

## 0.1 BITSS

The BITSS (Binary-Image Transition State Search) is an algorithm to find transition state and a reaction pathway starting from only reactant and product structure. It is a "bracketing" method, which is different from interpolation-based methods such as NEB, CI-NEB etc. in that it does not require an interpolation for the initial reaction path.

This means that it can converge to a transition state even if the PES is skewed. BITSS is also a gradient-only method and does not require a Hessian calculation, however, using a Hessian does improve convergence.

The BITSS method works by putting an energy constraint on both the reactant and product molecules that drives them towards each other until they meet at a saddle point, which is the transition state.

The BITSS energy for a pair of states (molecules in this case) is defined as:

$$E_{\text{BITSS}}(\vec{r}_1, \vec{r}_2) = E_1(\vec{r}_1) + E_2(\vec{r}_2) + \kappa_e(E_1 - E_2)^2 + \kappa_d(d - d_i)^2 \quad (1)$$

where  $E_1$  and  $E_2$  are the energies of the two individual molecules respectively. The vectors  $\vec{r}_1$  and  $\vec{r}_2$  are essentially the coordinates of each molecule. In the current *autodE* implementation, Cartesian coordinates of each atom are used. (Internal coordinates are also usable but difficult to implement in code, so it has not been done.)

$$\vec{r}_1 = (x_{1A}, y_{1A}, z_{1A}, x_{1B}, y_{1B}, \dots) \quad (2)$$

$$\vec{r}_2 = (x_{2A}, y_{2A}, z_{2A}, x_{2B}, y_{2B}, \dots) \quad (3)$$

where  $x_{2A}$  represents the x-coordinate of atom A of 2nd state (molecule) for example.

The distance  $d$  in eqn.-(1) can be expressed in multiple ways. In this implementation, we simply use the Euclidean distance in terms of Cartesian coordinates:

$$d = \sqrt{(x_{1A} - x_{2A})^2 + (y_{1A} - y_{2A})^2 + (z_{1A} - z_{2A})^2 + (x_{1B} - x_{2B})^2 \dots} \quad (4)$$

The BITSS optimiser minimises  $E_{\text{BITSS}}$  by varying all of the available coordinates ( $\vec{r}_1$  and  $\vec{r}_2$ ) simultaneously, i.e. it optimises both reactant and product together in each step.

In *autodE*, the BITSS optimiser can either use steepest descent or RFO for minimisation.

### 0.1.1 Analytic gradient

Gradient of the BITSS pair energy is:

$$\nabla E_{\text{BITSS}}(\vec{r}_1, \vec{r}_2) = \left( \frac{\partial E}{\partial x_{1A}}, \frac{\partial E}{\partial y_{1A}}, \frac{\partial E}{\partial z_{1A}}, \dots, \frac{\partial E}{\partial x_{2A}}, \frac{\partial E}{\partial y_{2A}}, \frac{\partial E}{\partial z_{2A}}, \dots \right) \quad (5)$$

Notice that the first half of the elements of gradient vector (eqn. 5) only depend on  $\vec{r}_1$  while the second half of the elements depend on  $\vec{r}_2$  only.

The gradient can be calculated for each term in  $E_{\text{BITSS}}$  as written in eqn.-1. First we handle the energy terms:

$$\begin{aligned} E_1 + E_2 + \kappa_e(E_1 - E_2)^2 &= E_1(1 - 2\kappa_e E_2) + \kappa_e E_1^2 + (\kappa_e E_2^2 + E_2) \\ &= E_2(1 - 2\kappa_e E_1) + \kappa_e E_2^2 + (\kappa_e E_1^2 + E_1) \end{aligned} \quad (6)$$

Gradient terms from eqn. (6) that only depend on  $\vec{r}_1$  would be

$$\nabla_{\vec{r}_1} E_1(1 - 2\kappa_e E_2) + 2\kappa_e E_1 \nabla_{\vec{r}_1} E_1 = \nabla_{\vec{r}_1} E_1(1 - 2\kappa_e E_2 + 2\kappa_e E_1)$$

and those that depend only on  $\vec{r}_2$  would be

$$\nabla_{\vec{r}_2} E_2(1 - 2\kappa_e E_1) + 2\kappa_e E_2 \nabla_{\vec{r}_2} E_2 = \nabla_{\vec{r}_2} E_2(1 + 2\kappa_e E_2 - 2\kappa_e E_1)$$

