Electronic Supplementary Material (ESI) for Physical Chemistry Chemical Physics.

Supporting Information: Reduced-dimensional surface hopping with offline–online computations

Zachary Morrow,[†] Hyuk-Yong Kwon,[‡] C. T. Kelley,[*,†] and Elena Jakubikova[*,‡]

[†]*Department of Mathematics, North Carolina State University, Raleigh, NC 27695*

[‡]*Department of Chemistry, North Carolina State University, Raleigh, NC 27695*

E-mail: tim_kelley@ncsu.edu; ejakubi@ncsu.edu

Contents

1 Interpolation

This section has been published as Supporting Information before,[S1] but we include it here for completeness and for the reader's convenience.

Sparse interpolation is a linear combination of full-tensor interpolation. Therefore, we begin by defining the specific one-dimensional rules, followed by the full-tensor interpolation operators.

1.1 One dimension: trigonometric interpolation

In one dimension, we have $m(l) = 3^l$ nodes at each level $l \geq 0$.[S2] The trigonometric interpolation nodes are

$$x_j^l = \frac{j-1}{m(l)}, \qquad 1 \leq j \leq m(l), \quad l \geq 0\,.$$

The basis functions are more complicated. Trigonometric interpolation with $2n+1$ points can resolve all modes up to mode n, so each frequency will have two associated nodes. Therefore, we begin by defining

$$\sigma(j) = \begin{cases} j/2, & j \text{ even} \\ -(1+j)/2, & j \text{ odd} \end{cases},$$

$$\phi_j(x) = \exp(2\pi \mathrm{i} \cdot \sigma(j) \cdot x),$$

where $\mathrm{i}^2 = -1$ (in Roman font). The one-dimensional interpolant of $f : [0,1] \to \mathbb{R}$ is

$$\mathcal{U}^l[f](x) = \sum_{j=1}^{m(l)} \mathrm{Re}(c_j^l\, \phi_j(x))$$

where the interpolation coefficients c_j^l are

$$c_j^l = \frac{1}{m(l)} \sum_{p=1}^{m(l)} f(x_p^l)\, \phi_j^*(x_p^l), \qquad 1 \le j \le m(l),$$

which is a normalized discrete Fourier transform. Here, ϕ_j^* denotes the complex conjugate of ϕ_j. Trigonometric interpolation converges uniformly as $l \to \infty$ as long as the target function f is periodic and uniformly continuous, and the convergence rate is faster as f gets smoother.[S2]

1.2 One dimension: Clenshaw–Curtis

The number of Clenshaw–Curtis points on level $l \ge 0$ is[S3,S4]

$$m(l) = \begin{cases} 1, & l = 0 \\ 2^l + 1, & l > 0 \end{cases}$$

and the points themselves are given by

$$x_j^l = \begin{cases} 0, & l = 0 \\ -\cos\left(\frac{j-1}{m(l)-1}\,\pi\right), & 1 \le j \le m(l),\ l > 0 \end{cases}.$$

The basis functions are Lagrange polynomials, defined as

$$\phi_i^l(x) = \prod_{j=1, j\neq i}^{m(l)} \frac{x - x_j}{x_i - x_j}, \qquad i = 1, \ldots, m(l)$$

which have the convenient property that $\phi_i(x_j) = \delta_{ij}$. With that property, it readily follows that the Clenshaw–Curtis interpolant of $f : [-1, 1] \to \mathbb{R}$ is

$$\mathcal{U}^l[f] = \sum_{j=1}^{m(l)} c_j^l \phi_j^l(x) \qquad c_j^l = f(x_j^l)\,.$$

Polynomial interpolation is exact for all polynomials of degree up to $m(l) - 1$. Clenshaw–Curtis interpolation converges uniformly as $l \to \infty$ as long as f is uniformly continuous, and the convergence rate gets faster as f becomes smoother.[S2,S5]

