libgrecp

a library for the evaluation of molecular integrals of the generalized effective core potential operator over Gaussian functions

A. V. Oleynichenko
A. V. Zaitsevskii

NRC "Kurchatov institute" – PNPI, Quantum Chemistry Laboratory

oleynichenko_av@pnpi.nrcki.ru
http://qchem.pnpi.spb.ru

23 December 2021

Bibliography: GRECP

generalized relativistic effective core potential

- Generation of Gaussian expansions of GRECPs:
  N. S. Mosyagin, A. V. Titov, Z. Latajka, *IJQC* 63, 1107 (1997)
  Generalized relativistic effective core potential: Gaussian expansions of potentials and pseudospinors for atoms Hg through Rn
- Theoretical grounds:
  A. V. Titov, N. S. Mosyagin, *IJQC* 71, 359 (1999)
  Generalized relativistic effective core potential: Theoretical grounds
- Accounting for Breit in GRECP:
  A. N. Petrov, N. S. Mosyagin, A. V. Titov, I. I. Tupitsyn, *J. Phys. B* 37, 4621 (2004)
  Accounting for the Breit interaction in relativistic effective core potential calculations of actinides
- QED model potentials:
  V. M. Shabaev, I. I. Tupitsyn, V. A. Yerokhin, *PRA* 88, 012513 (2013)
  Model operator approach to the Lamb shift calculations in relativistic many-electron atoms
- Pseudopotential library:
  http://www.qchem.pnpi.spb.ru/recp

Bibliography: integral evaluation

- L. E. McMurchie, E. R. Davidson, *J. Comp. Phys.* 44, 289 (1981)
  Calculation of integrals over ab initio pseudopotentials
- R. M. Pitzer, N. W. Winter, *IJQC* 40, 773 (1991)
  Spin-orbit (core) and core potential integrals
- C.-K. Skylaris *et al*, *CPL* 296, 445 (1998)
  An efficient method for calculating effective core potential integrals which involve projection operators
- R. Flores-Moreno *et al*, *J. Comp. Chem.* 27, 1009 (2006)
  Half-numerical evaluation of pseudopotential integrals
- R. A. Shaw, J. G. Hill, *JCP* 147, 074108 (2017)
  Prescreening and efficiency in the evaluation of integrals over *ab initio* effective core potentials
- R. A. Shaw, J. G. Hill, *J. Open Source Softw.*, 6(60), 3039 (2021)
  libecpint: A C++ library for the efficient evaluation of integrals over effective core potentials

Example: uranium atom

Consider the 64e small core pseudopotential for the U atom:

- outercore shells: 6sp, 5spd, 4$spdf$
- valence shells: 7sp, 6d, 5f

Transition energies, cm^{-1}		Absolute errors, cm^{-1}		
$5f^3 6d^1 7s^2 \rightarrow$	DFB	no Breit	GRECP	Val. RECP
$5f^3 7s^2 7p^1$	7516	-93	-1	-6
$5f^3 6d^2 7s^1$	13124	78	2	1
$5f^3 6d^1 7s^1 7p^1$	17200	14	1	-9
$5f^2 6d^2 7s^2$	4640	-779	53	551
$5f^2 6d^2 7s^1 7p^1$	23856	-764	54	543
$5f^4 7s^2$	15780	627	-45	-404
$5f^4 6d^1 7s^1$	30790	670	-42	-386
$5f^1 6d^3 7s^2$	31450	-1673	112	1231
$5f^1 6d^4 7s^1$	38781	-1550	115	1209

Data of N. S. Mosyagin

Implementations

		scalar	spin-orbit	outercore	open source	written in
ARGOS	1981	+	+	–	+	Fortran
MOLGEP	1991	+	+	+	–	Fortran
Turbomole	2005	+	+	–	–	Fortran
libECP	2015	+	–	–	+	C
libecpint	2021	+	–	–	+	C++
libgrecp	**2021**	+	+	+	+	**C**

- libgrecp is written in C99 *from scratch*
- testing: DIRAC, MOLGEP
- oriented at relativistic coupled cluster calculations
- no restrictions on max angular momentum of ECP and basis functions

Generalized relativistic effective core potential (GRECP)

$$
\begin{aligned}
\hat{U} = {} & U_{LJ}(r) \\
& + \sum_{lj} \left[U_{lj}(r) - U_{LJ}(r) \right] P_{lj} \\
& + \sum_{n_c} \sum_{lj} \left\{ \tilde{P}_{n_c lj} \left[U_{n_c lj}(r) - U_{lj}(r) \right] + \left[U_{n_c lj}(r) - U_{lj}(r) \right] \tilde{P}_{n_c lj} \right\} \\
& + \sum_{n_c n'_c} \sum_{lj} P_{n_c lj} \left[\frac{U_{n_c lj}(r) + U_{n'_c lj}(r)}{2} - U_{lj}(r) \right] P_{n'_c lj}
\end{aligned}
$$

- $P_l = \sum_m |lm\rangle \langle lm|$
- $P_{lj} = \sum_m |ljm\rangle \langle ljm|$
- $\tilde{P}_{n_c lj} = \sum_m |n_c ljm\rangle \langle n_c ljm|$
  $\rightarrow$ projectors onto outercore pseudospinors
  $\rightarrow$ depend on r

A. V. Titov, N. S. Mosyagin, *IJQC* 71, 359 (1999)

Generalized relativistic effective core potential (GRECP)

$$\hat{U} = U_L(r) + \sum_{l=0}^{L-1} [U_l(r) - U_L(r)] \; P_l + \sum_{l=1}^{L} \frac{2}{2l+1} U_l^{SO}(r) \; P_l \; \boldsymbol{\ell s}$$

$$+ \sum_{n_c} \sum_{l=0}^{L-1} \hat{U}_{n_c l}^{AREP} \; P_l + \sum_{n_c} \sum_{l=1}^{L} \frac{2}{2l+1} \hat{U}_{n_c l}^{SO} \; P_l \; \boldsymbol{\ell s}$$

$$\hat{U}_{n_c l}^{AREP} = \frac{l+1}{2l+1} \hat{V}_{n_c, l+} + \frac{l}{2l+1} \hat{V}_{n_c, l-}$$

$$\hat{U}_{n_c l}^{SO} = \frac{2}{2l+1} \left[\hat{V}_{n_c, l+} - \hat{V}_{n_c, l-} \right]$$

$$\hat{V}_{n_c lj} = (U_{n_c lj} - U_{lj}) \tilde{P}_{n_c lj} + \tilde{P}_{n_c lj} (U_{n_c lj} - U_{lj}) - \sum_{n'_c} \tilde{P}_{n_c lj} \left[\frac{U_{n_c lj} + U_{n'_c lj}}{2} - U_{lj} \right] \tilde{P}_{n'_c lj}$$

