

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 2024, this toolbox has significant new features and enhancements that enable: (1) Computing electronic energies and properties with a new generalization of density functional theory that captures static correlation, (2) Running the CoupledCluster and NuclearGradient commands on Windows that were previously available only on macOS and Linux, (3) Harnessing AI through a new command, Chat, that when prompted with a word or phrase, can provide invaluable information about molecules and compounds, electronic structure methods, as well as other scientific words or concepts, (4) Learning or teaching about the role of chemistry in art through a new curricular set of lessons entitled "The Chemistry of Art," and (5) Experiencing additional enhancements and improvements throughout the package.

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

Generalization of Density Functional Theory

A well known limitation of density functional theory (DFT) is its difficulty in predicting the energies and properties of molecules with static correlation. Static correlation is important to the accurate prediction of charges, van der Waals forces, barrier heights, and bi- and multiradicals. Introduced in QCT 2024 is a generalization of DFT that can treat static correlation. The new command, *RDMFunctional* adds a universal correction, based on the 1-electron reduced density matrix (1-RDM) rather than the density alone, to any DFT functional.

While the generalized DFT is applicable to a wide range of molecules, below you can see the command with the dissociation of the diatomic molecule N_2 . First, load the QuantumChemistry package

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

```
[AOLabels, ActiveSpaceCI, ActiveSpaceSCF, AtomicData, BondAngles, BondDistances, Charges, ChargesPlot, Chat, ContractedSchrodinger, CorrelationEnergy, CoupledCluster, DensityFunctional, DensityPlot3D, Dipole, DipolePlot, Energy, ExcitationEnergies, ExcitationSpectra, ExcitationSpectraPlot, ExcitedStateEnergies, ExcitedStateSpins, ExcitonDensityPlot, ExcitonPopulations, ExcitonPopulationsPlot, FullCI, GeometryOptimization, HartreeFock, Interactive, Isotopes, LiteratureSearch, MOCoefficients, MODiagram, MOEnergies, MOIntegrals, MOOccupations, MOOccupationsPlot, MOSymmetries, MP2, MolecularData, MolecularDictionary, MolecularGeometry, NuclearEnergy, NuclearGradient, OscillatorStrengths, Parametric2RDM, PlotMolecule, Populations, Purify2RDM, QuantumComputing, RDMI, RDM2, RDMFunctional, RTM1, ReadXYZ, Restore, Save, SaveXYZ, SearchBasisSets, SearchFunctionals, SkeletalStructure, SolventDatabase, Thermodynamics, TransitionDipolePlot, TransitionDipoles, TransitionOrbitalPlot,
```

TransitionOrbitals, Variational2RDM, VibrationalModeAnimation, VibrationalModes, Video]

Select a set of bond distances from the roots of the sixth-order Chebyshev polynomial

```
> bond_distances := map(x - x + 2.0, [fsolve(expand(ChebyshevT(8, x)))]);  
bond_distances := [1.01921472, 1.16853039, 1.44442977, 1.80490968, 2.19509032, 2.55557023,  
2.83146961, 2.98078528]
```

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

```
> molecules := [seq(["N", 0, 0, 0], ["N", 0, 0, R], R in bond_distances)];  
molecules := [[["N", 0, 0, 0], ["N", 0, 0, 1.01921472]], [[["N", 0, 0, 0], ["N", 0, 0, 1.16853039]], [[["N", 0,  
0, 0], ["N", 0, 0, 1.44442977]], [[["N", 0, 0, 0], ["N", 0, 0, 1.80490968]], [[["N", 0, 0, 0], ["N", 0, 0,  
2.19509032]], [[["N", 0, 0, 0], ["N", 0, 0, 2.55557023]], [[["N", 0, 0, 0], ["N", 0, 0, 2.83146961]],  
[[["N", 0, 0, 0], ["N", 0, 0, 2.98078528]]]
```