1.3 Full tensor

Let $\boldsymbol{i} \in \mathbb{N}_0^d$ be a multi-index drawn from the natural numbers including zero. For full-tensor interpolation, we have $m(i_k)$ points in dimension k, and we take the grid as the tensor product of one-dimensional grids at level i_k:

$$\{\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}}\} = \bigotimes_{k=1}^{d} \{x_{j_k}^{i_k}\},$$

which gives a lattice structure as the grid. Let

$$\phi_{\boldsymbol{j}}(\boldsymbol{x}) = \prod_{k=1}^{d} \phi_{j_k}(x_k), \qquad \boldsymbol{j} \in \mathbb{N}_0^d,\ \boldsymbol{x} \in \mathbb{R}^d,$$

where ϕ_{j_k} are the one-dimensional basis functions of your interpolation rule. The full-tensor interpolant of $f(\boldsymbol{x})$ is

$$\mathcal{U}^{\boldsymbol{i}}[f](\boldsymbol{x}) = \sum_{j_1=1}^{m(i_1)} \cdots \sum_{j_d=1}^{m(i_d)} c_{\boldsymbol{j}}^{\boldsymbol{i}}\, \phi_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x}), \tag{1}$$

where the full-tensor interpolation coefficients $c_{\boldsymbol{j}}^{\boldsymbol{i}}$ are

$$c_{\boldsymbol{j}}^{\boldsymbol{i}} = \frac{1}{\prod_{k=1}^{d} m(i_k)} \sum_{p_1=1}^{m(i_1)} \cdots \sum_{p_d=1}^{m(i_d)} f(\boldsymbol{x}_{\boldsymbol{p}}^{\boldsymbol{i}})\, \phi_{\boldsymbol{j}}^{*}(\boldsymbol{x}_{\boldsymbol{p}}^{\boldsymbol{i}}) \qquad \text{(trigonometric)}$$

$$c_{\boldsymbol{j}}^{\boldsymbol{i}} = f(\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}}) \qquad \text{(polynomial)}$$

To express $\mathcal{U}^{\boldsymbol{i}}[f]$ in adjoint form, we note that a linear operator $\mathcal{L}$ gets applied to the data $f(\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}})$ to produce $c_{\boldsymbol{j}}^{\boldsymbol{i}}$. For polynomial interpolation, $\mathcal{L}$ is the identity operator; for trigonometric interpolation, $\mathcal{L}$ is the discrete Fourier transform. Thus, for polynomial interpolation,

the adjoint form is identical to the form already presented. For trigonometric interpolation, however, we get

$$\mathcal{U}^{\boldsymbol{i}}[f] = \langle \boldsymbol{c}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle = \langle \mathcal{L}\boldsymbol{f}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle = \langle \boldsymbol{f}, \mathcal{L}^* \boldsymbol{\phi}(\boldsymbol{x}) \rangle = \langle \boldsymbol{f}, \mathcal{L}\boldsymbol{\phi}(\boldsymbol{x}) \rangle$$

where the last equality is justified since we take only the real part of the interpolant. So the adjoint functions $\psi_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x})$ are a discrete Fourier transform of the basis functions $\phi_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x})$, and we get

$$\mathcal{U}^{\boldsymbol{i}}[f](\boldsymbol{x}) = \sum_{j_1=1}^{m(i_1)} \cdots \sum_{j_d=1}^{m(i_d)} f(\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}})\, \psi_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x})\,.$$

1.4 Sparse interpolation

We first define the following relation between multi-indices:

$$\boldsymbol{i} \leq \boldsymbol{j} \quad \Longleftrightarrow \quad i_k \leq j_k \quad \forall 1 \leq k \leq d\,.$$

The most general form of a sparse interpolation operator is

$$G_\Theta = \sum_{\boldsymbol{i} \in \Theta} t_{\boldsymbol{i}}\, \mathcal{U}^{\boldsymbol{i}}$$

where $\mathcal{U}^{\boldsymbol{i}}$ is defined in Equation (1), Θ is a lower set,[1] and

$$\sum_{\boldsymbol{i} \leq \boldsymbol{j},\ \boldsymbol{j} \in \Theta} t_{\boldsymbol{j}} = 1, \qquad \forall \boldsymbol{i} \in \Theta.$$

To find Θ, we first begin with a desired exactness space $\Lambda(L)$, which is also lower set. The set $\Lambda(L)$ contains the frequencies/degrees for which G_Θ should be exact. In this paper, we have taken

$$\Lambda^{\boldsymbol{\alpha}}(L) = \{\boldsymbol{i} \in \mathbb{N}_0^d \mid \boldsymbol{\alpha} \cdot \boldsymbol{i} \leq L\} \tag{2}$$

[1] A set of multi-indices $S \subset \mathbb{N}_0^d$ is *lower* if and only if, for each $\boldsymbol{x} \in S$, we have $\{\boldsymbol{y} \in \mathbb{N}_0^d \mid \boldsymbol{y} \leq \boldsymbol{x}\} \subset S$.