A. V. Titov, N. S. Mosyagin, *IJQC* 71, 359 (1999)

Generalized relativistic effective core potential (GRECP)

12e-GRECP for Si by N.S.Mosyagin from 05.12.20
Pseudospinors from the 3s^2 3p^1 state

$|n_c lj\rangle$ *outercore pseudospinors*

```
  6    1                          2S1/2
 453.8239473297523       0.1534079305324878E-002
 81.78392245467846       0.1547338287858494E-001
 21.01756207682070       0.1653042571459425
 4.202187689530340      -0.4291694837841568
 1.832146025189554      -0.5564264215384425
0.7999057373996928      -0.1421685415780728
  9    2                          2P1/2                      2P3/2
 477.3315470425202       0.2113334997386953E-002  0.1918499782684725E-002
 122.3174655681572       0.1313815837665885E-001  0.1275315418716990E-001
 42.97638073927048       0.5162336374570296E-001  0.5085810372485870E-001
 17.59838144506293       0.1418791938851203       0.1407082656626522
 7.853460360176514       0.2692253991716999       0.2681159808207592
 3.659295865768477       0.3514829463498897       0.3514526455630522
 1.750902647135136       0.2856034294056517       0.2870071457181363
0.8310003070152262       0.1013370041109672       0.1027876524838510
0.3160304480358770       0.6896179598715287E-002  0.7081729074463639E-002
```

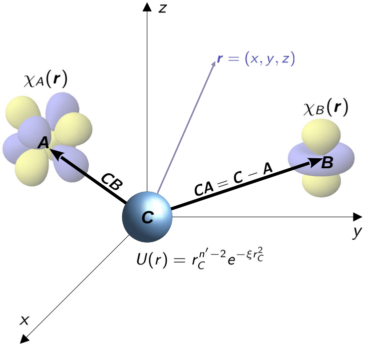
$U_l(r)$ / $U_L(r)$ / $U_l^{SO}(r)$ / $U_{nc,l}(r)$

```
 12    1                      3S-AREP                                      2S1/2
0   393.3602177650254     1.000000000000000     0.000000000000000     0.000000000000000
1   113.7058271536836     20.61434228311451     0.000000000000000     0.000000000000000
1   16.10913256549514     43.26766325656665     0.000000000000000     0.000000000000000
1   11.74489514767160    -26.00564060791052     0.000000000000000     0.000000000000000
1  0.7384624094940389    -1.767696976619511     0.000000000000000     0.000000000000000
2   7.435595881368535     12.81879139695621     0.000000000000000     0.000000000000000
2   1.704016034278843     1.724789989169324     0.000000000000000     0.000000000000000
2  0.8204571426674918     1.177798154185780     0.000000000000000     0.000000000000000
1   5.137451248761753     0.0                   0.000000000000000    .9380123541534502
1   3.653988234030431     0.0                   0.000000000000000   -1.454202516587702
2   6.495922514683006     0.0                   0.000000000000000     2.137038880995181
2   2.407546171026337     0.0                   0.000000000000000    .4924804917453065

 13    2                      3P-AREP               3P-ESOP               2P1/2                 2P3/2
1   6117.592663549806    -0.4205880832476154    0.4795688722317522     0.000000000000000     0.000000000000000
1   783.9977605773247    -0.8288329737421520     1.060775982167485     0.000000000000000     0.000000000000000
1   203.4274692639501    -0.6692270967733691    0.9407954603364033E-001 0.000000000000000    0.000000000000000
1   66.11279538566970    -0.5753535881882215    0.2999003218434382E-001 0.000000000000000    0.000000000000000
1   3.775507750937060     0.8202119815478000    0.5711114762129910E-001 0.000000000000000    0.000000000000000
1   2.117535857537166    -1.062330176459113     0.7979155991688458E-001 0.000000000000000    0.000000000000000
2   2.338276864774692     1.118051706876372     0.8489534538739463E-001 0.000000000000000    0.000000000000000
2  0.7630253652236829     0.4855751106136452E-003 0.1467293262304992E-003 0.000000000000000  0.000000000000000
2  0.9345397669109543E-001 0.4240612697226087E-004 0.2770340191290965E-005 0.000000000000000 0.000000000000000
2   6.592020250342957     0.                    0.                    .3430416383246921      0.3478030469624258
2   4.600761784919925     0.                    0.                   -0.5161737283857235    -0.5203252473759363
2   3.063252871982663     0.                    0.                    0.2219749178058450     0.2233588248259011
2   1.807398538859029     0.                    0.                   -0.1198529114654701E-001 -0.1290741233506232E-001
```

Semilocal part. Formulation of the problem

$$\hat{U} = U_L(r) + \sum_{l=0}^{L-1} [U_l(r) - U_L(r)] \; P_l + \sum_{l=1}^{L} \frac{2}{2l+1} U_l^{SO}(r) \; P_l \; \boldsymbol{\ell s}$$



$$\chi_A(\boldsymbol{r}) = N_A \, x_A^{n_A} y_A^{l_A} z_A^{m_A} \; e^{-\alpha_A |\boldsymbol{r}-\boldsymbol{A}|^2}$$

$$\chi_B(\boldsymbol{r}) = N_B \, x_B^{n_B} y_B^{l_B} z_B^{m_B} \; e^{-\alpha_B |\boldsymbol{r}-\boldsymbol{B}|^2}$$

$$x_A = x - A_x$$

Matrix elements of three types are required:

- $\langle \chi_A | U(r_C) | \chi_B \rangle$
- $\langle \chi_A | U(r_C) \; P_l | \chi_B \rangle$
- $\langle \chi_A | U(r_C) \; P_l \; \hat{\boldsymbol{\ell}} \; P_l \; | \chi_B \rangle$

The McMurchie-Davidson algorithm

Type 1 integrals: $\langle \chi_A | U(r) | \chi_B \rangle$

$$U_{AB} = \int \chi_A(\boldsymbol{r})\ r_C^{n'-2}\ e^{-\xi r_C^2}\ \chi_B(\boldsymbol{r})\ d\boldsymbol{r}_C$$

Basic idea: we reexpand functions χ_A and χ_B at the $\boldsymbol{C}$ point:

$$\boldsymbol{r}_A = \boldsymbol{r}_C + \boldsymbol{CA}$$

$$e^{-\alpha_A r_A^2} = e^{-\alpha_A r_C^2} e^{-2\alpha_A\ \boldsymbol{r}_C\ \cdot \boldsymbol{CA}} e^{-\alpha_A |\boldsymbol{CA}|^2}$$

(the same for χ_B). We substitute into the integral:

$$U_{AB} = \frac{D_{ABC}}{4\pi} \int x_A^{n_A} y_A^{l_A} z_A^{m_A} x_B^{n_B} y_B^{l_B} z_B^{m_B}\ r_C^{n'-2}\ e^{-\alpha r_C^2}\ e^{\boldsymbol{k}\cdot\boldsymbol{r}_C}\ d\boldsymbol{r}_C$$

$\alpha = \alpha_A + \alpha_B + \xi$
$\boldsymbol{k} = -2(\alpha_A \boldsymbol{CA} + \alpha_B \boldsymbol{CB})$
$D_{ABC} = N_A N_B\ e^{-\alpha_A |\boldsymbol{CA}|^2 - \alpha_B |\boldsymbol{CB}|^2}$

The McMurchie-Davidson algorithm

Type 1 integrals: $\langle \chi_A | U(r) | \chi_B \rangle$

$$U_{AB} = \frac{D_{ABC}}{4\pi} \int x_A^{n_A} y_A^{l_A} z_A^{m_A} x_B^{n_B} y_B^{l_B} z_B^{m_B} \; r_C^{n'-2} \; e^{-\alpha r_C^2} \; e^{\mathbf{k}\cdot\mathbf{r}_C} \; d\mathbf{r}_C$$

We use the $x_A = x_C + CA_x$ identity and the binomial expansion:

$$U_{AB} = \frac{D_{ABC}}{4\pi} \sum_{a=0}^{n_A} \sum_{b=0}^{l_A} \sum_{c=0}^{m_A} \sum_{d=0}^{n_B} \sum_{e=0}^{l_B} \sum_{f=0}^{m_B} \binom{n_A}{a} \binom{l_A}{b} \binom{m_A}{c} \binom{n_B}{d} \binom{l_B}{e} \binom{m_B}{f} \times$$
$$\times CA_x^{n_A-a} CA_y^{l_A-b} CA_z^{m_A-c} CB_x^{n_B-d} CB_y^{l_B-e} CB_z^{m_B-f} \times$$
$$\times \int x_C^{a+d} y_C^{b+e} z_C^{c+f} \; r_C^{n'-2} \; e^{-\alpha r_C^2} \; e^{\mathbf{k}\cdot\mathbf{r}_C} \; d\mathbf{r}_C$$

The McMurchie-Davidson algorithm

Type 1 integrals: $\langle \chi_A | U(r) | \chi_B \rangle$

$$\int x_C^{a+d} y_C^{b+e} z_C^{c+f} \; r_C^{n'-2} \; e^{-\alpha r_C^2} \; e^{\boldsymbol{k} \cdot \boldsymbol{r}_C} \; d\boldsymbol{r}_C$$

The plane-wave expansion in real spherical harmonics:

$$e^{\boldsymbol{k}\boldsymbol{r}_C} = 4\pi \sum_{\lambda=0}^{\infty} \sum_{\mu=-\lambda}^{+\lambda} M_\lambda(kr_C) \; S_{\lambda\mu}(\hat{k}) \; S_{\lambda\mu}(\hat{r}_C)$$

$M_\lambda(x)$ – modified spherical Bessel functions of the 1st kind
$S_{\lambda\mu}$ – real spherical harmonics
$\hat{k} = \boldsymbol{k}/|\boldsymbol{k}|$, $\hat{r}_C = \boldsymbol{r}_C/|\boldsymbol{r}_C|$ – angular variables for the $\boldsymbol{k}$ and $\boldsymbol{r}_C$ vectors, respectively

We use $x_C = r_C \hat{x}_C$ (+ the same for other projections):

$$4\pi \sum_{\lambda=0}^{\infty} \underbrace{\int r_C^{a+b+c+d+e+f+n'} e^{-\alpha r_C^2} M_\lambda(kr_C) \; dr_C}_{=Q_\lambda^N \text{ – radial integral}} \underbrace{\int \sum_{\mu=-\lambda}^{+\lambda} \hat{x}_C^{a+b} \hat{y}_C^{b+e} \hat{z}_C^{c+f} S_{\lambda\mu}(\hat{k}) S_{\lambda\mu}(\hat{r_C}) \; d\hat{r_C}}_{=\Omega_\lambda^{a+d,b+e,c+f} \text{ – angular integral}}$$

The McMurchie-Davidson algorithm

Type 1 integrals: $\langle \chi_A | U(r) | \chi_B \rangle$

$$U_{AB} = D_{ABC} \sum_{a=0}^{n_A} \sum_{b=0}^{l_A} \sum_{c=0}^{m_A} \sum_{d=0}^{n_B} \sum_{e=0}^{l_B} \sum_{f=0}^{m_B} \binom{n_A}{a} \binom{l_A}{b} \binom{m_A}{c} \binom{n_B}{d} \binom{l_B}{e} \binom{m_B}{f} \times$$

$$\times CA_x^{n_A-a} CA_y^{l_A-b} CA_z^{m_A-c} CB_x^{n_B-d} CB_y^{l_B-e} CB_z^{m_B-f} \times$$

$$\times \sum_{\lambda=0}^{\infty} Q_\lambda^{a+b+c+d+e+f+n'}(k,\alpha)\ \Omega_\lambda^{a+d,b+e,c+f}(\hat{k})$$

