

**Simulations of Light-Induced Molecular  
Transformations in Multiple Dimensions with  
Incremental Sparse Surrogates**

David Mokrauer, C. T. Kelley and Alexei Bykhovski

reprinted from

Journal of

**Algorithms &  
Computational  
Technology**

**Volume 6 • Number 4**

**December 2012**

Multi-Science Publishing Co. Ltd

# Simulations of Light-Induced Molecular Transformations in Multiple Dimensions with Incremental Sparse Surrogates

David Mokrauer, C. T. Kelley and Alexei Bykhovski<sup>\*†‡§</sup>

Submitted 06/10/2010; Revised 10/08/2011; Accepted 16/08/2012

## ABSTRACT

Simulations of light-induced molecular conformational transformations have traditionally been limited to a single degree of freedom because of the complexity of potential energy calculations. We propose a method of simulation that incrementally builds a surrogate for the potential energy function by computing gridpoints in parallel. We incorporate Smolyak's algorithm for sparse interpolation as the energy surrogates since it keeps the number of gridpoints at a manageable number. Our multi-dimensional algorithm is applied to the molecule 2-butene whose transition path is well known.

## 1. INTRODUCTION

The nuclei that bond to form a molecule will exist naturally in a geometric structure that has the least potential energy. A simple molecule is likely to only have a single conformation that is a minimum energy, but larger molecules are likely to have multiple structures which have such minima. When the potential energy of a molecule is modeled as a function of its nuclear coordinates, then the local minima of that function will be the stable conformations in which the molecule can exist. The geometry of a molecule that is made up of  $N$  atoms is uniquely determined by  $3N - 6$  nuclear

---

\*D. Mokrauer: BAE Systems, 8 NE Executive Park, Burlington, MA, 01803, USA  
(e-mail:david.mokrauer@baesystems.com)

†C. T. Kelley: Department of Mathematics, North Carolina State University, Raleigh, NC 27695 (USA) (e-mail: tim\_kelley@ncsu.edu)

‡A. Bykhovski: Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695 (USA) (e-mail: abykhov@ncsu.edu)

§This research was partially supported by ARO Grant number W911NF-07-1-0112.

§One of the authors (D. Mokrauer) is grateful for the mentoring assistance provided by Dr. D. L. Woolard, of the U.S. Army Research Office, in the preparation of this paper.

coordinates. Once a geometry has been specified, computing its potential energy requires finding the orbitals that the electrons associated with those atoms will occupy.

If a molecule can exist stably in more than one geometry, then it is possible that there is a reaction which will cause the molecule to transition from one conformation to the other. One way in which this can occur is through excitation. If enough energy is added to the system, an electron (or multiple electrons) may excite to a different orbital of higher energy. The quantities of energy that cause these excitations to occur are discrete amounts since this is a quantum phenomena. Once an electron occupies a new orbital, the forces between that electron and the rest of the molecule will have changed. It is possible that the potential energy is no longer a local minimum, and if this is the case, the molecule will reshape itself until a local minimum is reached. Since the molecule is in an excited state, some of the extra energy that caused the excitation will eventually be emitted and when that happens an electron will drop to a lower energy orbital and the molecule will relax again to another local minimum. If the current geometry differs from the original geometry, then we have found an energy path that will transition between the stable conformations.

Prior works simulate this transition with only one or two degrees of freedom [5]. In this work we develop an algorithm that can simulate these transitions in higher dimensions. Calculating the potential energy of the molecule as a function of its geometry is an expensive process which is not always successful. We addressed this problem in prior work with continuation and pre-processing [6]. Here we follow the natural relaxation of the molecule by incrementally interpolating the energy function in an area around the transition path. We improve the feasible degrees of freedom of the simulation by implementing a sparse interpolation method that is commonly used in high-dimensional numerical integration. All of our algorithms are applied to the molecule 2-butene which has a simple and well understood light-induced transformation.

Being able to discover molecules that exhibit this phenomena can help in the production of sensors. The emission of energy that occurs when an electron drops orbitals will be detectable as light of a specific frequency. If this reaction can be caused by the presence of other notable substances, then their presence can be detected by observing the light emission. Similarly this light emission can be used to cause a chain of biochemical reactions.

## 2. SIMULATION WITH 1 DEGREE OF FREEDOM

The potential energy of a molecule of  $N$  atoms is a function of  $3N - 6$  nuclear coordinates,  $p$ , and its quantum state,  $n$ .

$$E_n(p) \quad n = 0, 1, 2, \dots \quad (1)$$

The quantum states are well ordered

$$\{E_0(p) \leq E_1(p) \leq E_2(p) \leq \dots\}. \quad (2)$$

A stable geometry is a local minimum of the energy function on the ground state where  $n = 0$

$$E_0(p) \leq E_0(p + \delta) \quad \forall \delta < \epsilon. \quad (3)$$

A simple molecule which has two stable conformations is 2-Butene ( $C_4H_8$ ) and they are both pictured in figure 1. 2-Butene has 12 atoms and 30 coordinates, but the main difference between the two conformations is just a single coordinate. That coordinate is the rotation of the double bond between the angles labeled 2 and 3. We would like to simplify the energy functions of the molecule to be functions of this single coordinate. We partition the coordinate space,  $p$ , into the independent variable(s),  $x$ , and the remaining variables,  $\xi$ .

