

# Normal mode analysis of a single water molecule

---

## Specifications

### Geometry of the water molecule

We use the specification for the flexible TIP3P-FB water model (Wang, Martinez, Pande, J. Phys. Chem. Lett., 5:1885, 2014), which is used in the Amber MD force field. Distances are here given in nanometers.

```
In[1]:= distanceOH = Quantity[0.1011811, "nm"]
```

```
Out[1]= 0.101181 nm
```

```
In[2]:= distanceHH = Quantity[0.1638684, "nm"]
```

```
Out[2]= 0.163868 nm
```

The water molecule can be viewed as an isosceles triangle, with oxygen at the top and the two H atoms at the base. Determine first the distance of the oxygen atom from the base:

```
In[3]:= dOH = QuantityMagnitude[distanceOH]
```

```
Out[3]= 0.101181
```

```
In[4]:= dHH = QuantityMagnitude[distanceHH]
```

```
Out[4]= 0.163868
```

```
In[5]:= eq = (dHH / 2) ^ 2 + h ^ 2 == dOH ^ 2
```

```
Out[5]= 0.00671321 + h^2 == 0.0102376
```

```
In[6]:= soln = Solve[eq, h]
```

```
Out[6]= { {h → -0.0593667}, {h → 0.0593667} }
```

```
In[7]:= hsol = soln[[2, 1, 2]]
```

```
Out[7]= 0.0593667
```

Define now the positions of the atoms:

```
In[8]:= ROxygen = {0., 0., 0.}
```

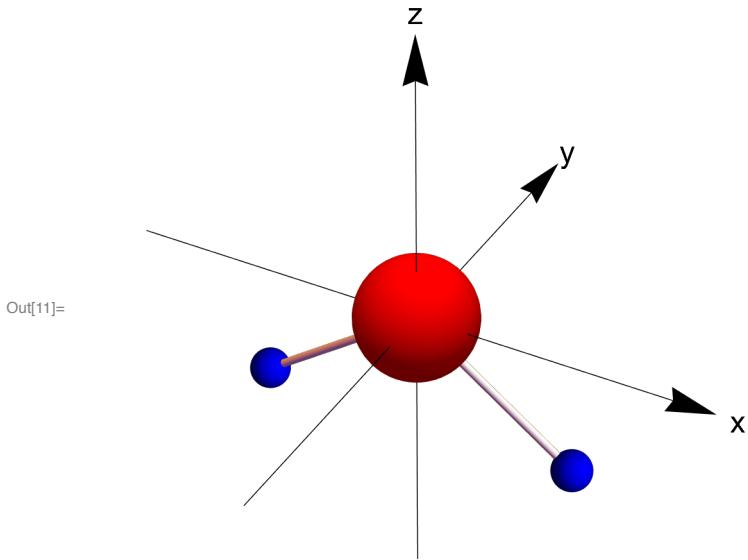
```
Out[8]= {0., 0., 0.}
```

```
In[9]:= RH1 = {dHH / 2, 0, -hsol}
```

```
Out[9]= {0.0819342, 0, -0.0593667}
```

```
In[10]:= RH2 = {-dHH / 2, 0, -hsol}
Out[10]= {-0.0819342, 0, -0.0593667}

In[11]:= Graphics3D[{Red, Sphere[R0xygen, 0.03]}, {Blue, Sphere[RH1, 0.01]}, {Blue, Sphere[RH2, 0.01]}, Tube[{R0xygen, RH1}, 0.002], Tube[{R0xygen, RH2}, 0.002], Arrow[{{-0.15, 0, 0}, {0.15, 0, 0}}], Arrow[{{0, -0.15, 0}, {0, 0.15, 0}}], Arrow[{{0, 0, -0.15}, {0, 0, 0.15}}], Text[Style["x", 16], {0.16, 0, 0}], Text[Style["y", 16], {0, 0.16, 0}], Text[Style["z", 16], {0, 0, 0.16}], Boxed → False]
```



