General **Romber**g FORTRAN95 code - **RomberG**

Guillem Pey
gpey@iciq.es

September 3, 2025

Author's note: *This documentation and software is based on the paper "M. Medved et al. / Journal of Molecular Structure: THEOCHEM 847 (2007) 39–46"*

1 A brief summary on the Rutishauser-Romberg methodology and its generalization

The original Romberg methodology was originally developed for numerical integration to improve the accuracy of the trapezoidal rule by linearly combining lower order trapezoids.

Following Romberg's work, Rutishauser proposed using the latter for numerical differentiation by exploiting an error, as in the order of magnitude, in the central differences formula. For it let us consider the Taylor series expansion of a function f(x) but evaluated at a nearby point displaced by h, *i.e.* f(x+h) and f(x-h):

$$\begin{cases} f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) + \dots \\ f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) + \dots \end{cases} \tag{1}$$

And subtracting f(x-h) from f(x-h) yields:

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{2h^3}{3!}f^{(3)}(x) + \frac{2h^5}{5!}f^{(5)}(x) + \dots \tag{2}$$

$$D(h) = \frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{3!}f^{(3)}(x) + \frac{h^4}{5!}f^{(5)}(x) + \dots \tag{3}$$

$$D(h) \simeq f'(x) + \mathcal{O}(h^2) \tag{4}$$

Consequently, having an error from the second decimal. On the other hand, if we consider D(h) expansions now using a $\frac{h}{2}$ step we can redfine the definition of the first derivative with a lower error:

$$D(h) = f'(x) + Ah^2 + Bh^4 + \dots \tag{5}$$

$$D(h/2) = f'(x) + A\frac{h^2}{4} + B\frac{h^4}{16} + \dots \tag{6}$$

We can multiply by four the D(h/2) expansion and substract the D(h) one so the first derivative of f(x) is defined as:

$$4D(h/2) = 4f'(x) + B\frac{h^4}{4} + \dots \tag{7}$$

$$D(h) = f'(x) + Ah^2 + Bh^4 + \dots \tag{8}$$

$$4D(h/2) - D(h) = 3f'(x) + \mathcal{O}(h^4) \tag{9}$$

$$\frac{4D(h/2) - D(h)}{3} = f'(x) + \mathcal{O}(h^4) \tag{10}$$

and reducing the magnitude of the error to $\mathcal{O}(\mathrm{h}^4)$. Then, combining f(x) computed at different multiples of h the previous expression can be iterated to reduce higher order contamination, known as the Romberg triangle:

$$D^{p,k} = \frac{4^D P^{p-1,k} - D^{p-1,k+1}}{4^p - 1} \quad p = 1, 2, ... \tag{11}$$

where p is the number of the Romberg iteration, k is related to the distance of the initial points from x=0, and D the derivative of our interest.
Using the previous methodology of subtracting Taylor series expansions it can be derived the following set of formulae for up to the fourth-order derivative:

$$f^{(1)} \approx \frac{f(2^k h) - f(-2^k h)}{2^{k+1} h} \tag{12}$$

$$f^{(2)} \approx \frac{f(2^k h) - 2f(0) + f(-2^k h)}{(2^k h)^2} \tag{13}$$

$$f^{(3)} \approx \frac{f(2^{k+1} h) - 2f(2^k h) + 2f(-2^k h) - f(-2^{k+1} h)}{2(2^k h)^3} \tag{14}$$

$$f^{(4)} \approx \frac{f(2^{k+1} h) - 4f(2^k h) + 6f(0) - 4f(-2^k h) - f(-2^{k+1} h)}{(2^k h)^4} \tag{15}$$

where these formulae can only be applied for step sizes increasing in powers of two. The general set of formulae applicable for step sizes a∈[1,2], and the ones implemented in this code and computed on the fly based on the data, are the following:

$$f^{(1)} \approx \frac{f(a^k h) - f(-a^k h)}{2a^k h} \tag{16}$$

$$f^{(2)} \approx \frac{f(a^k h) - 2f(0) + f(-a^k h)}{(a^k h)^2} \tag{17}$$

$$f^{(3)} \approx 3 \cdot \frac{f(a^{k+1} h) - af(a^k h) + af(-a^k h) - f(-a^{k+1} h)}{a(a^2 - 1)(a^k h)^3} \tag{18}$$

$$f^{(4)} \approx 12 \cdot \frac{f(a^{k+1} h) - a^2 f(a^k h) + 2(a^2 - 1)f(0) - a^2 f(-a^k h) - f(-a^{k+1} h)}{a^2(a^2 - 1)(a^k h)^4} \tag{19}$$

