ry chemist should be trained in the basics of computer technology, including topics such as terminals, PCs, workstations, networks, databases, operating systems, data exchange protocols. Many independent software packages are available, *e.g.* for data aquisition and manipulation, molecular modeling, text editing, *etc.* To a large extent these can be used without any knowledge of programming. However, development of new innovative programs of course requires modern programming knowhow. Furthermore, some chemists might be involved in the integration of various programs routinely used in a laboratory. There again programming is involved. The synergy between chemistry and programming skills will remain for a long time a positive factor in a well-functioning laboratory. In part these programming efforts may come from a trained chemist, while also the interaction with a computer department should be stimulated. In my own experience I have seen that for *ca.* ten years, most even simple applications had to be written in-house, while at present all can be bought. These trend will still increase, and certain 'dreams' will become available as powerful software, made by specialized vendors, which in turn indeed often work with programming chemists. More then programming skills, in the future a rapid adaptation to new hard- and software products will be asked from the 'chemist'.

Dr. *H. van de Waterbeemd*, Structure-Property Correlations Group, *F. Hoffmann-La Roche AG*, Basel

1) I had my first programming course at the ETH-Zürich 1964 'ALGOL'. After finishing this course I did very little programming in ALGOL and finally gave up completely. Later I acquired knowledge of BASIC by myself. I did a few very simple programs myself and then finally gave up completely in programming, because I still was interested in chemistry. Since *ca.* ten years, I'm now using personal computers, first MS-DOS and now *Macintosh.* I use it a lot, but I do not need to do my own programming. For some special problems I have asked specialists to give me a hand to do some programming.

2) 10% of the chemists in our institute do their own programming.

3) No, this percentage did not change a lot in the last years, but the difference is that now everybody uses computers, before only 10%, those who knew programming, used computers.

4) Yes, I think knowledge about computers is essential for every chemist today.

5) For 90% of the chemists today it suffices to know computers in the same way as they know how to drive their car. For them it is neither necessary to do their own programming, nor to make a tune up of their car themselves. The ability to use computers with existing programs is nowadays sufficient for most chemistry. Yet, there should be a possibility for those who want to go beyond that, to learn programming in a serious way.

Prof. Dr. *A. von Zelewsky*, Institut de chimie inorganique et analytique, Université de Fribourg

Chimia 46 (1992) 227–228
© Neue Schweizerische Chemische Gesellschaft
ISSN 0009–4293

Comparison of the Performance of a Program for Molecular Dynamics Simulations of Liquids on Different Computers

Rolf Eggenberger and **Hanspeter Huber***

In a previous 'Computational Chemistry Column' [1] *Th. Bally et.al.* carried out a comparison of the performance of two quantum chemistry programs on different computers. Such comparisons are helpful in several situations, *e.g.* if you plan to purchase a computer, if you ask for a grant of computer time on a national supercomputer or if you have a choice to run different programs on different computers. The previous comparison [1] showed for example that for quantum chemical calculations workstations are in general more cost-efficient than supercomputers. This situation might differ for another kind of problem. Therefore, we present here some results concerning the performance of a simulation program running on several machines including the new national supercomputer in Manno, the NEC SX-3.

Together with quantum chemistry calculations, simulations are undoubtly the major type of number-crunching applications in chemistry. For the nonspecialist, we present here a short description of the main features of a molecular dynamics simulation program for liquids. Imagine a number of, *e.g.* 500, molecules each set at a random position in a box with a velocity vector also chosen randomly. The lengths of the vectors are scaled according to the temperature. Now go ahead a certain time step, let's say one femtosecond. That means you have to move all particles along their velocity vectors by a displacement corresponding to the femtosecond. In addition, you change the velocity vectors according to the time-step and the corresponding acceleration vectors. The latter are obtained from the forces acting on the particles, which in turn are calculated for each particle from a sum over the forces due to all other particles. To evaluate these forces in the two-body approximation you have to supply the two-particle potential energy curve. Although the real program is slightly more complicated, its structure is rather simple compared to a quantum chemistry program. You loop over, *e.g.* 100 000, time-steps and the time-consuming part is the calculation of the forces at each step since you sum over pairs of particles, which number 124 750 (n•(n–1)/2) in the above example. The computer specialist will immediately recognize that such a program is suited to vectorization.