Type 1 radial integrals:

$$Q_\lambda^N(k,\alpha) = \int_0^{+\infty} r^N\ e^{-\alpha r^2}\ M_\lambda(kr)\ dr$$

$$\boldsymbol{k} = -2(\alpha_A \boldsymbol{CA} + \alpha_B \boldsymbol{CB})$$

$$\alpha = \alpha_A + \alpha_B + \xi$$

Type 1 angular integrals:

$$\Omega_\lambda^{IJK}(\hat{k}) = \sum_{\mu=-\lambda}^{+\lambda} S_{\lambda\mu}(\hat{k}) \int \hat{x}^I \hat{y}^J \hat{z}^K\ S_{\lambda\mu}(\hat{r})\ d\hat{r}$$

The McMurchie-Davidson algorithm

Type 2 integrals: $\langle \chi_A | U(r) P_l | \chi_B \rangle$

$$U^l_{AB} = \int \chi_A(\mathbf{r})\ r_C^{n'-2}\ e^{-\xi r_C^2} \sum_m |S_{lm}\rangle \langle S_{lm}|\ \chi_B(\mathbf{r})\ d\mathbf{r}_C =$$

$$= 4\pi D_{ABC} \sum_{a=0}^{n_A} \sum_{b=0}^{l_A} \sum_{c=0}^{m_A} \sum_{d=0}^{n_B} \sum_{e=0}^{l_B} \sum_{f=0}^{m_B} \binom{n_A}{a} \binom{l_A}{b} \binom{m_A}{c} \binom{n_B}{d} \binom{l_B}{e} \binom{m_B}{f} \times$$

$$\times CA_x^{n_A-a} CA_y^{l_A-b} CA_z^{m_A-c} CB_x^{n_B-d} CB_y^{l_B-e} CB_z^{m_B-f} \times$$

$$\times \sum_{\lambda=0}^{\infty} \sum_{\bar{\lambda}=0}^{\infty} Q^{a+b+c+d+e+f+n'}_{\lambda\bar{\lambda}}(k_A, k_B, \alpha) \sum_{m=-l}^{+l} \Omega^{abc}_{\lambda lm}(\hat{k})\ \Omega^{def}_{\bar{\lambda} lm}(\hat{k})$$

Type 2 radial integrals:

$$Q^N_{\lambda\bar{\lambda}}(k_A, k_B, \alpha) = \int_0^{+\infty} r^N\ e^{-\alpha r^2}\ M_\lambda(k_A r)\ M_{\bar{\lambda}}(k_B r)\ dr$$

Type 2 angular integrals:

$$\Omega^{abc}_{\lambda lm}(\hat{k}) = \sum_{\mu=-\lambda}^{+\lambda} S_{\lambda\mu}(\hat{k}) \int \hat{x}^a \hat{y}^b \hat{z}^c\ S_{\lambda\mu}(\hat{r})\ S_{lm}(\hat{r})\ d\hat{r}$$

The McMurchie-Davidson algorithm

Type 3 integrals (spin-orbit): $\langle \chi_A | U(r) P_l \boldsymbol{\ell} P_l | \chi_B \rangle$

$$SO_{AB}^{l} = i^{-1} \int \chi_A(\boldsymbol{r})\, r_C^{n'-2}\, e^{-\xi r_C^2} \left(\sum_m |S_{lm}\rangle \langle S_{lm}| \right) \boldsymbol{\ell} \left(\sum_m |S_{lm}\rangle \langle S_{lm}| \right) \chi_B(\boldsymbol{r})\, d\boldsymbol{r}_C =$$

$$= 4\pi D_{ABC} \sum_{a=0}^{n_A} \sum_{b=0}^{l_A} \sum_{c=0}^{m_A} \sum_{d=0}^{n_B} \sum_{e=0}^{l_B} \sum_{f=0}^{m_B} \binom{n_A}{a} \binom{l_A}{b} \binom{m_A}{c} \binom{n_B}{d} \binom{l_B}{e} \binom{m_B}{f} \times$$

$$\times CA_x^{n_A-a} CA_y^{l_A-b} CA_z^{m_A-c} CB_x^{n_B-d} CB_y^{l_B-e} CB_z^{m_B-f} \times$$

$$\times \sum_{\lambda=0}^{\infty} \sum_{\bar{\lambda}=0}^{\infty} Q_{\lambda\bar{\lambda}}^{a+b+c+d+e+f+n'}(k_A, k_B, \alpha) \sum_{m=-l}^{+l} \sum_{m'=-l}^{+l} \Omega_{\lambda l m}^{abc}(\hat{k})\, \Omega_{\bar{\lambda} l m}^{def}(\hat{k})\, \langle S_{lm} | \boldsymbol{\ell} | S_{lm'} \rangle$$

- type 2 radial integrals
- type 2 angular integrals
- matrix of the orbital angular momentum operator $\boldsymbol{\ell}$ in the S_{lm} basis

Angular integrals

Type 1 integrals:

$$\Omega_{\lambda}^{IJK}(\hat{k}) = \sum_{\mu=-\lambda}^{+\lambda} S_{\lambda\mu}(\hat{k}) \sum_{rst}^{\lambda} y_{rst}^{\lambda\mu} \int \hat{x}^{I+r} \hat{y}^{J+s} \hat{z}^{K+t} \, d\hat{r}$$

Type 2 integrals:

$$\Omega_{\lambda lm}^{abc}(\hat{k}) = \sum_{\mu=-\lambda}^{+\lambda} S_{\lambda\mu}(\hat{k}) \sum_{rst}^{\lambda} \sum_{uvw}^{l} y_{rst}^{\lambda\mu} y_{uvw}^{lm} \int \hat{x}^{a+r+u} \hat{y}^{b+s+v} \hat{z}^{c+t+w} \, d\hat{r}$$

