Maple Quantum Chemistry Toolbox

The Maple Quantum Chemistry Toolbox from RDMChem, a separate add-on product to Maple, is a powerful environment for the computation and visualization of the electronic structure of molecules. In Maple 2021, this toolbox has significant new features and enhancements: (1) a new command that provides a treasure trove of information about molecules−a molecular dictionary, (2) a new method for molecules that further enhances the package's suite of electronic structure solvers, (3) a new command for purifying density matrices with applications to the mitigation of errors in quantum computing, (4) a new optional parameter for the plotting of molecular orbitals that allows for customized colors, (5) new optional parameters for the generation of molecular integrals that support arbitrary molecular orbitals and active spaces, (6) a new addition to the ≈30 builtin lessons for classroom learning in undergraduate-to-graduate chemistry and physics, (7) updates to the interactive Maplet, and (8) additional enhancements to many methods and commands throughout the package.

Note that the Maple Quantum Chemistry Toolbox (QCT) is required in order to execute the examples in this worksheet.

Molecular Dictionary

A molecular dictionary appears in QCT 2021 with access to online data for more than 50, 000 molecules. The command *MolecularDictionary* accepts the name of a molecule as a Maple string or CID integer. For example, we retrieve the entries for the following two molecules: nitrogen ("dinitrogen") gas and the Cys-Ser peptide. Before we begin we load the *QuantumChemistry* package,

```
> with(QuantumChemistry);
```

```
[AOLabels, ActiveSpaceCI, ActiveSpaceSCF, AtomicData, BondAngles, BondDistances, Charges,
    ChargesPlot, ContractedSchrodinger, CorrelationEnergy, CoupledCluster, DensityFunctional,
    DensityPlot3D, Dipole, DipolePlot, Energy, ExcitationEnergies, ExcitationSpectra,
    ExcitationSpectraPlot, ExcitedStateEnergies, ExcitedStateSpins, FullCI, GeometryOptimization,
    HartreeFock, Interactive, Isotopes, MOCoefficients, MODiagram, MOEnergies, MOIntegrals,
    MOOccupations, MOOccupationsPlot, MOSymmetries, MP2, MolecularData,
    MolecularDictionary, MolecularGeometry, NuclearEnergy, NuclearGradient, OscillatorStrengths,
    Parametric2RDM, PlotMolecule, Populations, Purify2RDM, RDM1, RDM2, RTM1, ReadXYZ,
Restore, Save, SaveXYZ, SearchBasisSets, SearchFunctionals, SkeletalStructure, Thermodynamics,
TransitionDipolePlot, TransitionDipoles, TransitionOrbitalPlot, TransitionOrbitals,
Variational2RDM, VibrationalModeAnimation, VibrationalModes, Video]
```

For nitrogen gas we have

```
> MolecularDictionary("nitrogen");
```

```
Nitrogen: Dinitrogen is an elemental molecule consisting of two
trivalently-bonded nitrogen atoms. It has a role as a member of food
packaging gas and a food propellant. It is a diatomic nitrogen, a gas
molecular entity and an elemental molecule. It is a conjugate base of
a diazynium. -ChEBI

Nitrogen: Nitrogen is an element with atomic symbol N, atomic number
7, and atomic weight 14.01. -NCI Thesaurus (NCIt)
```

while for the Cys-Ser peptide we have

```
> MolecularDictionary("Cys-Ser");
```

```
Cys-Ser: Cys-Ser is a dipeptide composed of L-cysteine and L-serine
joined by a peptide linkage. It has a role as a metabolite. It is a
dipeptide, a secondary carboxamide, a primary alcohol, a primary amino
compound, a thiol and a carboxylic acid. It derives from a L-cysteine
and a L-serine. -ChEBI
```

Similarly, we can explore many, many other molecules. What about the molecule ozone that is sometimes generated before a thunderstorm?

```
> MolecularDictionary("ozone");
```

```
Ozone: Ozone is an elemental molecule with formula O3. An explosive,
pale blue gas (b.p. -112###) that has a characteristic, pleasant
odour, it is continuously produced in the upper atmosphere by the
action of solar ultraviolet radiation on atmospheric oxygen. It is an
antimicrobial agent used in the production of bottled water, as well
as in the treatment of meat, poultry and other foodstuffs. It has a
role as a member of greenhouse gas, a disinfectant, a tracer, an
electrophilic reagent, a mutagen, an oxidising agent and an antiseptic
drug. It is a member of reactive oxygen species, an elemental
molecule, a triatomic oxygen and a gas molecular entity. -ChEBI
```

Or the molecule ascorbic acid (Vitamin C)?

```
> MolecularDictionary("ascorbic acid");
```

```
Ascorbic acid: L-ascorbic acid is the L-enantiomer of ascorbic acid
and conjugate acid of L-ascorbate. It has a role as a water-soluble
vitamin, a vitamin C, a coenzyme, a flour treatment agent, a food
antioxidant, a plant metabolite and a cofactor. It is a conjugate acid
of a L-ascorbate. It is an enantiomer of a D-ascorbic acid. -ChEBI

Ascorbic acid: Ascorbic Acid is a natural water-soluble vitamin
(Vitamin C). Ascorbic acid is a potent reducing and antioxidant agent
that functions in fighting bacterial infections, in detoxifying
reactions, and in the formation of collagen in fibrous tissue, teeth,
bones, connective tissue, skin, and capillaries. Found in citrus and
other fruits, and in vegetables, vitamin C cannot be produced or
```

```
stored by humans and must be obtained in the diet. (NCI04) -NCI
Thesaurus (NCIt)
```

Or that important molecule in coffee?

```
> MolecularDictionary("caffeine");
```

```
Caffeine: Caffeine is a trimethylxanthine in which the three methyl
groups are located at positions 1, 3, and 7. A purine alkaloid that
occurs naturally in tea and coffee. It has a role as a central nervous
system stimulant, an EC 3.1.4.* (phosphoric diester hydrolase)
inhibitor, an adenosine receptor antagonist, an EC 2.7.11.1 (non-
specific serine/threonine protein kinase) inhibitor, a ryanodine
receptor agonist, a fungal metabolite, an adenosine A2A receptor
antagonist, a psychotropic drug, a diuretic, a food additive, an
adjuvant, a plant metabolite, an environmental contaminant, a
xenobiotic, a human blood serum metabolite, a mouse metabolite and a
mutagen. It is a purine alkaloid and a trimethylxanthine. -ChEBI

Caffeine: Caffeine is a methylxanthine alkaloid found in the seeds,
nuts, or leaves of a number of plants native to South America and East
Asia that is structurally related to adenosine and acts primarily as
an adenosine receptor antagonist with psychotropic and anti-
inflammatory activities. Upon ingestion, caffeine binds to adenosine
receptors in the central nervous system (CNS), which inhibits
adenosine binding. This inhibits the adenosine-mediated downregulation
of CNS activity; thus, stimulating the activity of the medullary,
vagal, vasomotor, and respiratory centers in the brain. This agent
also promotes neurotransmitter release that further stimulates the
CNS. The anti-inflammatory effects of caffeine are due the
nonselective competitive inhibition of phosphodiesterases (PDEs).
Inhibition of PDEs raises the intracellular concentration of cyclic
AMP (cAMP), activates protein kinase A, and inhibits leukotriene
synthesis, which leads to reduced inflammation and innate immunity. -
NCI Thesaurus (NCIt)
```

The database also contains many more technical molecules such as the antiviral drug remdesivir that was recently repurposed to treat Covid-19.

