

---

```

1  """Tutorial: SCF first hyperpolarizability"""
2
3  __author__    = "Eric J. Berquist"
4  __credit__    = ["Eric J. Berquist"]
5
6  __copyright__ = "(c) 2014-2017, The Psi4NumPy Developers"
7  __license__   = "BSD-3-Clause"
8  __date__      = "2017-12-19"

```

---

# 1 SCF First Hyperpolarizability

## 1.1 Introduction

In Tutorial 6a, the calculation of linear response properties from analytic derivative theory is presented, the foundation of which are the coupled-perturbed Hartree-Fock (CPHF) or coupled-perturbed self-consistent field (CPSCF) equations. Starting from analytic derivative theory provides a convenient physical picture: how does the total energy of a system change under the influence of one or more internal or external perturbations? Continuing the case of an external electric field, the total energy of a system can be represented with a series expansion:

$$E(\mathbf{E}) = \sum_{n=0}^{\infty} \frac{1}{n!} E^{(n)}(\mathbf{a}) \cdot (\mathbf{E} - \mathbf{a})^n,$$

where the electric field is  $\mathbf{E} = \vec{E} = (E_x, E_y, E_z)$  and  $\mathbf{a}$  is the expansion point. In practice, we always expand around  $\mathbf{a} = \mathbf{0}$ , so it is a Maclaurin series:

$$E(\mathbf{E}) = \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{E}^{(n)}(\mathbf{0}) \cdot \mathbf{E}^n.$$

Expanding the above to the first 4 explicit terms gives

$$E(\mathbf{E}) \approx E^{(0)}(\mathbf{0}) + \mathbf{E}^{(1)}(\mathbf{0}) \cdot \mathbf{E} + \frac{1}{2} \mathbf{E}^{(2)}(\mathbf{0}) \cdot \mathbf{E}^2 + \frac{1}{6} \mathbf{E}^{(3)}(\mathbf{0}) \cdot \mathbf{E}^3,$$

where we identify

$$E^{(0)} \rightarrow \text{the unperturbed ground-state energy} \quad (1)$$

$$E_a^{(1)} \rightarrow -\mu_a, \text{ the dipole moment} \quad (2)$$

$$E_{ab}^{(2)} \rightarrow -\alpha_{ab}, \text{ the polarizability} \quad (3)$$

$$E_{abc}^{(3)} \rightarrow -\beta_{abc}, \text{ the first hyperpolarizability} \quad (4)$$

The first hyperpolarizability is the leading-order term that describes the *nonlinear* response of a system to an external electric field. Each term in the series expansion increases the rank of the coefficient by one: the ground-state energy is a scalar, the dipole is a length 3 vector, the polarizability is a 3-by-3 matrix, and the first hyperpolarizability is a 3-by-3-by-3 tensor.

Translated into the language of analytic derivative theory, is it represented as  $\beta_{abc} = \left. \frac{\partial^3 E}{\partial E_a \partial E_b \partial E_c} \right|_{\mathbf{E}=\mathbf{0}}$ , though it is not yet clear how to take derivatives of the energy beyond what is presented in tutorial 6a. Additionally, nothing has been stated about time dependence; everything to this point has been the static case, where the strength of fields do not vary with time. We will first incorporate time dependence, and equations for derivative theory will result.

### 1.1.1 Notation

Before going further, some notational conventions should be mentioned. When used as field indices,  $a, b, c, \dots \in \{x, y, z\}$ , the three Cartesian directions.

For matrix indices,  $\mu, \nu, \lambda, \sigma, \dots$  label atomic orbitals (AOs)/basis functions,  $i, j, k, l, \dots$  label occupied molecular orbitals (MOs),  $a, b, c, d, \dots$  label unoccupied/virtual MOs, and  $p, q, r, s, \dots$  label all MOs. Einstein summation is used, so repeated indices are contracted over.