## Atomic masses

Here a.m.u. is the atomic mass unit

```
In[12]:= massH = ElementData["Hydrogen", "AtomicWeight"]
```

```
Out[12]= 1.008 u
```

```
In[13]:= massO = ElementData["Oxygen", "AtomicWeight"]
```

```
Out[13]= 15.999 u
```

```
In[14]:= UnitConvert[massH, "kg"]
```

```
Out[14]= 1.674 × 10-27 kg
```

```
In[15]:= UnitConvert[massO, "kg"]
```

```
Out[15]= 2.6567 × 10-26 kg
```

## Potential

This is the force constant of flexible TIP3P water potential of the Amber MD program

```
In[16]:= Na = Quantity[1, "AvogadroNumber"]
Out[16]= 1 Avogadro number

In[17]:= Ktip3pAmber = Quantity[553.0, "kcal/Angstroms^2"] / Na
Out[17]= 553. kcalth / (Avogadro number Å2)
```

This is the force constant in a.m.u/fs<sup>2</sup>

```
In[18]:= Ktip3p = UnitConvert[Ktip3pAmber, "amu/femtoseconds^2"]
Out[18]= 0.231375 u/fs2
```

The same force constant for O-H, and H-H

```
In[19]:= K12 = QuantityMagnitude[Ktip3p]
Out[19]= 0.231375
```

```
In[20]:= K13 = QuantityMagnitude[Ktip3p]
Out[20]= 0.231375
```

```
In[21]:= K23 = QuantityMagnitude[Ktip3p]
Out[21]= 0.231375
```

Combined position vector

```
In[22]:= r = {x1, y1, z1, x2, y2, z2, x3, y3, z3}
Out[22]= {x1, y1, z1, x2, y2, z2, x3, y3, z3}
```

Subvectors for O, H1, H2

```
In[23]:= rOxygen = r[[1 ;; 3]]
Out[23]= {x1, y1, z1}
```

```
In[24]:= rH1 = r[[4 ;; 6]]
Out[24]= {x2, y2, z2}
```

```
In[25]:= rH2 = r[[7 ;; 9]]
Out[25]= {x3, y3, z3}
```

Distances

```
In[26]:= dOH
Out[26]= 0.101181
```

```
In[27]:= dHH
Out[27]= 0.163868
```

Functional form of the potential

```
In[28]:=  $\phi = K_{12} (\text{Sqrt}[(rOxygen - rH1) \cdot (rOxygen - rH1)] - dOH)^2 +$ 
           $K_{13} (\text{Sqrt}[(rOxygen - rH2) \cdot (rOxygen - rH2)] - dOH)^2 +$ 
           $K_{23} (\text{Sqrt}[(rH2 - rH1) \cdot (rH2 - rH1)] - dHH)^2$ 

Out[28]=  $0.231375 \left( -0.101181 + \sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2} \right)^2 +$ 
           $0.231375 \left( -0.101181 + \sqrt{(x1 - x3)^2 + (y1 - y3)^2 + (z1 - z3)^2} \right)^2 +$ 
           $0.231375 \left( -0.163868 + \sqrt{(-x2 + x3)^2 + (-y2 + y3)^2 + (-z2 + z3)^2} \right)^2$ 
```

The potential for the equilibrium position vanishes

```
In[29]:=  $\phi /. \{x1 \rightarrow R0xygen[[1]], y1 \rightarrow R0xygen[[2]], z1 \rightarrow R0xygen[[3]], x2 \rightarrow RH1[[1]],$ 
           $y2 \rightarrow RH1[[2]], z2 \rightarrow RH1[[3]], x3 \rightarrow RH2[[1]], y3 \rightarrow RH2[[2]], z3 \rightarrow RH2[[3]]\}$ 

Out[29]= 0.
```