Spherical harmonic $S_{\lambda\mu}$ value at the $\hat{k}$ point:

$$S_{\lambda\mu}(\hat{k}) = \sum_{rst}^{\lambda} y_{rst}^{\lambda\mu} \; \hat{k}_x^r \hat{k}_y^s \hat{k}_z^t$$

Basic integrals are:

$$\int \hat{x}^i \hat{y}^j \hat{z}^k \, d\hat{r} = \begin{cases} 4\pi \frac{(i-1)!! \; (j-1)!! \; (k-1)!!}{(i+j+k+1)!!} & i, \; j, k \text{ even} \\ 0 & \text{otherwise} \end{cases}$$

Angular integrals

The y^{lm}_{rst} coefficients are calculated using the formula:

$$y^{lm}_{rst} = \sqrt{\frac{2l+1}{2\pi}\frac{(l-|m|)!}{(l+|m|)!}}\frac{1}{2^l l!}\sum_{i=j}^{(l-|m|)/2}\binom{l}{i}\binom{i}{j}\frac{(-1)^i(2l-2i)!}{(l-|m|-2i)!}\times$$

$$\times\sum_{k=0}^{j}\binom{j}{k}\binom{|m|}{r-2k}(-1)^{(|m|-r+2k)/2}\times$$

$$\times\begin{cases}1 & m>0 \text{ и } |m|-r \text{ even}\\ 1/\sqrt{2} & m=0 \text{ и } r \quad\quad \text{even}\\ 1 & m<0 \text{ и } |m|-r \text{ odd}\\ 0 & \text{otherwise}\end{cases}$$

$j = (r+s-|m|)/2$

y^{lm}_{rst} can be calculated only once and then tabulated

R. Flores-Moreno *et al*, *J. Comp. Chem.* 27, 1009 (2006)

Radial integrals

Type 1 radial integrals:

$$Q_{\lambda}^{N}(k,\alpha) = \int_{0}^{+\infty} r^{N}\ e^{-\alpha r^2}\ M_{\lambda}(kr)\ dr$$

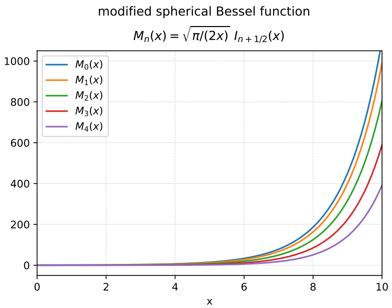
Type 2 radial integrals:

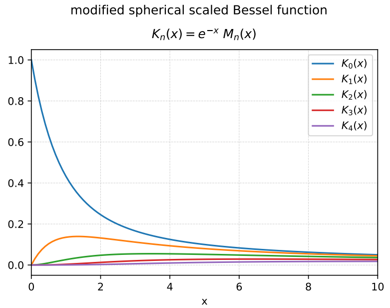
$$Q_{\lambda\bar{\lambda}}^{N}(k_A,k_B,\alpha) = \int_{0}^{+\infty} r^{N}\ e^{-\alpha r^2}\ M_{\lambda}(k_A r)\ M_{\bar{\lambda}}(k_B r)\ dr$$

$M_{\lambda}(x)$ – spherical modified Bessel functions
$\alpha = \alpha_A + \alpha_B + \xi$
$k_A = 2\alpha_A|\boldsymbol{CA}|$
$k_B = 2\alpha_B|\boldsymbol{CB}|$
$k = 2|\alpha_A \boldsymbol{CA} - 2\alpha_B \boldsymbol{CB}|$

L. E. McMurchie, E. R. Davidson, *J. Comp. Phys.* 44, 289 (1981)

Radial integrals





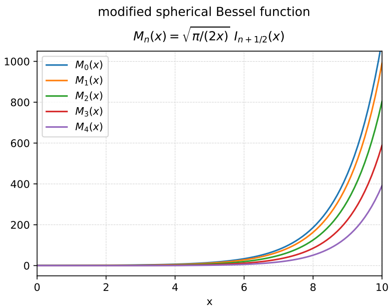
$$Q_\lambda^N(k,r) = \int_0^{+\infty} r^N\ e^{-\alpha r^2} M_\lambda(kr)\ dr$$

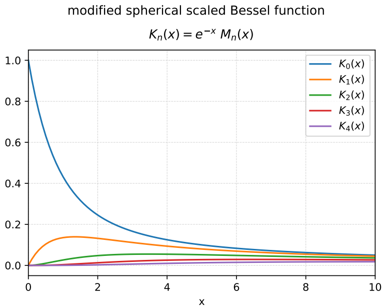
$$Q_\lambda^N(k,r) = \int_0^{+\infty} r^N\ e^{-\alpha r^2} e^{kr}\ \underbrace{e^{-kr} M_\lambda(kr)}_{K_\lambda(kr)}\ dr$$

$$e^{-\alpha_A |CA|^2} e^{-\alpha_B |CB|^2} Q_\lambda^N(k,r) = \int_0^{+\infty} r^N\ \underbrace{e^{-\alpha_A |CA|^2 - \alpha_B |CB|^2 - \alpha r^2 + kr}}_{\to 0}\ \underbrace{K_\lambda(kr)}_{\to 0}\ dr$$

R. Flores-Moreno *et al*, *J. Comp. Chem.* 27, 1009 (2006)

Radial integrals





Similarly for the type 2 radial integrals:

$$Q^N_{\lambda\bar{\lambda}}(k_A, k_B, r) = \int_0^{+\infty} r^N\ e^{-\alpha r^2} M_\lambda(k_A r)\ M_{\bar{\lambda}}(k_B r)\ dr$$

$$Q^N_{\lambda\bar{\lambda}}(k_A, k_B, r) = \int_0^{+\infty} r^N\ e^{-\alpha r^2} e^{k_A r} K_\lambda(k_A r)\ e^{k_B r} K_{\bar{\lambda}}(k_B r)\ dr$$

$$\int_0^{+\infty} r^N\ e^{-\alpha_A |CA|^2 - \alpha_A r^2 + k_A r} e^{-\alpha_B |CB|^2 - \alpha_B r^2 + k_B r} K_\lambda(k_A r)\ K_{\bar{\lambda}}(k_B r)\ dr$$