Now, realising that  $\nabla_{\vec{r}_1} E_1$  and  $\nabla_{\vec{r}_2} E_2$  are simply the molecular gradients, and concatenating the terms from  $\vec{r}_1$  and  $\vec{r}_2$  allows us to obtain the gradient contribution from energy.

The distance terms depends on both  $\vec{r}_1$  and  $\vec{r}_2$  terms.

$$\kappa_d(d - d_i)^2 = \kappa_d d^2 - 2\kappa_d d d_i + \kappa_d d_i^2$$

Gradient of the distance terms would be ( $d_i$  is a constant):

$$2\kappa_d(d - d_i)\nabla_{\vec{r}_1, \vec{r}_2} d$$

Now,  $d = \sqrt{(x_{1A} - x_{2A})^2 + (y_{1A} - y_{2A})^2 + (z_{1A} - z_{2A})^2 + \dots}$  So,

$$\frac{\partial d}{\partial x_{1A}} = \frac{1}{d}(x_{1A} - x_{2A}) \text{ and } \frac{\partial d}{\partial x_{2A}} = \frac{1}{d}(x_{2A} - x_{1A})$$

It is easy to see therefore,

$$\nabla_{\vec{r}_1} d = \frac{1}{d}(\vec{r}_1 - \vec{r}_2) \text{ and } \nabla_{\vec{r}_2} d = \frac{1}{d}(\vec{r}_2 - \vec{r}_1)$$

Concatenating these two arrays will provide  $\nabla_{\vec{r}_1, \vec{r}_2} d$ .

Summing the contributions of gradient from energy terms and distance terms will provide the gradient of BITSS energy.

### 0.1.2 Analytic Hessian

Optimising with only gradient is possible with gradient descent, but not recommended due to slow rate of convergence. Hessian is necessary for a quasi-Newton optimisation. Fortunately the Hessian for BITSS energy can be built up in terms of molecular Hessians and gradients.

The Hessian of BITSS energy can be written as:

$$\nabla^2 E_{\text{BITSS}} = \begin{bmatrix} \frac{\partial^2 E}{\partial x_{1A}^2} & \frac{\partial^2 E}{\partial x_{1A} \partial y_{1A}} & \cdots & \frac{\partial^2 E}{\partial x_{1A} \partial x_{2A}} & \frac{\partial^2 E}{\partial x_{1A} \partial y_{2A}} & \cdots \\ \frac{\partial^2 E}{\partial y_{1A} \partial x_{1A}} & \frac{\partial^2 E}{\partial y_{1A}^2} & \cdots & \frac{\partial^2 E}{\partial y_{1A} \partial x_{2A}} & \frac{\partial^2 E}{\partial y_{1A} \partial y_{2A}} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \frac{\partial^2 E}{\partial x_{2A} \partial x_{1A}} & \frac{\partial^2 E}{\partial x_{2A} \partial y_{1A}} & \cdots & \frac{\partial^2 E}{\partial x_{2A}^2} & \frac{\partial^2 E}{\partial x_{2A} \partial y_{2A}} & \cdots \\ \frac{\partial^2 E}{\partial y_{2A} \partial x_{1A}} & \frac{\partial^2 E}{\partial y_{2A} \partial y_{1A}} & \cdots & \frac{\partial^2 E}{\partial y_{2A} \partial x_{2A}} & \frac{\partial^2 E}{\partial y_{2A}^2} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{bmatrix} \quad (7)$$

The Hessian can again be separated in terms of the contributions from energy terms and distance terms. For energy terms, it is useful to divide the Hessian into four square blocks (visible in eqn.-(7)).

The upper-left block consists of terms that only depend on  $\vec{r}_1$ . Similarly, the lower-right block only depends on  $\vec{r}_2$ . The upper-right and lower-left blocks depend on both.