---

## Force constant matrix

### Forces

```
In[30]:= F = -Grad[\phi, r];
```

The sum of all forces is zero and there is no global translational acceleration

```
In[31]:= F0xy = F[[1 ;; 3]] // Simplify;
```

```
In[32]:= FH1 = F[[4 ;; 6]] // Simplify;
```

```
In[33]:= FH2 = F[[7 ;; 9]] // Simplify;
```

```
In[34]:= F0xy + FH1 + FH2
```

```
Out[34]= {0., 0., 0.}
```

The forces on the atoms are zero in the equilibrium configuration

```
In[35]:= F0 = F /. {x1 \rightarrow R0xygen[[1]], y1 \rightarrow R0xygen[[2]], z1 \rightarrow R0xygen[[3]], x2 \rightarrow RH1[[1]],
```

$$y2 \rightarrow RH1[[2]], z2 \rightarrow RH1[[3]], x3 \rightarrow RH2[[1]], y3 \rightarrow RH2[[2]], z3 \rightarrow RH2[[3]]}$$

```
Out[35]= {0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

### Matrix of force constants

The matrix of force constants is the second derivative of the potential at the equilibrium configuration

```
In[36]:= tmp = Grad[Grad[\phi, r], r] / 2;
```

```
In[37]:= MatrixForm[
K = tmp /. {x1 → R0xygen[[1]], y1 → R0xygen[[2]], z1 → R0xygen[[3]], x2 → RH1[[1]],
y2 → RH1[[2]], z2 → RH1[[3]], x3 → RH2[[1]], y3 → RH2[[2]], z3 → RH2[[3]]}]

Out[37]//MatrixForm=
```

$$\begin{pmatrix} 0.303444 & 0. & 0. & -0.151722 & 0. & 0.109932 & -0.151722 & 0. & -0.109932 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.159306 & 0.109932 & 0. & -0.0796532 & -0.109932 & 0. & -0.0796532 \\ -0.151722 & 0. & 0.109932 & 0.383097 & 0. & -0.109932 & -0.231375 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0.109932 & 0. & -0.0796532 & -0.109932 & 0. & 0.0796532 & 0. & 0. & 0. \\ -0.151722 & 0. & -0.109932 & -0.231375 & 0. & 0. & 0.383097 & 0. & 0.109932 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -0.109932 & 0. & -0.0796532 & 0. & 0. & 0. & 0.109932 & 0. & 0.0796532 \end{pmatrix}$$

## Eigenvalues and eigenvectors of the force constant matrix

The force constant matrix is symmetric and can thus be diagonalized - i.e. all its eigenvalues are real and its eigenvectors form an orthogonal basis of  $\mathbb{R}^{3N}$ . The matrix  $\mathbf{K}$  is found to be positive semi-definite :

```
In[38]:= evK = Eigenvalues[K] // Chop
Out[38]= {0.694126, 0.534819, 0.159306, 0, 0, 0, 0, 0, 0}
```

```
In[39]:= eveck = Eigenvectors[K] // Chop;
In[40]:= MatrixForm[QK = Transpose[eveck]]
```

```
Out[40]//MatrixForm=
```

$$\begin{pmatrix} 0 & -0.753244 & 0 & 0.235959 & 0.298184 & -0.473397 & 0.252841 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - \\ 0.315103 & 0 & -0.753244 & 0.382645 & 0.232634 & 0.163245 & -0.325804 & 0 & 0 \\ 0.652329 & 0.376622 & 0.272887 & -0.117979 & 0.466742 & -0.355115 & 0.0167748 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1. \\ -0.157552 & -0.272887 & 0.376622 & -0.105839 & 0.465269 & 0.326489 & -0.651609 & 0 & 0 \\ -0.652329 & 0.376622 & -0.272887 & -0.117979 & 0.466742 & -0.355115 & 0.0167748 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.452544 & -0.62485 & -0.636212 & 0 & 0 \\ -0.157552 & 0.272887 & 0.376622 & 0.871129 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
In[41]:= MatrixForm[Transpose[QK].QK // Chop]
Out[41]//MatrixForm=
```

$$\begin{pmatrix} 1. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1. & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1. & 0 \end{pmatrix}$$