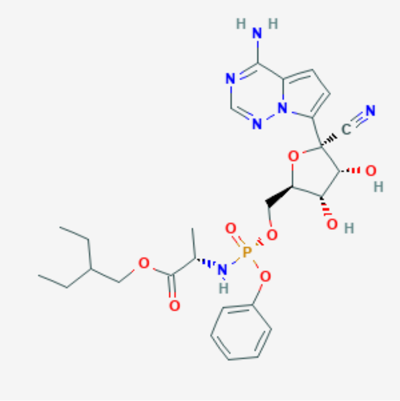
```
> MolecularDictionary("remdesivir");
```

```
Remdesivir: Remdesivir is a carboxylic ester resulting from the formal
condensation of the carboxy group of N-[(S)-{[(2R,3S,4R,5R)-5-(4-
aminopyrrolo[2,1-f][1,2,4]triazin-7-yl)-5-cyano-3,4-
dihydroxytetrahydrofuran-2-yl]methoxy}(phenoxy)phosphoryl]-L-alanine
with the hydroxy group of 2-ethylbutan-1-ol. A broad-spectrum
antiviral prodrug with potent in vitro antiviral activity against a
diverse panel of RNA viruses such as Ebola virus, MERS-CoV and SARS-
CoV. It is currently in Phase III clinical trials for the treatment of
Covid-19 in adults. It has a role as an antiviral drug, a prodrug and
an anticoronaviral agent. It is a carboxylic ester, a pyrrolotriazine,
a nitrile, a phosphoramidate ester, a C-nucleoside and an aromatic
amine. It derives from a GS-441524. -ChEBI

Remdesivir: Remdesivir is a prodrug of an adenosine triphosphate (ATP)
analog, with potential antiviral activity against a variety of RNA
viruses. Upon administration, remdesivir, being a prodrug, is
metabolized into its active form GS-441524. As an ATP analog,
GS-441524 competes with ATP for incorporation into RNA and inhibits
the action of viral RNA-dependent RNA polymerase. This results in the
termination of RNA transcription and decreases viral RNA production. -
NCI Thesaurus (NCIt)
```

The new *MolecularDictionary* command complements the existing commands *MolecularGeometry* and *SkeletalStructure* for retrieving molecular geometries and skeletal structures. For example, here is the skeletal structure for remdesivir

```
> SkeletalStructure("remdesivir");
```

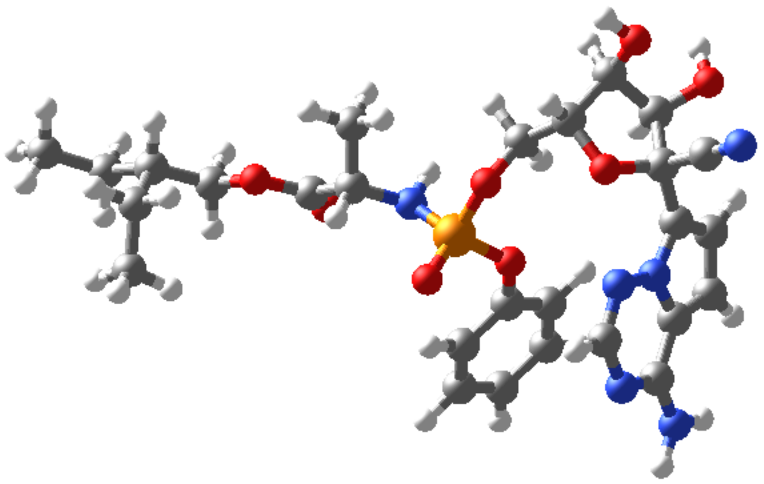


and its molecular geometry

```
> remdesivir := MolecularGeometry("remdesivir");
remdesivir := [["P", 0.71860000, 0.10970000, −0.76660000], ["O", −2.83790000, 2.01490000,
    −0.79800000], ["O", −4.60280000, 3.10730000, 2.14720000], ["O", −3.33010000,
    4.75570000, 0.36760000], ["O", −0.07640000, 1.52190000, −0.93430000], ["O",
    −0.38890000, −0.77740000, 0.01860000], ["O", 1.05700000, −0.46580000, −2.11880000],
    ["O", 5.56710000, 0.78540000, 0.04450000], ["O", 4.25640000, −0.39040000, 1.53130000],
    ["N", −3.90140000, −0.59850000, −0.58280000], ["N", −3.34010000, −0.39130000,
    −1.79850000], ["N", 1.97840000, 0.37370000, 0.27670000], ["N", −6.06950000, 2.93730000,
    −1.04000000], ["N", −3.45180000, −2.78050000, −2.09530000], ["N", −4.30780000,
    −4.26940000, −0.51260000], ["C", −3.95980000, 1.84290000, 0.10590000], ["C",
    −3.52950000, 2.58210000, 1.38960000], ["C", −2.58500000, 3.65710000, 0.88790000], ["C",
    −1.87900000, 2.94460000, −0.25310000], ["C", −4.21790000, 0.35750000, 0.34660000], ["C",
    −0.65550000, 2.15550000, 0.19440000], ["C", −5.14180000, 2.45600000, −0.53610000], ["C",
    −4.76810000, −0.28070000, 1.43770000], ["C", −4.24220000, −1.83550000, −0.09990000],
    ["C", −4.78420000, −1.67020000, 1.15620000], ["C", −3.99470000, −2.99540000,
    −0.92270000], ["C", 3.22850000, 0.90630000, −0.26190000], ["C", −3.14980000,
    −1.48650000, −2.47870000], ["C", 7.99230000, 0.90260000, 0.02990000], ["C", 6.73740000,
    0.33420000, 0.71240000], ["C", 8.01980000, 0.61290000, −1.48750000], ["C", 9.24690000,
    0.42250000, 0.78550000], ["C", 4.37400000, 0.34540000, 0.55990000], ["C", 3.21180000,
    2.42720000, −0.19330000], ["C", −0.18090000, −2.02330000, 0.54580000], ["C", 8.09250000,
−0.86050000, −1.86400000], ["C", 10.54730000, 1.00690000, 0.25290000], ["C", 0.74520000,
−2.88410000, −0.04300000], ["C", −0.89510000, −2.43210000, 1.67190000], ["C",
0.95730000, −4.15390000, 0.49430000], ["C", −0.68330000, −3.70160000, 2.20940000], ["C",
0.24290000, −4.56260000, 1.62060000], ["H", −2.96420000, 1.90360000, 2.04220000], ["H",
−1.91910000, 4.04370000, 1.66480000], ["H", −1.60480000, 3.62590000, −1.06620000],
["H", −0.92710000, 1.40590000, 0.94310000], ["H", 0.09090000, 2.82470000, 0.63360000],
["H", −4.22020000, 3.62550000, 2.87570000], ["H", −5.12360000, 0.19600000, 2.34060000],
["H", −2.69180000, 5.39520000, 0.00820000], ["H", −5.15100000, −2.45610000,
1.80150000], ["H", 1.77080000, 0.79490000, 1.18130000], ["H", 3.37230000, 0.56210000,
−1.29190000], ["H", 7.93990000, 1.99660000, 0.12380000], ["H", −2.69460000,
−1.38650000, −3.47770000], ["H", 6.70450000, 0.67470000, 1.75490000], ["H", 6.76100000,
−0.76200000, 0.71010000], ["H", 8.86860000, 1.13460000, −1.94470000], ["H", 7.12730000,
1.04590000, −1.95600000], ["H", 9.15420000, 0.70950000, 1.84080000], ["H", 9.31030000,
−0.67170000, 0.76940000], ["H", −4.72700000, −4.45630000, 0.39210000], ["H",
−4.12310000, −5.06830000, −1.10970000], ["H", 2.37960000, 2.84230000, −0.77100000],
["H", 4.13510000, 2.85010000, −0.60420000], ["H", 3.12470000, 2.78400000, 0.83960000],
["H", 7.22460000, −1.41360000, −1.49380000], ["H", 8.99880000, −1.33170000,
−1.47330000], ["H", 8.10960000, −0.96480000, −2.95390000], ["H", 10.77610000,
0.63290000, −0.74910000], ["H", 11.37950000, 0.72520000, 0.90630000], ["H", 10.50160000,
2.09980000, 0.21570000], ["H", 1.31380000, −2.60450000, −0.92210000], ["H",
−1.61490000, −1.76410000, 2.13670000], ["H", 1.67730000, −4.82510000, 0.03530000],
["H", −1.23940000, −4.01950000, 3.08640000], ["H", 0.40750000, −5.55130000, 2.03880000]]
```

