

**libgrpp**: a library for the evaluation of molecular integrals  
of the generalized relativistic pseudopotential operator over Gaussian functions

Alexander V. Oleynichenko  
A. Zaitsevskii, N. S. Mosyagin, A. N. Petrov, E. Eliav, A. V. Titov

*oleynichenko\_av@pnpi.nrcki.ru*  
*<http://qchem.pnpi.spb.ru>*

55 PNPI Winter School, Luga

17th March, 2023

What is the generalized relativistic pseudopotential method?

Integration algorithm and the `libgrpp` library

Accuracy assessment of generalized pseudopotentials in molecular problems

Appendix

## Bibliography: (generalized) relativistic pseudopotentials

- ▶ 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
- ▶ A. V. Titov, N. S. Mosyagin, *IJQC* 71, 359 (1999)  
Generalized relativistic effective core potential: Theoretical grounds
- ▶ 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
- ▶ M. Dolg, X. Cao, *Chem. Rev.* 112, 403 (2012)  
Relativistic Pseudopotentials: Their Development and Scope of Applications
- ▶ N. S. Mosyagin, A. V. Zaitsevskii, A. V. Titov, *IJQC*, e26076 (2019)  
Generalized relativistic effective core potentials for superheavy elements
- ▶ A. Zaitsevskii, N. S. Mosyagin, A. V. Oleynichenko, E. Eliav, *IJQC*, e27077 (2022)  
Generalized relativistic small-core pseudopotentials accounting for quantum electrodynamic effects: Construction and pilot applications

# Relativistic pseudopotentials (RPP) in a nutshell

- ▶ core electrons of a heavy atom are removed
- ▶ their effect on the valence electrons are simulated with the  $\hat{U}$  potential (the Pauli principle is accounted for)
- ▶ remaining electrons are described by the one- or two-component Schrödinger equation:

$$\hat{H}^{RPP} = \sum_i \left( -\frac{\Delta_i}{2} + \sum_{\alpha} \left( -\frac{z_{\alpha}}{r_{\alpha i}} + \hat{U}_{\alpha}(i) \right) \right) + \sum_{i>j} \frac{1}{r_{ij}}$$

$i, j$  – sum over electrons

$\alpha$  – sum over atoms in a molecule

$z_{\alpha}$  – effective charge of the  $\alpha$ -th nucleus,  $z_{\alpha} = Z_{\alpha} - N_{\text{core electrons}}$

- ▶ The  $\hat{U}$  operator can effectively account for:
  - ▶ Breit interaction between electrons
  - ▶ finite nuclear charge distribution (Fermi, ...)
  - ▶ QED corrections (self-energy + vacuum polarization)

(these effects are included during the RPP construction stage)

# Semilocal pseudopotentials

One  $U_{lj}$  potential for each partial wave with quantum numbers  $l, j$ :

$$\hat{U} = U_{LJ}(r) + \sum_{l=0}^L \sum_{j=|l-1/2|}^{l+1/2} [U_{lj}(r) - U_{LJ}(r)] P_{lj}$$

- ▶ projector  $P_{lj} = \sum_m |l j m\rangle \langle l j m|$  “extracts” states with given  $l, j$
- ▶ calculation of molecular integrals is relatively simple → are widely used in practical calculations
- ▶ work poorly for  $f$ -elements (lanthanides, actinides) and superheavy elements

# What is the generalized relativistic pseudopotential (GRPP)?

Generalized / Gatchina pseudopotentials depend also on the principal quantum number:  $\mathbf{U}_{lj} \rightarrow \mathbf{U}_{nlj}$

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

- ▶  $\tilde{P}_{n_c lj} = \sum_m |\tilde{\phi}_{n_c ljm}\rangle \langle \tilde{\phi}_{n_c ljm}|$  stand for projectors onto outercore *pseudospinors*
- ▶  $\tilde{P}_{n_c lj}$  depends on  $r \Rightarrow$  the  $\hat{U}^{GRPP}$  operator is non-local!

# What is the generalized relativistic pseudopotential (GRPP)?

Scalar-relativistic potential and effective spin-orbit interaction operators

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

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

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

# Example: uranium atom in the SCF approximation

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

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

Excitation energies, $\text{cm}^{-1}$	DFB	Absolute error, $\text{cm}^{-1}$				
		no QED	point nuc.	no Breit	GRPP	semilocal
$5f^3 6d^1 7s^2 \rightarrow$						
$5f^3 7s^2 7p^1$	7589	-72	-40	-93	-1	-6
$5f^3 6d^2 7s^1$	12990	133	96	78	2	1
$5f^3 6d^1 7s^1 7p^1$	17109	90	74	14	1	-9
$5f^2 6d^2 7s^2$	4809	-169	-85	-780	52	554
$5f^2 6d^2 7s^1 7p^1$	23920	-64	1	-765	53	546
$5f^4 7s^2$	15634	147	75	628	-44	-407
$5f^4 7s^1 7p^1$	30491	221	137	649	-45	-423
$5f^1 6d^3 7s^2$	31804	-354	-175	-1675	111	1238
$5f^1 6d^4 7s^1$	38957	-176	-49	-1552	113	1216

Excitation energies were derived from all-electron numerical SCF calculations for the states averaged over nonrelativistic configurations. Data by N. S. Mosyagin



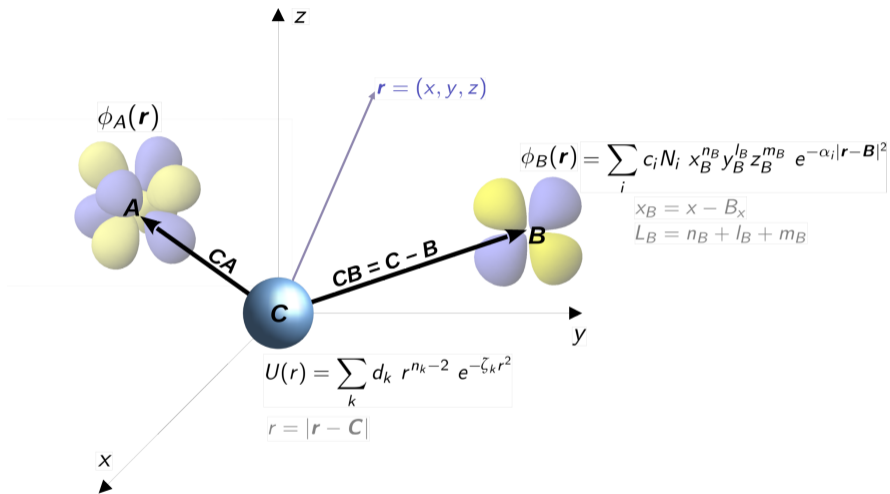
## Intermediate conclusions

- ▶ number of  $s$ -,  $p$ -,  $d$ -electrons is changed during the electronic transition → semilocal RPPs
- ▶ number of  $f$ -electrons is changed → full GRPP
- ▶ GRPP must take into account QED, Breit, and finite nuclear size effects
- ▶ the advantages of GRPP will be fully manifested in high-precision correlation calculations

