Numerical Analysis of Finite Difference Methods for Atoms

Charlotte Froese Fischer

August 19, 2019

1 Introduction

The equations that need to be solved for optimized Dirac-Hartree-Fock orbitals are of the form

$$w_a \begin{bmatrix} V(a;r)-\epsilon_{aa} & -c\left[\frac{d}{dr}-\frac{\kappa_a}{r}\right] \\ c\left[\frac{d}{dr}+\frac{\kappa_a}{r}\right] & V(a;r)-\epsilon_{aa}-2c^2 \end{bmatrix} \begin{bmatrix} P_a(r) \\ Q_a(r) \end{bmatrix} \tag{1}$$

$$= \sum_{b\neq a} \epsilon_{ab}\, \delta_{\kappa_a\kappa_b} \begin{bmatrix} P_b(r) \\ Q_b(r) \end{bmatrix}, \tag{2}$$

where $V(a;r) = V_{nuc}(r) + Y(a;r) + \bar{X}(a;r)$ is a potential consisting of nuclear, direct and exchange contributions arising from both diagonal and off-diagonal $\langle\Phi_\alpha|\mathcal{H}_{DC}|\Phi_\beta\rangle$ matrix elements [**?**]. In each κ-space, Lagrange related energy parameters $\epsilon_{ab} = \epsilon_{n_a n_b}$ are introduced to impose orthonormality constraints in the variational process.

When written as a system of equations it is customary to divide the equation by w_a , the generalized occupation number of the orbital. This is reasonable for a Dirac-Hartree-Fock calculation in which orbitals are occupied by one or more electrons, but in a multiconfiguration approximation, occupation numbers go to zero. In this case, the diagonal energy parameters for correlation orbitals become exceedingly large when, in fact, their contribution to the wave function becomes small. In this analysis we omit the division so that, for a given κ, the matrix of diagonal and off-diagonal energy parameters remains symmetric. In the above, the diagonal energy parameter is still the energy of a one-electron orbital.

2 The numerical grid.

Numerical grids are most efficient when grid points are equally spaced. Instead of varying the grid, a new variable is introduced, such that $t = f(r)$ and the grid is defined as $t_i = t_{i-1} + h$, $i = 2, \ldots, N$, where h is the step-size. Then t_1 is the closest grid point different from zero. In the 1980's the best numerical grid was not known but now the logarithmic grid for $t = \log(r)$ or, equivalently, the exponential grid $r = e^t$ are among the most efficient. With this transformation, a universal grid is easy to define as $t = log(Zr)$ so that for the hydrogenic equation, the same grid of equally spaced values of t are used. Note that, this transformation transforms the range $(0, \infty)$ to $(-\infty, \infty)$ so a small interval $(0, r_1)$ needs to be treated (if necessary) in the r variable and (t_1, t_p) defines the range of integration in the t

variable. Generally, at the origin, the asymptotic behaviour can be estimated and, because of the very small distances (in terms of the r variable), low-order methods can be used near the origin. Functions are assumed to be zero in the range (t_p, ∞) and a few more zeros can always be added to extend the range at large values of r.

Rounding errors are minimized if exact binary numbers can be used in generating the grid. The following defines an appropriate numerical grid. Note that step-size in the logarithic grid is $\bar{h} = log(1+h)$ or $e^{\bar{h}} = 1+h$.

```
!=====================================================================
   MODULE grid
!=====================================================================
      INTEGER, PARAMETER :: npt=600
      REAL(DOUBLE), PARAMETER :: H = 1.d0/16, RNT= 1.d0/2**20
      REAL(DOUBLE) :: eh,
      REAL(DOUBLE), DIMENSION(npt) :: R, RR, RM1, T
   CONTAINS
      SUBROUTINE  radial_grid(Z)
      REAL(double), intent (IN) :: z
      INTEGER :: i
      r(1) =  rnt/z        ; t(1) = log(r(1))   ; eh = log( (1+h)
      Do i 2, npt
        r(i) = (1+h)r(i-1) ; t(i) = t(i-1) + eh
      END Do
      rr = r*r; rm1=1.d0/r
   END MODULE grid
```

