



GOpt: A Package for Optimizing Molecular Geometries

Derrick X.T. Yang

McMaster University, Hamilton, ON, Canada

Problem

Transition State (Saddle Point) is **hard** to find.

$$H_{tot} = (T_n + V_n) + T_e + V_{ne} + V_{ee} = H_n + H_e$$
$$H_e \Psi(r, R) = E_e(R) \Psi(r, R)$$

R – nuclear coordinates

$E_e(R)$ – the electronic energy

$V_n(R)$ – nuclei-nuclei interaction

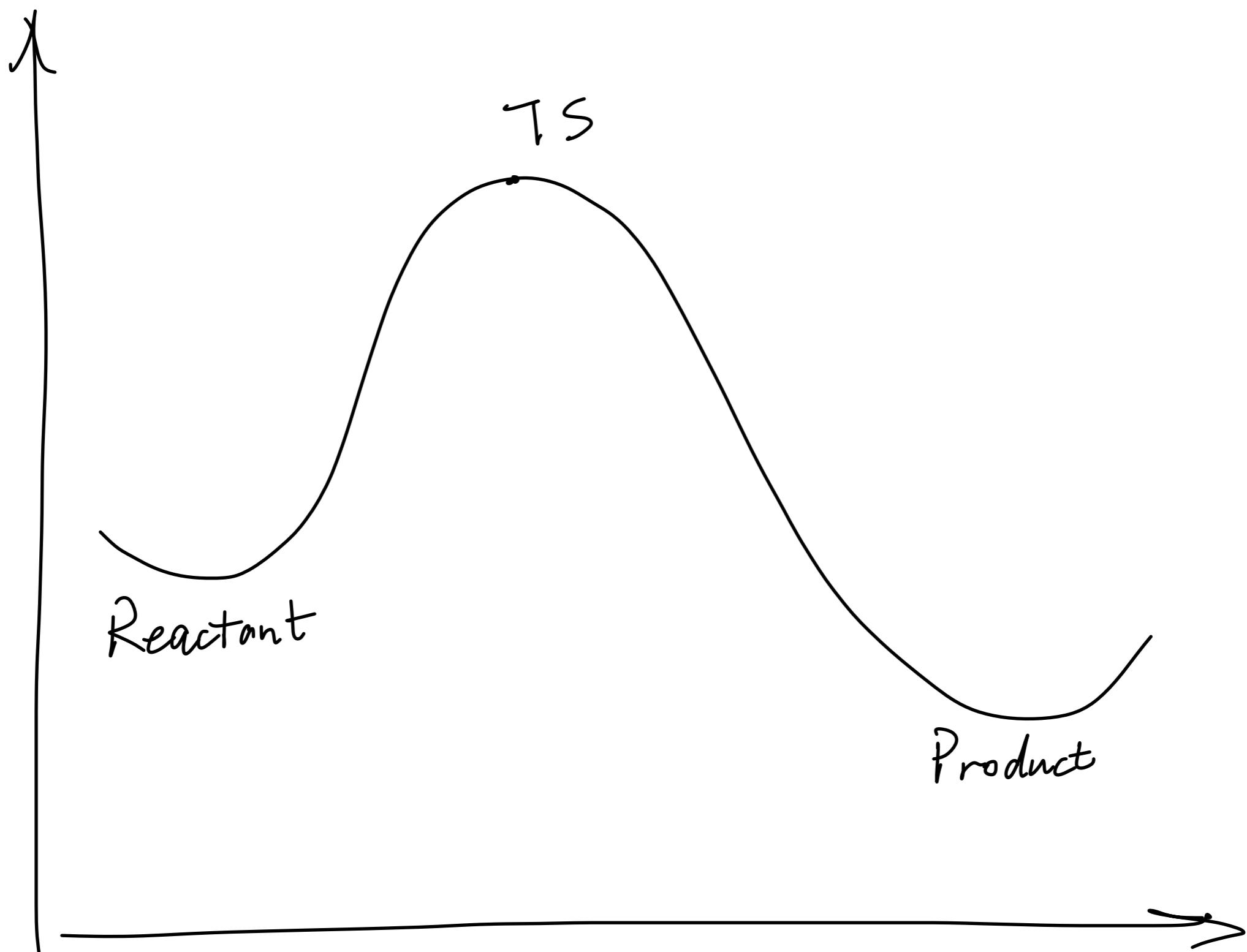
$E(R) = V_n(R) + E_e(R)$ is the PES

Gradient: $\nabla E = \left(\frac{\partial E}{\partial x_1}, \dots, \frac{\partial E}{\partial x_n} \right)$