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

```
> energies_dft := [seq(Energy(molecule, method = DensityFunctional, basis = "cc-pVDZ", xc  
= "B3LYP"), molecule in molecules)];  
energies_dft := [-109.50786339, -109.52331040, -109.37588586, -109.17741963, -109.03358249,  
-108.95146495, -108.90997138, -108.89291096]
```

and using polynomial interpolation, generate a polynomial in the bond distance R

```
> pes_dft := interp(bond_distances, energies_dft, R);  
pes_dft := -0.14339035 R7 + 2.20942302 R6 - 14.44738600 R5 + 51.96535307 R4 - 110.97891955 R3  
+ 140.38786698 R2 - 96.50876625 R - 81.97856683
```

Similarly, you can use the new *RDMFunctional* method in the *Energy* command

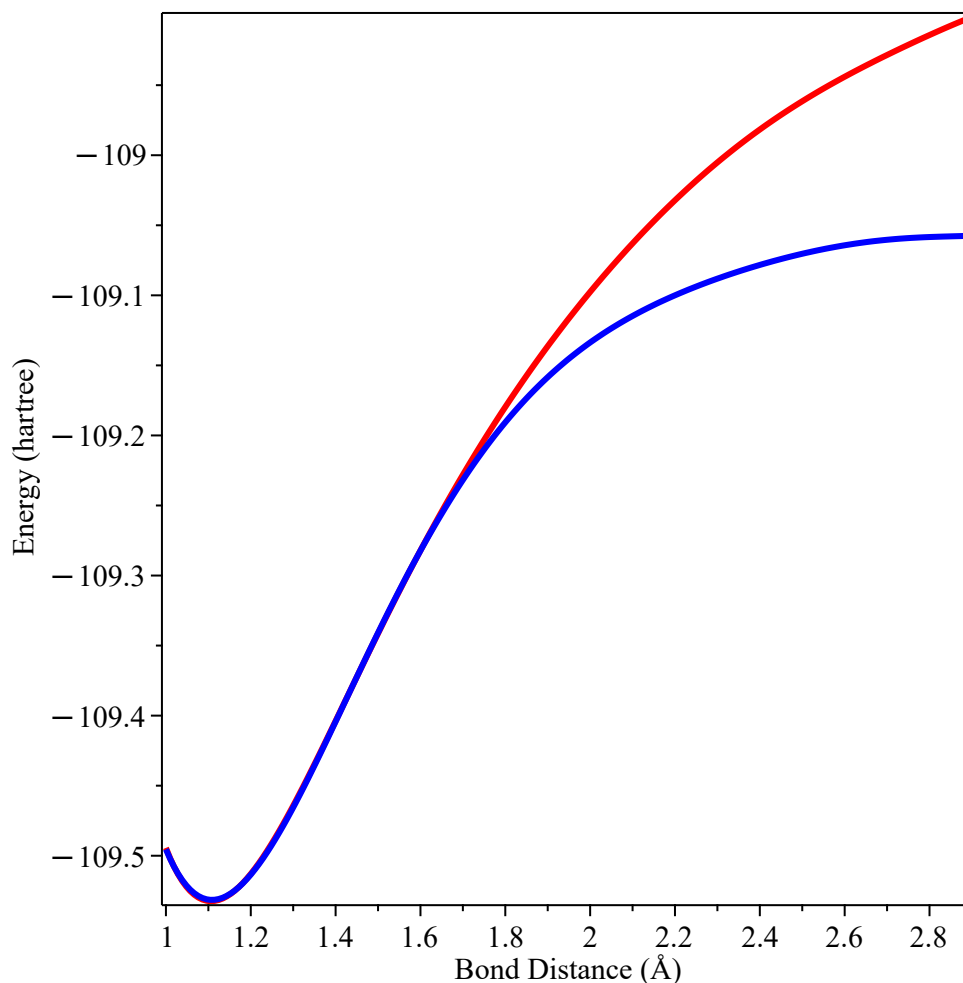
```
> energies_rdm := [seq(Energy(molecule, method = RDMFunctional, basis = "cc-pVDZ", xc  
= "B3LYP"), molecule in molecules)];  
energies_rdm := [-109.50786339, -109.52331040, -109.37588586, -109.18887170,  
-109.10062254, -109.06672079, -109.05806429, -109.05565802]
```

and generate a polynomial in R

```
> pes_rdm := interp(bond_distances, energies_rdm, R);  
pes_rdm := 0.00403016 R7 + 0.13029220 R6 - 2.19462464 R5 + 12.96198119 R4 - 38.72108443 R3  
+ 62.54064536 R2 - 51.32917602 R - 92.88749381
```

