An improved bead model method for calculation of hydrodynamic properties of rigid molecules of arbitrary shape

Supplemental Information

GRPY Userguide

P. J. Zuk[1,2,*], B. Cichocki[3], and P. Szymczak[3]

[1]Department of Biosystems and Soft Matter, Institute of Fundamental and Technological Research, Polish Academy of Sciences, Pawinskiego 5B, 02-106 Warsaw, Poland
[2]Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA
[3]Institute of Theoretical Physics, Faculty of Physics, University of Warsaw, Pasteura 5, 02-093 Warsaw
[*]Correspondence: pzuk@ippt.pan.pl

INTRODUCTION AND DESCRIPTION

The *GRPY* program is a *FORTRAN* code to calculate the hydrodynamic properties of rigid macromolecules using Generalized Rotne-Prager-Yamakawa approximation of hydrodynamic interactions (1, 2). The shape of a rigid macromolecule has to be represented with a set of beads having different radii that can overlap. The *GRPY* calculates amongst others, translational and rotational diffusion coefficients, rotational relaxation times, sedimentation coefficient, the position of mobility center and intrinsic viscosity of a macromolecule. For the user's convenience the *GRPY* accepts input in its own original format, *hydro++* program format (3, 4) and *US-SOMO* generated *bead_model* format (5–7). We also provide tools to create a hydrated atom (h-atom) bead model from pdb file.

AVAILABILITY, PERMISSIONS AND CITATIONS

The *GRPY* open source code is freely available for download and use under GNU General Public License version 3 from

- *http://www.fuw.edu.pl/ piotrek/sofware* website;
- GitHub *https://github.com/pjzuk/GRPY* repository with Zenodo listed DOI.

When using the program please cite

- this contribution (8);
- the generalized Rotne-Prager-Yamakawa method (1);
- the generalized Rotne-Prager-Yamakawa method for different sized beads (9);
- intrinsic viscosity of macromolecules with generalized Rotne-Prager-Yamakawa approximation (2).

PACKAGE CONTENT

The GRPY package contains

- user guide;
- source code for the single core *GRPY* program compilation including *LAPACK* procedures (10);
- source code for the parallel *plasmaGRPY* program compilation;
- compiled single core *GRPY* program for Linux and Windows;
- compiled parallel *plasmaGRPY* program for Linux;
- input examples in *GRPY*, *hydro++* and *bead_model* formats;
- PLASMA library (11) (needed to compile *plasmaGRPY*) installer.

Additional content

- scripts to generate a hydrated atom bead model from pdb file using the *MSMS* program (12).

PROGRAM COMPILATION AND EXECUTION

Precompiled executables

There are four precompiled executables ready to use after download:

- single core for Linux users:

```
GRPY/bin/GRPYLinux/GRPY.exe
```

- multi core for Linux users:

```
GRPY/bin/GRPYLinux/plasmaGRPY.exe
```

- single core 32bit version for Windows users:

```
GRPY/bin/GRPYWindows_x86/GRPY_x86.exe
```

- single core 64bit version for Windows users:

```
GRPY/bin/GRPYWindows_x64/GRPY_x64.exe
```

Compilation from source code

Source code for the single core version

```
GRPY/src/GRPY/GRPY.f
```

that needs additional BLAS/LAPACK routines, which are attached in the separate files

```
GRPY/src/lapackGRPY/lapack_dgsev.f
GRPY/src/lapackGRPY/lapack_dsyev.f
GRPY/src/lapackGRPY/lapblas_matinv.f.
```

For Linux users a suitable Makefile

```
Makefile
```

is provided to compile the *GRPY* program from the source using the *gfortran* compiler. To compile the program inside the `GRPY` folder execute

```
make -f Makefile linux
```

To clean/remove the compilation files

```
make -f Makefile clean
```

In addition, source code for the multiple core version is provided

```
GRPY/src/plasmaGRPY/plasmaGRPY.f
```

In order to compile *plasmaGRPY* program the PLASMA library has to be installed. The installer is provided in the

```
GRPY/ThirdParty/plasma-installer_2.8.0
```

folder. Please proceed with the instructions provided by the package and install the libraries in the default location. After successful installation of *PLASMA* library in order to compile *plasmaGRPY* run

```
make -f Makefile plasmaLinux
```

The users of other operating systems should modify these scripts for their needs.