## 1.2 Derivation

Again, write the total Hamiltonian as the sum of unperturbed and perturbed components

$$\hat{H}(\mathbf{E}, t) = \hat{H}^{(0)} + \hat{V}(\mathbf{E}, t) \quad (5)$$

$$\hat{V}(\mathbf{E}, t) = -\boldsymbol{\mu} \cdot \mathbf{E}(e^{\pm i\omega t} + 1) \quad (6)$$

where part of the external field now oscillates with some characteristic frequency  $\omega$ . This can be incorporated into the time-dependent Schrodinger equation, which for a stationary state obeys

$$\left[ \hat{H}^{(0)} + \hat{V}(\mathbf{E}, t) - i \frac{\partial}{\partial t} \right] \psi(t) = 0, \quad (7)$$

$$FC - i \frac{\partial}{\partial t} SC = SC\epsilon, \quad (8)$$

$$\frac{\partial}{\partial t} C^\dagger SC = 0, \quad (9)$$

where the full definition of the Fock matrix is

$$F_{\mu\nu} = h_{\mu\nu} + D_{\lambda\sigma} [2J_{\mu\nu\lambda\sigma} - K_{\mu\nu\lambda\sigma}]$$

and the density matrix is defined as

$$D_{\mu\nu} = C_{\mu p} n_{pq} C_{\nu q}^\dagger,$$

where the diagonal occupation number matrix  $n_{ii} = 2$  and  $n_{aa} = 0$  for RHF.

In general, the MO coefficients are perturbation- and time-dependent, but the basis functions themselves are not. This means that when the series expansion for the perturbation above is performed on other quantities, only  $F$ ,  $C$ ,  $\epsilon$ , and  $D$  are affected. For example, the Lagrangian multiplier matrix  $\epsilon$  can be expanded as

$$\epsilon(\mathbf{E}) = \epsilon^0 + E_a \epsilon^a + \frac{1}{2!} E_a E_b \epsilon^{ab} + \frac{1}{3!} E_a E_b E_c \epsilon^{abc} + \dots$$

where  $a, b, c, \dots \in \{x, y, z\}$ , and

$$\epsilon^a = e^{\pm i\omega t} \epsilon^a(\pm\omega) + \epsilon^a(0), \quad (10)$$

$$\epsilon^{ab} = e^{\pm 2i\omega t} \epsilon^{ab}(\pm\omega, \pm\omega) + e^{\pm i\omega t} \{ \epsilon^{ab}(0, \pm\omega) + \epsilon^{ab}(\pm\omega, 0) \} + \epsilon^{ab}(\pm\omega, \mp\omega) + \epsilon^{ab}(0, 0), \quad (11)$$

$$\epsilon^{abc} = e^{\pm 3i\omega t} \epsilon^{abc}(\pm\omega, \pm\omega, \pm\omega) + e^{\pm 2i\omega t} \{ \epsilon^{abc}(0, \pm\omega, \pm\omega) + \epsilon^{abc}(\pm\omega, 0, \pm\omega) + \epsilon^{abc}(\pm\omega, \pm\omega, 0) \} + e^{\pm i\omega t} \{ \epsilon^{abc}(\pm\omega, \pm\omega, \mp\omega) + \dots \} \quad (12)$$

showing that each order of the expansion consists of all possible phase combinations. For the first hyperpolarizability, only quantities with at most two field indices are required. Each permutationally unique subterm of the expansion corresponds to a different physical observable:

- (0)  $\rightarrow$  static polarizability  $\rightarrow \alpha(0; 0) = -Tr[H^a D^b(0)]$

- $(\pm\omega) \rightarrow$  dynamic polarizability  $\rightarrow \alpha(\mp\omega; \pm\omega) = -Tr[H^a D^b(\pm\omega)]$
- $(0, 0) \rightarrow$  static (first) hyperpolarizability  $\rightarrow \beta(0; 0, 0) = -Tr[H^a D^{bc}(0, 0)]$
- $(0, \pm\omega) \rightarrow$  electrooptic Pockels effect (EOPE)  $\rightarrow \beta(\mp\omega; 0, \pm\omega) = -Tr[H^a D^{bc}(0, \pm\omega)]$
- $(\pm\omega, \pm\omega) \rightarrow$  second harmonic generation (SHG)  $\rightarrow \beta(\mp 2\omega; \pm\omega, \pm\omega) = -Tr[H^a D^{bc}(\pm\omega, \pm\omega)]$
- $(\pm\omega, \mp\omega) \rightarrow$  optical rectification  $\rightarrow \beta(0; \pm\omega, \mp\omega) = -Tr[H^a D^{bc}(\pm\omega, \mp\omega)]$

where each property is calculated as the trace over the AO-basis dipole matrices  $H^a$  with the appropriate perturbed density. The task now comes down to calculating the necessary perturbed density for the phenomenon of interest. The second-order densities required for the four different first hyperpolarizabilities are (Karna eqs. III-2a to III-2d)