3 Transformation of the MCDHF equations

The transformation of the independent variable affects the transformation of MCDHF equations. With

$$t = log(r) \tag{3}$$

we have

$$\Delta t = (1/r)\Delta r \tag{4}$$

or

$$\frac{d}{dr} = \frac{1}{r}\frac{d}{dt} \tag{5}$$

Thus the MCDHF equations in the t-variable become

$$w_a \begin{bmatrix} V(a;t) - \epsilon_{aa} & -c\left[(\frac{1}{r})(\frac{d}{dt} - \kappa)\right] \\ c\left[(\frac{1}{r})(\frac{d}{dt} + \kappa)\right] & V(a;t) - \epsilon_{aa} - 2c^2 \end{bmatrix} \begin{bmatrix} P_a(t) \\ Q_a(t) \end{bmatrix} \tag{6}$$

$$= \sum_{b \neq a} \epsilon_{ab}\ \delta_{\kappa_a \kappa_b} \begin{bmatrix} P_b(t) \\ Q_b(t) \end{bmatrix}, \tag{7}$$

The equations here have been written in an eigenvalue form which can be implemented when orbitals are expanded in a basis such as B-splines. For numerical difference methods only direct terms can be included in $V(a,t)$ so that equations are solve with inhomogeneous terms as in

$$w_a \begin{bmatrix} D(a;t) - \epsilon_{aa} & -c\left[(\frac{1}{r})(\frac{d}{dt} - \kappa)\right] \\ c\left[(\frac{1}{r})(\frac{d}{dt} + \kappa)\right] & D(a;t) - \epsilon_{aa} - 2c^2 \end{bmatrix} \begin{bmatrix} P_a(t) \\ Q_a(t) \end{bmatrix} \tag{8}$$

$$= \begin{bmatrix} X_p(a,t) \\ X_q(a,t) \end{bmatrix} + \sum_{b \neq a} \epsilon_{ab}\, \delta_{\kappa_a \kappa_b} \begin{bmatrix} P_b(t) \\ Q_b(t) \end{bmatrix}, \tag{9}$$

GRASP2018 solves these systems of equations using integration methods. Briefly, if $y'(r) = f(r)$ then

$$y(b) - y(a) = \int_a^b f(r)dr \tag{10}$$

For equally spaced points,

$$\int_{t_{i-1}}^{t_{i+1}} f(r)dt = \frac{h}{90}\left(-f_{i+2} + 34f_{i+1} + 114f_i + 34f_{i-1} - f_{i-2}\right) + O(h^7) \tag{11}$$

Notice that this equation uses values outside the interval. At the origin, lower order methods can be used but equations whereas as at large r extra zero values may be introduced. Any modifications should result in a symmetric matrix. With this approach, the integration of the derivative is exact but integration of other functions is approximate. GRASP uses a lower fifth-order method.

But it would also be appropriate to approximate the derivatives using finite differences. A higher-order term derived from Stirling's formula is

$$F_i' = \frac{1}{60h}\left(F_{i+3} - 9F_{i+2} + 45F_{i+1} - 45Fi - 1 + 9F_{i-2} - F_{i-3}\right) \tag{12}$$

Applying these formulas to the MCDHF equations with their two point boundary conditions would produce linear systems of equations that are symmetric and could be solved using Lapack routines. Currently, the equations are solved using "shooting" methods (in and out integration with matching at a join). For homogeneous equations where a solution exists for only discrete eigenvalues, the matching of solutions can be used to predict the energy adjustment through solution continuty. Possibly such methods should still be used for occupied orbitals but for correlation orbitals with a large non-homogeneous term and no need for node counting, direct solution of the system of will overcome some of the present difficult.