PROGRAM INPUT

The input file for the *GRPY* program can be prepared in three ways

- original format of GRPY;
- *hydro*++ program format by de La Torre et al. (3, 4);
- *US-SOMO* generated *bead_model* format (5–7).

For all of the above cases, example input files are attached in the

```
GRPY/examples
```

folder.

GRPY input file

For each model of a rigid molecule a separate input file needs to be prepared. It consists of two blocks. The first block contains the general parameters of the model grouped in two columns. The first column contains the values of the parameters whereas the second column contains a short description. Importantly, the first column should have 32 characters. The first block contains

1. Model name - a string with the model name,
2. Temperature - solution temperature T in Celsius,
3. Solvent viscosity - solvent viscosity η_0 in Poise,
4. Molecular weight - molecular weight M_w in Daltons,
5. Specific volume - specific volume of macromolecule $\bar{v}$ in $\left[\frac{\text{cm}^3}{\text{g}}\right]$,
6. Unit length - unit length in which bead coordinates and radii are given (in [cm]),
7. Number of beads - number of beads N that the program will read from the list that follows (it cannot be larger than the number of listed beads)

The second block comprises four columns of floating numbers separated by spaces. In the ith row the first three columns contain the spatial coordinates (x_i, y_i, z_i) of the center of a bead, whereas the radius of the bead (a_i) is given in the fourth one. All four entries are expressed in the length unit specified in the first block. As an example we present an input file for a rigid dumbbell ($N = 2$) of two touching spheres of radii $a_1 = 20 \left[\text{Å}\right]$ and $a_2 = 30 \left[\text{Å}\right]$ with the smaller one at the origin of the coordinate system and the larger one placed at $x_2 = 0 \left[\text{Å}\right]$, $y_2 = 50 \left[\text{Å}\right]$, $z_2 = 0$. In the example given below there are three lines specifying bead coordinates, however the third line is not read by the program because the number of particles has been set to $N = 2$. On the other hand if N is larger than the number of provided lines the program will exit with an error. The particles are suspended in water of temperature 20° [C] and viscosity of 0.01 [P] (Poise). The molecular weight of the particle is 1000 [Da] and the specific volume is $0.9 \left[\frac{\text{cm}^3}{\text{g}}\right]$. The particle positions and radii are given in the units of $= 10^{-8}$ [cm].

```
rigid dumbbell                    Model name
20.                               Temperature
0.01                              Solvent viscosity
1.E+3                             Molecular weight
0.9                               Specific volume
1.0                               Solution density
1.E-8                             Unit length [cm]
2                                 Number of beads
  0.0     0.0     0.0     20.0
  0.0     50.0    0.0     30.0
  0.0     0.0     50.0    30.0
```

hydro++ input file

For the details about the structure of the *hydro++* input file please refer to the *hydro++* user manual (3, 4).

US-SOMO input file

For the details of the structure of the *bead_model* input file from *US-SOMO* program please refer to the user manual (5–7). The input file does not specify the temperature or viscosity of the suspension. The GRPY program makes the assumption that temperature is equal to $T = 20$ [C] and that the water viscosity is equal to $\eta_0 = 1$ [cP].

PROGRAM EXECUTION

For both Linux and Windows users the program needs to be executed from the command line. Here we give an example of the the single core *GRPY.exe* program execution

In case of *GRPY* input format, program is executed as

```
<path to GRPY.exe> <path to GRPY input file>
```

In case of *hydro++* input format the additional flag `-d` has to be added

```
<path to GRPY.exe> -d <path to hydro++ input file>
```

In case of *US-SOMO* generated *bead_model* format the additional flag `-u` has to be added

```
<path to GRPY.exe> -u <path to .bead_model input file>
```

In case of multi core *plasmaGRPY.exe* program the number of cores for calculation can be additionally specified using system variable `PROC_NUM`. If not specified, the program will use a single core. The usage of multiple different input files is the same as in the single core version. For example to run calculations using 5 cores on the *US-SOMO* file one enters

```
export PROC_NUM=5
<path to plasmaGRPY.exe> -u <path to .bead_model inupt file>
```

The number of cores used in calculation will not exceed the number of available physical cores. For details please refer to the PLASMA library user guide.