R. Flores-Moreno *et al*, *J. Comp. Chem.* 27, 1009 (2006)

The Log3 quadrature

The integral to be calculated:

$$I = \int_0^{+\infty} f(r)\ r^2\ dr$$

The integration grid consists of n_r points:

$$x_i = \frac{i}{n_r + 1}, \quad x_i \in (0,1)$$

$$r_i = -\alpha \ln(1 - x_i^3), \quad r_i \in (0, +\infty)$$

$$w_i = \frac{3\alpha^3 x_i^2 \ln^2(1 - x_i^3)}{(1 - x_i^3)(n_r + 1)}$$

$$I \approx \sum_i^{n_r} w_i\ f(r_i)$$

When expanding the grid to $n_r^{(2)} = n_r^{(1)} + 1$ points only the weights and $f(r)$ values at every second points are to be recalculated:

$$I^{(2)} \approx \frac{I^{(1)}}{2} + \sum_{i=1,3,5,\dots}^{n_r^{(2)}} w_i f(r_i)$$

Integral can be calculated with any pre-defined precision!

M. E. Mura, P. J. Knowles, *JCP*, 104, 9848 (1996); C.-K. Skylaris *et al*, *CPL* 296, 445 (1998)

Contracted ECPs and basis functions

Gaussian expansions are used for $U(r)$ in real calculations:

$$U(r) = \sum_i d_i \; r^{n_i - 2} \; e^{-\xi_i r^2}$$

Contracted Gaussian basis functions:

$$\chi_A(\mathbf{r}) = \sum_i c_i \; N_i \; x_A^n y_A^l z_A^m \; e^{-\alpha_i |\mathbf{r} - \mathbf{A}|^2} \qquad L_A = n + l + m$$

Radial integrals for contracted $U(r)$ and $\chi_A(\mathbf{r})$:

$$Q^N_{\lambda\bar{\lambda}} \quad \rightarrow \quad T^{N'}_{\lambda\bar{\lambda}} = \int_0^{+\infty} r^{N'+2} \; U(r) \; F^{\lambda}_A(r) \; F^{\bar{\lambda}}_B(r) \; dr$$

$$N' = 0, ..., L_A + L_B$$

$$F^{\lambda}_A(r) = \sum_i c_i \; N_i \; e^{-\alpha_A |\mathbf{CA}|^2 - k_{A,i} r^2} \; M_{\lambda}(k_{A,i} r)$$

- angular integrals do not depend on contractions!
- no advantages for the type 1 integrals Q^N_{λ}

R. Flores-Moreno *et al*, *J. Comp. Chem.* 27, 1009 (2006)

Contracted ECPs and basis functions

Algorithm

$$T^{N'}_{\lambda\bar{\lambda}} = \int_0^{+\infty} r^{N'}\, U(r)\, F_A^{\lambda}(r)\, F_B^{\bar{\lambda}}(r)\, r^2\, dr$$

$r_i^{N'}\, U(r_i)$	$F_A^{\lambda}(r_i)$	$F_B^{\bar{\lambda}}(r_i)$
$N' = 0, \ldots, L_A + L_B$	$\lambda = 0, \ldots, L_A + L_{ECP}$	$\bar{\lambda} = 0, \ldots, L_B + L_{ECP}$

↓

$$T^{N'}_{\lambda\bar{\lambda}} \approx \sum_i^{n_r} r_i^{N'}\, U(r_i)\, F_A^{\lambda}(r_i)\, F_B^{\bar{\lambda}}(r_i)\, w_i$$

↓

$$\sum_{\lambda\bar{\lambda}} \sum_m \Omega^{abc}_{\lambda lm}(\hat{k})\, \Omega^{def}_{\bar{\lambda} lm}(\hat{k})\, T^{a+b+c+d+e+f}_{\lambda\bar{\lambda}}$$

R. Flores-Moreno *et al*, *J. Comp. Chem.* 27, 1009 (2006)

Integrals with projectors onto outercore shells (GRECP)

Target integrals:

$$\langle\chi_A|\ \hat{U}^{AREP}_{n_c l}\ P_l\ |\chi_B\rangle \qquad\qquad \langle\chi_A|\ \hat{U}^{SO}_{n_c l}\ P_l\ \boldsymbol{\ell}\ P_l\ |\chi_B\rangle$$

We substitute the following expressions:

$$\hat{U}^{AREP}_{n_c l} = \frac{l+1}{2l+1}\hat{V}_{n_c,l+} + \frac{l}{2l+1}\hat{V}_{n_c,l-}$$

$$\hat{U}^{SO}_{n_c l} = \frac{2}{2l+1}\left[\hat{V}_{n_c,l+} - \hat{V}_{n_c,l-}\right]$$

the problem is reduced to the integrals

$$\langle\chi_A|\ \hat{V}_{n_c lj}\ P_l\ |\chi_B\rangle \qquad\qquad \langle\chi_A|\ \hat{V}_{n_c lj}\ P_l\ \boldsymbol{\ell}\ P_l\ |\chi_B\rangle$$

$$\hat{V}_{n_c lj} = (U_{n_c lj} - U_{lj})\,\tilde{P}_{n_c lj} + \tilde{P}_{n_c lj}\,(U_{n_c lj} - U_{lj}) - \sum_{n'_c}\tilde{P}_{n_c lj}\left[\frac{U_{n_c lj} + U_{n'_c lj}}{2} - U_{lj}\right]\tilde{P}_{n'_c lj}$$

Integrals with projectors onto outercore shells (GRECP)

