Nuclear Dynamics with the Sky3D Code

P.D. Stevenson

Department of Physics, University of Surrey, Guildford, Surrey, GU2 7XH, UK

Abstract. A description is presented of how to use the Sky3D time-dependent Hartree-Fock code to calculate giant monopole resonances. This requires modification to the code, and a step-by-step guide of how to make the necessary modification is given. An example of how to analyse the output of the code to obtain quantities of physics interest is included. Together, the modifications and the post-processing are intended to serve as a typical example of how the code, which was designed to be extendable to particular users' needs, can be extended.

1 Introduction

The time-dependent Hartree-Fock (TDHF) method [1] has, in the last few years, become a sufficiently mature technique that it may be used routinely in rather realistic situations describing a range of nuclear dynamics. A recently-published code, *Sky3D* [2] has made available a general purpose TDHF code, representing nuclei on a coordinate-space grid with no symmetry resetrictions. The code uses the effective Skyrme interaction, making it in fact really a Time-Dependent Density Functional Code (TDDFT), though the name TDHF is so ubiquitous the literature even for such Skyrme-based approaches, that this slightly inaccurate nomenclature is kept.

The paper describing the code [2] is open access, and comes with a link to the *Computer Physics Communications* program library whence the code can be downloaded, along with a substantial users manual, which goes into considerable detail about the program's inner workings. That being the case, such details are not repeated here, but rather an example of out-of-the-box running (in the static case) is given, along with an example of extending the code, which will be required for some cases of interest. The present paper should be read in conjunction with the paper describing the code.

Before embarking on the examples calculations, some mention will first be given about the kinds of application, with some references, that the code could be (and in some cases has been) applied to. A more comprehensive review of nuclear TDHF and related methods furnishes further examples [1]. Certain kinds of excited states of single nuclei may be explored. The most common example is that of giant multipole resonance states in the linear regime [3–11]. More sophisticaed examples of giant resonance type calculations include studying non-linearities and mixed strength functions [12, 13], skin vibrations [14], spining toroidal configurations [15] and clustered excitations [16, 17]. A considerable

body of literature describes the use of TDHF for nuclear dynamics involving more than one body. Some recent examples include cases of fusion [21–23], transfer reactions [24,25], fission [26–28], fragmentation [29] and deep-inelastic collisions [30, 31]. All such examples may be explored using the TDHF code, though in many cases, user-extensions to the code must be written. An example of such an extension – the case of giant monopole resonances – is presented in this paper.

2 Static Calculations

In order to perform a TDHF calculation, an initial state is required. This is obtained by running the *Sky3D* code in static mode, where it operates as a standard Skyrme-Hartree-Fock code, similar to other available codes such as *HFODD* [32], though much less fully-featured as such dedicated static solvers, as *Sky3D* is geared only to providing simple starting states for time-dependent calculations.

A suitable input file to generate the ground state of ^{18}O is given below:

```
&files wffile='O18' /
&force name='SkI4', pairing='NONE' /
&main mprint=100,mplot=100,
 mrest=100,writeselect='r',
 imode=1,tfft=T /
&grid nx=24,ny=24,nz=24,dx=1.0,dy=1.0,dz=1.0,
 periodic=F /
&static nprot=8, nneut=10,
 radinx=3.1,radiny=3.1,radinz=3.1,
 x0dmp=0.40,e0dmp=100.0,tdiag=T,tlarge=F,
 maxiter=2000,serr=2D-8 /
```

The instruction `imode=1` sets the operation of the code to be static Hartree-Fock. A particular Skyrme force (SkI4 [33]) is chosen, and the number of protons and neutrons set. For most purposes (except the inclusion of pairing), other parameters given in the input file for static runs can be left as in the file presented here.

The correct convergence of the wave functions to the true Hartree-Fock states in the static run can be checked in the file `conver.res`. One should take some care to ensure that not only has the total Hartree Fock energy converged, but that the fluctuations in the the single particle states are converged to small numbers. Figure 1 shows the convergence of the total energy for the input file given above, along with the average single particle energy fluctuation as given by the expression

$$\sqrt{\langle\psi|\hat{h}^2|\psi\rangle - \langle\psi|\hat{h}|\psi\rangle^2}. \tag{1}$$