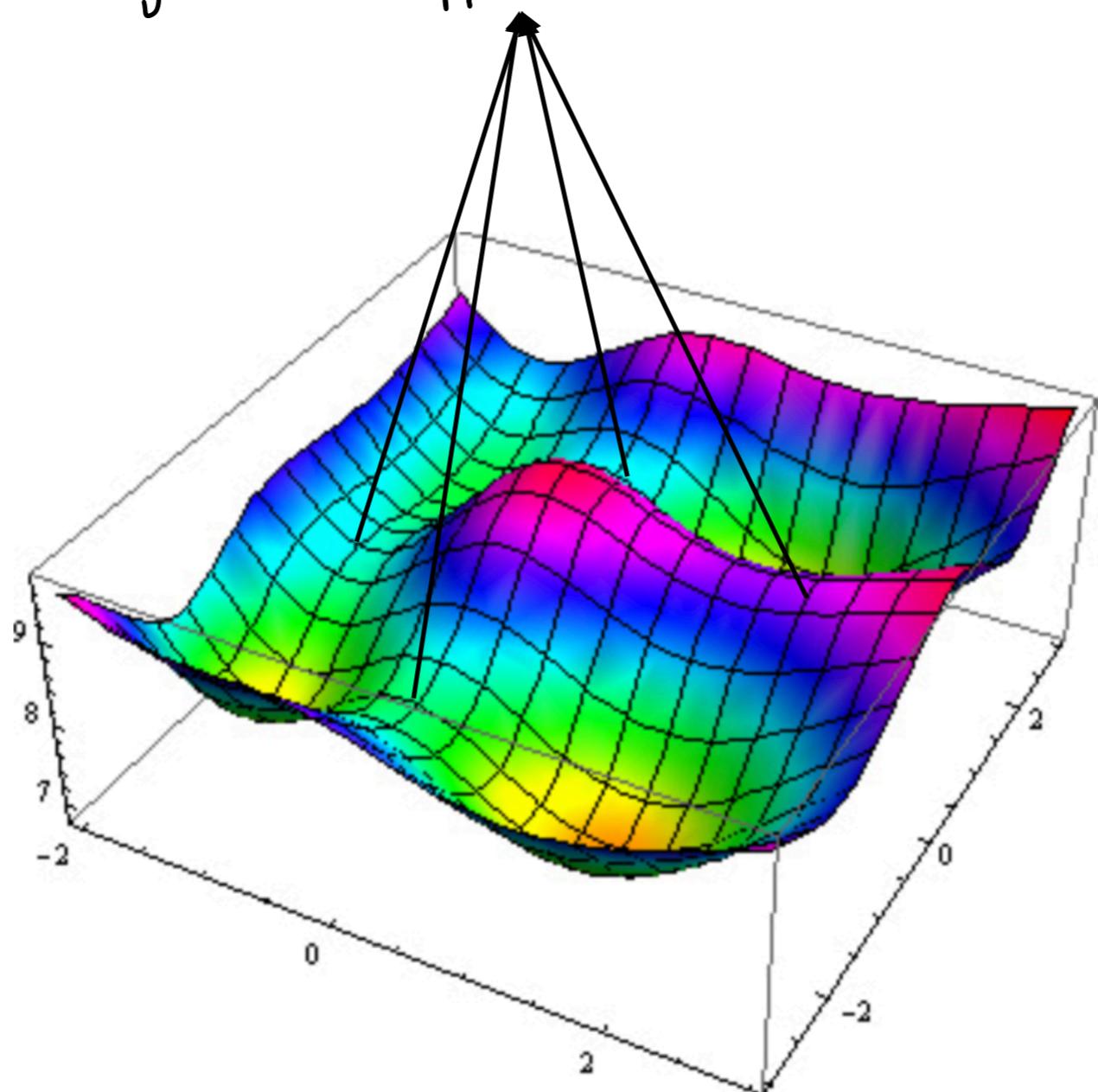
Hessian: $H_{ij}(E) = \frac{\partial^2 E}{\partial x_i \partial x_j}$

$\nabla E = 0$ Stationary Point

The eigenvalues of Hessian represent the curvature



four different TS



Purpose

To Build a robust and user-friendly package for
geometry optimization

Maths

Coordinate System

- Cartesian Coordinate system
- Conventional Internal Coordinate system
- Robust Redundant Coordinate system