However, we cannot practically use GRPP in calculations of molecules and clusters:

- ▶ MOLGEP has significant limitations (angular momentum, basis dimensionality)
- ▶ it is desirable to have access to GRPP in DIRAC

# Parametrization of pseudopotentials and basis functions



- ▶ max three-center integrals  $\langle \phi_A | \hat{U}_C | \phi_B \rangle$

## Molecular integrals over the semilocal part

$$\hat{U} = U_L(r) + \sum_{l=0}^{L-1} \underbrace{[U_l(r) - U_L(r)]}_{\Delta U_l(r)} P_l + \sum_{l=1}^L \frac{2}{2l+1} U_l^{SO}(r) P_l \ell_s$$

Three types of integrals to be calculated:

1.  $\langle \phi_A | U_L(r_C) | \phi_B \rangle$

- are reduced to overlap integrals and integrals over the  $1/r_C$  and  $1/r_C^2$  operators
- McMurchie-Davidson algorithm (re-expansion in the basis of Hermite Gaussians)

2.  $\langle \phi_A | \Delta U_l(r_C) P_l | \phi_B \rangle$

- McMurchie-Davidson algorithm for RPP integration (re-expansion of  $\phi_A$  and  $\phi_B$  at the RPP center)
- angular integrals are evaluated analytically
- radial integrals are evaluated numerically on a grid + pre-screening

3.  $\langle \phi_A | U_l^{SO}(r_C) P_l \ell | \phi_B \rangle$

- similar to type 2 integrals, but angular parts are slightly different

## GRPP integrals: non-local part

Integrals to be calculated:

$$\langle \phi_A | \hat{U}_{n_c l}^{AREP} P_l | \phi_B \rangle \quad \langle \phi_A | \hat{U}_{n_c l}^{SO} P_l \ell | \phi_B \rangle$$

After substituting expressions

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

the problem is reduced to the calculation of integrals:

$$\langle \phi_A | \hat{V}_{n_c l j} P_l | \phi_B \rangle \quad \langle \phi_A | \hat{V}_{n_c l j} P_l \ell | \phi_B \rangle$$

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

# GRPP integrals: non-local part

Scalar-relativistic part  $\langle \phi_A | \hat{V}_{n_c l j} P_l | \phi_B \rangle$

$$|\tilde{\phi}_{n_c l j m}\rangle = R_{n_c l j}(r) S_{l m}(\hat{r}) \quad \rightarrow \quad \tilde{P}_{n_c l j} = \sum_m |\tilde{\phi}_{n_c l j m}\rangle \langle \tilde{\phi}_{n_c l j m}|$$

$$1. \langle \phi_A | [U_{n_c l j} - U_{l j}] \tilde{P}_{n_c l j} P_l | \phi_B \rangle = \sum_{m=-l}^{+l} \underbrace{\langle \phi_A | U_{n_c l j} - U_{l j} | \tilde{\phi}_{n_c l j m}\rangle}_{\text{type 1 integral}} \times \underbrace{\langle \tilde{\phi}_{n_c l j m} | \phi_B \rangle}_{\text{overlap integral}}$$

$$2. \langle \phi_A | \tilde{P}_{n_c l j} [U_{n_c l j} - U_{l j}] P_l | \phi_B \rangle = \sum_{m=-l}^{+l} \langle \phi_A | \tilde{\phi}_{n_c l j m}\rangle \underbrace{\langle \tilde{\phi}_{n_c l j m} | U_{n_c l j} - U_{l j} | \phi_B \rangle}_{\text{type 1 integral}}$$

$$3. \langle \phi_A | \tilde{P}_{n_c l j} \left[ \frac{U_{n_c l j} + U_{n'_c l j}}{2} - U_{l j} \right] \tilde{P}_{n'_c l j} P_l | \phi_B \rangle =$$

$$= \sum_{m=-l}^{+l} \langle \phi_A | \tilde{\phi}_{n_c l j m}\rangle \times \underbrace{\langle \tilde{\phi}_{n_c l j m} | \frac{U_{n_c l j} + U_{n'_c l j}}{2} - U_{l j} | \tilde{\phi}_{n'_c l j m}\rangle}_{\text{purely radial integral}} \times \langle \tilde{\phi}_{n'_c l j m} | \phi_B \rangle$$

# GRPP integrals: non-local part

Effective spin-orbit interaction  $\langle \phi_A | \hat{V}_{nclj} P_l \ell | \phi_B \rangle$

$$4. \langle \phi_A | [U_{nclj} - U_{lj}] \tilde{P}_{nclj} P_l \ell | \phi_B \rangle = \sum_{m=-l}^{+l} \underbrace{\langle \phi_A | U_{nclj} - U_{lj} | \tilde{\phi}_{ncljm} \rangle}_{\text{type 1 integral}} \sum_{m'=-l}^{+l} \langle S_{lm} | \ell | S_{lm'} \rangle \langle \tilde{\phi}_{ncljm'} | \phi_B \rangle$$

$$5. \langle \phi_A | \tilde{P}_{nclj} [U_{nclj} - U_{lj}] P_l \ell | \phi_B \rangle = \sum_{m=-l}^{+l} \langle \phi_A | \tilde{\phi}_{ncljm} \rangle \sum_{m'=-l}^{+l} \langle S_{lm} | \ell | S_{lm'} \rangle \underbrace{\langle \tilde{\phi}_{ncljm'} | U_{nclj} - U_{lj} | \phi_B \rangle}_{\text{type 1 integral}}$$

$$6. \langle \phi_A | \tilde{P}_{nclj} \left[ \frac{U_{nclj} + U_{n'_clj}}{2} - U_{lj} \right] \tilde{P}_{n'_clj} P_l \ell | \phi_B \rangle =$$

$$= \sum_{m=-l}^{+l} \langle \phi_A | \tilde{\phi}_{ncljm} \rangle \times \underbrace{\langle \tilde{\phi}_{ncljm} | \frac{U_{nclj} + U_{n'_clj}}{2} - U_{lj} | \tilde{\phi}_{n'_cljm} \rangle}_{\text{purely radial integral}} \times \sum_{m'=-l}^{+l} \langle S_{lm} | \ell | S_{lm'} \rangle \langle \tilde{\phi}_{n'_cljm'} | \phi_B \rangle$$