**Correspondence*: PD Dr. H. Huber
Institut für Physikalische Chemie der
Universität Basel
Klingelbergstrasse 80
CH–4056 Basel

ry chemist should be trained in the basics of computer technology, including topics such as terminals, PCs, workstations, networks, databases, operating systems, data exchange protocols. Many independent software packages are available, *e.g.* for data aquisition and manipulation, molecular modeling, text editing, *etc.* To a large extent these can be used without any knowledge of programming. However, development of new innovative programs of course requires modern programming knowhow. Furthermore, some chemists might be involved in the integration of various programs routinely used in a laboratory. There again programming is involved. The synergy between chemistry and programming skills will remain for a long time a positive factor in a well-functioning laboratory. In part these programming efforts may come from a trained chemist, while also the interaction with a computer department should be stimulated. In my own experience I have seen that for *ca.* ten years, most even simple applications had to be written in-house, while at present all can be bought. These trend will still increase, and certain 'dreams' will become available as powerful software, made by specialized vendors, which in turn indeed often work with programming chemists. More then programming skills, in the future a rapid adaptation to new hard- and software products will be asked from the 'chemist'.

Dr. *H. van de Waterbeemd*, Structure-Property Correlations Group, *F. Hoffmann-La Roche AG*, Basel

1) I had my first programming course at the ETH-Zürich 1964 'ALGOL'. After finishing this course I did very little programming in ALGOL and finally gave up completely. Later I acquired knowledge of BASIC by myself. I did a few very simple programs myself and then finally gave up completely in programming, because I still was interested in chemistry. Since *ca.* ten years, I'm now using personal computers, first MS-DOS and now *Macintosh.* I use it a lot, but I do not need to do my own programming. For some special problems I have asked specialists to give me a hand to do some programming.

2) 10% of the chemists in our institute do their own programming.

3) No, this percentage did not change a lot in the last years, but the difference is that now everybody uses computers, before only 10%, those who knew programming, used computers.

4) Yes, I think knowledge about computers is essential for every chemist today.

5) For 90% of the chemists today it suffices to know computers in the same way as they know how to drive their car. For them it is neither necessary to do their own programming, nor to make a tune up of their car themselves. The ability to use computers with existing programs is nowadays sufficient for most chemistry. Yet, there should be a possibility for those who want to go beyond that, to learn programming in a serious way.

Prof. Dr. *A. von Zelewsky*, Institut de chimie inorganique et analytique, Université de Fribourg

Chimia 46 (1992) 227–228
© Neue Schweizerische Chemische Gesellschaft
ISSN 0009–4293

Comparison of the Performance of a Program for Molecular Dynamics Simulations of Liquids on Different Computers

Rolf Eggenberger and **Hanspeter Huber***

In a previous 'Computational Chemistry Column' [1] *Th. Bally et.al.* carried out a comparison of the performance of two quantum chemistry programs on different computers. Such comparisons are helpful in several situations, *e.g.* if you plan to purchase a computer, if you ask for a grant of computer time on a national supercomputer or if you have a choice to run different programs on different computers. The previous comparison [1] showed for example that for quantum chemical calculations workstations are in general more cost-efficient than supercomputers. This situation might differ for another kind of problem. Therefore, we present here some results concerning the performance of a simulation program running on several machines including the new national supercomputer in Manno, the NEC SX-3.

Together with quantum chemistry calculations, simulations are undoubtly the major type of number-crunching applications in chemistry. For the nonspecialist, we present here a short description of the main features of a molecular dynamics simulation program for liquids. Imagine a number of, *e.g.* 500, molecules each set at a random position in a box with a velocity vector also chosen randomly. The lengths of the vectors are scaled according to the temperature. Now go ahead a certain time step, let's say one femtosecond. That means you have to move all particles along their velocity vectors by a displacement corresponding to the femtosecond. In addition, you change the velocity vectors according to the time-step and the corresponding acceleration vectors. The latter are obtained from the forces acting on the particles, which in turn are calculated for each particle from a sum over the forces due to all other particles. To evaluate these forces in the two-body approximation you have to supply the two-particle potential energy curve. Although the real program is slightly more complicated, its structure is rather simple compared to a quantum chemistry program. You loop over, *e.g.* 100 000, time-steps and the time-consuming part is the calculation of the forces at each step since you sum over pairs of particles, which number 124 750 (n•(n–1)/2) in the above example. The computer specialist will immediately recognize that such a program is suited to vectorization.

**Correspondence*: PD Dr. H. Huber
Institut für Physikalische Chemie der Universität Basel
Klingelbergstrasse 80
CH–4056 Basel

Table. *Absolute and Relative Performance of the MD Simulation Program on Different Computers*

Computer	CPU-time/h	Rel. Perform.	Characteristics[a])
Macintosh IIci	1300	0.24	PC, Mac-OS, Language Systems Fortran (-opt = 3)
DEC VAX-8600	660	0.47	WS, VMS, VAX Fortran V5
CONVEX C120	334	0.93	ConvexOS V9.1, CONVEX Fortran (fc -O2),VP
DEC VAX-8800	310	**1.00**	MF, VMS, VAX Fortran V5
DEC VAX-6000	120	2.7	MF, VMS, VAX Fortran V5, VP
SG IRIS/INDIGO	92	3.4	PC, IRIX 4.0, SG Fortran (f77 -O4)
CONVEX C3210	87	3.6	ConvexOS V9.1, CONVEX Fortran (fc -O2 -pd8), VP
DECStation 5000/200	88	3.6	RS, RISC-ULTRIX, DEC Fortran (f77 -O4)
SG IRIS 4D/320	81	3.8	RS, IRIX 3.33, SG Fortran (f77 -O4)
IBM-RS/6000-550	30	10.	RS, AIX, XL Fortran (xlf -O)
CRAY Y-MP	8.0	39.	SC, UNICOS 6.0,cf77 5.0[b]),VP
NEC SX-3/22	2.6	120.	SC, SUPER-UX 2.11,f77sx[c]), VP