The potential energy curves from *DensityFunctional* (red) and *RDMFunctional* (blue) can be plotted together

```
> p_dft := plot(pes_dft, R = 1.0 .. 2.9, axes = boxed, labels = ["Bond Distance (Å)", "Energy (hartree)",  
    color = red, thickness = 3, labeldirections = [horizontal, vertical]) :  
    p_rdm := plot(pes_rdm, R = 1.0 .. 2.9, axes = boxed, labels = ["Bond Distance (Å)",  
    "Energy (hartree)", color = blue, thickness = 3) :  
    plots:-display({p_dft, p_rdm});
```



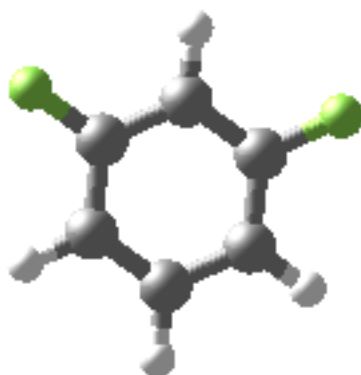
While the DFT dissociation curve (red) rises too quickly after 1.8 Å due to its incorrect treatment of static correlation, the RDMFunctional dissociation curve (blue) exhibits the correct behavior, leveling off as the bond breaks.

Enhanced Windows Edition

The Quantum Chemistry Toolbox has been updated on Windows to include the commands *CoupledCluster* and *NuclearGradient* that were previously available only on macOS and Linux. The *CoupledCluster* provides access to the coupled cluster methods, CCSD and CCSD(T), and *NuclearGradient* implements analytical gradients which significantly accelerate geometry optimizations.

Consider the molecule 1,3-difluorobenzene

```
> mol := MolecularGeometry("1,3-difluorobenzene");  
mol := [{"F", 2.36790000, 1.02540000, -0.00020000}, {"F", -2.36810000, 1.02530000, 0.00010000},  
        {"C", -0.00030000, 1.05310000, 0.00010000}, {"C", 1.20790000, 0.35580000, 0.00020000}, {"C",  
        -1.20810000, 0.35540000, -0.00020000}, {"C", 1.20820000, -1.03900000, 0.00010000}, {"C",  
        -1.20790000, -1.03940000, -0.00010000}, {"C", 0.00030000, -1.73660000, 0}, {"H",  
        -0.00040000, 2.13890000, 0.00010000}, {"H", 2.14900000, -1.58160000, 0.00010000}, {"H",  
        -2.14840000, -1.58220000, -0.00010000}, {"H", 0.00040000, -2.82270000, 0.00010000}]  
  
> PlotMolecule(mol);
```



On any supported platform including Windows you can use the command *CoupledCluster* to compute its ground-state energy and properties with coupled cluster with single and double excitations (CCSD)

```
> data := CoupledCluster(mol);
```

```

data := table (
  converged=1, populations=
  | 1 1.99904185
  | 2 1.94439981
  | 3 1.44799064
  | 4 1.79257854
  | 5 1.90224755
  | 6 1.99904230
  | ...
  | 44 element Vector[column]
, mo_coeff

=
| 1 0.70439770 -0.71208602 0.00327363 -0.00281368 -0.00128215 0.00097386 ...
| 2 -0.00477672 0.00383816 0.00086445 0.00204799 0.00089639 -0.00427394 ...
| 3 -0.00462148 0.00408573 0.00238255 -0.00108248 -0.00205104 0.00133458 ...
| 4 -0.00227506 0.00281433 0.00082170 -0.00245899 0.00153171 -0.00082123 ...
| 5 1.20677860 × 10-7 -1.52333907 × 10-7 -1.58924931 × 10-6 1.66461929 × 10-6 8.64645963 × 10-7 -7.22178108 × 10-7 ...
| 6 -0.71231999 -0.70416077 0.00326890 0.00281091 -0.00127921 -0.00097074 ...
| ...
| 44 × 44 Matrix
, dipole

=
| 1 0.00038269
| 2 -0.55574544
| 3 0.00004006
, e_tot=-423.26316684, e_corr=-0.45321282, aolabels=
| "0 F 1s"
| "0 F 2s"
| "0 F 2px"
| "0 F 2py"
| "0 F 2pz"
| "1 F 1s"
| ...
, mo_occ=
| 1 1.99999955
| 2 1.99999953
| 3 1.99997137
| 4 1.99996990
| 5 1.99996939
| 6 1.99996609
| ...
, charges=
| 1 -0.08625839
| 2 -0.08625334
| 3 -0.10260751
| 4 0.11452081
| 5 0.11447071
| 6 -0.08567199
| ...
| 12 element Vector[column]

e_tot_mp2=-423.18778258, rdm1