# Implementation of GRPP integrals: the libgrpp library

		scalar	spin-orbit	outercore	open source	written in
ARGOS	1981	+	+	-	+	Fortran
MOLGEP	1991	+	+	+	-	Fortran
Turbomole	2005	+	+	-	-	Fortran
libECP	2015	+	-	-	+	C
libecpint	2021	+	-	-	+	C++
<b>libgrpp</b>	<b>2022</b>	+	+	+	+	<b>C</b>

- ▶ **libgrpp** is written from scratch in C99
- ▶ no restrictions on maximum angular momenta of GRPP and basis functions
- ▶ analytic gradients of GRPP integrals
- ▶ **libgrpp** is inserted into the home version of DIRAC!

# Implementation of GRPP integrals: the libgrpp library

Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file <> Code

**aoleynichenko** new license: LGPL yesterday 17 commits

libgrpp	grpp gradients	3 weeks ago
test	command-line args for test_libgrpp_c	3 weeks ago
test_libgrpp_c	command-line args for test_libgrpp_c	3 weeks ago
test_libgrpp_f90	overlap and nucattr integrals in the test programs	last month
.gitignore	command-line args for test_libgrpp_c	3 weeks ago
CMakeLists.txt	grpp gradients	3 weeks ago
LICENSE	new license: LGPL	yesterday
README.md	Update README.md	3 weeks ago

README.md

## libgrpp

A library for the evaluation of molecular integrals of the generalized relativistic pseudopotential operator (GRPP) over Gaussian functions.

**About**

A library for the evaluation of molecular integrals of the generalized relativistic pseudopotential operator over Gaussian functions

Readme

LGPL-2.1 license

1 star

1 watching

0 forks

**Releases**

No releases published

[Create a new release](#)

**Packages**

No packages published

[Publish your first package](#)

**Languages**

<https://github.com/aoleynichenko/libgrpp>



# Implementation of GRPP integrals: the libgrpp library



symmetry



Article

## LIBGRPP: A Library for the Evaluation of Molecular Integrals of the Generalized Relativistic Pseudopotential Operator over Gaussian Functions

Alexander V. Oleynichenko <sup>1,†</sup>, Andrei Zaitsevskii <sup>1,2,†</sup>, Nikolai S. Mosyagin <sup>1,4</sup>, Alexander N. Petrov <sup>1,3,4</sup>, Ephraim Eliav <sup>4,\*,†</sup> and Anatoly V. Titov <sup>1,3,†</sup>

<sup>1</sup> Petersburg Nuclear Physics Institute Named by B.P. Konstantinov of National Research Center "Kurchatov Institute" (NRC "Kurchatov Institute"—PNPI), 1 Orlova Roscha, Gatchina 188300, Russia

<sup>2</sup> Department of Chemistry, M.V. Lomonosov Moscow State University, Leninskie Gory 1/3, Moscow 119991, Russia

<sup>3</sup> Department of Physics, Saint Petersburg State University, 7/9 Universitetskaya Nab., St. Petersburg 199034, Russia

<sup>4</sup> School of Chemistry, Tel Aviv University, Tel Aviv 6997801, Israel

\* Correspondence: ephraim@tau.ac.il

† These authors contributed equally to this work.

**Abstract:** Generalized relativistic pseudopotentials (GRPP) of atomic cores implying the use of different potentials for atomic electronic shells with different principal quantum numbers give rise to accurate and reliable relativistic electronic structure models of atoms, molecules, clusters, and solids. These models readily incorporate the effects of Breit electron–electron interactions and one-loop quantum electrodynamics effects. Here, we report the computational procedure for evaluating one-electron integrals of GRPP over contracted Gaussian functions. This procedure was implemented in a library of routines named LIBGRPP, which can be integrated into existing quantum chemistry software, thus enabling the application of various methods to solve the many-electron problem with GRPPs. Pilot applications to electronic transitions in the ThO and UO<sub>2</sub> molecules using the new library and intermediate-Hamiltonian Fock space relativistic coupled cluster method are presented. Deviations of excitation energies obtained within the GRPP approach from their all-electron Dirac–Coulomb–Gaunt counterparts do not exceed 50 cm<sup>−1</sup> for the 31 lowest-energy states of ThO and 110 cm<sup>−1</sup> for the 79 states of UO<sub>2</sub>. The results clearly demonstrate that rather economical tiny-core GRPP models can exceed in accuracy relativistic all-electron models defined by Dirac–Coulomb and Dirac–Coulomb–Gaunt Hamiltonians.

**Keywords:** generalized relativistic pseudopotentials; molecular integrals; Gaussian basis functions; relativistic coupled cluster theory; excited states; heavy-element compounds; high-precision electronic structure modeling; thorium oxide; uranium dioxide



**Citation:** Oleynichenko, A.V.; Zaitsevskii, A.; Mosyagin, N.S.; Petrov, A.N.; Eliav, E.; Titov, A.V. LIBGRPP: A Library for the Evaluation of Molecular Integrals of the Generalized Relativistic Pseudopotential Operator over Gaussian Functions. *Symmetry* **2023**, *15*, 197. <https://doi.org/10.3390/sym15010197>

Academic Editor: Markus Meringer

accuracy assessment of generalized pseudopotentials in molecular problems

# Bibliography: successful molecular applications of GRPPs

(pseudopotential integrals were calculated using MOLGEP)