$$D^{ab}(\pm\omega, \pm\omega) = C^{ab}(\pm\omega, \pm\omega)nC^{0\dagger} + C^a(\pm\omega)nC^{b\dagger}(\mp\omega) + C^b(\pm\omega)nC^{a\dagger}(\mp\omega) + C^0nC^{ab\dagger}(\mp\omega, \mp\omega), \quad (13)$$

$$D^{ab}(0, \pm\omega) = C^{ab}(0, \pm\omega)nC^{0\dagger} + C^a(0)nC^{b\dagger}(\mp\omega) + C^b(\pm\omega)nC^{a\dagger}(0) + C^0nC^{ab\dagger}(0, \mp\omega), \quad (14)$$

$$D^{ab}(\pm\omega, \mp\omega) = C^{ab}(\pm\omega, \mp\omega)nC^{0\dagger} + C^a(\pm\omega)nC^{b\dagger}(\pm\omega) + C^b(\mp\omega)nC^{a\dagger}(\mp\omega) + C^0nC^{ab\dagger}(\mp\omega, \pm\omega), \quad (15)$$

$$D^{ab}(0, 0) = C^{ab}(0, 0)nC^{0\dagger} + C^a(0)nC^{b\dagger}(0) + C^b(0)nC^{a\dagger}(0) + C^0nC^{ab\dagger}(0, 0). \quad (16)$$

Already a few important insights about the equations are revealed: Each perturbation index always carries its respective frequency, and the positive and negative frequencies are related by the Hermitian adjoint (except for  $C(-\omega) = -C^0U^\dagger(+\omega)$ , Karna eq. 40). We also see the appearance of terms like  $C^{ab}$ , which will require  $U^{ab}$  originating from the second-order CPHF. Computationally, this is undesirable due to the increased number of iterative calculations that must be performed, so we borrow a trick that most prominently appears in perturbation theory.

### 1.2.1 Wigner's $2n + 1$ rule

From Schaefer, page 25:

When the wavefunction is determined up to the  $n$ th order, the expectation value (electronic energy) of the the system is resolved, according to the results of perturbation theory, up to the  $(2n + 1)$ st order. This principle is called Wigner's  $2n + 1$  theorem [29-31].

	CI: MO/CI space	MCSCF: MO/CI space	RHF: MO space
Energy, $E$	$C_\mu^i, C_I$	$C_\mu^i, C_I$	$C_\mu^i$
First Derivative, $\frac{\partial E}{\partial a}$	$U^a, C_I$	$C_\mu^i, C_I$	$C_\mu^i$
Second Derivative, $\frac{\partial^2 E}{\partial a \partial b}$	$U^{ab}, \frac{\partial C_I}{\partial a}$	$U^a, \frac{\partial C_I}{\partial a}$	$U^a$
Third Derivative, $\frac{\partial^3 E}{\partial a \partial b \partial c}$	$U^{abc}, \frac{\partial C_I}{\partial a}$	$U^a, \frac{\partial C_I}{\partial a}$	$U^a$
Fourth Derivative, $\frac{\partial^4 E}{\partial a \partial b \partial c \partial d}$	$U^{abcd}, \frac{\partial^2 C_I}{\partial a \partial b}$	$U^{ab}, \frac{\partial^2 C_I}{\partial a \partial b}$	$U^{ab}$
Fifth Derivative, $\frac{\partial^5 E}{\partial a \partial b \partial c \partial d \partial e}$	$U^{abcde}, \frac{\partial^2 C_I}{\partial a \partial b}$	$U^{ab}, \frac{\partial^2 C_I}{\partial a \partial b}$	$U^{ab}$

Since the first hyperpolarizability is calculated as a third derivative of the energy, perturbed coefficients with only one field index should be required. From the above table, we can also see why SCF gradients ( $\frac{\partial E}{\partial R_A}$ , where  $R_A$  is the  $A$ -th Cartesian component of nucleus  $R$ ) avoid the need to solve for  $U$  matrices.