The spectral decomposition of the force constant matrix is  $\mathbf{K}=\mathbf{Q}\cdot\Lambda\cdot\mathbf{Q}^T$ , where the columns of  $\mathbf{Q}$  are the eigenvectors of  $\mathbf{K}$ :

```
In[42]:= MatrixForm[\Delta = DiagonalMatrix[evK]]
Out[42]//MatrixForm=
```

$$\begin{pmatrix} 0.694126 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0.534819 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.159306 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$
  

```
In[43]:= (K - QK.\Delta.Transpose[QK]) // Chop // MatrixForm
Out[43]//MatrixForm=
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## Normal modes

### Theory

We consider a potential function  $V[\mathbf{x}] \equiv V[\mathbf{x}_1, \dots, \mathbf{x}_N]$ , where  $\mathbf{x}_k$  is the position of atom  $k$  in a molecule of  $N$  atoms, and we write

$$\mathbf{x} = \mathbf{R} + \mathbf{u}$$

where  $\mathbf{R}$  is the set of equilibrium positions and  $\mathbf{u}$  the set of deviations with respect to these positions. We have then

$$\mathbf{F}[\mathbf{R}] = - \left( \frac{\partial V[\mathbf{x}]}{\partial \mathbf{x}} \right)_{\mathbf{x}=\mathbf{R}} = \mathbf{0},$$

i.e. the forces at the equilibrium positions vanish, and we may develop the potential up to second order

$$V[\mathbf{R} + \mathbf{u}] \approx V[\mathbf{R}] + \frac{1}{2} \mathbf{u}^T \cdot \left( \frac{\partial^2 V[\mathbf{x}]}{\partial \mathbf{x} \partial \mathbf{x}} \right)_{\mathbf{x}=\mathbf{R}} \cdot \mathbf{u}.$$

Setting  $V[\mathbf{R}] = 0$ , and defining the positive definite matrix of force constants

$$\mathbf{K} = \left( \frac{\partial^2 V[\mathbf{x}]}{\partial \mathbf{x} \partial \mathbf{x}} \right)_{\mathbf{x}=\mathbf{R}},$$

and the diagonal mass matrix  $\mathbf{M} = \text{diag}(M_1, M_1, M_1, \dots, M_N, M_N, M_N)$ , the equations of motion of the atoms in a molecule - here  $H_2O$  - for the deviations  $\mathbf{u}$  read

$$\mathbf{M} \cdot \ddot{\mathbf{u}} = -\mathbf{K} \cdot \mathbf{u},$$

where  $\mathbf{u}$  are the deviations of atomic positions  $\mathbf{x}$  with respect to some equilibrium position  $\mathbf{R}$ ,  
 $\mathbf{x} = \mathbf{R} + \mathbf{u}$ .

The masses can be removed from the calculation by introducing mass – weighted coordinates,  
 $\tilde{\mathbf{x}} = \mathbf{M}^{1/2} \cdot \mathbf{x}$ , such that

$$\ddot{\tilde{\mathbf{u}}} = -\tilde{\mathbf{K}} \cdot \tilde{\mathbf{u}}, \text{ where } \tilde{\mathbf{K}} = \mathbf{M}^{-1/2} \cdot \mathbf{K} \cdot \mathbf{M}^{-1/2}.$$

Note that  $\tilde{\mathbf{K}}$  has the units of a squared frequency. As  $\mathbf{K}$ , the matrix  $\tilde{\mathbf{K}}$  is positive semidefinite and can thus be diagonalized

$$\tilde{\mathbf{K}} = \tilde{\mathbf{Q}} \cdot \Omega^2 \cdot \tilde{\mathbf{Q}}^T,$$

where the columns of  $\tilde{\mathbf{Q}}$  are the eigenvectors of  $\tilde{\mathbf{K}}$  and

$$\Omega^2 = \text{diag}(\omega_1^2, \dots, \omega_{3N}^2)$$

is the diagonal matrix of squared eigenfrequencies. In the basis of the eigenvectors of  $\tilde{\mathbf{K}}$ , which are called the *normal modes*, the equations of motion become decoupled. Writing

$$\tilde{\mathbf{u}} = \tilde{\mathbf{Q}} \cdot \xi,$$

where  $\xi$  is the vector of normal coordinates, we have

$$\ddot{\xi} = -\Omega^2 \cdot \xi,$$

Since  $\Omega^2$  is diagonal, the solution for the components of can be written in the form

$$\xi_k(t) = \xi_k(0) \cos[\omega_k t] + \frac{\dot{\xi}_k(0)}{\omega_k} \sin[\omega_k t]$$

where  $\xi(0) = \mathbf{Q}^T \cdot \tilde{\mathbf{x}}[0] = \mathbf{Q}^T \cdot \mathbf{M}^{1/2} \cdot \mathbf{x}[0]$  and  $\dot{\xi}(0) = \mathbf{Q}^T \cdot \dot{\tilde{\mathbf{x}}}[0] = \mathbf{Q}^T \cdot \mathbf{M}^{1/2} \cdot \dot{\mathbf{x}}[0]$ .

## Mass-weighted coordinates and force constant matrix

To obtain the normal modes and frequencies for the water molecule we construct the diagonal mass matrix