which we can plot with the *PlotMolecule* command

```
> PlotMolecule(remdesivir);
```



With the dictionary entry, the skeletal structure, and the ball-and-stick plot we have all of the necessary background information to begin molecular calculations like the ones in the next section.

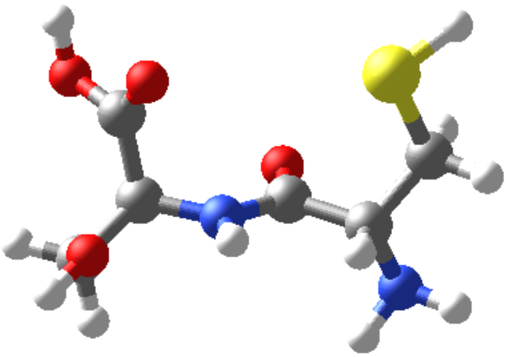
Computing with Molecules

In the QCT 2021 we implement a new method for the ground and excited states of molecules. The Schrödinger equation, developed by Erwin Schrödinger in 1925, is the key equation for solving quantum systems in non-relativistic physics. However, its exact solution scales exponentially with the number of electrons in the quantum system. To deal with this inefficiency, we solve a projection (or contraction) of the Schrödinger equation onto the space of only two electrons, known as a contracted Schrödinger equation. This contraction is especially useful for improving computational efficiency for strongly correlated molecules−molecules that deviate significantly from the mean-field (Hartree-Fock) picture of one electron in the field of the remaining electrons−for which many standard methods are often inaccurate.   Let us consider the peptide of the two amino acids L-cysteine and L-serine whose geometry we can import with the *MolecularGeometry* command

```
> mol := MolecularGeometry("Cys-Ser");
mol := [["S", −2.29090000, 2.10530000, −1.07690000], ["O", 2.52680000, −1.19180000,
    −1.63380000], ["O", −1.06460000, −0.04400000, 1.69560000], ["O", 2.66940000,
    1.27060000, 1.34030000], ["O", 1.68720000, 1.60900000, −0.67760000], ["N", 0.14240000,
    −0.64180000, −0.19310000], ["N", −3.00870000, −1.66870000, 0.29030000], ["C",
    1.44960000, −0.55430000, 0.41400000], ["C", −2.27570000, −0.56710000, −0.33790000],
    ["C", −1.02220000, −0.37450000, 0.51210000], ["C", 2.44060000, −1.50010000,
    −0.24810000], ["C", −3.16790000, 0.67650000, −0.36510000], ["C", 1.91400000, 0.88100000,
    0.28020000], ["H", 1.35260000, −0.79450000, 1.47910000], ["H", −2.01660000,
    −0.86110000, −1.36160000], ["H", 0.09180000, −0.84710000, −1.18670000], ["H",
    2.11130000, −2.53950000, −0.14810000], ["H", 3.43800000, −1.39630000, 0.19140000],
    ["H", −3.48660000, 0.95790000, 0.64500000], ["H", −4.06810000, 0.49580000,
    −0.96240000], ["H", −2.46140000, −2.52630000, 0.22140000], ["H", −3.87150000,
    −1.84260000, −0.22420000], ["H", 3.16370000, −1.81370000, −2.02500000], ["H",
    −3.30210000, 2.97810000, −0.95790000], ["H", 2.98800000, 2.19320000, 1.24240000]]
```

and plot with the *PlotMolecule* command

```
> PlotMolecule(mol);
```



In this example we showcase the use of two active spaces, an active space of 6 electrons in 6 orbitals to account for any potential static (strong) electron correlation and a larger active space to account for additional (dynamic) electron correlation. (This second active space is chosen to be [16,16] to make the calculation quick but much larger second active spaces are possible.)

```
> data_peptide := ContractedSchrodinger(mol, active = [6, 6], active_cse = [16, 16]);
```

$$\mathit{data\_peptide} := \mathrm{table}\left(\left[\mathit{dipole} = \begin{bmatrix} 0.95773786 \\ -2.37658291 \\ -1.59751383 \end{bmatrix}, \mathit{rdm2} = [[[1.99517719, 0.00959108, \right.\right.$$

$$-0.00154178, 0.00053321, 0.00006690, 0.00008319, \ldots],$$

$$[0.00959108, 3.88234515, 0.02055790, -0.00200413, -0.00047973, -0.00010074, \ldots],$$
$$[-0.00154178, 0.02055790, 3.98740585, 0.00175947, 0.00015780, -0.00003534, \ldots],$$
$$[0.00053321, -0.00200413, 0.00175947, 3.99272582, 0.00008538, -0.00016065, \ldots],$$
$$[0.00006690, -0.00047973, 0.00015780, 0.00008538, 3.99026788, -9.62404863\ 10^{-6}, \ldots],$$
$$[0.00008319, -0.00010074, -0.00003534, -0.00016065, -9.62404863\ 10^{-6}, 3.95134203,$$
$$\ldots],$$
$$[\vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \ddots]]],$$

$$\begin{bmatrix} \text{slice of } 16 \times 16 \times 16 \times 16 \text{ Array} \end{bmatrix}, \mathit{active\_orbitals} = [53\,..58], \mathit{aolabels} = \underset{\text{81 element Vector[column]}}{\begin{bmatrix} \text{"0 S 1s"} \\ \text{"0 S 2s"} \\ \text{"0 S 3s"} \\ \text{"0 S 2px"} \\ \text{"0 S 2py"} \\ \text{"0 S 2pz"} \\ \vdots \end{bmatrix}},$$

$$\mathit{mo\_coeff\_canonical} = [[[0.99415756, -1.72490244\ 10^{-8}, 1.24166682\ 10^{-6},$$
$$2.48859275\ 10^{-8}, -8.07782124\ 10^{-8}, 1.28471104\ 10^{-6}, \ldots],$$
$$[0.01712856, 6.34133083\ 10^{-8}, -4.91194013\ 10^{-6}, -8.63520473\ 10^{-8}, 3.07623526\ 10^{-7},$$
$$-4.71769622\ 10^{-6}, \ldots],$$
$$[-0.00254766, -2.22737125\ 10^{-7}, 0.00001191, 3.81324524\ 10^{-7}, -8.80379494\ 10^{-7},$$
$$0.00002131, \ldots],$$
$$[-0.00019443, 1.05474583\ 10^{-7}, -4.24974316\ 10^{-6}, -1.29319579\ 10^{-7}, 1.08607113\ 10^{-6},$$
$$-4.99715484\ 10^{-6}, \ldots],$$
$$[-8.09146400\ 10^{-6}, -5.32699855\ 10^{-8}, 5.78429997\ 10^{-6}, 1.14184281\ 10^{-7},$$