```

Similarly, on Windows you can now compute the analytical nuclear gradient for 1,3-difluorobenzene in seconds

```
> grad := NuclearGradient(mol, method = CoupledCluster);
```

```

grad :=
| 1 -0.04192547 -0.02497885 -0.00001438
| 2 0.04183445 -0.02481230 0.00001422
| 3 -0.00017756 -0.02222527 0.00001213
| 4 0.01089222 0.00559782 0.00003886
| 5 -0.01070347 0.00550302 -0.00005214
| ...
| 12 × 3 Matrix

```

which substantially accelerates molecular geometry optimization using the *GeometryOptimization* command.

Power of AI through New Command Chat

QCT 2024 combines a modern quantum chemistry package with the power of generative AI through natural language models. With the new command *Chat* you can ask AI to define a molecule, drug, or compound or to explain a scientific concept, terminology, or method. (Note that before using *Chat*, you need to review and agree to the AI Terms of Use.)

For example, you can ask *Chat* for information about buckeyballs

> *Chat*("Buckminsterfullerene");

"Buckminsterfullerene, also known as a buckyball, is a specific type of molecular formation that includes 60 carbon atoms arranged in a spherical structure, similar in shape to a soccer ball. It was named after Richard Buckminster Fuller, a renowned architect known for designing structures similar to the shape of this molecule.

This particular molecule is a type of fullerene, a category of carbon structures which also includes carbon nanotubes. Buckminsterfullerenes are significant in scientific research due to their unique physical and chemical properties. They are used in a variety of applications, ranging from conductors and superconductors to medical applications.

The buckminsterfullerene was discovered in 1985 by a team of scientists including Harry Kroto, Richard Smalley, and Robert Curl, for which they were awarded the Nobel Prize in Chemistry in 1996."

or the antibiotic drug penicillin

> *Chat*("penicillin");

"Penicillin is a group of antibiotics that are used to treat and prevent a wide variety of bacterial infections. These antibiotics were among the first medications to be effective against many previously serious diseases, such as tuberculosis and certain types of pneumonia. They are derived from a type of fungi called *Penicillium* fungi and are used in both human and veterinary medicine.

Penicillin works by interfering with the ability of bacteria to form cell walls. The cell walls are vital for the bacteria's survival because they keep the contents of the bacteria together and protect the bacteria from its surroundings. When the cell wall is damaged, the contents of the bacteria leak out and this kills the bacteria.

Penicillin is most effective against gram-positive bacteria, including *Streptococcus pneumoniae*, *Staphylococcus aureus*, and certain types of strep throat and skin infections. However, many types of bacteria have become resistant to penicillin due to widespread use, which has led to the development of other types of antibiotics.

It's also important to note that some people can have an allergic reaction to penicillin, which can range from a rash to a severe anaphylactic reaction."

If you want to know more about penicillin, we can just ask for more

> *Chat(more)*;

"Penicillin is a group of antibiotics that are commonly used to treat different types of infections caused by bacteria. It was the first antibiotic to be discovered and is one of the most widely used in medicine today. The medicine works by destroying the wall that surrounds the bacteria, stopping them from growing and multiplying, and ultimately killing them.

Penicillin was discovered in 1928 by a Scottish scientist named Alexander Fleming. He realized that a type of mold, *Penicillium notatum*, produced a substance that could kill bacteria like streptococci and staphylococci. This discovery revolutionized medicine and has saved countless lives since its introduction.