```
In[44]:= mH = QuantityMagnitude[massH]; mO = QuantityMagnitude[massO];
```

```
In[45]:= MatrixForm[M = DiagonalMatrix[{mO, mO, mO, mH, mH, mH, mH, mH}]];
```

The square root of M can be obtained by replacing the masses on the diagonal of K by their square roots or by Cholesky decomposition

```
In[46]:= MatrixForm[sqrtM = CholeskyDecomposition[M]];
```

This is the mass-weighted force constant matrix

```
In[47]:= MatrixForm[Ktilde = Inverse[sqrtM].K.Inverse[sqrtM]]
Out[47]//MatrixForm=
```

$$\begin{pmatrix} 0.0189664 & 0. & 0. & -0.0377809 & 0. & 0.0273747 & -0.0377809 & 0. & -0.0273747 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.00995728 & 0.0273747 & 0. & -0.0198348 & -0.0273747 & 0. & -0.0198348 \\ -0.0377809 & 0. & 0.0273747 & 0.380057 & 0. & -0.10906 & -0.229539 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0.0273747 & 0. & -0.0198348 & -0.10906 & 0. & 0.0790211 & 0. & 0. & 0. \\ -0.0377809 & 0. & -0.0273747 & -0.229539 & 0. & 0. & 0.380057 & 0. & 0.10906 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -0.0273747 & 0. & -0.0198348 & 0. & 0. & 0. & 0.10906 & 0. & 0.0790211 \end{pmatrix}$$

## Normal frequencies and normal modes

The eigenvalues of  $\tilde{\mathbf{K}}$  are the squared eigenfrequencies (here in  $1/\text{fs}^2 = (10^{15} \text{ Hz})^2$ ,

```
In[48]:= Eigenvalues[Ktilde] // Chop
Out[48]= {0.634161, 0.248505, 0.0644126, 0, 0, 0, 0, 0, 0}
```

These are the normal frequencies (in  $10^{15} \text{ Hz}$ )

```
In[49]:= NMfreq = Sqrt[Eigenvalues[Ktilde] // Chop]
Out[49]= {0.796342, 0.498503, 0.253796, 0, 0, 0, 0, 0, 0}
```

and these the normal modes:

```
In[50]:= MatrixForm[Q = Transpose[Eigenvectors[Ktilde] // Chop]]
Out[50]//MatrixForm=
```

$$\begin{pmatrix} 0 & 0.276264 & 0 & 0.17484 & 0.378793 & -0.359166 & 0.787797 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0694627 & 0 & -0.327233 & 0.353809 & -0.249101 & -0.776548 & -0.312786 & 0 \\ 0.691695 & -0.550315 & 0.146828 & -0.348278 & 0.0497754 & -0.231384 & 0.140855 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1. \\ -0.138369 & 0.398739 & 0.651844 & -0.452432 & -0.125051 & -0.389836 & -0.157022 & 0 \\ -0.691695 & -0.550315 & -0.146828 & -0.348278 & 0.0497754 & -0.231384 & 0.140855 & 0 \\ 0 & 0 & 0 & 0 & 0.8797 & -0.0944685 & -0.466052 & 0 \\ -0.138369 & -0.398739 & 0.651844 & 0.630048 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The first three columns are the normal modes corresponding to frequencies  $\omega_k > 0$ , corresponding to vibrations, and the remaining ones correspond to frequencies  $\omega_k = 0$ , i.e. to global translations and rotations which do not lead to a deformation of the molecule.

## Geometric animation for a single mode

Chose the mode, an amplitude for the animation, and the number of frames for a (purely geometric) cyclic motion

```
In[51]:= iPICK = 1; ε = 0.03; Nsteps = 100;
```

The is the resulting “time” step

```
In[52]:= Δt = 2 π / (Nsteps - 1);
```

Vector of initial positions:

```
In[53]:= x0 = Flatten[{ROxygen, RH1, RH2}]
Out[53]= {0., 0., 0., 0.0819342, 0, -0.0593667, -0.0819342, 0, -0.0593667}
```

Animation of the positions

```
In[54]:= ε = Table[If[k == iPICK, ε * Sin[t], 0], {k, 1, 9}]
```

```
Out[54]= {0.03 Sin[t], 0, 0, 0, 0, 0, 0, 0, 0}
```

```
In[55]:= MatrixForm[xMode[t_] = x0 + Inverse[sqrtM].Q.ε // Chop]
```

```
Out[55]//MatrixForm=
```

$$\begin{pmatrix} 0 \\ 0 \\ 0.000520986 \sin[t] \\ 0.0819342 + 0.0206683 \sin[t] \\ 0 \\ -0.0593667 - 0.00413455 \sin[t] \\ -0.0819342 - 0.0206683 \sin[t] \\ 0 \\ -0.0593667 - 0.00413455 \sin[t] \end{pmatrix}$$

Extract positions for the graphics commands

```
In[56]:= ROxygenMode[t_] = xMode[t][[1 ;; 3]];
```

```
In[57]:= RH1Mode[t_] = xMode[t][[4 ;; 6]];
```

```
In[58]:= RH2Mode[t_] = xMode[t][[7 ;; 9]];
```

Make the list of frames

```
In[59]:= FigList = Table[Graphics3D[{Red, Sphere[ROxygenMode[k * Δt], 0.03],
                                 {Blue, Sphere[RH1Mode[k * Δt], 0.01]}, {Blue, Sphere[RH2Mode[k * Δt], 0.01]},
                                 Tube[{ROxygenMode[k * Δt], RH1Mode[k * Δt]}, 0.002],
                                 Tube[{ROxygenMode[k * Δt], RH2Mode[k * Δt]}, 0.002],
                                 Arrow[{{-0.15, 0, 0}, {0.15, 0, 0}}], Arrow[{{0, -0.15, 0}, {0, 0.15, 0}}],
                                 Arrow[{{0, 0, -0.15}, {0, 0, 0.15}}], Text[Style["x", 16], {0.16, 0, 0}],
                                 Text[Style["y", 16], {0, 0.16, 0}], Text[Style["z", 16], {0, 0, 0.16}]},
                                 Boxed → False, ImageSize → Medium], {k, 0, Nsteps - 1}];
```

Animation

```
In[60]:= ListAnimate[FigList, AnimationRunning → False]
```