The resulting wave functions are stored in the output file given by `wffile` in the above input file, i.e. `O18` in this case.
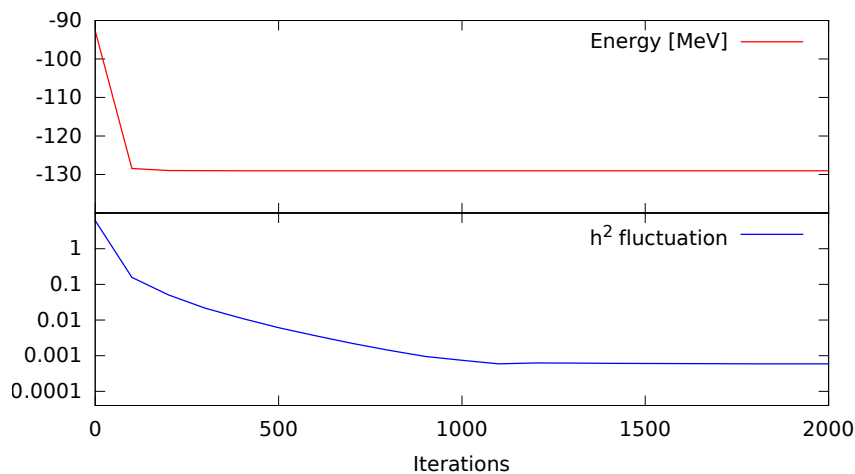
Figure 1. Convergence of the total Hartree-Fock energy (top panel) and in the fluctuations in the single particle wave functions (lower panel) for the case of ^{18}O.

3 Dynamic Calculations and Code Extension

The version of *Sky3D* that comes from the Computer Physics Communications library includes the ability to apply an external field, but the only multipole operator is the quadrupole Q_{20} tensor. Here, an example is given of how to extend the code to include monopole resonances, along with a sample run.

The file `external.f90` contains the relevant code which gives the multipolar boosts to the nucleus. To adapt it to include monopole resonances, the following steps are needed

First, a new variable should be declared to denote the strength of the monopole kick. In analogy to the existing strength variable for the Q_{20} kick, `amplq0`, we declare and initialise `amplm=0.0_db` in the same declaration statement in which `amplq0` is declared.

Second, `amplm` should be added to the `extern` namelist so that the user can input the strength of the monopole boost in the input file.

Third, the boost operator itself should be applied. The quadrupole boost is set up in the initialisation subroutite `getin_external`, via the generation of the array `extfield` which contains the external field to be applied to the nucleus. The monopole boost can simply be added to this field, so that in principle one may simultaneously apply both monopole and quadrupole boosts. The spatial form of the monopole boost is given conventionally as

$$f_M(r) \propto r^2. \tag{2}$$

In the code, the variable `facr` works out the relevant boost for a particular coordinate (inside a nested `DO`-loop over all spatial points). Separate neutron and

proton factors are calculated to deal with isovector and isocalar boosts, which we can make use of.

There are two different options for how the field is applied - *periodic* or *damped*, corresponding to one that matches periodic boundary conditions, and one that is masked around the nucleus via a three-dimensional Fermi function. Only one method is used in a particular run, and for the purposes of this extension, the monopole is implemented for the damped method. This is accomplished by changing lines 63 and 64 from

```
facr=amplq0 *(2.D0*z(iz)**2-x(ix)**2-y(iy)**2) &
   /(1.0D0+EXP((SQRT(x(ix)**2+y(iy)**2+z(iz)**2)-radext)/widext))
```

to

```
  facr=(amplq0 *(2.D0*z(iz)**2-x(ix)**2-y(iy)**2) &
     +amplm * (x(ix)**2+y(iy)**2+z(iz)**2) ) &
   /(1.0D0+EXP((SQRT(x(ix)**2+y(iy)**2+z(iz)**2)-radext)/widext))
```

These changes suffice to initialise the nucleus in a monopole kick. The extent and form of the masking function are determined by the `radext` and `widext` variables. A suitable input file to calculate a isoscalar giant monopole interaction is

```
&files wffile='restart'/
&force name='SkI4', pairing='NONE' /
&main mprint=20,mplot=100,
 mrest=100,writeselect='rc',
 imode=2,tfft=T,nof=1 /
&grid nx=24,ny=24,nz=24,dx=1,dy=1,dz=1,
periodic=F /
&dynamic nt=5000, dt=0.2, mxpact=6, texternal=T/
&extern ipulse=0,isoext=0,amplm=0.001,radext=4.0,
        widext=1.0, textfield_periodic=F /
&fragments filename=1*'../static/O18',fix_boost=T,
     fcent(1,1)=0,0,0    /
```