$-1.06986327\ 10^{-7}$, $5.37207860\ 10^{-6}$, …],

[0.00007132, $2.73299767\ 10^{-8}$, $-5.35704684\ 10^{-6}$, 0., $-1.27056647\ 10^{-8}$,
$-2.09471754\ 10^{-6}$, …],

[⋮, ⋮, ⋮, ⋮, ⋮, ⋮,]]],

$$\left[\begin{array}{c} \\ \\ \\ \text{81 × 81 Matrix} \\ \\ \\ \\ \end{array}\right], mo\_occ = \underset{\text{81 element Vector[column]}}{\left[\begin{array}{c} 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ \vdots \end{array}\right]}, mo\_coeff =$$

[[[0.00003123, −0.00005739, −0.00664267, 0.00173122, −0.36729192,
−0.00008007, …],

[−0.00010573, 0.00016643, 0.01935902, −0.00495663, 1.04674729, 0.00016717, …],

[0.00030301, −0.00002700, −0.00858121, 0.00326181, 0.05391625, −0.00390713, …],

[−0.00006997, 0.32316494, 0.86077536, −0.34892667, −0.01409257, −0.00054301, …],

[0.00001196, 0.24976950, 0.27393065, 0.90976738, 0.00003609, −0.00077274, …],

[−0.00001461, 0.89746001, −0.38621579, −0.12755137, 0.00490922, 0.00040701, …],

[⋮, ⋮, ⋮, ⋮, ⋮, ⋮,]]],

$$\left[\begin{array}{c} \\ \\ \\ \text{81 × 81 Matrix} \\ \\ \\ \\ \end{array}\right], charges = \underset{\text{25 element Vector[column]}}{\left[\begin{array}{c} 0.10504005 \\ -0.30624021 \\ -0.20852064 \\ -0.28630119 \\ -0.26480265 \\ -0.37476210 \\ \vdots \end{array}\right]}, rdm1 = [[[1.99736371,$$

0.00982622, −0.00145516, 0.00082201, 0.00006191, 0.00009142, …],

```
[0.00982622, 1.94302579, 0.01041239, −0.00099437, −0.00024361, −0.00005003, … ],
[−0.00145516, 0.01041239, 1.99627560, 0.00084865, 0.00007966, −0.00001798, … ],
[0.00082201, −0.00099437, 0.00084865, 1.99889520, 0.00004593, −0.00008309, … ],
[0.00006191, −0.00024361, 0.00007966, 0.00004593, 1.99778101, −4.92868782 10^-6, … ],
[0.00009142, −0.00005003, −0.00001798, −0.00008309, −4.92868782 10^-6, 1.97826992,
… ],
[⋮, ⋮, ⋮, ⋮, ⋮, ⋮,  ]]],

                                    [ 1.99990877 ]
                                    [ 1.99656772 ]
                                    [ 1.85424360 ]
[ 16 × 16 Matrix ], populations =   [ 1.98480911 ]  , e_tot = −1027.51381995, e_corr
                                    [ 1.97903916 ]
                                    [ 1.99673222 ]
                                    [     ⋮      ]
                               81 element Vector[column]

= −0.12499606 ]])
```

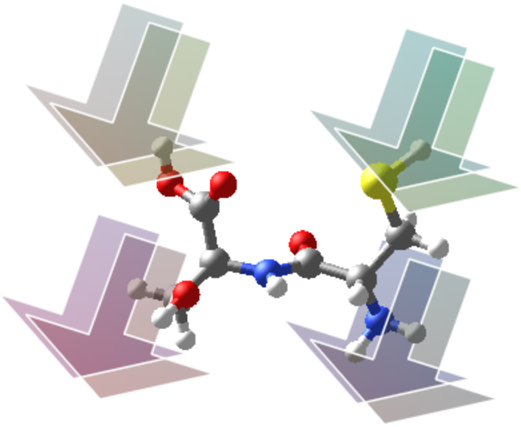
In the table of computed properties we observe that the peptide has a nonzero dipole moment

```
> data_peptide[dipole];
```

```
 [  0.95773786 ]
 [ −2.37658291 ]
 [ −1.59751383 ]
```

The direction of the dipole moment can be readily visualized with the *DipolePlot* command

```
> DipolePlot(mol, method = ContractedSchrodinger, active = [6, 6], active_cse = [16, 16]);
```

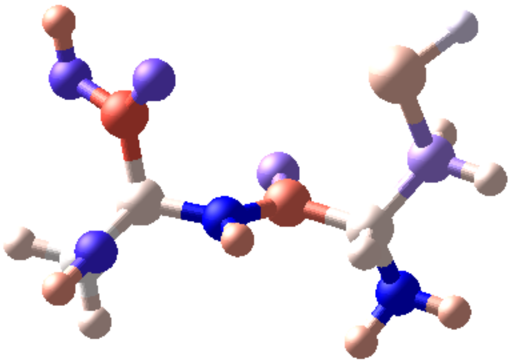


We can also compute the atomic charges which are closely associated with the dipole moment

```
> seq([mol[i][1], data_peptide[charges][i]], i = 1 ..nops(mol));
["S", 0.10504005], ["O", −0.30624021], ["O", −0.20852064], ["O", −0.28630119], ["O",
    −0.26480265], ["N", −0.37476210], ["N", −0.39527767], ["C", 0.04398837], ["C",
    0.03851104], ["C", 0.26632682], ["C", 0.00879806], ["C", −0.17897052], ["C", 0.29797645],
    ["H", 0.09534526], ["H", 0.04389074], ["H", 0.20310443], ["H", 0.05844607], ["H",
    0.06152166], ["H", 0.07400052], ["H", 0.05447756], ["H", 0.15148599], ["H", 0.15137748],
    ["H", 0.18850391], ["H", −0.04234736], ["H", 0.21442792]
```

Only one of the carbons is negatively charged and only two of the carbons are fairly positively charged which we can also see from a *ChargesPlot*

```
> ChargesPlot(mol, method = ContractedSchrodinger, active = [6, 6], active_cse = [16, 16]);
```



where the positively charged atoms are shown in red and the negatively charged atoms are shown in blue.

Error Mitigation in Quantum Computing

All of the one- and two-electron properties of molecules are computable from the two-electron reduced density matrix (2-RDM). The 2-RDM of a molecule obeys a set of stringent constraints that are necessary for it to represent an *N*-electron wave function−constraints known as *N*-representability conditions. A 2-RDM that is computed by an approximate method or measured in the presence of noise as on a quantum computer can violate some of these conditions. In QCT 2021 we introduce a new algorithm through the command *Purify2RDM* that purifies the 2-RDM to obey an important set of *N*-representability conditions. *Purify2RDM* uses an advanced type of optimization, known as semidefinite programming, to project the 2-RDM onto the set of approximately *N*-representable 2-RDMs. The command is especially useful in molecular simulations on quantum computers where noise generates errors in the measured 2-RDMs. The correction of errors in quantities measured on a quantum computer is known as error mitigation. Here we demonstrate the command by using it to purify an approximate 2-RDM of hydrogen fluoride generated from second-order many-body perturbation (MP2) theory. First, we generate the data from applying the *MP2* command to the geometry of hydrogen fluoride where we include the keyword parameter *return_rdm*= "*rdm1_and_rdm2*" to return the 2-RDM.

```
> hf := MolecularGeometry("hydrogen fluoride");
            hf := [["F", 0, 0, 0], ["H", 0.43580000, -0.14770000, -0.81880000]]
```

```
> data_hf := MP2(hf, return_rdm = "rdm1_and_rdm2") :
```