Scalar-relativistic part $\langle \chi_A | \hat{V}_{n_c lj} P_l | \chi_B \rangle$

$$|n_c ljm\rangle = R_{n_c lj}(r) S_{lm}(\hat{\mathbf{r}}) \quad \rightarrow \quad \tilde{P}_{n_c lj} = \sum_m |n_c ljm\rangle \langle n_c ljm|$$

1. $$\langle \chi_A | \; [U_{n_c lj} - U_{lj}] \, \tilde{P}_{n_c lj} P_l \; |\chi_B\rangle = \sum_m \underbrace{\langle \chi_A | \; [U_{n_c lj} - U_{lj}] \, P_l \; | n_c ljm\rangle}_{\text{type 2 integral}} \langle n_c ljm | \chi_B \rangle$$

2. $$\langle \chi_A | \; \tilde{P}_{n_c lj} \; [U_{n_c lj} - U_{lj}] \, P_l \; |\chi_B\rangle = \sum_m \langle \chi_A | n_c ljm\rangle \underbrace{\langle n_c ljm | \, [U_{n_c lj} - U_{lj}] \, P_l \; |\chi_B\rangle}_{\text{type 2 integral}}$$

3. $$\langle \chi_A | \; \tilde{P}_{n_c lj} \; \left[\frac{U_{n_c lj} + U_{n'_c lj}}{2} - U_{lj} \right] \tilde{P}_{n'_c lj} P_l \; |\chi_B\rangle =$$

$$= \sum_m \langle \chi_A | n_c ljm\rangle \underbrace{\langle n_c ljm | \left[\frac{U_{n_c lj} + U_{n'_c lj}}{2} - U_{lj} \right] | n'_c ljm\rangle}_{\text{radial integral} \;\rightarrow\; \text{quadrature}} \langle n'_c ljm | \chi_B \rangle$$

Integrals with projectors onto outercore shells (GRECP)

Effective spin-orbit interaction $\langle \chi_A | \hat{V}_{n_c lj} P_l \boldsymbol{\ell} P_l | \chi_B \rangle$

4. $\langle \chi_A | \; [U_{n_c lj} - U_{lj}] \, \tilde{P}_{n_c lj} P_l \; | \chi_B \rangle =$

$$= \sum_m \underbrace{\langle \chi_A | \; [U_{n_c lj} - U_{lj}] \, P_l \; | n_c ljm \rangle}_{\text{type 2 integral}} \sum_{m'} \langle S_{lm} | \boldsymbol{\ell} | S_{lm'} \rangle \langle n_c ljm' | \chi_B \rangle$$

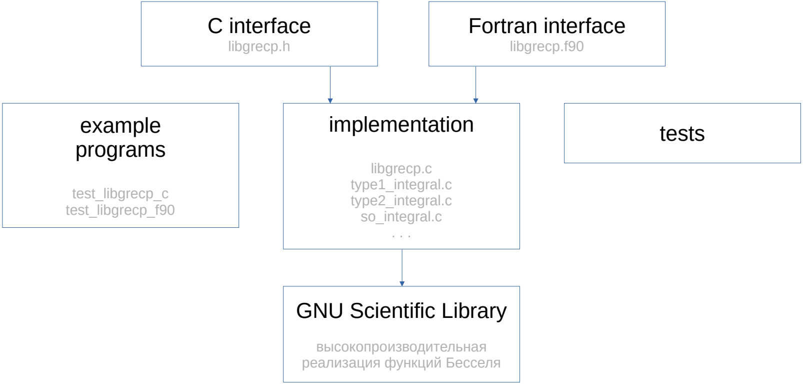
5.
$\langle \chi_A | \tilde{P}_{n_c lj} \; [U_{n_c lj} - U_{lj}] \, P_l \; | \chi_B \rangle = \sum_m \langle \chi_A | n_c ljm \rangle \underbrace{\langle n_c ljm | \, [U_{n_c lj} - U_{lj}] \, P_l \boldsymbol{\ell} P_l | \chi_B \rangle}_{\text{type 3 integral (SO)}}$

6. $\langle \chi_A | \; \tilde{P}_{n_c lj} \; \left[\frac{U_{n_c lj} + U_{n'_c lj}}{2} - U_{lj} \right] \tilde{P}_{n'_c lj} P_l \; | \chi_B \rangle =$

$$= \sum_m \langle \chi_A | n_c ljm \rangle \underbrace{\langle n_c ljm | \left[\frac{U_{n_c lj} + U_{n'_c lj}}{2} - U_{lj} \right] | n'_c ljm \rangle}_{\text{radial integral} \; \to \; \text{quadrature}} \sum_{m'} \langle S_{lm} | \boldsymbol{\ell} | S_{lm'} \rangle \langle n'_c ljm | \chi_B \rangle$$

The libgrecp library

Structure of the library



https://www.gnu.org/software/gsl/

The libgrecp library

Data structures. Pseudopotential

$$U_{lj}(r) = \sum_i d_i \ r^{n_i - 2} \ e^{-\xi_i r^2}$$

```
typedef struct {
    int L;
    int J;
    int num_primitives;
    int *powers;
    double *coeffs;
    double *alpha;
} libgrecp_ecp_t;
```

```
// constructor
libgrecp_ecp_t *libgrecp_new_ecp(
    int L, int J, int num_primitives,
    int *powers, double *coeffs, double *alpha
);

// destructor
void libgrecp_delete_ecp(libgrecp_ecp_t *ecp);
```

The libgrecp library

Data structures. Gaussian basis functions (shells)

$$\chi_A(\boldsymbol{r}) = \sum_i c_i \; N_i \; x_A^n y_A^l z_A^m \; e^{-\alpha_i |\boldsymbol{r}-\boldsymbol{A}|^2}$$

```
typedef struct {
    int L;
    int cart_size;
    int *cart_list;
    int num_primitives;
    double *coeffs;
    double *alpha;
    double origin[3];
} libgrecp_shell_t;
```

```
// constructor
libgrecp_shell_t *libgrecp_new_shell(
    double *origin, int L,
    int num_primitives, double *coeffs, double *alpha
);

// destructor
void libgrecp_delete_shell(libgrecp_shell_t *shell);
```

example: the d-shell

cart_size = 6

$$\text{cart\_list} = [\, \underbrace{2,0,0}_{d_{xx}},\ \underbrace{1,1,0}_{d_{xy}},\ \underbrace{1,0,1}_{d_{xz}},\ \underbrace{0,2,0}_{d_{yy}},\ \underbrace{0,1,1}_{d_{yz}},\ \underbrace{0,0,2}_{d_{zz}} \,]$$