In this input file, we are running in `imode=2`, meaning a time-dependent calculation. The `&dynamic` namelist controls the total number of time steps (`5000`) along with the time step-size (`0.2 fm/c`)

The output of the code most useful for analysing the response of the nucleus to the isoscalar monopole boost are the files containing the various moments, by default called `monopoles.res`, `dipoles.res`, and `quadrupoles.res`. For the basic analysis, that would e.g. lead to the strength function, one typically follows the same kind of moment as provided the kick. The Fourier transform of this moment is essentially the strength function for the giant resonance [34], as given by linear response theory;

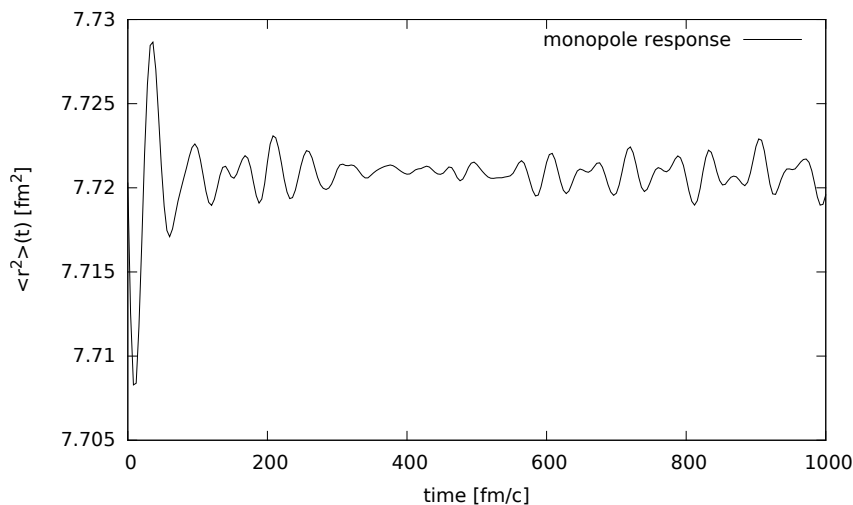$$S(E) = \sum_{\nu} |\langle \nu | F | 0 \rangle|^2 \delta(E - E_\nu). \tag{3}$$

27

Figure 2. The time-dependent expectiation value of r^2 in response to a monopole kick in ^{18}O.

The contents of `monopoles.res` file includes, out of the box, the time, the r.m.s. neutron radius, the r.m.s. proton radius, the matter radius and the isovector radius (i.e. the difference between proton and neutron radii). The matter radius is what is needed for the linear response, but the output in `monopoles.res` is too low in precision for the purposes of dealing with linear response in the small-amplitude limit. Of course, one can straightforwardly alter the code to produce enough precision in this file, but in fact the necessary data is available in the `quadrupoles.res` file, whose columns include the expectation values of x^2, y^2 and z^2 separately for protons and neutrons. For example, the column labelled `x^2(n)` gives the quantity $\frac{1}{N}\int d^3r\rho_n x^2$. If we denote this column x_n^2, and similarly for the other columns, then the response to the monopole kick can be calculated as

$$R(t) = (Nx_n^2 + Ny_n^2 + Nz_n^2 + Zx_p^2 + Zy_p^2 + Zz_p^2)/A. \tag{4}$$

This can hence be conveniently post-processed from the output without modification, and the result of this quantity is shown in Figure 2.

The figure shows the characteristic short-time damping of a resonant state followed by revenant vibrations after around $t = 400$fm/c due to rebounding of outgoing flux from the walls of the coordinate space box [35].