Since this software is aimed at computing the electric field derivatives of non-linear optical properties, *i.e.* the electronic energy, the dipole moment, the polarizability matrix, the first hyperpolarizability tensor, an so on, the definitions of the just stated properties read as

$$\mu_i := -\frac{\partial E(F)}{\partial F_i} \tag{20}$$

$$\alpha_{ij} := -\frac{\partial^2 E(F)}{\partial F_i \partial F_j} = \frac{\partial \mu_i}{\partial F_j} \tag{21}$$

$$\beta_{ijk} := -\frac{\partial^3 E(F)}{\partial F_i \partial F_j \partial F_k} = \frac{\partial^2 \mu_i}{\partial F_j \partial F_k} = \frac{\partial \alpha_{ij}}{\partial F_k} \tag{22}$$

$$\gamma_{ijkl} := -\frac{\partial^3 E(F)}{\partial F_i \partial F_j \partial F_k \partial F_l} = \frac{\partial^3 \mu_i}{\partial F_j \partial F_k \partial F_l} = \frac{\partial^2 \alpha_{ij}}{\partial F_k \partial F_l} = \frac{\partial \beta_{ijk}}{\partial F_l} \tag{23}$$

Hence, the derivatives coming minus-derivatives of the electronic energy define the the non-linear optical property. As a consequence, the derivatives Eqs.(19) when using the energy must be negative. The Romberg procedure from Eq.(11).

2 Prerequisits of RomberG.exe

RomberG.exe is a FORTRAN95 flag-based script. FORTRAN95 is not capable of reading flags alone, that is why this code has been built using this module. Therefore, the first requirement is having installed, or downloaded, in your working directory before compiling RomberG.f95.

Next is compiling RomberG.f95 along with the getopt module with the following command line:

```
$ gfortran f90getopt.F90 -ffree-line-length-none -o ROMBERG.exe ROMBERG.f95
```

The "-ffree-line-length-none" compiling flag is mandatory - looking at the source code the reader is going to see that it clearly surpasses the limit of 80 characters per line. Once executed the previous command line, you are going to have a ROMBERG.exe in your working directory.

Moreover, it is necessary to have all fchk files of the molecule in a directory named after the molecule. It is recommended to name the fchk files as: *basename_field-direction_magnitude-of-the-field*. For instance, a valid fchk filename is "mol3_X-0.0004" or "ethaney0.0064".

If the user desires to generate RomberG-ready .fchk files without any trouble, there is also the 'makeromberg.py' Python script. This script works under the assumption that there is a com and log file in the directory under this name-template 'sp(*name-of-the-molecule*)0.com' with the keyword 'Field=X+000':

```
%nproc=4
%mem=8GB
%chk=spH2O0.chk
%oldchk=H2O.chk
# B3LYP/6-311+G(d,p) Polar NoSymm guess=read geom=check Field=X+000
...
```

This simple example uses as the name of the molecule 'H2O'. For further details, the user may read the 'makeromberg' script.