There are several types of penicillin, including penicillin G, penicillin V, amoxicillin, and ampicillin, among others. Each is used to treat different kinds of infections, some of which can be life-threatening, like pneumonia, meningitis, and sepsis.

While penicillin is a very important antibiotic, not everyone can use it. Some people are allergic to penicillin or may experience side effects such as diarrhea, nausea, rash, or yeast infections.

Additionally, overuse of antibiotics like penicillin has led to the development of antibiotic-resistant bacteria, which is a major concern in healthcare today."

You can also inquire about any scientific concept or terminology; for example, we can ask about entanglement

> *Chat("entanglement")*;

"Entanglement is a phenomenon in quantum physics where two or more particles become linked and instantaneously affect each other's states or properties, regardless of the distance between them. When particles become entangled, the state of one particle is directly related to the state of the other. Even when the particles are separated by large distances, a change in one of the particles results in an immediate change in the other.

This concept is often illustrated through the example of a pair of gloves: if you put a pair of gloves in two boxes (one in each), and then sent the boxes to opposite sides of the universe, the moment you opened one box and found a left-handed glove, you would immediately know that the other box must contain a right-handed glove, despite not observing it.

In real quantum entanglement, the particles do not have definite states until they are measured, and the measurement of one particle immediately and inevitably influences the state of the other particle, no matter the distance between the particles.

This phenomenon was described by Albert Einstein as "spooky action at a distance" due to its baffling and counterintuitive nature. It forms one of the core principles of quantum mechanics, with broad implications for quantum computing and secure communications, among other things."

or even an electronic structure method like density functional theory

> *Chat*("density functional theory");

"Density Functional Theory (DFT) is a computational technique used in physics and chemistry to investigate the electronic structure of many-body systems, particularly atoms, molecules, and the condensed phases.

This method is based on the principle that the ground state properties of a many-electron system can be determined through an energy functional of the electron density. In simpler terms, DFT calculates the properties of a system by examining the behavior of its electron density rather than of its individual electrons.

The advantage of using DFT in comparison to other methods is its better balance between accuracy and computational cost. While not as accurate as methods that treat every electron interaction explicitly, it is capable of making approximations that are in general accurate enough for many scientific applications, and much less computer-intensive.

DFT has become a popular method for researchers, especially in the field of quantum chemistry and solid-state physics, to study the electronic properties of matter. It is widely used to study a variety of physical and chemical phenomena, including the properties of solids, the nature of chemical bonds, and the structures and reactions of molecules."

The command *Chat* allow you to connect to the power of AI without leaving either Maple or the Quantum Chemistry package.