Finally, one would like to be able to turn the time-dependent response into a strength function. The *Sky3D* distribution comes with a utiliy program (in the testing foloder) called `spectral_analysis.f90`. It turns the time-dependent moment and transforms it to an energy spectrum. One needs to subtract the $t = 0$ value of the moment in order to centre it around zero.
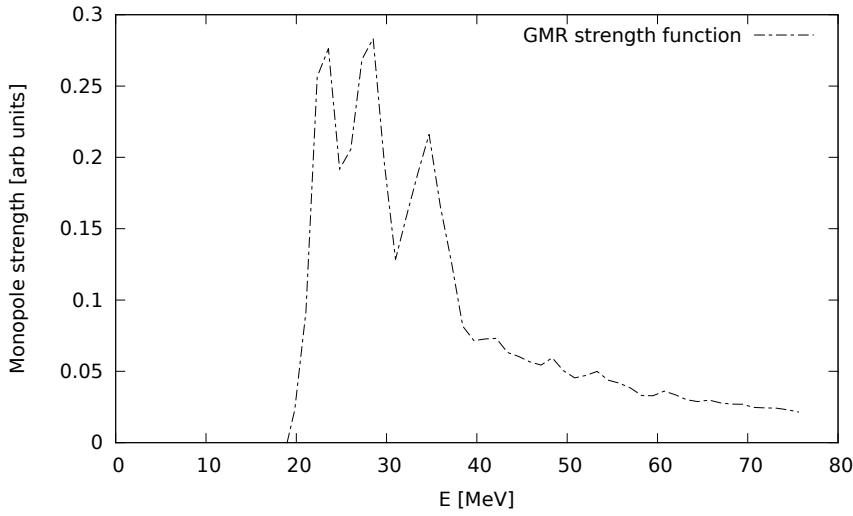
Figure 3. Monopole strength in ^{18}O using the SkI4 force

The resulting output of the code gives the strength function, as well as the power spectrum. A straightforward running of the expression (4) through the `spectral_analysis` code is shown in Figure 3.

4 Conclusion

This paper has reported on some aspects of the *Sky3D* code, in particular giving a practical example of how it can be extended to give useful functionality beyond what is present in the standard distribution, as well as giving an example of post-processing the output of the code. Extending to other forms of multipole excitations could be performed along very similar lines. The user manual [2] gives suggestions for other ways to extend the code.

Acknowledgements

References

[1] C. Simenel, *Eur. Phys. J. A* **48** (2012) 152;
http://dx.doi.org/10.1140/epja/i2012-12152-0

[2] J.A. Maruhn, P.-G. Reinhard, P.D. Stevenson and A.S. Umar, *Comput. Phys. Commun.* **85** (2014) 2195;
http://dx.doi.org/10.1016/j.cpc.2014.04.008

[3] M. Tohyama and A.S. Umar, *Phys. Lett. B* **549** (2002) 72;
http://dx.doi.org/10.1016/S0370-2693(02)02885-X

[4] P.D. Stevenson, M.R. Strayer, J. Rikovska Stone and W.G. Newton, *Int. J. Mod. Phys. E* **13** (2004);
http://dx.doi.org/10.1142/S0218301304001928

[5] T. Nakatsukasa and K. Yabana, *Phys. Rev. C* **71** (2005) 024301;
http://dx.doi.org/10.1103/PhysRevC.71.024301

[6] A.S. Umar and V. Oberacker, *Phys. Rev. C* **71** (2005) 034310;
http://dx.doi.org/10.1103/PhysRevC.71.034314

[7] M. P. Brine, P. D. Stevenson, J.A. Maruhn, and P.-G. Reinhard, *Int. J. Mod. Phys. E* **15** (2006) 14171423;
http://dx.doi.org/10.1142/S0218301306005009

[8] P.D. Stevenson, D. Almehed, P.-G. Reinhard and J.A. Maruhn, *Nucl. Phys. A* **788** (2007) 343c;
http://dx.doi.org/10.1016/j.nuclphysa.2007.01.091

[9] T. Nakatsukasa and K. Yabana, *Nucl. Phys. A* **788** (2007) 349c;
http://dx.doi.org/10.1016/j.nuclphysa.2007.01.064

[10] T. Inakura, T. Nakatsukasa and K. Yabana, *Int. J. Mod. Phys. E* **18** (2009) 2088;
http://dx.doi.org/10.1142/S0218301309014342

[11] C.I. Pardi and P.D. Stevenson, in: *Nuclear Theory* **32**, *Proceedings of the 32nd International Workshop on Nuclear Theory*, ed. A. Georgieva and N. Minkov (Heron Press, Sofia, 2013) p3.

[12] C. Simenel and Ph. Chomaz, *Phys. Rev. C* **68** (2005) 024302;
http://link.aps.org/doi/10.1103/PhysRevC.68.024302

[13] D. Almehed and P.D. Stevenson, *AIP Conference Proceedings* **802** (2005) 305.

