1 Introduction to the program Tnum-Tana

The derivation or computation of the kinetic energy operator (KEO) using curvilinear coordinates for large systems is a difficult task. With this program, you can calculate the KEO of a given system by using a numerical approach (Tnum) or an analytical approach (Tana).

1.1 Tnum branch

The Tnum program enables ones to calculate numerically and exactly kinetic energy operator (KEO), $\hat{T}$, or the metric tensor (covariant, $\mathbf{g}$, or contravariant, $\mathbf{G}$, components) using several kinds of curvilinear coordinates for a molecular system of any size, without build in limitation. It can deal with reduced dimensionality KEO (rigid or flexible constraints or other approximations).

The molecular system has N_{at} atoms. Let $\mathbf{Q}_{dyn} = [Q^1_{dyn}, Q^2_{dyn}, \cdots, Q^{n+m}_{dyn}]$ be the curvilinear coordinates associated with the internal deformation. Where n is the number of active coordinates and m is the number of inactive coordinates, $n + m = 3N_{at} - 6$. n can be smaller than $3N_{at} - 6$ when reduced dimensionality is used, i.e. the quantum dynamics is performed only with n coordinates. In full dimensionality $m = 0$. The exact KEO, including the deformation, the Coriolis coupling and the overall rotation is given by,

$$\hat{T} = \underbrace{\hat{T}_{\text{def}}}_{\text{Deformation}} + \underbrace{\hat{T}_{\text{cor}}}_{\text{Corlioris}} + \underbrace{\hat{T}_{\text{rot}}}_{\text{Rotation}}$$

There are several equivalent expressions for the deformation term of the KEO, $\hat{T}_{\text{def}}$, see Eqs.(1-3):

$$\hat{T}_{\text{def}} = -\frac{\hbar^2}{2}\sum_{i,j}\Sigma^{ij}(\mathbf{Q}_{dyn})\frac{\partial^2 \mathbf{g}}{\partial Q^i_{dyn}\partial Q^j_{dyn}} - \frac{\hbar^2}{2}\sum_{i,j}\left(\Sigma^{ij}(\mathbf{Q}_{dyn})\frac{\partial ln\left(\rho(\mathbf{Q}_{dyn})\right)}{\partial Q^i_{dyn}} + \frac{\partial \Sigma^{ij}(\mathbf{Q}_{dyn})}{\partial Q^i_{dyn}}\right)\frac{\partial \mathbf{g}}{\partial Q^j_{dyn}} + v_{\text{ext}}(\mathbf{Q}_{dyn}) \tag{1}$$

or

$$\hat{T}_{\text{def}} = \sum_{i\le j} f_2^{ij}(\mathbf{Q}_{dyn})\frac{\partial^2 \mathbf{g}}{\partial Q^i_{dyn}\partial Q^j_{dyn}} + \sum_j f_1^j(\mathbf{Q}_{dyn})\frac{\partial \mathbf{g}}{\partial Q^j_{dyn}} + v_{\text{ext}}(\mathbf{Q}_{dyn}) \tag{2}$$

or

$$\hat{T}_{\text{def}} = -\frac{\hbar^2}{2}\sum_{i,j}\frac{1}{\rho(\mathbf{Q}_{dyn})}\frac{\partial \mathbf{g}}{\partial Q^i_{dyn}}\left(\Sigma^{ij}(\mathbf{Q}_{dyn})\rho(\mathbf{Q}_{dyn})\right)\frac{\partial \mathbf{g}}{\partial Q^j_{dyn}} + v_{\text{ext}}(\mathbf{Q}_{dyn})\,. \tag{3}$$

The Coriolis term, $\hat{T}_{\text{cor}}$, is given by

$$\hat{T}_{\text{cor}} = -i\hbar\sum_{j,\alpha}\Gamma^{\alpha j}(\mathbf{Q}_{dyn})\frac{\partial \mathbf{g}}{\partial Q^j_{dyn}}\ddot{\mathcal{P}}_\alpha - i\frac{\hbar}{2}\sum_{j,\alpha}\left(\Gamma^{\alpha j}(\mathbf{Q}_{dyn})\frac{\partial ln(\rho(\mathbf{Q}_{dyn}))}{\partial Q^j_{dyn}} + \frac{\partial \Gamma(\mathbf{Q}_{dyn})}{\partial Q^j_{dyn}}\right)\ddot{\mathcal{P}}_\alpha\,.$$

The rotational term, $\hat{T}_{\text{rot}}$, is given by

$$\hat{T}_{\text{rot}} = \frac{1}{2}\sum_{\alpha,\beta}\mu^{\alpha\beta}(\mathbf{Q}_{dyn})\ddot{\mathcal{P}}_\alpha\ddot{\mathcal{P}}_\beta\,.$$

Where $i, j = 1, \cdots, n$, $\alpha, \beta = x, y, z$ and $d\tau_{\text{def}} = \rho(\mathbf{Q}_{dyn})dQ^1_{dyn}dQ^2_{dyn}\cdots Q^n_{dyn}$.

1.2 Tana branch

The Tana program is an extension of the Tnum code. The two programs share the same data structure. The branch enables to compute the analytical expression of the KEO using the polyspherical coordinates. This expression is expression as,

$$\hat{T} = \frac{1}{2}\sum_{i,j=1}^{N_{at}-1} M_{ij}\hat{\vec{P}}_i^\dagger\cdot\hat{\vec{P}}_j$$

where $\mathbf{M}$ is the mass-matrix