Second, we reshape the data stored in a 4-index tensor into a 2-index matrix

```
> D2 := Matrix(36, 36, shape = symmetric) :
  D2( .. ) := ArrayTools:-Permute(data_hf[rdm2], [1, 3, 2, 4]);
D2 :=
      [ 1.99999763   -0.00001813  -0.00001559  0.          0.          0.          ... ]
      [-0.00001813    3.99657180   0.00684113  0.          0.          0.          ... ]
      [-0.00001559    0.00684113   3.97665772  0.          0.          0.          ... ]
      [ 0.            0.           0.          3.99912733  0.          0.          ... ]
      [ 0.            0.           0.          0.          3.99912733  0.          ... ]
      [ 0.            0.           0.          0.          0.          0.02850872  ... ]
      [ ⋮             ⋮            ⋮           ⋮           ⋮           ⋮               ]
                                                                      36 × 36 Matrix
```

and compute the eigenvalues of the 2-RDM with the *Eigenvalues* command in the *LinearAlgebra* package

```
> eigenvals := LinearAlgebra:-Eigenvalues(D2);
```

$$eigenvals := \begin{bmatrix} -0.01431570 \\ 0.01425436 \\ 0.01425436 \\ 0.01425436 \\ 0.01425436 \\ 0.01425436 \\ \vdots \end{bmatrix}$$

36 element Vector[column]

Notice that the lowest eigenvalue of the approximate 2-RDM is negative (-0.0143), but because each eigenvalue represents the probability of being in a two-electron quantum state, all of the eigenvalues of the 2-RDM should be non-negative! Finally, we use the *Purify2RDM* command to correct this violation as well as more subtle violations of the physical requirements for the 2-RDM

```
> data_hf_purified := Purify2RDM(data_hf[rdm2]);
```

$$data\_hf\_purified := \text{table}\Bigg(\Bigg[rdm2$$

$$= \begin{bmatrix} 1.99841341 & -0.00001418 & -0.00001325 & 0. & 0. & 0. \\ -0.00001418 & 3.99395109 & 0.00625701 & 0. & 0. & 0. \\ -0.00001325 & 0.00625701 & 3.97572714 & 0. & 0. & 0. \\ 0. & 0. & 0. & 3.99625863 & 0. & 0. \\ 0. & 0. & 0. & 0. & 3.99626171 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0.03251581 \end{bmatrix},$$

slice of 6 × 6 × 6 × 6 Array

$rdm1$

$$= \left[\begin{matrix} 1.99923645 & -0.00001520 & -0.00001377 & 0. & 0. & 0. \\ -0.00001520 & 1.99771644 & 0.00312075 & 0. & 0. & 0. \\ -0.00001377 & 0.00312075 & 1.98863166 & 0. & 0. & 0. \\ 0. & 0. & 0. & 1.99886787 & 0. & 0. \\ 0. & 0. & 0. & 0. & 1.99886724 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0.01668030 \end{matrix}\right]]])$$

To check the correction of the 2-RDM, as before we rearrange the outputted 4-index tensor into a 2-index matrix

```
> D2p := Matrix(36, 36, shape=symmetric) :
  D2p( .. ) := ArrayTools:-Permute(data_hf_purified[rdm2], [1, 3, 2, 4]);
```

$$D2p := \left[\begin{matrix} 1.99841341 & -0.00001418 & -0.00001325 & 0. & 0. & 0. & \ldots \\ -0.00001418 & 3.99395109 & 0.00625701 & 0. & 0. & 0. & \ldots \\ -0.00001325 & 0.00625701 & 3.97572714 & 0. & 0. & 0. & \ldots \\ 0. & 0. & 0. & 3.99625863 & 0. & 0. & \ldots \\ 0. & 0. & 0. & 0. & 3.99626171 & 0. & \ldots \\ 0. & 0. & 0. & 0. & 0. & 0.03251581 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{matrix}\right]$$

36 × 36 Matrix

which we diagonalize with the *LinearAlgebra:-Eigenvalues* command

```
> eigenvals := LinearAlgebra:-Eigenvalues(D2p);
```

$$eigenvals := \left[\begin{matrix} -6.77922778\ 10^{-13} \\ 0.00474144 \\ 0.01414059 \\ 0.01691224 \\ 0.01708308 \\ 0.01708939 \\ \vdots \end{matrix}\right]$$

36 element Vector[column]

All of the eigenvalues, we find, are now non-negative.

3-Dimensional Density Plots

The command *DensityPlot3D* receives new optional keywords bonds and colors that control the appearance of chemical bonds and the color of the orbital densities. Consider the diatomic molecule hydrogen fluoride

```
> hf := MolecularGeometry("hydrogen fluoride");
```

$$hf := [["F", 0, 0, 0], ["H", 0.43580000, -0.14770000, -0.81880000]]$$

After performing an electronic structure calculation such as its solution by the new *ContractedSchrodinger* command

```
> data_hf := ContractedSchrodinger(hf, active = {3, 6});
```

$$data\_hf := \mathrm{table}\left(\left[\left[dipole = \begin{bmatrix} 0.53092969 \\ -0.17994106 \\ -0.99753380 \end{bmatrix}, rdm2 \right.\right.\right.$$

$$= \underset{\text{slice of } 6 \times 6 \times 6 \times 6 \text{ Array}}{\begin{bmatrix} 1.99999855 & -0.00001411 & 0. & 0. & -2.37126205\ 10^{-6} & -0.00026105 \\ -0.00001411 & 3.99898150 & 0. & 0. & -0.00020921 & 0.00267332 \\ 0. & 0. & 3.99891455 & 0. & 0. & 0. \\ 0. & 0. & 0. & 3.99891455 & 0. & 0. \\ -2.37126205\ 10^{-6} & -0.00020921 & 0. & 0. & 3.93554043 & 0.00235321 \\ -0.00026105 & 0.00267332 & 0. & 0. & 0.00235321 & 0.06763978 \end{bmatrix}}, active\_orbitals = [5 ..6], aolabels$$

$$= \begin{bmatrix} \text{"0 F 1s"} \\ \text{"0 F 2s"} \\ \text{"0 F 2px"} \\ \text{"0 F 2py"} \\ \text{"0 F 2pz"} \\ \text{"1 H 1s"} \end{bmatrix}, mo\_coeff\_canonical = \begin{bmatrix} 0.99473292 & -0.26291579 & 0. & 0. & -0.00515494 & 0.08099256 \\ 0.02241815 & 1.02658132 & 0. & 0. & 0.13417827 & -0.52752059 \\ 0.00118886 & -0.05522715 & 0.81831225 & -0.33922852 & -0.31885323 & -0.38324331 \\ -0.00040292 & 0.01871742 & 0.45295313 & 0.87755590 & 0.10806476 & 0.12988765 \\ -0.00223367 & 0.10376318 & 0.35383403 & -0.33885050 & 0.59907533 & 0.72005421 \\ -0.00566635 & -0.01122934 & 0. & 0. & -0.55903946 & 1.06456063 \end{bmatrix}, mo\_occ$$