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

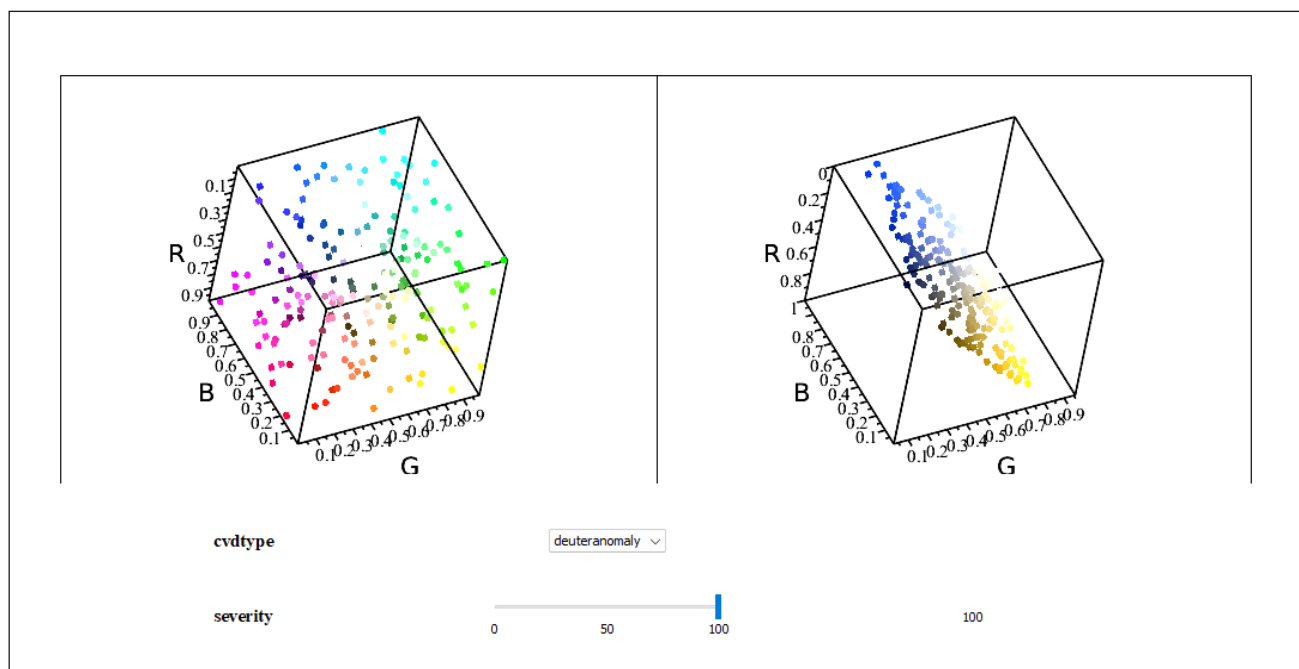
Thermodynamics), Thermodynamics (Physics), Quantum Mechanics (Physics), Computational Chemistry, and Quantum Chemistry as well as Advanced Placement (AP) and International Baccalaureate (IB) chemistry courses. Topics include atomic structure, chemical bonding, the Maxwell-Boltzmann distribution, heat capacity, enthalpy, entropy, free energy, particle-in-a-box, vibrational normal modes, infrared spectroscopy, as well as advanced electronic structure methods. Additional resources and lessons are available at the [Great Quantum Chemistry Dictionary](#) and in the Maple [Application Center](#). Use of the QCT in the classroom is described in a [recent paper in J. Chem. Ed.](#)

QCT 2024 includes a new curricular set of lessons entitled "The Chemistry of Art." For example, the following sections provide two short excerpts from lessons in which (1) the visual effects of color vision deficiency (CVD) are explored through an interactive embedded component and (2) quantum calculations of the indigo and tyrian purple dyes are performed to compare their absorption spectra.

▼ Color Vision Deficiency (CVD)

Using Maple, you can simulate the effect of CVD on the perception of a sample color palette plotted in the RGB color space. Select a CVD type, and then slowly increase the severity from 0 to 100 and watch as the colorspace is distorted and each color in the palette changes. You may find it useful to rotate the plots as you increase the severity. If you personally experience a form of color blindness, you can still describe the effect of the simulation as the points representing the colors in the gamut contract into a plane. You should also recall what property is plotted along each axis.

```
> colors := [seq(RandomTools:-Generate(list(float(range = 0..1, digits = 4, method = uniform),
3)), i = 1..200)]:
Explore(plots:-display(Vector[row]([ColorTools:-SpatterPlot3d(colors, symbol = "box", space
= "RGB", labelfont = [Helvetica, 16], orientation = [-20, 45, -38]), ColorTools:-
SpatterPlot3d(map(ColorTools:-CVDSimulation, colors, cvdtype, severity), symbol = "box",
space = "RGB", labelfont = [Helvetica, 16], orientation = [-20, 45, -38]))), severity = 0
..100, parameters = [cvdtype = ["deuteranomaly", "protanomaly", "tritanomaly"]], placement
= bottom, size = [1000, 350]);
```



▼ Quantum Calculations of Indigo and Tyrian Purple Dyes

Consider two related dyes: indigo and tyrian purple:

> *indigo* := [["C", -2.82265000, 0.73898000, 0.00005000], ["C", -3.96802000, 1.53537000, 0.], ["C", -5.20393000, 0.88612000, -0.00005000], ["C", -5.31138000, -0.51504000, -0.00005000], ["C", -4.16296000, -1.30350000, -0.00001000], ["C", -2.91836000, -0.67319000, 0.00003000], ["H", -6.29247000, -0.97833000, -0.00009000], ["H", -4.21764000, -2.38780000, -0.00001000], ["H", -3.90356000, 2.61878000, 0.00001000], ["H", -6.10933000, 1.48632000, -0.00009000], ["C", -1.55139000, -1.21054000, 0.00005000], ["C", -0.68036000, 0.00041000, 0.00006000], ["N", -1.48727000, 1.11979000, 0.00017000], ["O", -1.14282000, -2.37426000, 0.00004000], ["H", -1.10083000, 2.05448000, -0.00008000], ["C", 0.68036000, -0.00040000, -0.00006000], ["N", 1.48727000, -1.11979000, -0.00017000], ["C", 2.82264000, -0.73898000, -0.00005000], ["C", 2.91836000, 0.67319000, -0.00003000], ["C", 1.55140000, 1.21054000, -0.00004000], ["H", 1.10081000, -2.05448000, 0.00011000], ["O", 1.14282000, 2.37426000, -0.00004000], ["C", 3.96802000, -1.53537000, -0.], ["C", 4.16297000, 1.30350000, -0.], ["C", 5.31138000, 0.51504000, 0.00004000], ["C", 5.20393000, -0.88612000, 0.00005000], ["H", 6.10932000, -1.48633000, 0.00009000], ["H", 3.90355000, -2.61878000, 0.], ["H", 4.21765000, 2.38780000, 0.], ["H", 6.29247000, 0.97832000, 0.00007000]] :

```

> tyrianpurple := [[ "C", -7.53629000, 2.44628000, 0.02964000], [ "C", -6.37087000,
3.18191000, 0.04692000], [ "C", -5.15824000, 2.48164000, 0.00715000], [ "C",
-5.11665000, 1.07281000, -0.02689000], [ "C", -6.29914000, 0.33034000,
-0.01590000], [ "C", -7.48107000, 1.04288000, 0.02018000], [ "H", -4.16101000,
0.55333000, -0.05401000], [ "H", -6.29526000, -0.75397000, -0.03064000], [ "H",
-6.39556000, 4.26515000, 0.09313000], [ "Br", -3.53176000, 3.45097000, 0.01113000],
[ "C", -8.87511000, 0.56228000, 0.04867000], [ "C", -9.70911000, 1.82064000,
0.05880000], [ "N", -8.84729000, 2.89698000, 0.13866000], [ "O", -9.25455000,
-0.59150000, 0.06050000], [ "H", -9.17649000, 3.82873000, -0.08865000], [ "C",
-11.03865000, 1.75648000, -0.03735000], [ "N", -11.90080000, 0.68029000,
-0.11662000], [ "C", -13.21148000, 1.13137000, -0.00440000], [ "C", -13.26634000,
2.53479000, 0.00535000], [ "C", -11.87231000, 3.01505000, -0.02598000], [ "H",
-11.57160000, -0.25121000, 0.11179000], [ "O", -11.49263000, 4.16874000,
-0.03827000], [ "C", -14.37716000, 0.39607000, -0.01880000], [ "C", -14.44796000,
3.24766000, 0.04478000], [ "C", -15.63061000, 2.50553000, 0.05891000], [ "C",
-15.58949000, 1.09670000, 0.02446000], [ "Br", -17.21626000, 0.12784000, 0.02504000],
[ "H", -14.35292000, -0.68716000, -0.06523000], [ "H", -14.45146000, 4.33197000,
0.05995000], [ "H", -16.58602000, 3.02527000, 0.08893000] ] :

> plots:-display( Vector[row]([ PlotMolecule(indigo), PlotMolecule(tyrianpurple) ]));