where $\boldsymbol{\alpha}$ is the anisotropy vector. For the mixed-basis grid, we used $\boldsymbol{\alpha} = (1, 2, 2)$ on the trigonometric part and $\boldsymbol{\alpha} = \mathbf{1}$ on the polynomial part.[2] After defining $\Lambda(L)$, we construct the optimal Θ as[S2,S4]

$$\Theta_{opt}(L) = \{\boldsymbol{i} \in \mathbb{N}_0^d \mid \boldsymbol{m}(\boldsymbol{i} - \mathbf{1}) \in \Lambda(L)\} \qquad \text{(polynomial)}$$

$$\Theta_{opt}(L) = \{\boldsymbol{i} \in \mathbb{N}_0^d \mid (\boldsymbol{m}(\boldsymbol{i} - \mathbf{1}) + \mathbf{1})/2 \in \Lambda(L)\} \qquad \text{(trigonometric)}$$

where $(\boldsymbol{m}(\boldsymbol{i}))_k = m(i_k)$.

By defining the set of all sparse grid indices

$$\Theta_m = \bigcup_{\boldsymbol{i} \in \Theta} \{\boldsymbol{j} \in \mathbb{N}_0^d \mid \mathbf{1} \leq \boldsymbol{j} \leq \boldsymbol{m}(\boldsymbol{i})\}$$

and using either the internal or adjoint form of $\mathcal{U}^{\boldsymbol{i}}$, we can also write $G_\Theta[f]$ as

$$G_\Theta[f] = \sum_{\boldsymbol{j} \in \Theta_m} w_{\boldsymbol{j}} \phi_{\boldsymbol{j}}(\boldsymbol{x}) = \sum_{\boldsymbol{j} \in \Theta_m} f(\boldsymbol{x}_{\boldsymbol{j}}) \tilde{\psi}_{\boldsymbol{j}}(\boldsymbol{x})$$

where

$$w_{\boldsymbol{j}} = \sum_{\substack{\boldsymbol{i} \in \Theta, \\ \boldsymbol{j} \leq \boldsymbol{m}(\boldsymbol{i})}} t_{\boldsymbol{i}}\, c_{\boldsymbol{j}}^{\boldsymbol{i}}, \qquad \tilde{\psi}_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x}) = \sum_{\substack{\boldsymbol{i} \in \Theta, \\ \boldsymbol{j} \leq \boldsymbol{m}(\boldsymbol{i})}} t_{\boldsymbol{i}}\, \psi_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x})\,.$$

[2] We did not conjure up these values *a priori*; we iteratively arrived at them as a result of refining the grid (see Section 3 below).

2 Details of electronic structure calculations

All S_0 electronic structure calculations occur in Gaussian 16.We have included two example input files, corresponding to a linear and nonlinear structure. The difference between the two is that the `modredundant` section includes the `L` keyword for linear structures. For nonlinear structures, we use `A` and `D` instead of `L`.

All calculations of excitation energies and nonadiabatic coupling vectors occur in Orca as some of them employ open-shell ground state as a reference, for which the TD-DFT and CIS implementations in Orca are more suitable.

2.1 Example 1: linear structure (S_0)

```
%chk=chkfiles/scan_zmat_1.chk
%mem=32GB
%nprocshared=16
# stable=opt b3lyp/6-311g(d) pop=always integral=grid=ultrafine
scf=(xqc,vtl,maxconventionalcycles=512) symmetry=none

azomethane stability 1

0 1
 N      0.00000000      0.00000000      0.00000000
 N      1.21001503      0.00000000      0.00000000
 C      1.94413820      1.36087129      0.00000000
 C     -1.80000000      0.00000000      0.00000000
 H      3.01529271      1.16666079      0.00000000
 H      1.65967212      1.92152238     -0.89311136
 H      1.65967124      1.92152000      0.89311218
 H     -2.30534096      0.96713099      0.00000000
 H     -1.92067884     -0.57054140     -0.90903927
 H     -1.92068356     -0.57054030      0.90903849

--Link1--
%chk=chkfiles/scan_zmat_1.chk
%mem=32GB
%nprocshared=16
# opt=(newton,tight,modredundant,maxstep=1,calcall,maxcycles=30)
geom=check guess=read b3lyp/6-311g(d) pop=always integral=grid=ultrafine
scf=(xqc,vtl,maxconventionalcycles=512) symmetry=none

azomethane opt 1

0 1

B 1 4 F
D 1 2 3 5 F
L 2 1 4 3 F
D 8 4 1 3 F

--Link1--
%chk=chkfiles/scan_zmat_1.chk
%mem=32GB
%nprocshared=16
# stable=opt geom=check guess=read b3lyp/6-311g(d) pop=always
```

```
integral=grid=ultrafine scf=(xqc,vtl,maxconventionalcycles=512)
symmetry=none

azomethane stability 2

0 1

--Link1--
%chk=chkfiles/scan_zmat_1.chk
%mem=32GB
%nprocshared=16
# opt=(newton,tight,modredundant,maxstep=1,calcall,maxcycles=30) freq
guess=read geom=check b3lyp/6-311g(d) pop=always integral=grid=ultrafine
scf=(xqc,vtl,maxconventionalcycles=512) symmetry=none

azomethane opt 2

0 1

--Link1--
%chk=chkfiles/scan_zmat_1.chk
%mem=32GB
%nprocshared=16
# stable=opt geom=check guess=read b3lyp/6-311g(d) pop=always
integral=grid=ultrafine scf=(xqc,vtl,maxconventionalcycles=512)
symmetry=none

azomethane stability 3

0 1
```