2001	HgH, HgH <sup>+</sup>	N. S. Mosyagin et al, <i>J. Chem. Phys.</i> 115, 2007 (2001)
2002	TiF	A. N. Petrov et al, <i>Phys. Rev. Lett.</i> 88, 073001(2002)
2005	HI <sup>+</sup>	T. A. Isaev et al, <i>Phys. Rev. Lett.</i> 95, 163004 (2005)
2005	PbO	A. N. Petrov et al, <i>Phys. Rev. A</i> 72, 022505 (2005)
2006	CnH, CnH <sup>+</sup>	N. S. Mosyagin et al, <i>J. Chem. Phys.</i> 124, 224302 (2006)
2009	TiF <sup>-</sup>	L. V. Skripnikov et al, <i>Opt. Spectrosc.</i> 106, 790 (2009)
2009	Yb <sub>2</sub>	N. S. Mosyagin et al, arXiv:0901.0077v5 [physics.chem-ph] (2009)
2013	Ca <sub>2</sub>	N. S. Mosyagin et al, <i>Int. J. Quantum Chem.</i> 113, 2277 (2013)
2013	RaO	A. D. Kudashov et al, <i>Phys. Rev. A</i> 87, 020102(R) (2013)
2013	ThO	L. V. Skripnikov et al, <i>J. Chem. Phys.</i> 139, 221103 (2013)
2014	RaF	A. D. Kudashov et al, <i>Phys. Rev. A</i> , 90, 052513 (2014)

it works well for *s*, *p*, *d*-elements... what about actinides?

# How to assess an accuracy of GRPP?

**Problem:** taking into account QED and Breit in 4c calculations is extremely difficult

**Solution:** to construct a special GRPP for testing only (N. S. Mosyagin)

- + atomic Dirac-Hartree-Fock-Gaunt calculation (4c)
- + Gaussian nuclear charge distribution (instead of Fermi)
- retardation
- QED contributions

**Reference 4c calculation:** Dirac-Coulomb-Gaunt (DCG-x2cmmf)

more: J. Sikkema et al, *J. Chem. Phys.* 131, 124116 (2009); <https://doi.org/10.1063/1.3239505>

**Correlation calculations:**

- ▶ relativistic Fock space coupled cluster method (FS-RCCSD)
- ▶ intermediate Hamiltonian for incomplete model spaces (IH-IMMS)

A. Zaitsevskii et al, *IJQC*, e27077 (2022), <https://doi.org/10.1002/qua.27077>

- ▶ the EXP-T program package

<http://qchem.pnpi.spb.ru/expt>

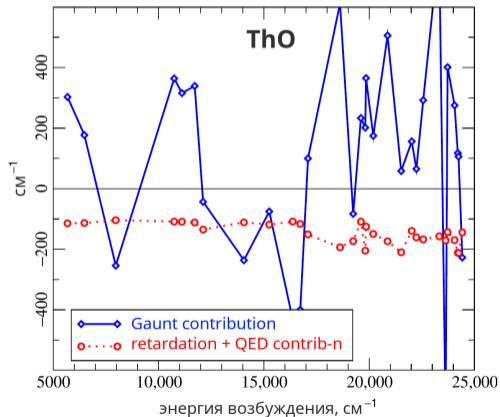
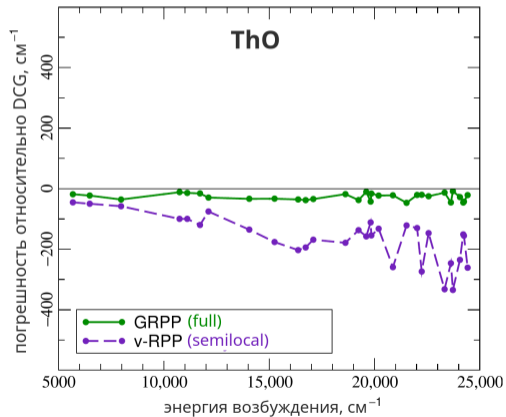
# Vertical excitation energies of ThO

FS-RCCSD scheme:  $\text{ThO}^{2+} (0h0p) \rightarrow \text{ThO}^+ (0h1p) \rightarrow \text{ThO} (0h2p)$

Active space: 24 lowest virtual Kramers pairs of  $\text{ThO}^{2+}$

Main model space: CAS 2e / 12 spinors,  $\approx 7s + 6d$  Th

Basis set: [19s17p15d15f5g4h3i] (Th), aug-cc-pVQZ-DK (O)



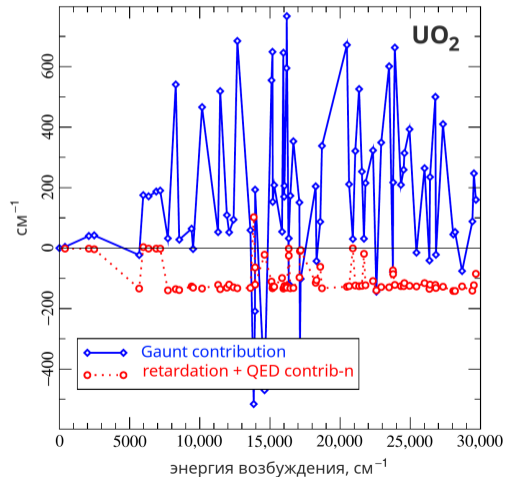
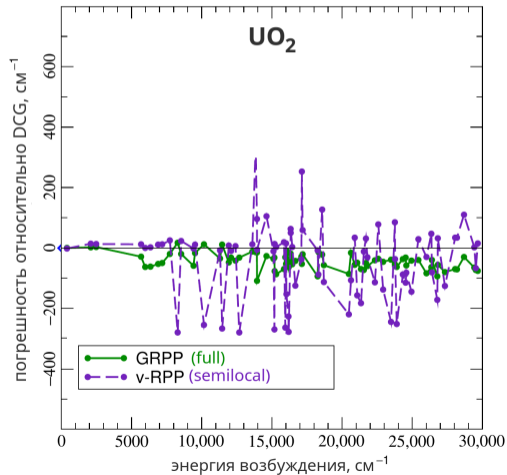
# Vertical excitation energies of $\text{UO}_2$

FS-RCCSD scheme:  $\text{UO}_2^{2+} (0h0p) \rightarrow \text{UO}_2^+ (0h1p) \rightarrow \text{UO}_2 (0h2p)$

Active space: 24 lowest virtual Kramers pairs of  $\text{UO}_2^{2+}$

Main model space:  $\approx 7s5f, 5f^2, 6d5f, 7p5f$  U

Basis set: [25s21p19d14f5g4h3i] (U), aug-cc-pVQZ-DK (O)



# Results

- ▶ Deviation from DCG:

		GRPP	semilocal	DC	Ret.+QED
ThO	max abs	46	335	802	212
	rms	29	181	341	151
UO <sub>2</sub>	max abs	110	345	767	142
	rms	51	128	316	112

- ▶ the error of GRPP is balanced for all electronic states
- ▶ the Dirac-Coulomb Hamiltonian is inherently less accurate than even a semi-local potential
- ▶ the contributions of retardation and QED effects are greater than the error of GRPP
- ▶ our future: pseudopotentials accounting for QED

GRPP seems to be the most precise Hamiltonian for real-life molecular calculations?

