

# Modern Quantum Chemistry, Szabo & Ostlund

## HW

WSR

May 1, 2021

### Contents

<b>A</b>	<b>Integral Evaluation with 1s Primitive Gaussians</b>	<b>2</b>
<b>B</b>	<b>2-Electron Self-consistent-field Program</b>	<b>2</b>
<b>C</b>	<b>Analytic Derivative Methods and Geometry Optimization</b>	<b>2</b>
C.1	Introduction . . . . .	2
C.2	General Considerations . . . . .	2
C.3	Analytic Derivatives . . . . .	2
C.4	Optimization Techniques . . . . .	2
C.5	Some Optimization Algorithms . . . . .	2
Ex C.1	. . . . .	2
Ex C.2	. . . . .	3
Ex C.3	. . . . .	3
Ex C.4	. . . . .	3
C.6	Transition States . . . . .	3
C.7	Constrained Variation . . . . .	3
<b>D</b>	<b>Molecular Integrals for H<sub>2</sub> as a Function of Bond Length</b>	<b>3</b>

## A Integral Evaluation with 1s Primitive Gaussians

## B 2-Electron Self-consistent-field Program

## C Analytic Derivative Methods and Geometry Optimization

### C.1 Introduction

### C.2 General Considerations

### C.3 Analytic Derivatives

### C.4 Optimization Techniques

### C.5 Some Optimization Algorithms

#### Ex C.1

(a)

$$\begin{aligned}\mathbf{H} &= \begin{pmatrix} \frac{\partial^2 E}{\partial x^2} & \frac{\partial^2 E}{\partial x \partial y} \\ \frac{\partial^2 E}{\partial y \partial x} & \frac{\partial^2 E}{\partial y^2} \end{pmatrix} \\ &= \begin{pmatrix} K & K'' \\ K'' & K' \end{pmatrix}\end{aligned}\tag{C.5.1}$$

$$\begin{aligned}\mathbf{f}(\mathbf{X}) &= \begin{pmatrix} \frac{\partial E}{\partial x} \\ \frac{\partial E}{\partial y} \end{pmatrix} \\ &= \begin{pmatrix} K(x-a) + K''y \\ K'(y-b) + K''x \end{pmatrix} \\ &= \begin{pmatrix} K & K'' \\ K'' & K' \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} Ka \\ K'b \end{pmatrix} \\ &= \begin{pmatrix} K & K'' \\ K'' & K' \end{pmatrix} \mathbf{X} - \begin{pmatrix} Ka \\ K'b \end{pmatrix}\end{aligned}\tag{C.5.2}$$

$$\begin{aligned}\mathbf{q} &= -\mathbf{H}^{-1}\mathbf{f} \\ &= -\mathbf{X} + \begin{pmatrix} K & K'' \\ K'' & K' \end{pmatrix}^{-1} \begin{pmatrix} Ka \\ K'b \end{pmatrix} \\ &= -\mathbf{X} + \frac{1}{KK' - K''^2} \begin{pmatrix} K' & -K'' \\ -K'' & K \end{pmatrix} \begin{pmatrix} Ka \\ K'b \end{pmatrix} \\ &= -\mathbf{X} + \frac{1}{KK' - K''^2} \begin{pmatrix} KK'a - K'K''b \\ -KK''a + KK'b \end{pmatrix} \\ &= -\mathbf{X} + \frac{1}{KK' - K''^2} \begin{pmatrix} KK' & -K'K'' \\ -KK'' & KK' \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}\end{aligned}\tag{C.5.3}$$

(b) Since  $\mathbf{q} = \mathbf{X}_e - \mathbf{X}$ ,

$$\begin{aligned}\mathbf{X}_e &= \frac{1}{KK' - K''^2} \begin{pmatrix} KK' & -K'K'' \\ -KK'' & KK' \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \\ &= \frac{1}{0.1 - K''^2} \begin{pmatrix} 0.1 & -0.1K'' \\ -K'' & 0.1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix}\end{aligned}\tag{C.5.4}$$

$K''$	$\mathbf{X}_e = (x_e, y_e)$
0	(3.000, 2.000)
0.010	(2.983, 1.702)
0.030	(2.967, 1.110)

## Ex C.2

$$\begin{aligned}\mathbf{H} &= \begin{pmatrix} K & K'' \\ K'' & K' \end{pmatrix} \\ &= \begin{pmatrix} 1.000 & 0.030 \\ 0.030 & 0.100 \end{pmatrix}\end{aligned}\tag{C.5.5}$$

$$\begin{aligned}\mathbf{f} &= \begin{pmatrix} K & K'' \\ K'' & K' \end{pmatrix} \mathbf{X} - \begin{pmatrix} Ka \\ K'b \end{pmatrix} \\ &= \begin{pmatrix} 1.000 & 0.030 \\ 0.030 & 0.100 \end{pmatrix} \begin{pmatrix} 3.3 \\ 1.8 \end{pmatrix} - \begin{pmatrix} 1.000 \times 3.00 \\ 0.100 \times 2.00 \end{pmatrix} \\ &= \begin{pmatrix} 0.354 \\ 0.079 \end{pmatrix}\end{aligned}\tag{C.5.6}$$

$$\begin{aligned}\mathbf{q} &= -\mathbf{H}^{-1}\mathbf{f} \\ &= -\begin{pmatrix} 1.000 & 0.030 \\ 0.030 & 0.100 \end{pmatrix}^{-1} \begin{pmatrix} 0.354 \\ 0.079 \end{pmatrix} \\ &= (-0.333, -0.690)\end{aligned}\tag{C.5.7}$$

thus

$$\begin{aligned}\mathbf{X}_e &= \mathbf{q} + \mathbf{X} \\ &= (2.967, 1.110)\end{aligned}\tag{C.5.8}$$

which agrees with the result in Ex C.1(b).

**Ex C.3** A program is written to solve this problem, which is `C-3.py`.

For example, run the program by `python C-3.py 0.03`, and the Nelder-Mead optimization steps will be printed for  $K'' = 0.03$ .

**Ex C.4** A program is written to solve this problem, which is `C-4.py`.

For example, run the program by `python C-4.py`, and the MS optimization steps will be printed.

## C.6 Transition States

## C.7 Constrained Variation

## D Molecular Integrals for $\text{H}_2$ as a Function of Bond Length