Using eqn.-(6), the upper-left block can be written as:

$$\begin{aligned} (1 - 2\kappa_e E_2) \nabla_{\vec{r}_1}^2 E_1 + \kappa_e \nabla_{\vec{r}_1}^2 (E_1^2) \\ \nabla_{\vec{r}_1}^2 (E_1^2) = \begin{bmatrix} \frac{\partial^2 E_1^2}{\partial x_{1A}^2} & \frac{\partial^2 E_1^2}{\partial x_{1A} \partial y_{1A}} & \cdots \\ \frac{\partial^2 E_1^2}{\partial y_{1A} \partial x_{1A}} & \frac{\partial^2 E_1^2}{\partial y_{1A}^2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \\ = 2 \times \begin{bmatrix} (\frac{\partial E_1}{\partial x_{1A}})^2 + E_1 \frac{\partial^2 E_1}{\partial x_{1A}^2} & \frac{\partial E_1}{\partial x_{1A}} \frac{\partial E_1}{\partial y_{1A}} + E_1 \frac{\partial^2 E_1}{\partial x_{1A} \partial y_{1A}} & \cdots \\ \frac{\partial E_1}{\partial y_{1A}} \frac{\partial E_1}{\partial x_{1A}} + E_1 \frac{\partial^2 E_1}{\partial y_{1A} \partial x_{1A}} & (\frac{\partial E_1}{\partial y_{1A}})^2 + E_1 \frac{\partial^2 E_1}{\partial y_{1A}^2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (8) \\ = 2(\nabla_{\vec{r}_1} E_1)^T (\nabla_{\vec{r}_1} E_1) + 2E_1 \nabla_{\vec{r}_1}^2 E_1 \end{aligned}$$

which gives,

$$\begin{aligned} (1 - 2\kappa_e E_2) \nabla_{\vec{r}_1}^2 E_1 + 2\kappa_e E_1 \nabla_{\vec{r}_1}^2 E_1 + 2\kappa_e (\nabla_{\vec{r}_1} E_1)^T (\nabla_{\vec{r}_1} E_1) \\ = (1 - 2\kappa_e E_2 + 2\kappa_e E_1) \nabla_{\vec{r}_1}^2 E_1 + 2\kappa_e (\nabla_{\vec{r}_1} E_1)^T (\nabla_{\vec{r}_1} E_1) \end{aligned} \quad (9)$$

Similarly for the lower-right square, all the terms depend only on  $\vec{r}_2$ , so the Hessian of the energy terms can be written as

$$\begin{aligned} & (1 - 2\kappa_e E_1) \nabla_{\vec{r}_2}^2 E_2 + 2\kappa_e E_2 \nabla_{\vec{r}_2}^2 E_2 + 2\kappa_e (\nabla_{\vec{r}_2} E_2)^T (\nabla_{\vec{r}_2} E_2) \\ & = (1 - 2\kappa_e E_1 + 2\kappa_e E_2) \nabla_{\vec{r}_2}^2 E_2 + 2\kappa_e (\nabla_{\vec{r}_2} E_2)^T (\nabla_{\vec{r}_2} E_2) \end{aligned} \quad (10)$$

Now, the upper-right square of the original Hessian, the terms depend on both  $\vec{r}_1$  and  $\vec{r}_2$ . Looking at eqn.-(6), it is clear that only the term  $-2\kappa_e E_1 E_2$  depends on both. Derivatives of other terms will be zero. So, the upper-right square can be written as:

$$\begin{aligned} & -2\kappa_e \begin{bmatrix} \frac{\partial E_1}{x_{1A}} \frac{\partial E_2}{x_{2A}} & \frac{\partial E_1}{x_{1A}} \frac{\partial E_2}{y_{2A}} & \cdots \\ \frac{\partial E_1}{y_{1A}} \frac{\partial E_2}{x_{2A}} & \frac{\partial E_1}{y_{1A}} \frac{\partial E_2}{y_{2A}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \\ & = -2\kappa_e (\nabla_{\vec{r}_1} E_1)^T (\nabla_{\vec{r}_2} E_2) \end{aligned}$$