[14] P.M. Goddard, N. Cooper, V. Werner, G. Rusev, P. D. Stevenson, A. Rios, C. Bernards, A. Chakraborty, B. Crider, J. Glorius, R.S. Ilieva, J.H. Kelley, E. Kwan, E.E. Peters, N. Pietrella, R. Raut, C. Romig, D. Savran, L. Schnorrenberger, M.K. Smith, K. Sonnabend, A.P. Tonchev, W. Tornow and S.W. Yates, *Phys. Rev. C* **88** (2013) 064308;
http://link.aps.org/doi/10.1103/PhysRevC.88.064308

[15] T. Ichikawa, J.A. Maruhn, N. Itagaki, K. Matsuyanagi, P.-G. Reinhard and S. Ohkubo, *Phys. Rev. Lett.* **109** (2012) 232503; http://dx.doi.org/10.1103/PhysRevLett.109.232503

[16] T. Ichikawa, J.A. Maruhn, N. Itagaki and S. Okhubo, *Phys. Rev. Lett.* **107** (2011) 112501;
http://dx.doi.org/10.1103/PhysRevLett.107.112501

[17] T. Ichikawa, N. Itagaki, N. Loebl, J.A. Maruhn, V.E. Oberacker, S. Ohkubo, B. Schuetrumpf and A. S. Umar, *EPJ Web of Conferences* **17** (2011) 07002;
http://dx.doi.org/10.1051/epjconf/20111707002

[18] R. Keser, A.S. Umar and V.E. Oberacker, *Phys. Rev. C* **85** (2012) 044606;
http://dx.doi.org/10.1103/PhysRevC.85.044606

[19] A.S. Umar and V.E. Oberacker, *Phys. Rev. C* **74** (2006) 061601; http://dx.doi.org/10.1103/PhysRevC.74.061601

[20] C. Simenel and B. Avez, *Int. J. Mod. Phys. E* **17** (2008) 31;
http://dx.doi.org/10.1142/S0218301308009525

[21] A.S. Umar, M.R. Strayer and P.-G. Reinhard, *Phys. Rev. Lett.* **56** (1986) 2793;
http://dx.doi.org/10.1103/PhysRevLett.56.2793

[22] A. S. Umar, C. Simenel and V. E. Oberacker, *Phys. Rev. C* **89** (2014) 034611;
http://dx.doi.org/10.1103/PhysRevC.89.034611

[23] C. Simenel, R. Keser, A.S. Umar and V. E. Oberacker, *Phys. Rev. C* **88** (2013) 024617;
http://dx.doi.org/10.1103/PhysRevC.88.024617

[24] C. Simenel, *Phys. Rev. Lett.* **105** (2010) 192701;
http://link.aps.org/doi/10.1103/PhysRevLett.105.192701

[25] K. Sekizawa and K. Yabana, http://link.aps.org/doi/10.1103/PhysRevC.88.014614*Phys. Rev. C* **88**, 014614 (2013)

[26] A.S. Umar, V.E. Oberacker, J.A. Maruhn and P.-G. Reinhard, *J. Phys. G* **37** (2010) 064037;
http://dx.doi.org/10.1088/0954-3899/37/6/064037

[27] P.M. Goddard, *PhD thesis, University of Surrey* (2014)

[28] C. Simenel and A.S. Umar, *Phys. Rev. C* **89** (2014) 031601;
http://dx.doi.org/10.1103/PhysRevC.89.031601

[29] Y. Iwata, T. Otsuka, J.A. Maruhn and N. Itagaki, *EPJ Web of Conferences* **2** (2010) 13002;
http://dx.doi.org/10.1051/epjconf/20100213002

[30] P.D. Stevenson, Sara Fracasso and E.B. Suckling, *J. Phys. Conf. Ser.* **381** (2012) 012105;
http://dx.doi.org/10.1088/1742-6596/381/1/012105

[31] C. Golabek and C. Simenel, *Phys. Rev. Lett.* **103** (2009) 042701;
http://link.aps.org/doi/10.1103/PhysRevLett.103.042701

[32] N. Schunck, J. Dobaczewski, J. McDonnell, W. Satuła, J.A. Sheikh, A. Staszczak, M. Stoitsov and P. Toivanen, *Comput. Phys. Commun.* **183** (2013) 166;
http://dx.doi.org/10.1016/j.cpc.2011.08.013

[33] P.-G. Reinhard and H. Flocard, *Nucl. Phys. A* **584** (1995) 467;
http://dx.doi.org/10.1016/0375-9474(94)00770-N

[34] D. Almehed and P.D. Stevenson, *J. Phys. G* **31** (2005) S1819; http://dx.doi.org/10.1088/0954-3899/31/10/079

[35] C.I. Pardi, P.D. Stevenson and K. Xu, *Phys. Rev. E* **89** (2014) 033312;
http://dx.doi.org/10.1103/PhysRevE.89.033312