This discussion has assumed that an estimated energy is known. If the linear system of equation for the unknowns is written as

$$w_a \begin{bmatrix} H^{pp} - \epsilon_{aa} & H^{pq} \\ H^{qp} & H^{qq} - \epsilon_{aa} \end{bmatrix} \begin{bmatrix} P_a \\ Q_a \end{bmatrix} = \begin{bmatrix} X_p \\ X_q \end{bmatrix}. \tag{13}$$

where P_a, Q_a are column vectors of the values of $P(a;r)$ and $Q(a;r)$ at the grid points. Then, given initial estimates of the orbitals, the value of $\epsilon_a = (P_a, Q_a) \cdot (H(P_a, Q_a) + (X_p, X_q))$.

In the above, the order of the (P, Q) column vector is $(P(1), P(2, \ldots, Q(1), Q(2) \ldots)$. The GRASP view is closely tied to a the differential equation view in which the order is $(P(1), Q(1), P(2), Q(2), \ldots$ and the matrix would consist of 2×2 matrices. It is not clear that the latter has any advantages.

Some studies would be useful. Most likely, two methods would be needed for solving the systems of equations along with one-electron orbital energies.

Experience to date indicates that the size of the system of equations will be less that 600 and could be solved as an eigenvalue problem, when homogeneous,and also when an initial estimate is available and the non-homogeneous term is small, as for occupied orbitals, and that systems of equations be solved for correlation orbitals an w_a is small.

4 The functions $Y^k(i, i'; r)$

Because $Y^k(i, i'; r)$ functions occur frequently in atomic structure calculations, it is desirable to compute then as efficiently as possible. By definition, the functions are defined as

$$Y^k(i,i';r) = \int_0^r \left(\frac{t}{r}\right)^k f(t)dt + \int_r^\infty \left(\frac{r}{t}\right)^{k+1} f(t)dt \tag{14}$$

or, equivalently,

$$Y^k(i,i';r) = Z^k(i,i';r) + W^k(i,i';r). \tag{15}$$

where $f(r) = P(i;r)P(i';r) + Q(i;r)Q(i';r)$. According to this definition, the calculation of Y^k functions requires the integration of integrands with factors t^k or t^{k+1}. For large k, the finite difference approximation may not be accurate when approximated by polynomials of lower degree.

Hartree was developing *equations with exchange* when Fock published his paper and thus was pre-empted. But he was also trying to find efficient methods for computing these Y^k functions using mechanical calculators before publishing the equations. These methods are computationally most efficient and stable methods today.

Hartree showed that the Y^k and Z^k functions are solutions of a pair of first-order equations, namely

$$\frac{d}{dr}Z^k(i,i';r) = f(r) - \frac{k}{r}Z^k(i,i';r), \quad Z^k(i,i';0) = 0 \tag{16}$$

$$\frac{d}{dr}Y^k(i,i';r) = \frac{1}{r}\left((k+1)Y^k(i,i';r) - (2k+1)Z^k(i,i';r)\right), \quad Y^k(i,i';\infty) = Z^k(i,i',\infty), \tag{17}$$

The first equation can be integrated outward, the second inward.

Transforming to the $t = log(Zr)$ variable and multiplying by r, we get

$$\frac{d}{dt}Z^k(i,i';t) = rf(t) - kZ^k(i,i';t) \tag{18}$$

$$\frac{d}{dt}Y^k(i,i';t) = (k+1)Y^k(i,i';t) - (2k+1)Z^k(i,i';t), \tag{19}$$

with boundary conditions $Z^k(i,i';\infty) = 0$ and $Y^k(i,i';\infty) = Z^k(1,i';\infty$, respectively.

It was shown by Worsley that these equations have simple integrating factors, namely, e^{kt} and $e^{-(k+1)t}$, respectively. On integrating, we get the equations

$$Z^k_{i+m} = e^{-m\bar{h}k}Z^k_i + \int_{t_i}^{t_{i+m}} f(t)e^{k(t-t_i)}dt, i = 1, 2, \ldots \tag{20}$$

$$Y^k_i = e^{-m\bar{h}(k+1)}Y^k_{i+m} + (2k+1)\int_{t_i}^{t_{i+m}} (Z^k e^{-(k+1)(t-t_i)}dt, i = \ldots 2, 1. \tag{21}$$