Coordinate Transfer

Transformation from Cartesian to Internal Coordinates:

$$b_{ij} = \frac{\partial q_i}{\partial x_j}$$

$$B\delta x = \delta q$$

Manifold projection method to locate the
realizable cartesian coordinates:

$$x(q^{target}) = \arg \underbrace{\min}_x (q(x) - q^{target})^T W (q(x) - q^{target})$$

- Jacobian matrix from Cartesian to redundant internal coordinates,

$$b_{ij} = \frac{\partial q_i}{\partial x_j}$$

$$\delta q = B \delta x$$

$$g_q = (B^T)^+ g_x$$

$$H_q = (B^T)^+ (H_x - K) B^+$$

$$\delta x = B^+ \delta q$$

$$g_x = B^T g_q$$

$$H_x = B^T H_q B + K$$

Reduced Coordinates

- Most chemical reaction can be characterized by a few key internal coordinates.
- Select key internal coordinates those are associated with bond-breaking and bond-forming process.

Delocalized Reduced Space

Step 1: choose the $3N_{\text{atoms}} - 6(5)$ non-singular vectors

$$a^{(i)} = [a_1^i, a_2^i, \dots, a_{N_{int}}^{(i)}]^T \quad i = 1, 2, \dots, 3N_{\text{atoms}} - 6(5)$$