### 1.3 Final expressions

To this point, most work has been in the AO basis, but it is conceptually easier to work in the MO basis, in particular due to the use of the  $\epsilon$  equations (Karna eq. 34)

$$\epsilon^a(\pm\omega) = G^a(\pm\omega) + \epsilon^0 U^a(\pm\omega) - U^a(\pm\omega) \epsilon^0 \pm \omega U^a(\pm\omega),$$

where the  $G$  matrices are the MO-basis Fock matrices

$$G^{ab\dots} = C^{0\dagger} F^{ab\dots} C^0,$$

and the  $U$  matrices are the MO-basis perturbation parameters

$$C^{ab\dots} = C^0 U^{ab\dots},$$

which will be discussed in the implementation. The final expression for the static hyperpolarizability is (Karna eq. VII-4)

$$\beta_{abc}(0;0,0) = Tr[n\{U^a(0)G^b(0)U^c(0)+U^c(0)G^b(0)U^a(0)+U^b(0)G^c(0)U^a(0)+U^a(0)G^c(0)U^b(0)+U^c(0)G^a(0)U^b(0)+U^b(0)G^a(0)U^c(0)\}]$$

By noticing that each term corresponds to a unique permutation of the field indices, it can be rewritten as

$$\beta_{abc} = Tr \left[ n \sum \mathcal{P}(d,e,f) U^d G^e U^f \right] - Tr \left[ n \sum \mathcal{P}(d,e,f) U^d U^e \epsilon^f \right],$$

where the permutation indices are initially assigned as  $d = a, e = b, f = c$ . The frequency notation has also been dropped, since each  $abc$  (and therefore each  $def$ ) will always carry the appropriate field index, making this the most general form of the first hyperpolarizability. If the indices  $abc$  are also permuted, then all 27 components of the first hyperpolarizability tensor will be computed.

### 1.4 Computational Procedure

The basic quantities we need are the matrices  $C, \mu, F, \epsilon$ , and  $U$ . The MO coefficients  $C$  are already obtained from the ground-state calculation, along with  $\epsilon^0$  (the MO energies) and  $F^0$  (the AO-basis Fock matrix). The dipole matrices  $\mu$  are needed for the linear response (polarizability) calculation, which results in response vectors that compose the off-diagonal blocks of the  $U^a$  matrix.  $G^a$  is obtained from  $F^a$ , which comes from performing a single Fock build with the perturbed density  $D^a$ . Finally,  $\epsilon^a$  can be constructed.

Although the expressions so far are general for any frequency and first-order non-linear optical response, the tutorial implementation will cover the static case. For second-harmonic generation, see the reference implementation.

---

```

1 import numpy as np
2 np.set_printoptions(3, linewidth=100, suppress=True)      # when we inspect the vectors/matrices,
3                                                         # use a prettier format for printing
4 import psi4

```

---

The energy and density convergence criteria are tightened from defaults, as response properties are sensitive to the quality of the ground-state wavefunction.

---

```

1 mol = psi4.geometry('''
2     O
3     H 1 0.9435
4     H 1 0.9435 2 105.9443
5     symmetry c1

```

---

```

6  '''
7  psi4.set_options({
8      "basis": "aug-cc-pVDZ",
9      "scf_type": "direct",
10     "df_scf_guess": False,
11     "e_convergence": 1e-9,
12     "d_convergence": 1e-9,
13 })

```

---

```

1  # This is to enable testing outside of the notebook environment.
2  import sys
3  try:
4      get_ipython()
5      sys.path.append('../Response-Theory/Self-Consistent-Field')
6  except NameError:
7      import os.path
8      dirname = os.path.dirname(os.path.abspath(__file__))
9      sys.path.append(os.path.join(dirname, '../Response-Theory/Self-Consistent-Field'))
10
11  from helper_CPHF import helper_CPHF

```

---

The helper encapsulates the solution of the ground-state wavefunction followed by the frequency-(in)dependent linear response equations,

$$\left[ \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{A}^* \end{pmatrix} - \omega_f \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Delta} \\ -\boldsymbol{\Delta}^* & -\boldsymbol{\Sigma}^* \end{pmatrix} \right] \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} = \begin{pmatrix} \mathbf{V} \\ -\mathbf{V}^* \end{pmatrix},$$

either directly (via matrix inversion in the MO basis) or iteratively (via repeated matrix-vector products using Fock builds). For a HF/DFT reference with canonical orbitals, the above equations reduce to

$$\left[ \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{pmatrix} - \omega_f \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} \end{pmatrix} \right] \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} = \begin{pmatrix} \mathbf{V} \\ -\mathbf{V} \end{pmatrix}.$$

In the static limit ( $\omega_f = 0$ ), the whole superoverlap matrix vanishes, and the CPHF equations can be reduced to those used in tutorial 6a.