In other words, the first equation is integrated outwards and the second inwards. For our grid, since we have $e^{\bar{h}} = (1+h)$, parts of the calculations simplify and are essentially exact at the binary level. At the same time, if we also introduce the variable s such that $t = t_i + s$, then these simplify the equations some more, namely

$$Z^k_{i+m} = (1+h)^{-mk} Z^k_i + \int_0^{2\bar{h}} f(t) e^{k\bar{h}s} ds, i = 1, 2, \ldots \quad (22)$$

$$Y^k_i = (1+h)^{-m(k+1)} Y^k_{i+m} - (2k+1) \int_0^{2\bar{h}} Z^k e^{-(k+1)(s)} ds, i = \ldots 2, 1. \quad (23)$$

Both require an integration over m intervals. For the first non-zero interval (t_1, t_2), a formula for $m = 1$ can be used since, in terms of the r variable, the grid-points are very close together and a low order approximation may be sufficient. Thereafter, the ATSP package uses $m = 2$ along with the Simpson's rule with an error h^5 and can be improved (after the value at t_3 has been computed) with the consequence that the errors on the odd grid points are different from the error at the even grid point.

This algorithm for solving for $Y^k(1, i'; r)$ is stable for $k > 0$ in that, as the integration proceeds, the error is reduced by the integration factor. The most sensitive case is $k = 0$.

For this value of k, the calculation at large r is an othonormality integral so that $Z^k(i, i'; \infty) = 1$ for $i = i'$ and 0 otherwise. Thus numerical errors are readily detected. The ATSP package corrected the $Z^0(i, i; t)$ values since in this case the error for large values of r is easy to determine from the last few values. An investigation is appropriate for the relativistic case.

5 Slater integrals

By definition, the Slater integral,

$$R^k(ij, i'j') = \int_0^\infty f(r) \frac{1}{r} Y^k(i, i'; r) dr \quad (24)$$

where $f(r) = P(j; r)P(j'; r) + Q(j; r)Q(j'; r)$. To my knowledge, there has been no effort made to evaluate the accuracy of Slater integrals. For the hydrogenic, point nucleus case, where exact orbitals are none. It would be interesting to compare the accuracy of the early GRASP with a revised algorithm.

In many applications, such as a configuration interaction, many Slater integrals are needed and often used many times, especially in large calculations. One effective strategy is to generate lists of all possible Slater integrals for a given orbital set and evaluate all integrals in advance. This is particularly effective in parallel runs since the task can easily be distributed among the processors.

Lists of integrals are ordered by k. Typically the integrals are ordered $i <= j; i' <= j'; i \leq i'$. GRASP generates values of these integrals effectively "on demand" in which Y^k integrals are computed repeatedly. A more efficient algorithm would be:

```
For i =1, n_orb
  For i' = i, n_orb
     Compute Y^k(i, i'; r)
     For j= i, n_orb
```

```
For j' = j, n_orb
  Compute  R^k(ij;i'j') integrals
```

Current practice in GRASP is to compute each R^k integral independently on demand using only integration which requires the generation of r^k factors for each integral. With orbitals up to $l = 6$, k may be as large as 12. Such factors are not needed for differential equation methods, making the differential equation method more efficient computationally.

6 Transverse Photon integrals

Closely related to Slater integrals, are the integrals associated with the transverse photon (TP) correction. In this case, there are many types of integrals and a similar scheme could be used. The structure could be something like

```
For i =1, n_orb
 For i' = i, n_orb
    Compute Bessel(i, i')
    Compute Z^k(i, i'; r)
    For j= i, n_orb
       For j' = j, n_orb
         Compute Bessel(j,j')
         Compute all types of TP(ij:i'j') integrals
```

The evaluation of Breit integrals on a logarithmic grid with differential equation methods needs to be evaluated.

There are many types of integrals for the transverse photon correction. The current implementation has 6 types of integrals. For the more complex integrals, GRASP computes all integrals essentially independently without reusing intermediate results. There are two basic routines, zkf.f90 and yzk.f90, that use algorithms based on $r**K$ factors and need to be reviewed.