3 Execution of RomberG.exe and other optionalities

The optionalities of RomberG.exe can be displayed executing the following command line:

```
$ ./ROMBERG.exe -h {name_of_the_molecule}
```

and it is going to print in screen the following:

```
Several optionalities are available for RomberG.exe:
-i, –input : Input property - Energy/energy/E ; Dipole/dipole/M ; Alpha/alpha/A ; Beta/beta/B
-o, –output : Output property - Dipole/dipole/M ; Alpha/alpha/A ; Beta/beta/B ; Gamma/gamma/G
-F, –totalfields : Number of total fields probed. Integer required.
-l, –longitudinal : Whether to compute longitudinal properties or not.
-I, –isotropic : Whether to compute isotropic properties or not.
-p, –printP : Print the derivatives when computing the Romberg triangle for each component.
-h, –help : Displays this message.

REMEMBER: The execution of RomberG.exe requires the name of the molecule, even for
displaying this message.
```

For instance, a regular command line execution could be

```
$ ./ROMBERG.exe -i alpha -o gamma -F 13 {name_of_the_molecule}
```

this exemple of command line is going get the elements of the polarizability matrix of *.fchk.

4 Output of RomberG.exe

RomberG.exe is designed not only to compute the formulae on Eqs.(19) but also computing isotropic- and longitudinal-based properties.

Without specifying either the flags of isotropic and/or longitudinal, only the Romberg triangles are going to be printed along with the tensors composed with the propoerties with the minimum Romberg error using the Gaussian's fchk format.

Specifying the longitudinal flag enables the comutation of the following properties:

$$\mu = \sqrt{\mu_x^2 + \mu_y^2 + \mu_z^2} \tag{24}$$

$$\alpha = \frac{\alpha_{xx} + \alpha_{yy} + \alpha_{zz}}{3} \tag{25}$$

And prints the respective output tensor following Gaussian's format-checkpoint format. Toggling the isotropic flag activates the computation of the two previous equations and the ones below. All of them are rederived according following permutation symmetry:

$$\Delta\alpha = \frac{\sqrt{2}}{2}((\alpha_{xx} - \alpha_{yy})^2 + (\alpha_{yy} - \alpha_{zz})^2 + (\alpha_{zz} - \alpha_{xx})^2 + 12(\alpha_{xy}^2 + \alpha_{yz}^2 + \alpha_{xz}^2)) \tag{26}$$

$$\beta_{vec} = \sqrt{\sum_i^{x,y,z} (\sum_j^{x,y,z} \beta_{ijj})^2} \tag{27}$$

$$\beta_{||} = \frac{3}{5\mu} \sum_{i,j}^{x,y,z} \mu_i \cdot \beta_{ijj} \tag{28}$$

$$\gamma_{||} = \frac{1}{5} \sum_{i,j}^{x,y,z} \gamma_{iijj} \tag{29}$$

$$\bar{\gamma} = \gamma_{xxxx} + \gamma_{yyyy} + \gamma_{zzzz} + 12 \cdot (\gamma_{xxyy} + \gamma_{xxzz} + \gamma_{yyzz}) \tag{30}$$

Since the code is open-source if the user wants to implement other formulae other than the ones presented here or correct the factors multiplying, they are free to modify the original code. For further corrections, implementations, corrections, requests, etc., feel free to contact through the email at the beginning of the document.

5 Summary of top-to-bottom execution of RomberG.exe

This section is a summary that covers from the optimization of the interest molecule to the execution of RomberG.exe

1. (*Recommended*) Optimization of the molecule. Use the basename of the molecule, *i.e.* H2O or CH3OH, for the following steps. It is also recommended to move the all the optimization files, except for the .chk, elsewhere.

2. Prepare the 'Polar NoSymm Field=X+000' calculation according to the last textbox of Section 2 and following the name convention - 'sp(NAME)0.com'. You may not read from the previous checkpoint file.

3. Execute makeromberg.py. For instance, $ python makeromberg.py H2O .
4. Submit the calculations to your cluster.
5. In the parent directory of the molecule calculations, compile RomberG.f95 as stated before and execute following the guidelines presented in Section 3.
6. You may delete the checkpoint files to save on disk space.

6 References

NO MIRAR, ENCARA S'HAN DE PREPRAR LES REFERÈNCIES

[1] wikipedia.org/wiki/Scrum_(software_development)
[2] scruminc.com/why-three-questions-in-daily-scrum/
[3] scruminc.com/new-scrum-the-book/
[4] hbr.org/2007/07/the-making-of-an-expert