---

```

1  solver = helper_CPHF(mol)
2  solver.run()

```

---

```

1  Number of occupied orbitals: 5
2  Number of basis functions: 41
3
4  Tensor sizes:
5  ERI tensor          0.02 GB.
6  oNNN MO tensor      0.00 GB.
7  ovov Hessian tensor 0.00 GB.
8
9  Forming Hessian...
10 ...formed Hessian in 0.473 seconds.
11
12 Inverting Hessian...
13 ...inverted Hessian in 0.007 seconds.

```

---

Because the calculation of  $\beta$  requires  $U^a$ , we also obtain linear response properties from a quadratic response calculation. This holds for any order of response, where lower-order response functions are automatically obtained from higher-order response calculations.

---

```
1 print(np.around(solver.polar, 4))
```

---

```
1 [[ 7.2587 -0.      0.    ]
2  [-0.      8.7969  0.    ]
3  [ 0.      0.      7.854 ]]
```

---

---

```
1 # epsilon~{0}
2 moenergies = solver.epsilon
3 C = np.asarray(solver.C)
4 Co = solver.Co
5 Cv = solver.Cv
6 nbf, norb = C.shape
7 nocc = Co.shape[1]
8 nvir = norb - nocc
9 nov = nocc * nvir
10 # the response vectors X_x, X_y, X_z; Y_x, Y_y, Y_z not needed separately for static response
11 x = np.asarray(solver.x)
12 ncomp = x.shape[0]
13 # reuse the AO-basis dipole integrals
14 integrals_ao = np.asarray([np.asarray(dipole_ao_component)
15                               for dipole_ao_component in solver.tmp_dipoles])
16 print("dimension of response vectors from linear response: {}".format(x.shape))
17 # for dynamic response, this will be (2 * nov)
18 assert x.shape[1] == nov
```

---

```
1 dimension of response vectors from linear response: (3, 180)
```

---

The foundation of the CPHF equations is that the right-hand side  $\mathbf{V}$  is a perturbation on the wavefunction causing single excitations from the occupied orbitals to virtual orbitals, the coefficients of which are in the response vectors  $\mathbf{X}$ ; the vectors  $\mathbf{Y}$  describe single deexcitations. Because the full (square)  $\mathbf{U}$  matrices are required, all MO-based quantities must be of shape  $[N_{orb}, N_{orb}]$  rather than  $[N_{occ}, N_{vir}]$ .

---

```
1 # form full MO-basis dipole integrals
2 integrals_mo = np.empty(shape=(ncomp, norb, norb))
3 for i in range(ncomp):
4     integrals_mo[i, ...] = (C.T).dot(integrals_ao[i, ...]).dot(C)
```

---

Similarly,  $\mathbf{X}$  and  $\mathbf{Y}$  form the off-diagonal blocks of the  $\mathbf{U}$  matrices. They are usually stored as in DALTON, where each vector is of length  $2N_{ov}$ , with  $\mathbf{X}$  on top of  $\mathbf{Y}$ .

---

```
1 # repack response vectors to [norb, norb]; 1/2 is due to X + Y
2 U = np.zeros_like(integrals_mo)
3 for i in range(ncomp):
4     U[i, :nocc, nocc:] = 0.5 * x[i, ...].reshape(nocc, nvir)
5     U[i, nocc:, :nocc] = -0.5 * x[i, ...].reshape(nocc, nvir).T
```

---

A minor implementation detail: because this was not a frequency-dependent calculation, only  $\mathbf{X} + \mathbf{Y}$  needs to be calculated; as they are identical, this leads to the prefactor of 1/2.

---

```
1 # form G matrices from perturbation and generalized Fock matrices; do
2 # one more Fock build for each response vector
3 jk = psi4.core.JK.build(solver.scf_wfn.basisset())
```