# Results

- ▶ the `libgrpp` library for GRPP integrals
- ▶ 4c models are no longer necessary for studying the structure and spectra of molecules?
- ▶ all efforts must be focused on improving the (FS-)CCSDT code
  
- ▶ a vast field of potential applications for the GRPP method is opening up:
  - actinide molecules ( $\text{UF}_6$ ,  $\text{AcF}$ ,  $\text{PaO}^+$ , ...)
  - cluster models of impurities (perovskites, ...)
  - superheavy elements



thanks to

Yu. V. Lomachuk  
L. V. Skripnikov

## Appendix

## 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
- ▶ B. Gao, A. J. Thorvaldsen, K. Ruud, *IJQC* 111, 858 (2011)  
GEN1INT: A unified procedure for the evaluation of one-electron integrals over Gaussian basis functions and their geometric derivatives
- ▶ R. A. Shaw, J. G. Hill, *JCP* 147, 074108 (2017)  
Prescreening and efficiency in the evaluation of integrals over *ab initio* effective core potentials
- ▶ A. V. Oleynichenko, A. Zaitsevskii, N. S. Mosyagin, A. N. Petrov, E. Eliav, A. V. Titov, *Symmetry*, 15, 197 (2023)  
LIBGRPP: A library for the evaluation of molecular integrals of the generalized relativistic pseudopotential operator over Gaussian functions

# Gatchina pseudopotentials

Effective potentials and basis sets

Group #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
Period																			
1	1 H																	2 He	
2	3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne	
3	11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar	
4	19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr	
5	37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe	
6	55 Cs	56 Ba	57 La	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn	
7	87 Fr	88 Ra	** Ac	104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Nh	114 Fl	115 Mc	116 Lv	117 Ts	118 Og	
8	119	120	*** Uu																
* Lanthanides			57 La	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb	71 Lu		
** Actinides			89 Ac	90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No	103 Lr		
***			121	122	123														

<https://github.com/aoleynichenko/libgrpp>

# Gatchina pseudopotentials

## GRPP file format

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	0.5005810372485870E-001
17.59838144506293		0.1418751938851203	0.1407082656626522
7.852460360176514		0.2692539091716999	0.2681159808207592
3.659295865768477		0.3514829463498897	0.3514526455630622
1.750902647135136		0.295603294056517	0.2870071457181363
0.8310003070152262		0.1013370041109672	0.1027876524838510
0.3168304480358770		0.6896179598715287E-002	0.7081729074463639E-002

$Inclj>$   
outercore  
pseudospinors

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

12	1	3S-AREP		2S1/2	
0	393.3602177650254	1.0000000000000000	0.0000000000000000	0.0000000000000000	
1	113.7058271536836	20.61434228311451	0.0000000000000000	0.0000000000000000	
1	16.10913256549514	43.26766325656665	0.0000000000000000	0.0000000000000000	
1	11.74489514767160	-26.09564060791052	0.0000000000000000	0.0000000000000000	
1	0.7384624094940389	-1.767696976619511	0.0000000000000000	0.0000000000000000	
2	7.435595881368535	12.81879139695621	0.0000000000000000	0.0000000000000000	
2	1.704016034278843	1.724789989169324	0.0000000000000000	0.0000000000000000	
0	8204571426674918	1.177798154185780	0.0000000000000000	0.0000000000000000	
1	5.137451248761753	0.0	0.0000000000000000	0.9380123541534502	
1	3.653988234030431	0.0	0.0000000000000000	-1.454202516587702	
2	6.495922514683006	0.0	0.0000000000000000	2.137038880995181	
2	2.407546171026337	0.0	0.0000000000000000	0.4924804917453065	
13	2	3P-AREP	3P-ESOP	2P1/2	2P3/2
1	6117.592663549806	-0.4205880832476154	0.4795688722317522	0.0000000000000000	0.0000000000000000
1	783.9977605773247	0.8288329737421520	1.060775982167485	0.0000000000000000	0.0000000000000000
1	203.4274692639501	-0.6692270967733691	0.9407954603364033E-001	0.0000000000000000	0.0000000000000000
1	66.11279538566970	0.573535881882215	0.2999003218434382E-001	0.0000000000000000	0.0000000000000000
1	3.775507750937060	0.8202119815478000	0.5711114762129910E-001	0.0000000000000000	0.0000000000000000
1	2.117535857537166	-1.062330176459113	0.7979155991688458E-001	0.0000000000000000	0.0000000000000000
2	2.338276864774692	1.118051706876372	0.8489534538739463E-001	0.0000000000000000	0.0000000000000000
0	7639253652236829	0.4855751106136452E-003	0.1467293262304992E-003	0.0000000000000000	0.0000000000000000
2	0.9345397669109543E-001	0.4240612697226007E-004	0.7770340191290965E-005	0.0000000000000000	0.0000000000000000
2	6.502020250342957	0.0	0.0	0.3430415383246921	0.3470030463624258
2	4.600761784913925	0.0	0.0	0.5161737283857235	-0.5203252473759303
2	3.063252871982663	0.0	0.0	0.2219749178055450	0.2233588248259011
2	1.807398538859029	0.0	0.0	0.1198529114654701E-001	-0.1290741233506232E-001