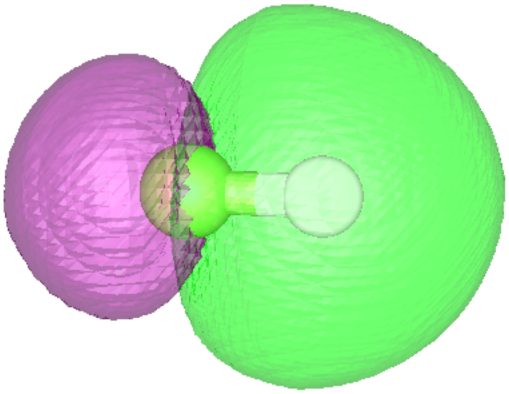
$$= \begin{bmatrix} 1.99999929 \\ 1.99949606 \\ 1.99946197 \\ 1.99946197 \\ 1.96776686 \\ 0.03381385 \end{bmatrix}, mo\_coeff = \begin{bmatrix} 1.00200779 & 0.23353502 & 0. & 0. & 0.00593699 & 0.08130923 \\ -0.00760983 & -1.02609756 & 0. & 0. & -0.13640394 & -0.52831187 \\ 0.00278727 & 0.05464163 & -0.38106604 & 0.79968719 & 0.31922319 & -0.38301091 \\ -0.00094465 & -0.01851897 & 0.85297431 & 0.49770043 & -0.10819014 & 0.12980888 \\ -0.00523684 & -0.10266308 & -0.35668403 & 0.33584920 & -0.59977041 & 0.71961756 \\ -0.00563677 & 0.00927133 & 0. & 0. & 0.55841585 & 1.06490689 \end{bmatrix}, charges$$

$$= \begin{bmatrix} -0.18049460 \\ 0.18049461 \end{bmatrix}, rdm1 = \begin{bmatrix} 1.99999882 & -0.00001457 & 0. & 0. & 5.95981985\ 10^{-6} & -0.00025893 \\ -0.00001457 & 1.99949535 & 0. & 0. & -0.00007970 & 0.00136508 \\ 0. & 0. & 1.99946197 & 0. & 0. & 0. \\ 0. & 0. & 0. & 1.99946197 & 0. & 0. \\ 5.95981985\ 10^{-6} & -0.00007970 & 0. & 0. & 1.96776634 & 0.00117498 \\ -0.00025893 & 0.00136508 & 0. & 0. & 0.00117498 & 0.03381554 \end{bmatrix}, populations$$

$$\left.\left.\left. = \begin{bmatrix} 1.99911327 \\ 1.95146256 \\ 1.83401939 \\ 1.98045846 \\ 1.41544091 \\ 0.81950539 \end{bmatrix}, e\_tot = -98.60014967, e\_corr = -0.02765420 \right]\right]\right)$$

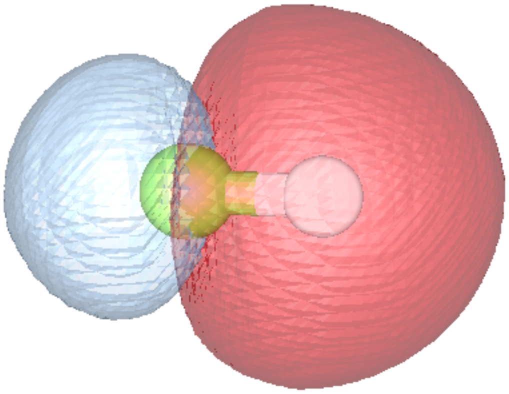
we can visualize the orbitals with *DensityPlot3D*. In QCT 2020 the density plot always uses green and purple colors for the orbitals

```
> DensityPlot3D(hf, data_hf, orbitalindex = 5, densitycutoff = 0.0005, orientation = [-97, -4, -75]);
```



Now we can use the keyword color to change the default colors to any color supported in Maple's *ColorTools* package. For example,

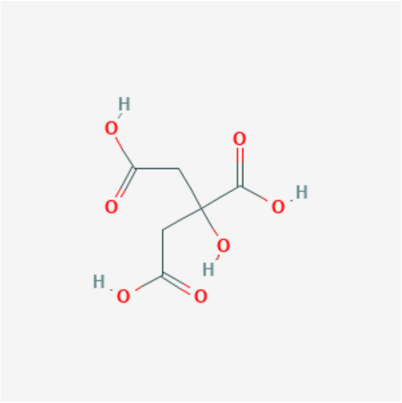
```
> DensityPlot3D(hf, data_hf, orbitalindex = 5, densitycutoff = 0.0005, colors = ["Nautical Red",
      "Nautical Light Blue"]);
```



Molecular Integrals

The QCT makes advanced features like obtaining the molecular integrals as simple as a single command MO*Integrals*. In QCT 2021 we add two new optional keywords *initial_mo* and *active* to the *MOIntegrals* command that makes generating molecular integrals in any MO basis set with or without an active space as easy as possible. Let's consider the molecule citric acid whose skeletal structure, geometry, and dictionary entry are

```
> SkeletalStructure("citric acid");
```



```
> citricacid := MolecularGeometry("citric acid");
```

```
citricacid := [["O", 0.02960000, 0.20950000, 1.60690000], ["O", −0.85580000, 1.79790000,
    −1.49620000], ["O", −1.35540000, 2.26890000, 0.66220000], ["O", −2.39080000,
    −2.30460000, −0.78710000], ["O", 3.53510000, −0.40330000, −0.92730000], ["O",
    −2.55590000, −0.62650000, 0.73170000], ["O", 2.54380000, −1.12450000, 0.98350000], ["C",
    0.10070000, 0.36200000, 0.19410000], ["C", −0.49050000, −0.87910000, −0.49710000], ["C",
    1.53740000, 0.70960000, −0.22660000], ["C", −0.76500000, 1.57730000, −0.15870000], ["C",
    −1.90980000, −1.23050000, −0.11540000], ["C", 2.57670000, −0.35680000, 0.03000000],
    ["H", 0.11200000, −1.75550000, −0.22860000], ["H", −0.45740000, −0.76140000,
    −1.58660000], ["H", 1.88420000, 1.60800000, 0.30000000], ["H", 1.55890000, 0.93610000,
    −1.30040000], ["H", 0.46200000, 0.97710000, 2.01860000], ["H", −1.42930000, 2.56690000,
    −1.70150000], ["H", −3.30780000, −2.52520000, −0.51720000], ["H", 4.19140000,
    −1.11160000, −0.75460000]]
```

and

```
> MolecularDictionary("citric acid");
```

```
Citric acid: Citric acid is a tricarboxylic acid that is propane-1,2,3
-tricarboxylic acid bearing a hydroxy substituent at position 2. It is
an important metabolite in the pathway of all aerobic organisms. It
has a role as a food acidity regulator, a chelator, an antimicrobial
agent and a fundamental metabolite. It is a conjugate acid of a
citrate(1-) and a citrate anion. -ChEBI

Citric acid: Anhydrous Citric Acid is a tricarboxylic acid found in
citrus fruits. Citric acid is used as an excipient in pharmaceutical
preparations due to its antioxidant properties. It maintains stability
of active ingredients and is used as a preservative. It is also used
as an acidulant to control pH and acts as an anticoagulant by
chelating calcium in blood. -NCI Thesaurus (NCIt)
```