2.2 Example 2: nonlinear structure (S_0)

```
%chk=chkfiles/scan_zmat_2.chk
%mem=32GB
%nprocshared=16
# stable=opt b3lyp/6-311g(d) pop=always integral=grid=ultrafine
scf=(xqc,vtl,maxconventionalcycles=512) symmetry=none

azomethane stability 1

0 1
 N      0.00000000      0.00000000      0.00000000
 N      1.23874400      0.00000000      0.00000000
 C      1.80434700      1.38280600      0.00000000
 C      0.00000000     -1.80000000      0.00000000
 H      2.89084000      1.30705300      0.00000000
 H      1.47283200      1.92607900     -0.88854700
 H      1.47283200      1.92607900      0.88854700
 H     -1.02105000     -2.18591100      0.00000000
 H      0.54010800     -2.04428300     -0.90535200
 H      0.54010800     -2.04428300      0.90535200

--Link1--
%chk=chkfiles/scan_zmat_2.chk
%mem=32GB
%nprocshared=16
# opt=(newton,modredundant,maxstep=1,calcall,maxcycles=30) geom=check
guess=read b3lyp/6-311g(d) pop=always integral=grid=ultrafine
scf=(xqc,vtl,maxconventionalcycles=512) symmetry=none

azomethane opt 1

0 1

B 1 4 F
D 1 2 3 5 F
D 4 1 2 3 F
A 2 1 4 F
D 8 4 1 2 F

--Link1--
%chk=chkfiles/scan_zmat_2.chk
%mem=32GB
%nprocshared=16
```

```
# stable=opt geom=check guess=read b3lyp/6-311g(d) pop=always
integral=grid=ultrafine scf=(xqc,vtl,maxconventionalcycles=512)
symmetry=none

azomethane stability 2

0 1

--Link1--
%chk=chkfiles/scan_zmat_2.chk
%mem=32GB
%nprocshared=16
# opt=(newton,modredundant,maxstep=1,calcall,maxcycles=30) freq
guess=read geom=check b3lyp/6-311g(d) pop=always integral=grid=ultrafine
scf=(xqc,vtl,maxconventionalcycles=512) symmetry=none

azomethane opt 2

0 1

--Link1--
%chk=chkfiles/scan_zmat_2.chk
%mem=32GB
%nprocshared=16
# stable=opt geom=check guess=read b3lyp/6-311g(d) pop=always
integral=grid=ultrafine scf=(xqc,vtl,maxconventionalcycles=512)
symmetry=none

azomethane stability 3

0 1
```

2.3 Example 3: excited states

The Orca input file `azo.inp` (with stability checks) is shown below:

```
!B3LYP 6-311g(d) TightSCF
%CIS
   HFNacme true
END
%SCF
   MaxIter 100
   HFType UHF
   STABPerform true
END
%TDDFT
   NROOTS   10
END
* xyzfile 0 1 geomfiles/azo.xyz
```

The file `geomfiles/azo.xyz` contains the geometry specified in Cartesian coordinates:

```
10
azomethane
 N          0.00000000    0.00000000    0.00000000
 N          1.22915061    0.00000000    0.00000000
 C          1.87275627    1.37687217    0.00000000
 C         -1.61279073    0.01401353   -0.00653607
 H          2.62923902    1.39536387   -0.79264170
 H          1.13534982    2.18398066   -0.12897983
 H          2.39160980    1.47760052    0.95776512
 H         -1.88615071    0.21731351    1.02800251
 H         -1.91964069    0.83289353   -0.66933412
 H         -1.97129227   -0.96664721   -0.33283514
```

3 Final refinement

Table S1 shows the parameters used in the mixed-basis grid after all refinement was completed. Table S2 contains the stationary points and energies (relative to the *trans-* minimum) for the final mixed-basis grid. These stationary points were calculated on the unrestricted S_0 PES. The energy relative to the global minimum is within ∼5% or 1 kcal/mol of the Gaussian-optimized value. Furthermore, since the predicted mixed-basis energies are all smaller than the Gaussian-optimized values, the relative barriers remain comparable.