PROGRAM OUTPUT

All the information presented in the program output are the same for the *GRPY*, *hydro++* and *US-SOMO* inputs however the outputs for *GRPY* and *US-SOMO* inputs are displayed to the standard output whereas for the *hydro++* input the output is written to the file with filename `<file name for output file>-GRPY.dat`, where `<file name for output file>` is specified in the *hydro++* input file analogous to the original *hydro++* program.

Here we present an example of a program output from the calculations for the non-symmetric dumbbell, for which the input file is presented above and included in the *examples* folder

```
                                                                        GRPY program

                                                    Hydrodynamic properties of the macromolecule
                                           based on the Generalized Rotne-Prager-Yamakawa method

                                                         from the:     GRPY input file

                                                     Rotational diffusion coefficient:  3.514E+03 [s^-1]
                                           Rotational relaxation times associated with
                                                  the Brownian relaxation of a vector:
                                                               -> Relaxation time (1):  1.284E-04 [s]
                                                               -> Relaxation time (2):  1.284E-04 [s]
                                                               -> Relaxation time (3):  1.816E-04 [s]
                                           Rotational relaxation times associated with
                              the Brownian relaxation of a traceless symmetric tensor:
                                                               -> Relaxation time (1):  3.899E-05 [s]
                                                               -> Relaxation time (2):  6.053E-05 [s]
                                                               -> Relaxation time (3):  5.319E-05 [s]
                                                               -> Relaxation time (4):  5.319E-05 [s]
                                                               -> Relaxation time (5):  3.899E-05 [s]
                                                     Harmonic mean (correlation) time:  4.743E-05 [s]
                Sedimentation coefficient (Mw Dlt (1. - (vbar*rho))/(nA kB T)):  2.493E-03 [Svedberg]

                                            High frequency intrinsic viscosity eta oo:  2.584E+05 [cm^3/g]
                                             Zero frequency intrinsic viscosity eta 0:  2.837E+05 [cm^3/g]

                     calculated using the origin
                       of the coordinate system:
  0.000E+00  0.000E+00  0.000E+00
                          as the reference point
        in the standard reference frame:
        e1:  1.000E+00  0.000E+00  0.000E+00
        e2:  0.000E+00  1.000E+00  0.000E+00
        e3:  0.000E+00  0.000E+00  1.000E+00

                                           Translational diffusion coefficient:  8.251E-08 [cm^2/s]

                          6x6 diffusion matrix:        Dtt  Dtr

                                                       Drt  Drr

  9.137E-08  0.000E+00  0.000E+00     0.000E+00  0.000E+00  9.479E-03
  0.000E+00  6.480E-08  0.000E+00     0.000E+00  0.000E+00 -0.000E+00
  0.000E+00  0.000E+00  9.137E-08    -9.479E-03  0.000E+00 -0.000E+00

  0.000E+00  0.000E+00 -9.479E-03     2.753E+03  0.000E+00 -0.000E+00
  0.000E+00  0.000E+00  0.000E+00     0.000E+00  5.035E+03 -0.000E+00
  9.479E-03 -0.000E+00 -0.000E+00    -0.000E+00 -0.000E+00  2.753E+03

   calculated using the mobility center:
     x [cm]     y [cm]     z [cm]
  0.000E+00  3.443E-06  0.000E+00
                 as the reference point
                in the reference frame:
        e1:  0.000E+00  0.000E+00  1.000E+00
        e2:  1.000E+00  0.000E+00  0.000E+00
        e3:  0.000E+00  1.000E+00  0.000E+00

                                Translational diffusion coefficient:  6.075E-08 [cm^2/s]

                 6x6 diffusion matrix:        Dtt  Dtr

                                              Drt  Drr

  5.873E-08  0.000E+00  0.000E+00     0.000E+00  1.142E-18  0.000E+00
  0.000E+00  5.873E-08  0.000E+00    -2.283E-18  0.000E+00  0.000E+00
  0.000E+00  0.000E+00  6.480E-08     0.000E+00  0.000E+00  0.000E+00

  0.000E+00 -2.283E-18  0.000E+00     2.753E+03  0.000E+00  0.000E+00
  1.142E-18  0.000E+00  0.000E+00     0.000E+00  2.753E+03  0.000E+00
  0.000E+00  0.000E+00  0.000E+00     0.000E+00  0.000E+00  5.035E+03

                                    Radius of the sphere with equal:

                                Translational diffusion coefficient:  3.534E-06 [cm]
                                   Rotational diffusion coefficient:  3.579E-06 [cm]
                          Rotational relaxation times associated with
                                 the Brownian relaxation of a vector:
                                              -> Relaxation time (1):  4.988E-06 [cm]
                                              -> Relaxation time (2):  4.988E-06 [cm]
                                              -> Relaxation time (3):  5.598E-06 [cm]
                          Rotational relaxation times associated with
            the Brownian relaxation of a traceless symmetric tensor:
                                              -> Relaxation time (1):  3.352E-06 [cm]
                                              -> Relaxation time (2):  3.882E-06 [cm]
                                              -> Relaxation time (3):  3.718E-06 [cm]
                                              -> Relaxation time (4):  3.718E-06 [cm]
                                              -> Relaxation time (5):  3.352E-06 [cm]
                                                Mean relaxation time:  3.579E-06 [cm]
                           High frequency intrinsic viscosity eta oo:  3.447E-06 [cm]
                            Zero frequency intrinsic viscosity eta 0:  3.557E-06 [cm]
```