$U_{nc,l}(r)$

$U_l(r)$

$U_L(r)$

$U_l^{SO}(r)$

# The McMurchie-Davidson scheme for type 2 integrals

Basic idea: re-expansion of the  $\phi_A$  and  $\phi_B$  basis functions at the point  $C$  at which the pseudopotential operator is centered

$$\begin{aligned}
 \langle \phi_A | \Delta U_l(r) P_l | \phi_B \rangle &= \int_0^\infty \sum_m \langle \phi_A | S_{lm} \rangle_\Omega \cdot \Delta U_l(r) \cdot \sum_{m'} \langle \phi_B | S_{lm'} \rangle_{\Omega'} r^2 dr = \\
 &= 16\pi^2 \sum_{a=0}^{n_A} \sum_{b=0}^{l_A} \sum_{c=0}^{m_A} \sum_{d=0}^{n_B} \sum_{e=0}^{l_B} \sum_{f=0}^{m_B} \begin{pmatrix} n_A \\ a \end{pmatrix} \begin{pmatrix} l_A \\ b \end{pmatrix} \begin{pmatrix} m_A \\ c \end{pmatrix} \begin{pmatrix} n_B \\ d \end{pmatrix} \begin{pmatrix} l_B \\ e \end{pmatrix} \begin{pmatrix} m_B \\ f \end{pmatrix} \times \\
 &\times CA_x^{n_A-a} CA_y^{l_A-b} CA_z^{m_A-c} CB_x^{n_B-d} CB_y^{l_B-e} CB_z^{m_B-f} \times \\
 &\times \sum_{\lambda_1}^{\lambda_{1,\max}} \sum_{\lambda_2}^{\lambda_{2,\max}} \underbrace{T_{\lambda_1 \lambda_2}^{a+b+c+d+e+f}(\phi_A, \phi_B)}_{\text{radial type 2 integral}} \times \sum_{m=-l}^{+l} \underbrace{\Omega_{\lambda_1 lm}^{abc}(\hat{k}_A) \Omega_{\lambda_2 lm}^{def}(\hat{k}_B)}_{\text{angular type 2 integrals}}
 \end{aligned}$$

$$P_l = \sum_m |S_{lm}\rangle \langle S_{lm}|$$

$$\lambda_{1,\max} = l + a + b + c, \quad \lambda_{2,\max} = l + d + e + f \\
 \hat{k}_A = \frac{CA}{|CA|}, \quad \hat{k}_B = \frac{CB}{|CB|}, \quad CA = C - A, \quad CB = C - B$$

## Angular type 2 integrals

- ▶ Angular integrals are evaluated analytically:

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

- ▶ The value of a real spherical harmonic  $S_{\lambda\mu}$  at the  $\hat{k}$  point:

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

- ▶ USPs integrals:

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

## Angular type 2 integrals

The  $y_{rst}^{lm}$  coefficients are calculated using the formula:

$$y_{rst}^{lm} = \sqrt{\frac{2l+1}{2\pi} \frac{(l-|m|)!}{(l+|m|)!} \frac{1}{2^l l!}} \sum_{i=j}^{(l-|m|)/2} \binom{l}{i} \binom{i}{j} \frac{(-1)^i (2l-2i)!}{(l-|m|-2i)!} \times$$
$$\times \sum_{k=0}^j \binom{j}{k} \binom{|m|}{r-2k} (-1)^{(|m|-r+2k)/2} \times$$
$$\times \begin{cases} 1 & m > 0 \text{ and } |m| - r \text{ even} \\ 1/\sqrt{2} & m = 0 \text{ and } r \text{ even} \\ 1 & m < 0 \text{ and } |m| - r \text{ odd} \\ 0 & \text{otherwise} \end{cases}$$

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

libgrpp calculates  $y_{rst}^{lm}$  only once and then stores them in a table



## Radial type 2 integrals

- ▶ Radial type 2 integral:

$$T_{\lambda_1 \lambda_2}^N(\phi_A, \phi_B) = \int_0^\infty r^{N+2} \Delta U_l(r) F_A^{\lambda_1}(r) F_B^{\lambda_2}(r) dr$$

- ▶ Auxiliary functions  $F_A^\lambda(r)$  and  $F_B^\lambda(r)$  allow one to work directly with contracted basis functions:

$$F_A^\lambda(r) = \sum_i c_i N_i e^{-\alpha_i |\mathbf{CA}|^2 - k_{Ai} r^2} M_\lambda(k_{Ai} r)$$

$M_\lambda(x)$  – spherical modified Bessel functions

$$k_{Ai} = 2\alpha_{Ai} |\mathbf{CA}|$$

- ▶ It is more convenient to use scaled Bessel functions  $K_\lambda(x) = e^{-x} M_\lambda(x)$ :

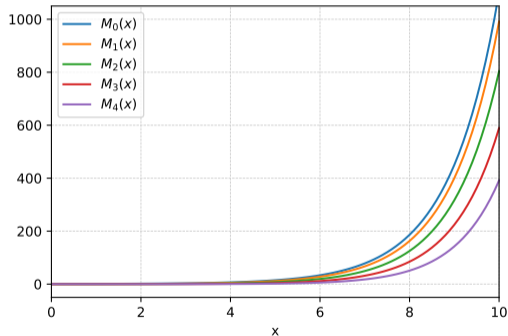
$$F_A^\lambda(r) = \sum_i c_i N_i \cdot e^{-\alpha_{Ai} |\mathbf{CA}|^2 - k_{Ai} r^2 + k_{Ai} r} K_\lambda(k_{Ai} r)$$

# Radial type 2 evaluation

Scaled modified spherical Bessel functions  $K_\lambda(x)$

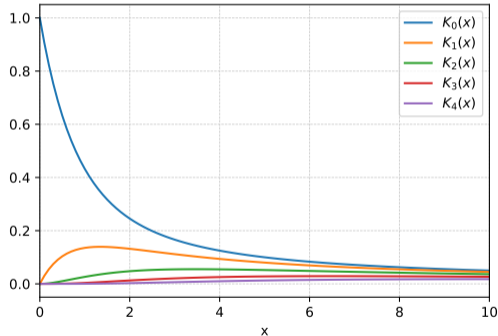
modified spherical Bessel function

$$M_n(x) = \sqrt{\pi/(2x)} I_{n+1/2}(x)$$



modified spherical scaled Bessel function

$$K_n(x) = e^{-x} M_n(x)$$



# Radial type 2 integrals

## The Log3 quadrature

Integral to be evaluated:

$$I = \int_0^{+\infty} f(r) r^2 dr \approx \sum_i^{n_r} w_i f(r_i)$$

Radial grid consists of  $n_r$  points:

$$x_i = \frac{i}{n_r + 1}, \quad x_i \in (0, 1) \quad r_i = -\alpha \ln(1 - x_i^3), \quad r_i \in (0, +\infty) \quad w_i = \frac{3\alpha^3 x_i^2 \ln^2(1 - x_i^3)}{(1 - x_i^3)(n_r + 1)}$$