[a]) PC = Personal computer, WS = Workstation, RS = Risc-workstation, MF = Mainframe, SC = Supercomputer, VP = Vector processor.
[b]) Options: FFLAGS = -Zv -c -Wf '-e amsx -dp -i 46 -o vector,aggress' LIBS = -lsci.
[c]) Options: FFLAGS = -float2 -w double 16 -b -e2 LIBS = -l-lblas2 -l-leispack2 -l-llinpack2.

The program used for this comparison was written in FORTRAN77 for an *ab initio* study of neon in the condensed phase [2]. The CPU-times given for 235 270 steps have been obtained from very long runs which were needed to evaluate transport properties in equilibrium simulations from a *Green-Kubo* integral. The *Verlet* leap frog algorithm for a cubic box with periodical boundary conditions and the minimum image convention were used [3]. The program was written in such a way that it uses little RAM as well as disk space, making it possible to run it on very different machines. For the time correlation functions an algorithm to be used in connection with tapes was adapted so that the data needed can be accumulated during the simulation. On scalar computers *Verlet* neighbour lists were used. The *Table* shows the absolute and relative performance on different computers and gives a minimal description of the computer type. The relative performance is taken as the speed relative to the DEC VAX-8800 main-frame computer.

All the calculations were performed on a single processor, even on machines where several processors were available. It is seen that the relative performance of the supercomputers is in this case better than for the quantum chemical calculations, although the cost/performance ratio is probably still in favour of the workstations. To optimize the overall speed, the vector machines usually have to do unnecessary calculations because a vectorization is not (or only partially) possible when 'if-statements' are present in loops. *E.g.*, the vector processors on the two Convex machines and on the VAX-6000 hardly accelerated the calculation. This means that a Mflop-comparison would give a better performance for vector machines, but it would not be a very realistic comparison. On the other hand, it is possible that the performance could still be improved on the vector machines by rewriting the program, trying to vectorize the list-algorithm. A further question is whether the relative performance on scalar and vector machines depends much on the number of particles in such a simulation. For a check, we performed two short calculations with 2048 particles. The one on a scalar machine (INDIGO) utilizing the *Verlet*-list-algorithm takes 7.5 times longer, the one on the vector machine (NEC) without this algorithm 14.2 times longer than the simulation with 500 particles. This should be compared with a scaling by n^2 of 16.8. This means that longer vectors hardly help to decrease the n^2-scaling.

We should perhaps comment on three machines which are quite new on the market. The INDIGO from Silicon Graphics, which is marketed as personal computer but reaches the performance of several workstations, the Convex3210 and the NEC SX-3. The latter is of interest because supercomputers with vectorprocessors show usually very different performances in different benchmarks. This comparison gives probably one of the first numbers from 'real life'.

In conclusion, we should mention some problems associated with this comparison. As we did not want to waste computer time, these measurements were taken from runs needed for our research [2]. The conditions (*e.g.* the pressure of the liquid neon) were slightly different for different runs, resulting in estimated errors of 5–10% in the above comparison. Similar or even larger errors might occur from inaccurate CPU-time measurements (on some multi-user computers the reported time might change up to 20% depending on the load of the machine). The program might not be optimized on each machine. In particular one might ask whether a fair comparison should use a different algorithm on a scalar than on a vector machine. We decided to use different algorithms, but on different scalar machines we used the same code and the same is essentially valid for different vector processors (some details were modified and on the Convex the optimized linpack/eispack library was not available).

We thank the staff of the computer centers for their assistance and the *Schweizerische Nationalfonds zur Förderung der Wissenschaften* for support of the underlying research project (20-29924.90).

[1] Th. Bally, P.-A. Carrupt, J. Weber, *Chimia* **1991**, *45*, 352.
[2] R. Eggenberger, S. Gerber, H. Huber, D. Searles, *Chem. Phys.* **1991**, *156*, 395; R. Eggenberger, S. Gerber, H. Huber, D. Searles, M. Welker, *Mol. Phys.* **1992**, accepted for publication; *Chem. Phys.* **1992**, submitted for publication.
[3] M.P. Allen, D.J. Tildesley, 'Computer Simulations of Liquids', Clarendon Press, Oxford, 1987.