We can generate the molecular orbitals (MOs) in a 26 electrons in 26 orbitals ([26,26]) active space−the 13 highest occupied orbitals (26 electrons) and the 13 lowest unoccupied orbitals−with the *active* keyword

```
> data := MOIntegrals(citricacid, active = [26, 26]);
```

```
data :=table([core_energy = −1529.62094994, electron_repulsion_integrals =

    [[[0.32124226, −0.04379042, 0.20081119, 0.01659544, 0.00657583, 0.18256574, …
],
[ −0.04379042, 0.02453868, 0.00758620, −0.00654389, −0.00485144, 0.01476096, … ],
[ 0.20081119, 0.00758620, 0.25495886, −0.00356739, 0.01135458, 0.22723658, … ],
[ 0.01659544, −0.00654389, −0.00356739, 0.00863329, 0.00834643, −0.01179976, … ],
[ 0.00657583, −0.00485144, 0.01135458, 0.00834643, 0.02351705, −0.01102714, … ],
[ 0.18256574, 0.01476096, 0.22723658, −0.01179976, −0.01102714, 0.24691087, … ],
[ ⋮, ⋮, ⋮, ⋮, ⋮, ⋮, ]]],
[ 351 × 351 Matrix ], one_electron_integrals =
[[[ −5.46419447, 0.04500267, 0.04516621, −0.03997982, 0.03204238, 0.03425667,
… ],
[ 0.04500267, −5.59252962, 0.00822404, −0.07764675, −0.02976393, 0.02205225, … ],
[ 0.04516621, 0.00822404, −5.57828947, 0.21529969, −0.09906170, 0.11139708, … ],
[ −0.03997982, −0.07764675, 0.21529969, −5.33772277, −0.01304712, −0.17961000, …
],
[ 0.03204238, −0.02976393, −0.09906170, −0.01304712, −5.63110516, 0.02139972, … ],
[ 0.03425667, 0.02205225, 0.11139708, −0.17961000, 0.02139972, −6.04287921, … ],
[ ⋮, ⋮, ⋮, ⋮, ⋮, ⋮, ]]],
[ 26 × 26 Matrix ]]])
```

These molecular integrals are in the basis set of the Hartree-Fock MOs. We can easily use any set of MOs with the keyword *initial_mo*. For example, we can run a density functional theory (DFT) calculation and use the DFT MOs

```
> data_dft := DensityFunctional(citricacid);
```

$$data\_dft := \mathrm{table}\Bigg(\Bigg[\, dipole = \begin{bmatrix} 1.22937495 \\ 0.34548694 \\ -2.88440739 \end{bmatrix}, mo\_energy = \underset{\text{73 element Vector[column]}}{\begin{bmatrix} -18.93433581 \\ -18.92919366 \\ -18.92723760 \\ -18.86197401 \\ -18.85931351 \\ -18.85557936 \\ \vdots \end{bmatrix}}, group = \text{"C1"},$$

$$aolabels = \underset{\text{73 element Vector[column]}}{\begin{bmatrix} \text{"0 O 1s"} \\ \text{"0 O 2s"} \\ \text{"0 O 2px"} \\ \text{"0 O 2py"} \\ \text{"0 O 2pz"} \\ \text{"1 O 1s"} \\ \vdots \end{bmatrix}}, mo\_occ = \underset{\text{73 element Vector[column]}}{\begin{bmatrix} 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ 2.00000000 \\ \vdots \end{bmatrix}}, mo\_coeff = [\,[$$

$[\,-6.38999609\ 10^{-6}, -0.00012416, -9.76123403\ 10^{-6}, 0.99343933, -0.00231932,$
$-0.00098545, \ldots\,],$
$[\,8.60202467\ 10^{-6}, -0.00001166, 5.97688063\ 10^{-6}, 0.02937309, -0.00006269, 0.00001183,$
$\ldots\,],$
$[\,-0.00001345, 0.00006578, 0.00001446, 0.00192123, -7.75098185\ 10^{-6},$
$-1.15742764\ 10^{-6}, \ldots\,],$
$[\,9.71038929\ 10^{-6}, -0.00006084, 0.00001439, 0.00346363, 0.00007291, 0.00002746, \ldots\,],$
$[\,-4.37784910\ 10^{-6}, 0.00001683, -6.59953268\ 10^{-6}, -0.00199713, 0.00009735,$
$-3.05088877\ 10^{-6}, \ldots\,],$
$[\,0.00014013, 0.99337390, -0.00039270, 0.00013271, -0.00114194, -5.96394950\ 10^{-6}, \ldots$
$],$
$[\,\vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \,]\,]\,],$

$$\left[\begin{array}{c} 73 \times 73 \text{ Matrix} \end{array}\right], \mathit{charges} = \underset{\text{21 element Vector[column]}}{\left[\begin{array}{c} -0.26832299 \\ -0.25961742 \\ -0.22920018 \\ -0.25606741 \\ -0.25348756 \\ -0.22983320 \\ \vdots \end{array}\right]}, \mathit{rdm1} =$$

$$[[[2.10743080, -0.45465645, 0.04394344, 0.07974249, -0.04920243, -0.00008310, \ldots],$$
$$[-0.45465645, 2.01314474, -0.24569555, -0.44631459, 0.25511638, 0.00157429, \ldots],$$
$$[0.04394344, -0.24569555, 1.74743818, -0.34214859, -0.10861086, 0.01338712, \ldots],$$
$$[0.07974249, -0.44631459, -0.34214859, 1.31742236, -0.17855416, -0.01442339, \ldots],$$
$$[-0.04920243, 0.25511638, -0.10861086, -0.17855416, 0.84012318, 0.01264903, \ldots],$$
$$[-0.00008310, 0.00157429, 0.01338712, -0.01442339, 0.01264903, 2.10633832, \ldots],$$
$$[\vdots, \vdots, \vdots, \vdots, \vdots, \vdots,]]],$$

$$\left[\begin{array}{c} 73 \times 73 \text{ Matrix} \end{array}\right], \mathit{populations} = \underset{\text{73 element Vector[column]}}{\left[\begin{array}{c} 1.99739216 \\ 1.84407136 \\ 1.80662593 \\ 1.50285471 \\ 1.11737882 \\ 1.99720322 \\ \vdots \end{array}\right]}, \mathit{e\_tot} = -749.37832783, \mathit{mo\_symmetry}$$

$$= \underset{\text{73 element Vector[column]}}{\left[\begin{array}{c} \text{"A"} \\ \text{"A"} \\ \text{"A"} \\ \text{"A"} \\ \text{"A"} \\ \text{"A"} \\ \vdots \end{array}\right]}, \mathit{converged} = 1 \Bigg]\Bigg)$$

The *initial_mo* keyword is assigned to a list of the MO coefficients and MO symmetries

```
> data := MOIntegrals(citricacid, active = [26, 26], initial_mo = [data_dft[mo_coeff],
      data_dft[mo_symmetry]]);
```

data ≔table([*core_energy* = −1529.62767538, *electron_repulsion_integrals* =

[[[0.27429700, 0.04786344, 0.22536388, 0.00250729, 0.00619826, 0.17533006, ...],
[0.04786344, 0.05771343, −0.00070182, 0.00380999, 0.00465537, −0.03154236, ...],
[0.22536388, −0.00070182, 0.23047547, 0.00062981, 0.00258952, 0.20176427, ...],
[0.00250729, 0.00380999, 0.00062981, 0.00331238, −0.00094396, −0.00433470, ...],
[0.00619826, 0.00465537, 0.00258952, −0.00094396, 0.00655314, −0.01143280, ...],
[0.17533006, −0.03154236, 0.20176427, −0.00433470, −0.01143280, 0.28105130, ...],
[⋮, ⋮, ⋮, ⋮, ⋮, ⋮,]]],

[351 × 351 Matrix], *one_electron_integrals* =

[[[−5.53503379, −0.01859793, −0.01110019, 0.03261685, −0.05019326,
−0.02613277, ...],
[−0.01859793, −5.43159388, 0.01401383, −0.04763806, 0.03925754, 0.07596055, ...],
[−0.01110019, 0.01401383, −5.50707977, 0.21427881, 0.03373941, 0.03250823, ...],
[0.03261685, −0.04763806, 0.21427881, −5.37204653, 0.03538810, −0.09885000, ...],
[−0.05019326, 0.03925754, 0.03373941, 0.03538810, −5.65020028, −0.00027176, ...],
[−0.02613277, 0.07596055, 0.03250823, −0.09885000, −0.00027176, −6.10164655, ...],
[⋮, ⋮, ⋮, ⋮, ⋮, ⋮,]]],

$$\left[26 \times 26 \text{ Matrix} \right]\Big]\Big]\Big)$$

Voila! We have the MO integrals with respect to the DFT MOs in a [26,26] active space.