```

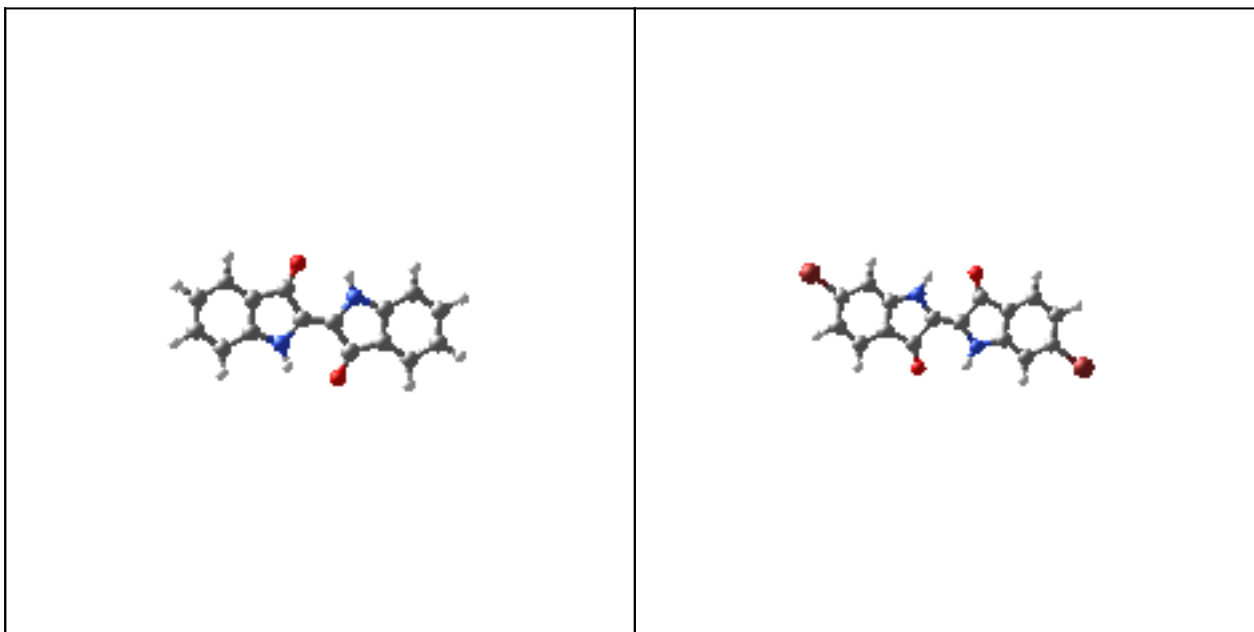


Figure: Structures of indigo (left) and tyrian purple (right).

The only difference between the two structures is that tyrian purple has bromines at the 6 and 6' positions. This does not affect the number of π -electrons or the length of the conjugated chain, so the particle-in-a-box model would treat these two dyes as being identical!

For more complex compounds, one must use more sophisticated methods to calculate the energy levels. Here we use the Quantum Chemistry Toolbox to calculate the ground and excited states of each of these dyes.

After performing DFT and TDDFT calculations for both molecules in the lesson (see the lesson for the explicit calculations), you obtain the following two tables of excitation spectra with the *ExcitationSpectra* command:

Indigo

```
spectra_indigo_b3lyp := ExcitationSpectra(indigo, method = DensityFunctional, basis = "6-31g", nstates = [3, 3], showtable);
```

State	Energy	Wavelength	Spin	Oscillator
1	1.03972808 eV	1192.46753\ 221 nm	Triplet	0.44895520
2	2.32548387 eV	533.154406\ 26 nm	Singlet	0.28403090
3	2.35743686 eV	525.927967\ 49 nm	Triplet	0.08185276
4	2.42658941 eV	510.940156\ 24 nm	Triplet	4.81491290 × 10
5	2.79208571 eV	444.055843\ 58 nm	Singlet	2.51377265 × 10
6	3.46710287 eV	357.601727\ 17 nm	Singlet	0.00641236

and

Tyrian Purple

spectra_tyrianpurple_b3lyp := ExcitationSpectra(tyrianpurple, method = DensityFunctional, basis = "6-31g", nstates = [3, 3], showtable);

State	Energy	Wavelength	Spin	Oscillator
1	1.18482603 eV	1046.43377 \ 112 nm	Singlet	0.01063373
2	1.46300077 eV	847.465019 \ 88 nm	Triplet	0.42879167
3	2.46467321 eV	503.045177 \ 81 nm	Triplet	5.64622357 × 10
4	2.59177389 eV	478.375824 \ 95 nm	Singlet	0.30183798
5	2.93685950 eV	422.165913 \ 57 nm	Triplet	1.77823855
6	2.94297131 eV	421.289181 \ 32 nm	Singlet	0.00001806

Looking for the singlet excitations with the largest oscillator strengths, you can predict transitions at 533 nm and 478 nm for indigo and tyrian purple with the relative ordering in reasonable agreement with experimental measurements. Their differences in absorption are responsible for their differences in color!