Step 2: $b^{(j)} = P\hat{e}^{(j)} = BB^+\hat{e}^{(j)} \quad j = 1, 2, \dots, R$

Form a Grammian by $G_{ij} = b_i^T \cdot b_j$

Use the eigenvectors to form reduced space

$$\tilde{v}^{(j)} \quad j = 1, 2, \dots, R$$

Step 3: The non-reduced space: $d^{(j)} = (I - P_{reduced})a^{(j)}$

Form a Grammian by $F_{ij} = d_i^T \cdot d_j$

Use the eigenvectors to form non-reduced space

$$\tilde{v}^{(R+j)} \quad j = 1, 2, \dots, 3N_{atoms} - 6(5) - R$$

Step 4: Form V space: $\tilde{v}^{(j)} + \tilde{v}^{(R+j)}$

Conversion of information

Cartesian \iff *Internal*

$$b_{ij} = \frac{\partial q_i}{\partial x_j}$$

$$\delta q = B \delta x$$

$$\delta x = B^+ \delta q$$

$$g_x = B^T g_q$$

$$g_q = (B^T)^+ g_x$$

$$H_x = B^T H_q B + K$$

$$H_q = (B^T)^+ (H_x - K) B^+$$

$$K_{jh} = \sum_i [g_q]_i b_{ijk}$$

Internal \iff *Reduced Internal*

$$V_{ij} = \frac{\partial q_i}{\partial v_j}$$

$$\Delta v = V^T \Delta q$$

$$\Delta q = V \Delta v$$

$$g_v = V^T g_q$$

$$g_q = V g_v$$

$$H_v = V^T H_q V$$

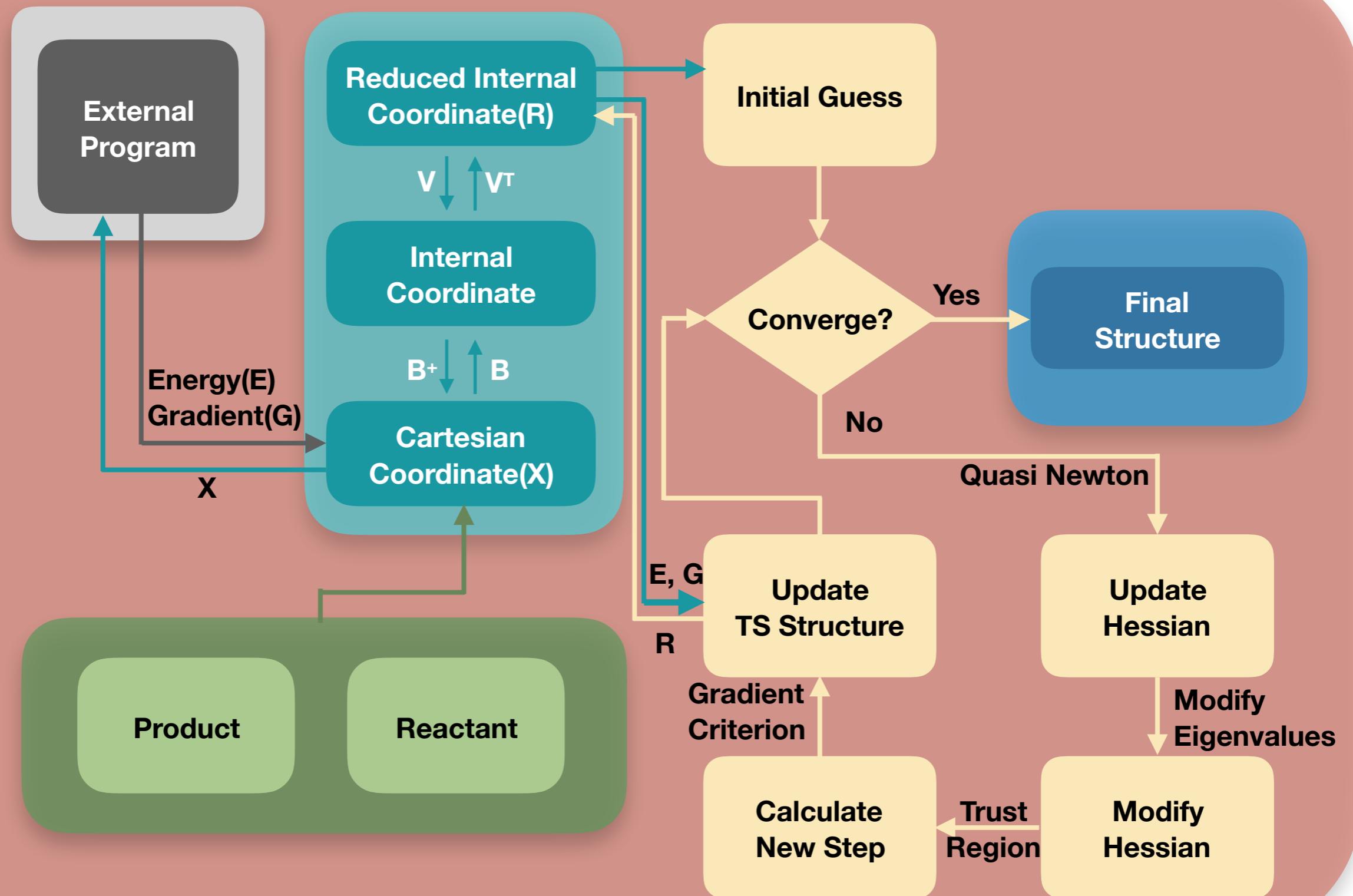
$$H_q = V H_v V^T$$

Optimization

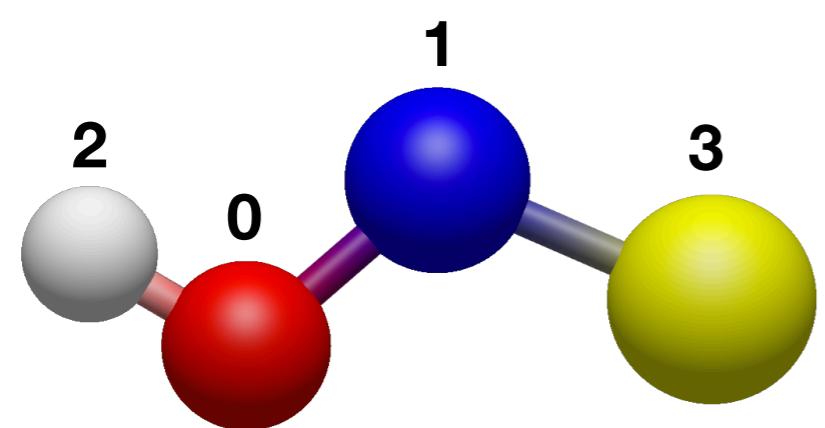
- Quasi Newton method: BFGS, SR1, PSB etc.
- Trust radius method:

trust-region image potential(TRIM)