Similarly, the lower-left square can be written as:

$$= -2\kappa_e (\nabla_{\vec{r}_2} E_2)^T (\nabla_{\vec{r}_1} E_1)$$

Putting these together, the Hessian from the energy terms:

$$\begin{bmatrix} (1 - 2\kappa_e E_2 + 2\kappa_e E_1) \nabla_{\vec{r}_1}^2 E_1 + 2\kappa_e (\nabla_{\vec{r}_1} E_1)^T (\nabla_{\vec{r}_1} E_1) & -2\kappa_e (\nabla_{\vec{r}_1} E_1)^T (\nabla_{\vec{r}_2} E_2) \\ -2\kappa_e (\nabla_{\vec{r}_2} E_2)^T (\nabla_{\vec{r}_1} E_1) & (1 - 2\kappa_e E_1 + 2\kappa_e E_2) \nabla_{\vec{r}_2}^2 E_2 + 2\kappa_e (\nabla_{\vec{r}_2} E_2)^T (\nabla_{\vec{r}_2} E_2) \end{bmatrix}$$

Realising that  $\nabla_{\vec{r}_1}^2 E_1$  and  $\nabla_{\vec{r}_2}^2 E_2$  are the molecular Hessians, which can be calculated with a QM method (preferably a fast semi-empirical method) allows us to calculate the Hessian of energy terms.

Now, the distance term is  $\kappa_d (d - d_i)^2$ . If we express the concatenation of the  $\vec{r}_1$  and  $\vec{r}_2$  as  $\vec{z}$  then,

$$\nabla_{\vec{r}_1, \vec{r}_2} d = \frac{1}{d} \begin{bmatrix} +\mathbf{I}_n & -\mathbf{I}_n \\ -\mathbf{I}_n & +\mathbf{I}_n \end{bmatrix} (\vec{z})^T = \frac{1}{d} \mathbf{A} (\vec{z})^T$$

So Hessian of distance will be

$$\nabla^2 d = \frac{1}{d} (\mathbf{A} - (\nabla d)^T (\nabla d))$$

Now, Hessian of distance terms is:

$$\nabla^2 (d^2) - 2d_i \nabla^2 d$$

$$\begin{aligned}
\nabla^2(d^2) &= \begin{bmatrix} \frac{\partial^2 d^2}{\partial x_{1A}^2} & \frac{\partial^2 d^2}{\partial x_{1A} \partial y_{1A}} & \cdots \\ \frac{\partial^2 d^2}{\partial y_{1A} \partial x_{1A}} & \frac{\partial^2 d^2}{\partial y_{1A}^2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \\
&= 2 \begin{bmatrix} (\frac{\partial d}{\partial x_{1A}})^2 + d \frac{\partial^2 d}{\partial x_{1A}^2} & \frac{\partial d}{\partial x_{1A}} \frac{\partial d}{\partial y_{1A}} + d \frac{\partial^2 d}{\partial x_{1A} \partial y_{1A}} & \cdots \\ \frac{\partial d}{\partial y_{1A}} \frac{\partial d}{\partial x_{1A}} + d \frac{\partial^2 d}{\partial y_{1A} \partial x_{1A}} & (\frac{\partial d}{\partial y_{1A}})^2 + d \frac{\partial^2 d}{\partial y_{1A}^2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (11) \\
&= 2(\nabla d)^T (\nabla d) + 2\nabla^2 d
\end{aligned}$$

So putting it together gives the Hessian of distance terms:

$$2(\nabla d)^T (\nabla d) + 2\nabla^2 d (1 - 2d_i)$$

Summing the Hessian contributions from the energy terms and the distance terms provides the Hessian of the BITSS energy.

### 0.1.3 Minimisation

BITSS optimisation requires the BITSS potential to be minimised. With the gradient and Hessian available, we can optimise by using RFO steps, and update the Hessian with BFGS/Bofill formula when not calculating Hessian analytically.