Using the Package in the Classroom

The Maple Quantum Chemistry Toolbox includes approximately 30 lessons that can be used in chemistry and physics courses from advanced high school courses through the graduate level. These lessons and associated curricula provide instructors and students with real-time quantum chemistry computations and visualizations that quickly deepen understanding of molecular concepts. Detailed lesson plans and curricula are provided for Introductory (General) Chemistry, Physical Chemistry (Quantum Mechanics and Thermodynamics), Thermodynamics (Physics), Quantum Mechanics (Physics), Computational Chemistry, and Quantum Chemistry as well as Advanced Placement (AP) and International Baccalaureate (IB) chemistry courses. Topics include atomic structure, chemical bonding, the Maxwell-Boltzmann distribution, heat capacity, enthalpy, entropy, free energy, particle-in-a-box, vibrational normal modes, infrared spectroscopy, as well as advanced electronic structure methods. Use of the QCT in the classroom is described in a recent paper in J. Chem. Ed. QCT 2021 includes a new lesson for Physical Chemistry and Undergraduate Quantum Mechanics on Vibrational Motion and the Harmonic Oscillator.   In the following two subsections we show snippets of the lesson

Reduced mass

Here we set the variables A and B to the strings of the atoms in the diatomic molecule (By changing these values, you can use the worksheet to treat other diatomic molecules!)

```
> A := "H";
  B := "F";
```

```
                                   A := "H"
                                   B := "F"
```

We use the *AtomicData* command in the Quantum Chemistry package to obtain the masses as well as other data

```
> dataA := AtomicData(A);
```

```
dataA :=table([ionizationenergy = 13.59840000 eV, name = hydrogen, meltingpoint = 13.81000000 K,
    electronegativity = 2.10000000, atomicweight = 1.00794000 amu, atomicnumber = 1, boilingpoint
     = 20.28000000 K, names = {hydrogen}, symbol = H])
```

```
> dataB := AtomicData(B);
dataB := table([ionizationenergy = 17.42280000 eV, name = fluorine, electronaffinity
     = 3.40119000 eV, meltingpoint = 53.53000000 K, electronegativity = 3.98000000, atomicweight
     = 18.99840320 amu, atomicnumber = 9, boilingpoint = 85.03000000 K, names = {fluorine}, symbol
     = F])
```

Set the masses

```
> mA := dataA[atomicweight];
                              mA := 1.00794000 amu
> mB := dataB[atomicweight];
                              mB := 18.99840320 amu
```

Compute the reduced mass in amu

$$> mu0 := \frac{mA \cdot mB}{mA + mB};$$

$$\mu0 := 0.95715895 \text{ amu}$$

Convert the reduced mass from amu to kg

$$> \mathrm{mu} := convert(mu0, units, 'kg');$$

$$\mu := 1.58940092\ 10^{-27} \text{ kg}$$

Bond length

To compute the equilibrium bond length, we select a set of bond distances from the roots of the sixth-order Chebyshev polynomial that are suitable for interpolation

```
> bonds := map(x → x/5 + 1.05, [fsolve(expand(ChebyshevT(6, x)))]);
      bonds := [0.85681483, 0.90857864, 0.99823619, 1.10176381, 1.19142136, 1.24318516]
```

We define a list of molecular geometries with each geometry corresponding to one of the bond distances

```
> molecules := map(R→ [[A, 0, 0, 0], [B, 0, 0, R]], bonds);
molecules := [[["H", 0, 0, 0], ["F", 0, 0, 0.85681483]], [["H", 0, 0, 0], ["F", 0, 0, 0.90857864]],
     [["H", 0, 0, 0], ["F", 0, 0, 0.99823619]], [["H", 0, 0, 0], ["F", 0, 0, 1.10176381]], [["H", 0, 0, 0],
     ["F", 0, 0, 1.19142136]], [["H", 0, 0, 0], ["F", 0, 0, 1.24318516]]]
```

The energies for each geometry may be then readily computed with the *Energy* command in the Quantum Chemistry package.

```
> energies := map(Energy, molecules, basis = "cc-pVDZ");
```

$$energies := [-100.01686137, -100.01964383, -100.01018182, -99.98693789, -99.96201956, -99.94683591]$$

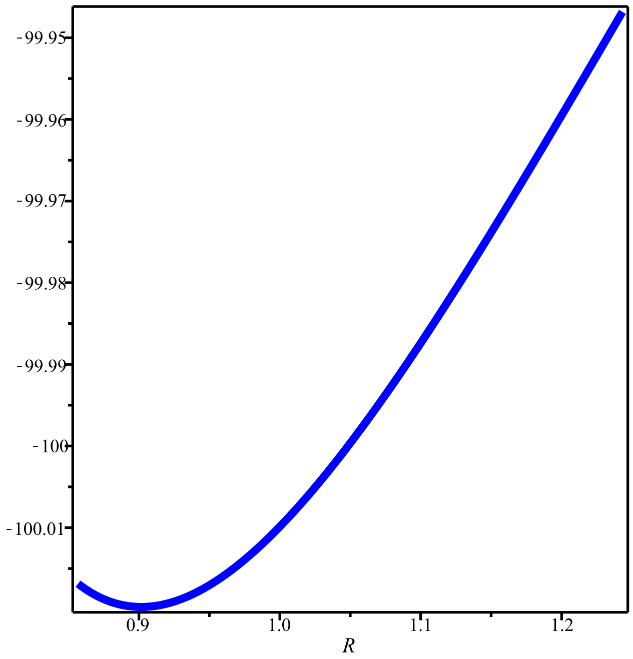
We use polynomial interpolation to generate a polynomial in the bond distance *R*

```
> pes := interp(bonds, energies, R);
```

$$pes := -2.84449888\, R^5 + 17.02510690\, R^4 - 41.28045230\, R^3 + 50.73972465\, R^2 - 31.33797004\, R - 92.31177999$$

The potential energy surface (curve) can be plotted

```
> plot(pes(R), R = bonds[1]..bonds[-1], axes = boxed, labels = ['R','E'], color = blue, thickness = 3);
```



Finally, we differentiate the potential energy curve with respect to *R* and set the derivative to zero.

```
> eq := diff(pes, R) = 0;
```

$$eq := -14.22249441\, R^4 + 68.10042762\, R^3 - 123.84135691\, R^2 + 101.47944931\, R - 31.33797004 = 0$$

Solving the resulting equation yields the equilibrium bond length

```
> R_eq := fsolve(eq, R = bonds[1]..bonds[-1]);
```

$$R\_eq := 0.90161288$$