förderung durch Kohn und Pople 1998 mit dem Nobelpreis für Chemie ausgezeichnet wurde). Während eines längeren Gastaufenthaltes in New Orleans liess

sich Andreas Görling von Prof. Levy in die theoretische Analyse der Grundlagen der Dichtefunktionale einführen. Andreas Görling hat dann sowohl den grundlegenden Formalismus wie die anwendungstechnischen Rechenverfahren weiterentwickelt und wurde mit diesen Ergebnissen, zurück in München, im Jahre 1995 habilitiert. Seither hat er weitere Beiträge zur verbesserten dichtefunktionaltheoretischen Behandlung von Grund- und insbesondere von angeregten Zuständen in kleineren Molekülen wie in ausgedehnten Festkörpern, mittels selbstwechselwirkungsfreier Austauschpotentiale und unter Lösung des Symmetriedilemmas, neuerdings auch zur Untersuchung nichtlinearer optischer Phänomene, erarbeitet. Die vornehmlich in *Physical Review* erschienenen Arbeiten wurden am 11. September 2000 auf dem 36. Symposium für Theoretische Chemie in Litschau, Niederösterreich, mit dem 'Hans G.A. Hellmann-Preis 2000' ausgezeichnet.