The output file provides the following information

1. Rotational diffusion coefficient D^r - where $D^r = \frac{1}{3}\mathrm{Tr}\mathbf{D}^r$ with $\mathbf{D}^r$ defined as in Eq. (8) of Ref. (8);
2. Rotational relaxation times for rank 1 tensor τ_i - calculated in Eq. (18) of Ref. (8);
3. Rotational relaxation times for rank 2 tensor f_i^{-1}: - calculated as in Eq. (C.1) of Ref. (8);
4. Harmonic mean (correlation) time: - harmonic mean calculated as in Eq. (32) of Ref. (8);
5. Sedimentation coefficient s - calculated as in Eq. (22) of Ref. (8);
6. High-frequency intrinsic viscosity $[\eta]_\infty$ - calculated as in Eq. (27) of Ref. (8);

7. Zero-frequency intrinsic viscosity $[\eta]_0$ - calculated as in Eq. (28) of Ref. (8);
8. Translational diffusion coefficient $D_s^t(\mathbf{R}_0)$ calculated as in Eq. (19) of Ref. (8) in the original coordinate system (in which the positions of the beads were given in the input file)
9. 6×6 diffusion matrix divided into four submatrices $\mathbf{D}^{tt}, \mathbf{D}^{tr}, \mathbf{D}^{rt}, \mathbf{D}^{rr}$ calculated in the original coordinate system
10. Position of the mobility center of the particle $\mathbf{R}_{mc}$ see Ref. (8);
11. The reference frame $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3$ in which the rotational diffusion matrix $\boldsymbol{D}^{rr}$ is diagonal (given by the eigenvectors of $\boldsymbol{D}^{rr}$)
12. Long time diffusion coefficient $D_l^t = D_s^t(\mathbf{R}_{mc})$ equal to the short time diffusion coefficient calculated with mobility center as the reference point $\mathbf{R}_0 = \mathbf{R}_{mc}$;
13. 6×6 diffusion matrix divided into four submatrices $\mathbf{D}^{tt}, \mathbf{D}^{tr}, \mathbf{D}^{rt}, \mathbf{D}^{rr}$ calculated in the reference frame $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3$ with mobility center as the reference point $\mathbf{R}_0 = \mathbf{R}_{mc}$;
14. List of radii of equivalent spheres that have the same hydrodynamic properties as the macromolecule

HYDRATED ATOM BEAD MODEL

In addition to the *GRPY* program we provide scripts that can be used to construct the hydrated atom (h-atom) model described in the main text. The scripts are present in the

```
hydrAtom
```

folder. This is not a stand alone code and surface detection has to be done with MSMS program (12). The script

```
hydrAtom/pdbToHydrAtom
```

can be used to convert the pdb file to the *.xyzr* file containing positions and radii of surface atoms with the added hydration layer. Inside this script absolute paths to the `hydrAtom` folder and folder containing *MSMS* program have to be specified. The script

```
hydrAtom/pdb_to_xyzrn_mod
```

is a slightly modified version of *pdb_to_xyzrn* script added to *MSMS* program, which allows it to be executed correctly from another directory using the *bash* shell. Python (version 2.7) script

```
hydrAtom/msmsVertNameTohydrAtom.py
```