Table S1: Specification of final grids.

Trig grid	Polynomial grid
$d = 3$	$d = 2$
$L = 5$	$L = 8$
$\boldsymbol{\alpha} = (1, 2, 2)$	$\boldsymbol{\alpha} = \mathbf{1}$

Table S2: Locations of minima and transition states, as well as energy barriers relative to the *trans-* minimum on the unrestricted S_0 surface.

Structure	Surrogate	Optimized $\boldsymbol{q}$	ΔE (kcal/mol)
trans- minimum	Gaussian	$[180.00, 122.20, 122.20, 1.46, 113.00]$	0
	mixed	$[-179.71, 121.91, 121.84, 1.46, 116.06]$	0
cis- minimum	Gaussian	$[0.00, -119.38, -60.33, 1.48, 120.21]$	10.48
	mixed	$[0.95, -118.43, -50.84, 1.49, 124.18]$	9.57
Inversion TS	Gaussian	$[-180.00, -97.52, 121.18, 1.39, 180.00]$	51.02
	mixed	$[-92.16, 123.10, 121.72, 1.40, 180.00]$	50.12
Torsion TS	Gaussian	$[-89.50, 145.77, 55.27, 1.47, 117.79]$	45.71
	mixed	$[-87.46, 138.92, 141.04, 1.46, 118.78]$	43.16

4 Animations

A suite of animations is in a ZIP archive as part of the ESI. Each of these animations used the mixed interpolation basis with the final level of refinement (7215 nodes).

Five of these animations, named `XQ_i.mp4`, where i ranges from 1 to 5, are animations of each geometry component at a slice chosen from a uniform distribution. Each file animates a different design variable over its domain while holding the other slice components constant. The purpose of these five animations is to demonstrate visually the Cartesian geometry change as a function of the design variables, particularly when the N^2–N^1–C^4 bond angle is larger than 180°. They also demonstrate the smoothly varying nature of $\boldsymbol{X}(\boldsymbol{q})$ during rotation, dissociation, and inversion.

The file `nearly_linear.mp4` motivates our definition of q_2 at nearly linear geometries. If we did not adjust for the changing dihedral in our definition of q_2, we would end up with a linear structure where the main dihedral rotation still produces changes in the geometry. Our definition of q_2 prevents multivalued geometries in the limit $q_5 \to 180$ by keeping the methyl group properly oriented.

Lastly, in `FSSH.mp4`, we animate one surface hopping trajectory. Hops occur at $t = 150$ fs ($S_1 \to S_0$), $t = 151$ fs ($S_0 \to S_1$), $t = 155$ fs ($S_1 \to S_0$), $t = 410$ fs ($S_0 \to S_1$), and $t = 445$ fs ($S_1 \to S_0$). The purpose of this animation is to show the reconstruction of the geometry as part of the MD run itself, rather than as isolated investigations of $\boldsymbol{X}(\boldsymbol{q})$. It also demonstrates relaxation to $q_1 \approx \pm 90$ on S_1, followed by relaxation to a minimum on S_0 after hopping.

References

(S1) Morrow, Z.; Kwon, H.-Y.; Kelley, C. T.; Jakubikova, E. Efficient Approximation of Potential Energy Surfaces with Mixed-Basis Interpolation. *Submitted* **2021**, in review.

(S2) Morrow, Z.; Stoyanov, M. A Method for Dimensionally Adaptive Sparse Trigonometric Interpolation of Periodic Functions. *SIAM J. Sci. Comput.* **2020**, *42*, A2436–A2460.

(S3) Clenshaw, C. W.; Curtis, A. R. A Method for Numerical Integration on an Automatic Computer. *Numer. Math.* **1960**, *2*, 197–205.

(S4) Stoyanov, M. K.; Webster, C. G. A Dynamically Adaptive Sparse Grids Method for Quasi-Optimal Interpolation of Multidimensional Functions. *Comput. Math. A* **2016**, *71*, 2449–2465.

(S5) Gautschi, W. *Numerical Analysis*, 2nd ed.; Birkhäuser Basel, 2012.