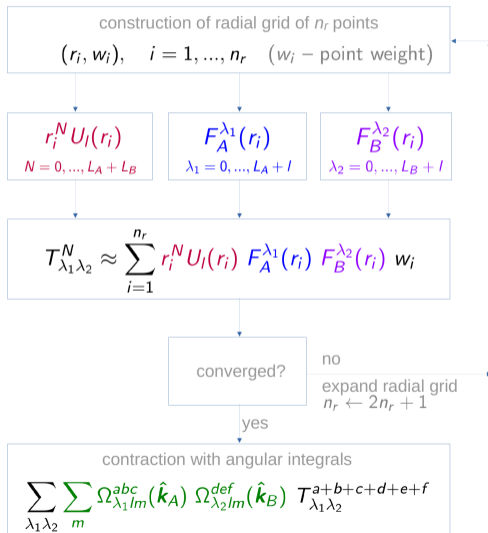
When expanding the grid to  $n_r^{(2)} = n_r^{(1)} + 1$  radial points:

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

**The radial integral can be computed with any required accuracy!**

# Radial type 2 integrals

## Flowchart



## The McMurchie-Davidson scheme for type 3 integrals (spin-orbit operator)

$$\begin{aligned}
 \langle \phi_A | U_l^{SO}(r) P_l \ell P_l | \phi_B \rangle &= i^{-1} \int \phi_A(\mathbf{r}) U_l^{SO}(r_C) \left( \sum_m |S_{lm}\rangle \langle S_{lm}| \right) \ell \left( \sum_m |S_{lm}\rangle \langle S_{lm}| \right) \phi_B(\mathbf{r}) d\mathbf{r}_C = \\
 &= 16\pi^2 \sum_{a=0}^{n_A} \sum_{b=0}^{l_A} \sum_{c=0}^{m_A} \sum_{d=0}^{n_B} \sum_{e=0}^{l_B} \sum_{f=0}^{m_B} \begin{pmatrix} n_A \\ a \end{pmatrix} \begin{pmatrix} l_A \\ b \end{pmatrix} \begin{pmatrix} m_A \\ c \end{pmatrix} \begin{pmatrix} n_B \\ d \end{pmatrix} \begin{pmatrix} l_B \\ e \end{pmatrix} \begin{pmatrix} m_B \\ f \end{pmatrix} \times \\
 &\times CA_x^{n_A-a} CA_y^{l_A-b} CA_z^{m_A-c} CB_x^{n_B-d} CB_y^{l_B-e} CB_z^{m_B-f} \times \\
 &\times \sum_{\lambda=0}^{\lambda_{1,\max}} \sum_{\bar{\lambda}=0}^{\lambda_{2,\max}} T_{\lambda\bar{\lambda}}^{a+b+c+d+e+f}(\phi_A, \phi_B) \times \sum_{m=-l}^{+l} \sum_{m'=-l}^{+l} \Omega_{\lambda lm}^{abc}(\hat{\mathbf{k}}) \Omega_{\bar{\lambda} lm}^{def}(\hat{\mathbf{k}}) \langle S_{lm} | \ell | S_{lm'} \rangle
 \end{aligned}$$

- ▶ radial type 2 integrals
- ▶ angular type 2 integrals
- ▶ integrals of the angular momentum operator  $\ell$  in the basis of real spherical harmonics  $S_{lm}$

## The `libgrpp` library programming interface

# The libgrpp library programming interface

## Data structures: pseudopotentials

$$U_{ij}(r) = \sum_k d_k r^{n_k-2} e^{-\zeta_k r^2}$$

```
1 typedef struct {
2     int L;
3     int J;
4     int num_primitives;
5     int *powers;
6     double *coeffs;
7     double *alpha;
8 } libgrpp_potential_t;
```

```
1 // constructor
2 libgrpp_potential_t *libgrpp_new_potential(
3     int L, int J, int num_primitives, int *powers, double *coeffs, double *alpha
4 );
5
6 // destructor
7 void libgrpp_delete_potential(libgrpp_potential_t *potential);
```

# The libgrpp library programming interface

## Data structures: the full GRPP operator

```
1 typedef struct {
2     int n_arep;
3     int n_esop;
4     int n_oc_shells;
5     libgrpp_potential_t *U_L;           // local potential
6     libgrpp_potential_t **U_arep;      // semi-local scalar-relativistic potential
7     libgrpp_potential_t **U_esop;      // semi-local effective spin-orbit potential
8     libgrpp_potential_t **U_oc;        // outercore potentials
9     libgrpp_shell_t **oc_shells;       // outercore pseudospinors
10 } libgrpp_grpp_t;
```

```
1 // constructor
2 libgrpp_grpp_t *libgrpp_new_grpp();
3
4 // destructor
5 void libgrpp_delete_grpp(libgrpp_grpp_t *);
6
7 void libgrpp_grpp_set_local_potential(libgrpp_grpp_t *grpp, libgrpp_potential_t *pot);
8 void libgrpp_grpp_add_averaged_potential(libgrpp_grpp_t *grpp, libgrpp_potential_t *pot);
9 void libgrpp_grpp_add_spin_orbit_potential(libgrpp_grpp_t *grpp, libgrpp_potential_t *pot);
10 void libgrpp_grpp_add_outercore_potential(
11     libgrpp_grpp_t *grpp, libgrpp_potential_t *pot, libgrpp_shell_t *oc_shell
12 );
```



# The libgrpp library programming interface

Data structures: shells of Cartesian basis functions

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

```
1 typedef struct {
2     int L;
3     int cart_size;
4     int *cart_list;
5     int num_primitives;
6     double *coeffs;
7     double *alpha;
8     double origin[3];
9 } libgrpp_shell_t;
```

```
1 // constructor
2 libgrpp_shell_t *libgrpp_new_shell(
3     double *origin, int L, int num_primitives, double *coeffs, double *alpha
4 );
5
6 // destructor
7 void libgrpp_delete_shell(libgrpp_shell_t *shell);
```

Example: the *d*-shell

cart\_size = 6

cart\_list = [ 2, 0, 0, 1, 1, 0, 1, 0, 1, 0, 2, 0, 0, 1, 1, 0, 0, 2 ]

