Computer Physics Communications 191 (2015) 221–234

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Efficient hybrid-symbolic methods for quantum mechanical calculations☆

T.C. Scott*, Wenxing Zhang

College of Physics and Optoelectronics, Taiyuan University of Technology, Shanxi 030024, China

ARTICLE INFO

Article history:
Received 10 December 2014
Received in revised form
10 February 2015
Accepted 13 February 2015
Available online 24 February 2015

Keywords:
Quantum chemistry
Symbolic manipulation
Efficient code generation
Basis sets
Maple
Macrofort
MATLAB
Scilab

ABSTRACT

We present hybrid symbolic–numerical tools to generate optimized numerical code for rapid prototyping and fast numerical computation starting from a computer algebra system (CAS) and tailored to any given quantum mechanical problem. Although a major focus concerns the quantum chemistry methods of H. Nakatsuji which has yielded successful and very accurate eigensolutions for small atoms and molecules, the tools are general and may be applied to any basis set calculation with a variational principle applied to its linear and non-linear parameters.

Program summary

Program title: EVAN

Catalogue identifier: AEVU_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEVU_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html

No. of lines in distributed program, including test data, etc.: 99020

No. of bytes in distributed program, including test data, etc.: 5388574

Distribution format: tar.gz

Programming language: Maple[1], MATLAB, Scilab [2], FORTRAN.

Computer: Ranging from laptop to CLUSTER system.

Operating system: Linux system which supports Maple and MATLAB (or Scilab).

RAM: 64 bytes

Classification: 4, 5, 16.

External routines: Macrofort[3,4], BLAS, LAPACK[5]

Nature of problem: Develop and showcase general tools for analyzing and solving ab initio quantum chemistry problems, in particular an efficient means of generating an accurate and high performance optimized program for any specific problem.

Solution method: Use a Computer Algebra system (CAS) to analyze any given problem, study its symmetries, explore basis sets that best match its natural properties, seek preliminary solutions and when needed, use hybrid symbolic–numerical tools to generate optimized code in MATLAB (or Scilab) or FORTRAN (or C), etc., code which is tailored to a specific problem combined with high-performance numerical routines for solving the given problem. Particular attention is spent on matrix elements for a resulting Hamiltonian and using accessible eigensolvers, as well as their accuracy and performance.

Additional comments: Subroutines Generated by Maple or Macrofort

Running time: Variable depending on problem size and speed of processors but good algorithmic complexity

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: tcscott@gmail.com (T.C. Scott).

http://dx.doi.org/10.1016/j.cpc.2015.02.009

References:

[1] L. Bernardin, P. Chin, P. DeMarco, K. O. Geddes, D. E. G. Hare, K. M. Heal, G. Labahn, J. P May, J. McCarron, M. B. Monagan, D. Ohashi and S. M. Vorkoetter, Maple Programming Guide, Toronto: Maplesoft, a division of Waterloo Maple, Inc., 2012.

[2] C. Gomez (Ed.), C. Bunks, J.-P Chancelier, M. Goursat, R. Nikoukhah and S. Steer, Engineering and Scientific Computing with Scilab, Birkhauser, 1999.

[3] C. Gomez, Macrofort: a fortran code generator in maple, INRIA report 119, 1990.

[4] P. Capolsini, C. Gomez, Macroc and Macrofort, C and FORTRAN code generation within Maple, MapleTech 3, no. 3, 1996.

[5] J. Dongarra, LAPACK Linear Algebra Package, http://www.netlib.org/lapack

1. Introduction

Whether it be in the areas of quantum chemistry, atomic and molecular physics, solid-state physics or even pharmacology, the calculation of atomic and molecular properties using *ab initio* quantum mechanical techniques is still an area of active and fundamental research. Moreover, the horizons of these areas have become broader over time. Quadruple and multiple precision are seriously contemplated. More than just an issue of getting more accuracy, computational quantum mechanics now looks beyond the ground state, beyond non-relativistic results, beyond time-independent problems and beyond the standard square-integrable boundary conditions over all space.

Today, we have a number of general tried-and-tested softwares such as GAUSSIAN [1], Turbomole [2], and MOLPRO [3], most of which are based on Gaussian functions and provide non-relativistic time-independent ground state energies within chemical accuracy. However, for greater accuracy, many have had to use specialized codes, often with specialized adapted basis sets that best match the natural properties of the wave function of a given system of interest. For example, in the case of relativistic electron structure calculations of small atoms and molecules, the variational approach has provided the most accurate energies for a comprehensive regime of quantum states by Drake and co-workers [4,5] as well as J.D. Morgan III and co-workers [6] using Hylleraas or Frankowski–Pekeris basis functions. We note that the integrals for the matrix elements for the latter involved special handling, a combination of manipulations by hand and by a CAS before conversion into numerical code.

Realizing that increased accuracy for solving the Schrödinger equation of a given system requires tweaking and specialization, it thus becomes desirable to establish *general* tools to create *specialized* software for handling individual systems. Thus, *code generation* tools based on *hybrid symbolic numerics* are of interest. The practical merits brought about by such an approach are huge. We can make a preliminary mathematical analysis of the given problem using a CAS and even consider solving a small trial version of the problem and use code generation tools to build up a larger numerical program for a *rapid prototyping* system like MATLAB or a fast numerical computation language like FORTRAN for larger calculations. In effect, this is a part of a natural evolution of quantum chemistry.

A CAS allows for multiple (high) precision but is very limited in speed and memory when compared to fast rapid prototyping programs or numerical computation languages. Not surprisingly, the largest industry of computation is therefore numerical. Note that Scilab is free and consequently an accessible alternative to MATLAB whose syntax is very similar and even identical for individual assignments.

An *ab initio* method which has gained notice was pioneered by Hiroshi Nakatsuji [7,8]. It encapsulated notions of full CI expansions (an accurate but computational intense procedure beyond the SCF Hartree–Fock approximation [9]) with Krylov subspaces (spaces spanned by the successive application of a particular operator $\hat{A}$ on a particular vector $\mathbf{b}$ [10]). Since its introduction, it has been applied to a number of small atoms and molecules with considerable success and accuracy (see [11,12] and references therein). Referred to as the "Free ICI" method (Free iterative configuration or complement interaction), the results have been impressive but there is little description of the computational details and timings required. Nakatsuji has claimed an accuracy as high as 32 and even 41–45 digits for the ground state of the Helium atom using a refined version of his approach, namely his "Free Complement" (FC) method [12]. Together with the benchmark of Charles Schwartz [13] at 45 digits, these are the most accurate non-relativistic ground state energies for the Helium atom. The goal of the present work is to provide *accessible tools* to replicate some of Nakatsuji's results and demystify some of his methods.

The present work is outlined as follows. Firstly, we outline Nakatsuji's method and identify its essentials. Since this is a variational approach, we then focus on optimizing its parameters, both linear and non-linear. Then we describe the symbolic tools for analysis and the code generation tools and outline our methodology. We then apply Nakatsuji's approach using these methods to three test cases i.e.:

1. Hydrogen Atom [7].
2. Helium Atom [8].
3. Hydrogen Atom in a Spherical cavity [14].

The first and third applications are done entirely within Maple. However, the second application involves a combination of Maple and a software for rapid prototyping and numerical computation, respectively MATLAB and FORTRAN. Nakatsuji's methods have evolved and thus the first two applications refer to his earlier work. Note that the third application is a novel application of Nakatsuji's method not seen before. These examples serve as relatively small demonstrations to computationally illustrate Nakatsuji's method and yield a rounded portrait of the hybrid-symbolic tools used. Concluding remarks are given at the end.