---

```

4  jk.initialize()
5  G = np.empty_like(U)
6  R = psi4.core.Matrix(nbf, nocc)
7  npR = np.asarray(R)
8  for i in range(ncomp):
9      V = integrals_mo[i, ...]
10
11      # eqn. (III-1b)
12      # Note: this simplified handling of the response vector
13      # transformation for the Fock build is insufficient for
14      # frequency-dependent response. 1/2 is due to X + Y
15      jk.C_clear()
16      L = Co
17      npR[...] = x[i, ...].reshape(nocc, nvir).dot(np.asarray(Cv).T).T
18      jk.C_left_add(L)
19      jk.C_right_add(R)
20      jk.compute()
21      J = 0.5 * np.asarray(jk.J()[0])
22      K = 0.5 * np.asarray(jk.K()[0])
23
24      # eqn. (21b)
25      F = (C.T).dot(4 * J - K.T - K).dot(C)
26      G[i, ...] = V + F
27
28      # form epsilon matrices, eqn. (34)
29      E = G.copy()
30      omega = 0
31      for i in range(ncomp):
32          eoU = (moenergies[..., np.newaxis] + omega) * U[i, ...]
33          Ue = U[i, ...] * moenergies[np.newaxis, ...]
34          E[i, ...] += (eoU - Ue)
35
36      # Assume some symmetry and calculate only part of the tensor.
37      # eqn. (VII-4)
38      hyperpolarizability = np.zeros(shape=(6, 3))
39      off1 = [0, 1, 2, 0, 0, 1]
40      off2 = [0, 1, 2, 1, 2, 2]
41      for r in range(6):
42          b = off1[r]
43          c = off2[r]
44          for a in range(3):
45              t11 = 2 * np.trace(U[a, ...].dot(G[b, ...]).dot(U[c, ...])[:nocc, :nocc])
46              t12 = 2 * np.trace(U[a, ...].dot(G[c, ...]).dot(U[b, ...])[:nocc, :nocc])
47              t13 = 2 * np.trace(U[c, ...].dot(G[a, ...]).dot(U[b, ...])[:nocc, :nocc])
48              tr1 = np.trace(U[c, ...].dot(U[b, ...]).dot(E[a, ...])[:nocc, :nocc])
49              tr2 = np.trace(U[b, ...].dot(U[c, ...]).dot(E[a, ...])[:nocc, :nocc])
50              tr3 = np.trace(U[c, ...].dot(U[a, ...]).dot(E[b, ...])[:nocc, :nocc])
51              tr4 = np.trace(U[a, ...].dot(U[c, ...]).dot(E[b, ...])[:nocc, :nocc])
52              tr5 = np.trace(U[b, ...].dot(U[a, ...]).dot(E[c, ...])[:nocc, :nocc])
53              tr6 = np.trace(U[a, ...].dot(U[b, ...]).dot(E[c, ...])[:nocc, :nocc])
54              t1 = t11 + t12 + t13
55              tr = tr1 + tr2 + tr3 + tr4 + tr5 + tr6
56              hyperpolarizability[r, a] = -2 * (t1 - tr)

```

---

```

1  ref_static = np.array([
2      [ 0.00000001,  0.00000000, -0.10826460],
3      [ 0.00000000,  0.00000000, -11.22412215],
4      [ 0.00000000,  0.00000000, -4.36450397],
5      [ 0.00000000,  0.00000000, -0.00000001],
6      [-0.10826460, -0.00000001,  0.00000000],
7      [-0.00000001, -11.22412215,  0.00000000]
8  ])
9  assert np.allclose(ref_static, hyperpolarizability, rtol=0.0, atol=1.0e-3)

```

```

10 print('\nFirst dipole hyperpolarizability (static):')
11 print(hyperpolarizability)

```

---

```

1 First dipole hyperpolarizability (static):
2 [[ -0.         -0.         -0.10826]
3  [ -0.         -0.        -11.22412]
4  [ -0.         -0.        -4.3645 ]
5  [ -0.         -0.         0.         ]
6  [ -0.10826    0.         -0.         ]
7  [  0.         -11.22412  -0.         ]]

```

---

## 1.5 References

### 1.5.1 Primary equations and implementation

- Karna, S. P.; Dupuis, M. *J. Comput. Chem.* **1991**, *12*, 487–504, DOI: 10.1002/jcc.540120409

### 1.5.2 $2n + 1$ rule

- Yamaguchi, Y.; Goddard, J. D.; Osamura, Y.; Schaefer III, H. F., *A New Dimension to Quantum Chemistry: Analytic Derivative Methods in Ab Initio Molecular Electronic Structure Theory*; International Series of Monographs on Chemistry; Oxford University Press: 1994
- Epstein, S. T. *J. Chem. Phys.* **1968**, *48*, 4725–4726, DOI: 10.1063/1.1668053
- Epstein, S. T. *Chem. Phys. Lett.* **1980**, *70*, 311–312, DOI: 10.1016/0009-2614(80)85340-1

### 1.5.3 Additional reading

- Neese, F. *Coord. Chem. Rev.* **2009**, *253*, 526–563, DOI: 10.1016/j.ccr.2008.05.014