$\underbrace{\quad}_{d_{xx}} \quad \underbrace{\quad}_{d_{xy}} \quad \underbrace{\quad}_{d_{xz}} \quad \underbrace{\quad}_{d_{yy}} \quad \underbrace{\quad}_{d_{yz}} \quad \underbrace{\quad}_{d_{zz}}$

# The libgrpp library programming interface

Radially-local integrals  $\langle \phi_A | U(r) | \phi_B \rangle$

C:

```
1 void libgrpp_type1_integrals(  
2     libgrpp_shell_t *shell_A, libgrpp_shell_t *shell_B,  
3     double *rpp_origin, libgrpp_potential_t *potential,  
4     double *matrix  
5 );
```

Example: integrals for the  $d/f$  shell pair:

	$f_{xxx}$	$f_{xxy}$	$f_{xxz}$	$f_{xyy}$	$f_{xyz}$	$f_{xzz}$	$f_{yyy}$	$f_{yyz}$	$f_{yzz}$	$f_{zzz}$
$d_{xx}$	[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 libgrpp library programming interface

Radially-local integrals  $\langle \phi_A | U(r) | \phi_B \rangle$

Fortran 90:

```
1  subroutine libgrpp_type1_integrals(                               &
2      origin_A, L_A, num_primitives_A, coeffs_A, alpha_A,      &
3      origin_B, L_B, num_primitives_B, coeffs_B, alpha_B,      &
4      rpp_origin, rpp_nprim, rpp_pow, rpp_coef, rpp_alpha,      &
5      matrix                                                    &
6  )
7
8  ! shell centered on A
9  integer(4) :: L_A, num_primitives_A
10 real(8)    :: origin_A(*), coeffs_A(*), alpha_A(*)
11
12 ! shell centered on B
13 integer(4) :: L_B, num_primitives_B
14 real(8)    :: origin_B(*), coeffs_B(*), alpha_B(*)
15
16 ! pseudopotential expansion
17 integer(4) :: rpp_nprim, rpp_pow(*)
18 real(8)    :: rpp_origin(*), rpp_coef(*), rpp_alpha(*)
19
20 ! output
21 real(8)    :: matrix(*)
```

# The libgrpp library programming interface

Integrals with angular projectors  $\langle \phi_A | \Delta U_i(r) P_l | \phi_B \rangle$

C:

```
1 void libgrpp_type2_integrals(  
2     libgrpp_shell_t *shell_A, libgrpp_shell_t *shell_B,  
3     double *rpp_origin, libgrpp_potential_t *potential,  
4     double *matrix  
5 );
```

Fortran 90:

```
1 subroutine libgrpp_type2_integrals(                                &  
2     origin_A, L_A, num_primitives_A, coeffs_A, alpha_A, &  
3     origin_B, L_B, num_primitives_B, coeffs_B, alpha_B, &  
4     rpp_origin, rpp_L, rpp_num_primitives,                    &  
5     rpp_powers, rpp_coeffs, rpp_alpha,                        &  
6     matrix                                                    &  
7 )
```

# The libgrpp library programming interface

Semi-local spin-orbit integrals:  $\langle \phi_A | U_i^{SO}(r) P_i \ell P_i | \phi_B \rangle$

C:

```
1 void libgrpp_spin_orbit_integrals(  
2     libgrpp_shell_t *shell_A, libgrpp_shell_t *shell_B,  
3     double *rpp_origin, libgrpp_potential_t *potential,  
4     double *so_x_matrix, double *so_y_matrix, double *so_z_matrix  
5 );
```

Fortran 90:

```
1 subroutine libgrpp_spin_orbit_integrals(                &  
2     origin_A, L_A, num_primitives_A, coeffs_A, alpha_A, &  
3     origin_B, L_B, num_primitives_B, coeffs_B, alpha_B, &  
4     rpp_origin, rpp_ang_momentum, rpp_num_primitives,  &  
5     rpp_powers, rpp_coeffs, rpp_alpha,                 &  
6     so_x_matrix, so_y_matrix, so_z_matrix              &  
7 )
```

# The libgrpp library programming interface

Non-local integrals (with projectors onto outercore shells):  $\langle \phi_A | \hat{U}_{ncI}^{AREP} P_I | \phi_B \rangle$  and  $\langle \phi_A | \hat{U}_{ncI}^{SO} P_I e P_I | \phi_B \rangle$

C:

```
1 void libgrpp_outercore_potential_integrals(  
2     libgrpp_shell_t *shell_A, libgrpp_shell_t *shell_B,  
3     double *rpp_origin, int num_oc_shells,  
4     libgrpp_potential_t **oc_potentials, libgrpp_shell_t **oc_shells,  
5     double *arep, double *so_x, double *so_y, double *so_z  
6 );
```

Fortran 90:

```
1 subroutine libgrecp_outercore_potential_integrals(           &  
2     origin_A, L_A, num_primitives_A, coeffs_A, alpha_A,    &  
3     origin_B, L_B, num_primitives_B, coeffs_B, alpha_B,    &  
4     ecp_origin, num_oc_shells, oc_shells_L, oc_shells_J,   &  
5     ecp_num_primitives, ecp_powers, ecp_coeffs, ecp_alpha, &  
6     oc_shells_num_primitives, oc_shells_coeffs, oc_shells_alpha, &  
7     arep_matrix, so_x_matrix, so_y_matrix, so_z_matrix     &  
8 )
```

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

# The libgrpp library programming interface

Integrals over the full GRPP operator (C only):  $\langle \phi_A | \hat{U}^{GRPP} | \phi_B \rangle$

C:

```
1 void libgrpp_full_grpp_integrals(  
2     libgrpp_shell_t *shell_A, libgrpp_shell_t *shell_B,  
3     libgrpp_grpp_t *grpp_operator, double *grpp_origin,  
4     double *arep_matrix, double *so_x_matrix, double *so_y_matrix, double *so_z_matrix  
5 );
```

# The libgrpp library programming interface

Analytic gradients of GRPP integrals (C only)

$$\frac{d}{dR} \langle \phi_A | \hat{U}^{GRPP} | \phi_B \rangle$$

C:

```
1 void libgrpp_full_grpp_integrals_gradient(  
2     libgrpp_shell_t *shell_A, libgrpp_shell_t *shell_B,  
3     libgrpp_grpp_t *grpp_operator, double *grpp_origin,  
4     double *point_3d, // derivative wrt this point  
5     double **grad_arep, double **grad_so_x, double **grad_so_y, double **grad_so_z  
6 );
```