The libgrecp library

Radially local integrals $\langle \chi_A | U(r) | \chi_B \rangle$

C:

```
void libgrecp_type1_integrals(
    libgrecp_shell_t *shell_A, libgrecp_shell_t *shell_B,
    double *ecp_origin, libgrecp_ecp_t *ecp,
    double *matrix
);
```

Example: integrals for the pair of d- and f-shells:

	f_{xxx}	f_{xxy}	f_{xxz}	f_{xyy}	f_{xyz}	f_{xzz}	f_{yyy}	f_{yyz}	f_{yzz}	f_{zzz}
d_{xx}	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
d_{xy}	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
d_{xz}	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
d_{yy}	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]
d_{yz}	[41]	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]	[50]
d_{zz}	[51]	[52]	[53]	[54]	[55]	[56]	[57]	[58]	[59]	[60]

The libgrecp library

Radially local integrals $\langle \chi_A | U(r) | \chi_B \rangle$

Fortran 90:

```
subroutine libgrecp_type1_integrals_shells(                   &
    origin_A, L_A, num_primitives_A, coeffs_A, alpha_A,       &
    origin_B, L_B, num_primitives_B, coeffs_B, alpha_B,       &
    ecp_origin, ecp_nprim, ecp_pow, ecp_coef, ecp_alpha,      &
    matrix                                                    &
)

! shell centered on A
integer(4) :: L_A, num_primitives_A
real(8)    :: origin_A(*), coeffs_A(*), alpha_A(*)

! shell centered on B
integer(4) :: L_B, num_primitives_B
real(8)    :: origin_B(*), coeffs_B(*), alpha_B(*)

! effective core potential expansion
integer(4) :: ecp_nprim, ecp_pow(*)
real(8)    :: ecp_origin(*), ecp_coef(*), ecp_alpha(*)

! output
real(8)    :: matrix(*)
```

The libgrecp library

Semilocal integrals $\langle \chi_A | U(r) P_l | \chi_B \rangle$

C:

```
void libgrecp_type2_integrals(
    libgrecp_shell_t *shell_A, libgrecp_shell_t *shell_B,
    double *ecp_origin, libgrecp_ecp_t *ecp,
    double *matrix
);
```

Fortran 90:

```
subroutine libgrecp_type2_integrals_shells(                 &
    origin_A, L_A, num_primitives_A, coeffs_A, alpha_A, &
    origin_B, L_B, num_primitives_B, coeffs_B, alpha_B, &
    ecp_origin, ecp_L, ecp_num_primitives,               &
    ecp_powers, ecp_coeffs, ecp_alpha,                  &
    matrix                                               &
)
```

The libgrecp library

Semilocal effective spin-orbit operator: $\langle \chi_A | U^{SO}(r) P_l \boldsymbol{\ell} P_l | \chi_B \rangle$

C:

```c
void libgrecp_spin_orbit_integrals(
    libgrecp_shell_t *shell_A, libgrecp_shell_t *shell_B,
    double *ecp_origin, libgrecp_ecp_t *ecp,
    double *so_x_matrix, double *so_y_matrix, double *so_z_matrix
);
```

Fortran 90:

```fortran
subroutine libgrecp_spin_orbit_integrals_shells(        &
    origin_A, L_A, num_primitives_A, coeffs_A, alpha_A, &
    origin_B, L_B, num_primitives_B, coeffs_B, alpha_B, &
    ecp_origin, ecp_ang_momentum, ecp_num_primitives,   &
    ecp_powers, ecp_coeffs, ecp_alpha,                  &
    so_x_matrix, so_y_matrix, so_z_matrix               &
)
```

The libgrecp library

Integrals with projectors onto outercore shells (GRECP-specific):

$\langle \chi_A | \hat{U}^{AREP}_{n_c l} P_l | \chi_B \rangle$ и $\langle \chi_A | \hat{U}^{SO}_{n_c l} P_l \boldsymbol{\ell} P_l | \chi_B \rangle$

C:

```
void libgrecp_outercore_potential_integrals(
    libgrecp_shell_t *shell_A, libgrecp_shell_t *shell_B,
    double *ecp_origin, int num_oc_shells,
    libgrecp_ecp_t **oc_potentials, libgrecp_shell_t **oc_shells,
    double *arep, double *so_x, double *so_y, double *so_z
);
```

Fortran 90:

```
subroutine libgrecp_outercore_potential_integrals_shells(        &
    origin_A, L_A, num_primitives_A, coeffs_A, alpha_A,          &
    origin_B, L_B, num_primitives_B, coeffs_B, alpha_B,          &
    ecp_origin, num_oc_shells, oc_shells_L, oc_shells_J,         &
    ecp_num_primitives, ecp_powers, ecp_coeffs, ecp_alpha,       &
    oc_shells_num_primitives, oc_shells_coeffs, oc_shells_alpha, &
    arep_matrix, so_x_matrix, so_y_matrix, so_z_matrix           &
)
```

$$\hat{V}_{n_c lj} = (U_{n_c lj} - U_{lj})\tilde{P}_{n_c lj} + \tilde{P}_{n_c lj}(U_{n_c lj} - U_{lj}) - \sum_{n'_c} \tilde{P}_{n_c lj} \left[\frac{U_{n_c lj} + U_{n'_c lj}}{2} - U_{lj} \right] \tilde{P}_{n'_c lj}$$

Future plans

- further testing
- interface to the DIRAC program package

  → actinide compounds
  → cluster modelling
  → transactinide atoms: E121, E122, E123
- optimizations:

  → screening of radial integrals
  → other radial quadratures
- Python interface
- after the first publication the source code will be available on GitHub

thanks to

N. S. Mosyagin and A. V. Titov