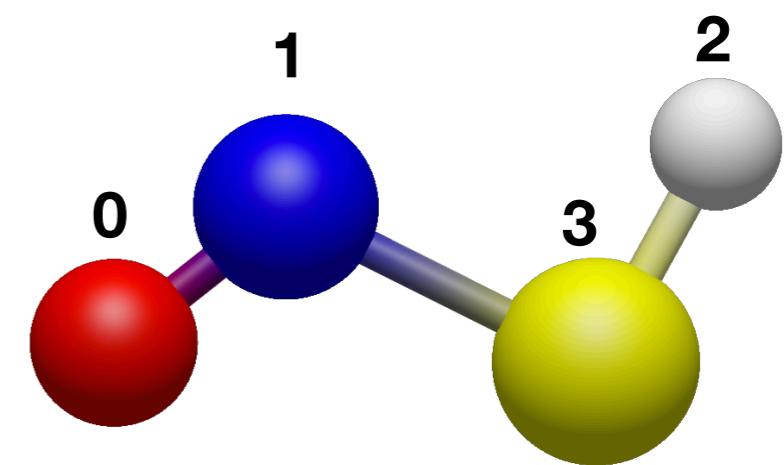
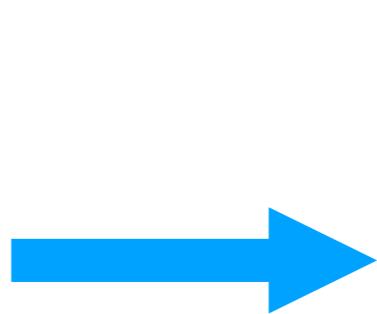
Rational function optimization(RFO)



Codes



HONS



ONSH

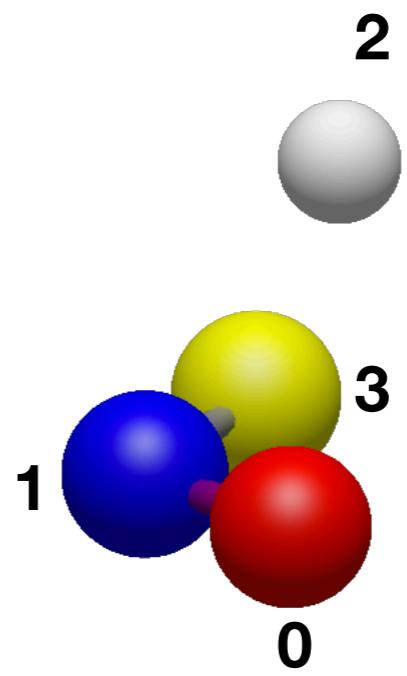
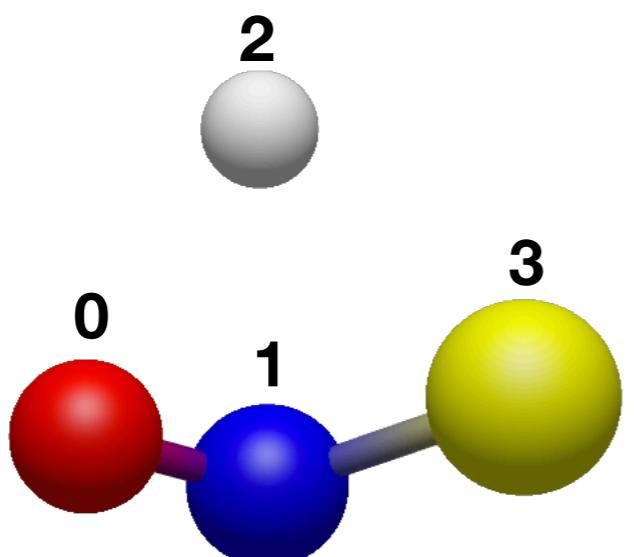
- Generate Coordinates

```
from GOpt.ts_construct import TSConstruct
from pkg_resources import Requirement, resource_filename

# construct TS initial guess instance
ts_const = TSConstruct.from_file('HONS.xyz', 'ONSH.xyz')
ts_const.auto_generate_ts(ratio=0.5, mode='mix')
ts_init = ts_const.ts

#inspect ic
ts_init.ic

[Bond-(0, 1)-(2.3609172959792777),
 Bond-(0, 2)-(4.139676632099312),
 Bond-(1, 3)-(3.2792707809283166),
 Angle-(1, 0, 2)-(0.983115802564365),
 Angle-(0, 1, 3)-(1.9997530497727223),
 Dihed-(2, 0, 1, 3)-(1.430751046555983),
 Bond-(2, 3)-(4.465214373482998),
 Angle-(1, 3, 2)-(0.8729279563567633),
 Dihed-(0, 1, 3, 2)-(-1.4947847271876917)]
```