chooses the surface atoms from the *.vert* file resulting from execution of the *MSMS* program and adds the hydration radius. In the very beginning of the file there is a dictionary with the amount of hydration waters associated with the residue. In case a residue which is not listed appears, the program does not hydrate the residue and the radii of constituent atoms do not change. Information about unknown residue is displayed to the standard error stream. Additionally if a *GRPY* header file (the first block of of native *GRPY* input file) is present the complete input file for *GRPY* is generated. An example for an *1znf.pdb* file is attached in the

```
hydrAtom/examples
```

folder. After specifying appropriate paths one can execute the script (*1znf.pdb* and *1znf.header* files are provided)

```
<path to pdbToHydrAtom script>  <path to 1znf.pdb file>
```

For example from inside the

```
hydrAtom
```

folder execute

```
./pdbToHydrAtom ./example/1znf.pdb
```

This will result in the output files

- `example/1znf_noHET.pdb` - pdb file without HETATM lines

- `example/1znf_noHET.xyzrn` - x,y,z,radius,name file from *pdb_to_xyzrn_mod* script
- `example/1znf_noHET.xyzrn_msms.face` and *1znf_noHET.xyzrn_msms.vert* files from *MSMS* program
- `example/1znf_hydAtom.xyzr` - x,y,z,radius file with hydrated radii of surface atoms
- `example/1znf_hydAtom_GRPY` - *GRPY* program input file

also the file `example/1znf_hydAtom_GRPY_out` resulting from the *GRPY* program is provided.

REFERENCES

1. Wajnryb, E., K. A. Mizerski, P. J. Zuk, and P. Szymczak, 2013. Generalization of the Rotne–Prager–Yamakawa mobility and shear disturbance tensors. *J. Fluid Mech.* 731:R3.

2. Zuk, P. J., B. Cichocki, and P. Szymczak, 2017. Intrinsic viscosity of macromolecules within the generalized Rotne–Prager–Yamakawa approximation. *J. Fluid. Mech.* 822:R2.

3. de la Torre, J. G., S. Navarro, M. L. Martinez, F. Diaz, and J. L. Cascales, 1994. HYDRO: a computer program for the prediction of hydrodynamic properties of macromolecules. *Biophys. J.* 67:530–531.

4. de la Torre, J. G., G. del Rio Echenique, and A. Ortega, 2007. Improved calculation of rotational diffusion and intrinsic viscosity of bead models for macromolecules and nanoparticles. *J. Phys. Chem. B* 111:955–961.

5. Rai, N., M. Nöllmann, B. Spotorno, G. Tassara, O. Byron, and M. Rocco, 2005. SOMO (SOlution MOdeler): differences between X-ray-and NMR-derived bead models suggest a role for side chain flexibility in protein hydrodynamics. *Structure* 13:723–734.

6. Brookes, E., B. Demeler, C. Rosano, and M. Rocco, 2010. The implementation of SOMO (SOlution MOdeller) in the UltraScan analytical ultracentrifugation data analysis suite: enhanced capabilities allow the reliable hydrodynamic modeling of virtually any kind of biomacromolecule. *Eur. Biophys. J.* 39:423–435.

7. Brookes, E., B. Demeler, and M. Rocco, 2010. Developments in the US-SOMO Bead Modeling Suite: New Features in the Direct Residue-to-Bead Method, Improved Grid Routines, and Influence of Accessible Surface Area Screening. *Macromol. Biosci.* 10:746–753.

8. Zuk, P. J., B. Cichocki, and P. Szymczak, 2017. GRPY - an accurate bead method for calculation of hydrodynamic properties of rigid biomacromolecules. *Biophys. J. (submitted)* .

9. Zuk, P. J., E. Wajnryb, K. A. Mizerski, and P. Szymczak, 2014. Rotne-Prager-Yamakawa approximation for different-sized particles in application to macromolecular bead models. *J. Fluid Mech.* 741:R5.

10. Anderson, E., Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, et al., 1999. LAPACK Users' guide. SIAM.

11. Agullo, E., J. Dongarra, B. Hadri, J. Kurzak, J. Langou, J. Langou, H. Ltaief, P. Luszczek, and A. YarKhan, 2009. Plasma users guide. Technical report, Technical report, ICL, UTK.

12. Sanner, M. F., A. J. Olson, and J.-C. Spehner, 1996. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers* 38:305–320.