Initial Guess

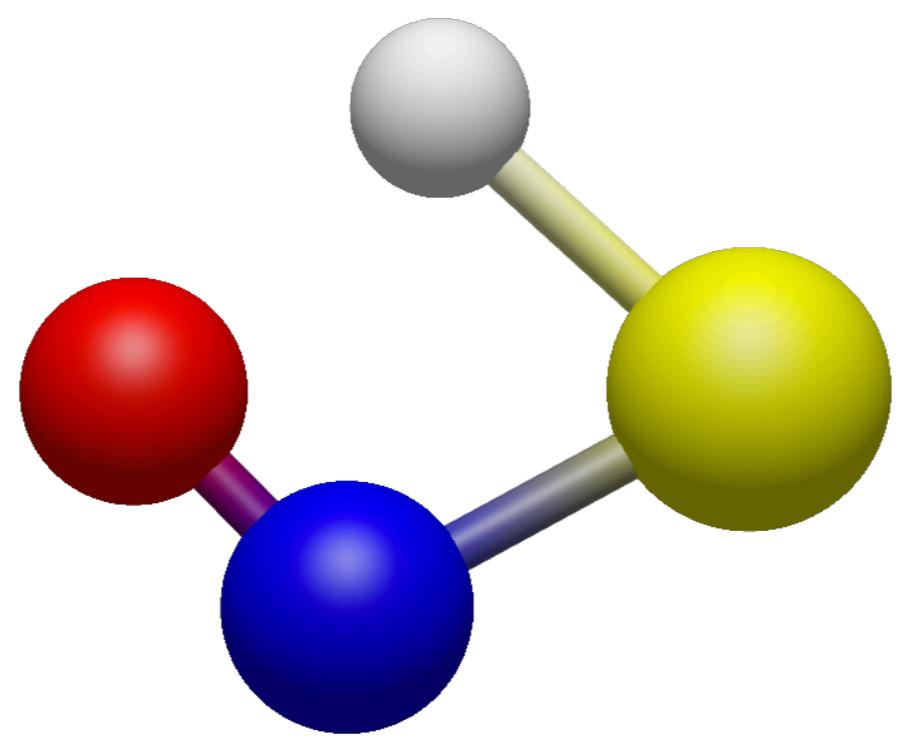
- Select Key iC

```
# select key internal coordinates
ts_init.select_key_ic(1, 6)
# V matrix would be automatically generated
# You can inspect V matrix by property instance.vspace
ts_init.ic

[Bond-(0, 2)-(4.139676632099312),
Bond-(2, 3)-(4.465214373482998),
Bond-(1, 3)-(3.2792707809283166),
Angle-(1, 0, 2)-(0.983115802564365),
Angle-(0, 1, 3)-(1.9997530497727223),
Dihed-(2, 0, 1, 3)-(1.430751046555983),
Bond-(0, 1)-(2.3609172959792777),
Angle-(1, 3, 2)-(0.8729279563567633),
Dihed-(0, 1, 3, 2)-(-1.4947847271876917)]
```

- Select optimizer and run

```
# initialize optimizer
solver = OptLoop(structure=ts_init, task='saddle')
solver.start_optimization(iterations=40,
                           neg_h=1,
                           result='ts.xyz',
                           output='ts_log.xyz')
```



Transition State

Optimization Path

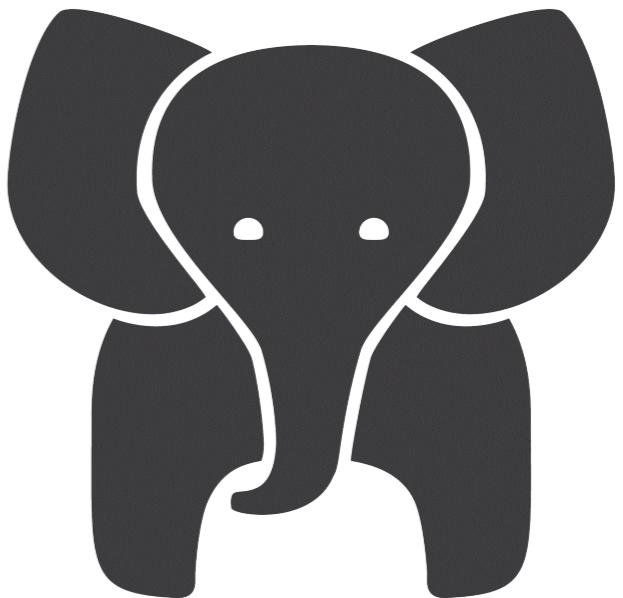
Conclusions

- GOpt is an open-source package, and it would be part of HORTON 3.
- GOpt uses reduced internal coordinates to refine chemical information of chemical reactions.
- GOpt is also a standalone general-purpose geometry optimization package. It can work along with any quantum chemistry software.

Appreciations



**NSERC
CRSNG**



HORTON