$$p = (x, \xi) \quad (4)$$

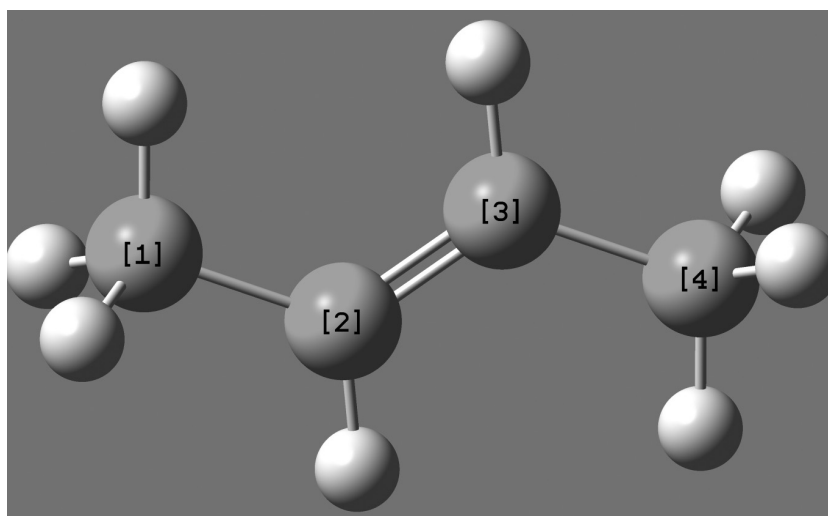
The simplification of the problem from thirty variables to a single variable allowed the simulation to feasibly be performed on a single computer. In order to compute the energy for a given value of the single coordinate, all of the other coordinates must be optimized so that the potential energy at that value is a minimum

$$E_n(x) = \min_{\xi} E_n(x, \xi) \quad (5)$$

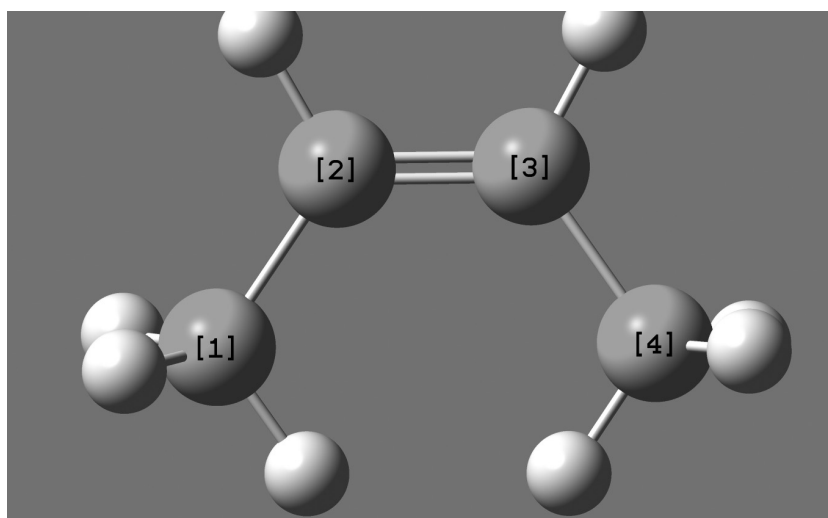
In other words, computing potential energy means performing a constrained optimization. One can perform the simulation by computing potential energy at a few values between the desired minima and building a surrogate for the function by interpolation

$$E_n(x) \approx E_n^s(x). \quad (6)$$

Once the surrogate for the potential energy surface (PES) is attained, the path is easily found by gradient descent as exemplified by figure 2. This path only



(a) Trans 2-Butene



(b) Cis 2-Butene

Figure 1. Butene molecule,  $C_4H_8$ .

required two quantum states  $E_0(x)$  and  $E_1(x)$ . We will not need to visit any other quantum states in this work. Simulation of this reaction for 2-butene with this single design variable has been done previously by Luo et al [5]. For many

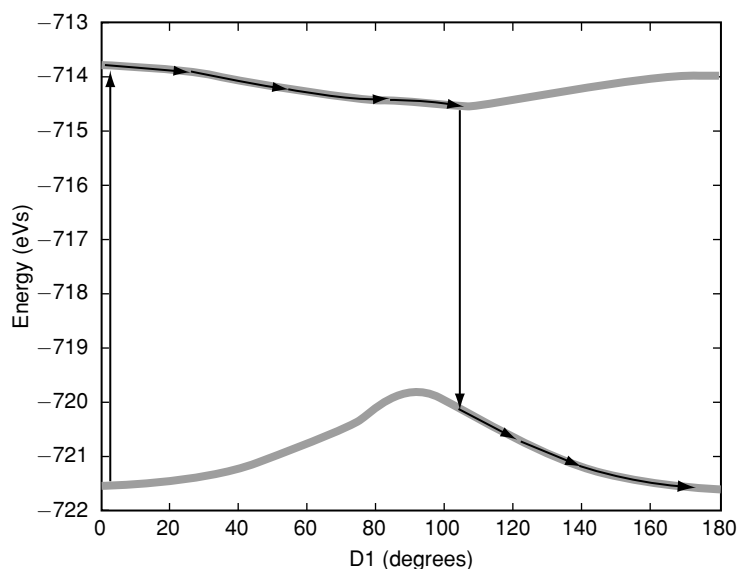


Figure 2. 2-Butene transition path in a single degree of freedom.

molecules the assumption of a single dominant coordinate will be unsuccessful in finding a path between minima. Not only does increasing the size of the molecule increase the possibility that it will require more degrees of freedom, but computing the potential energy becomes exponentially more costly.

### 2.1. Full PES in 2 Degrees of Freedom

In our prior work [7] we chose a fixed step length for each variable and computed the energy of the molecule at each point on the square grid of length  $360^\circ \times 360^\circ$ . This allowed us to use the default cubic spline in Matlab to interpolate the entire PES. Once the entire surrogate had been constructed we excited the molecule to a new quantum state. The relaxation of the molecule was simulated by continuous steepest descent [2, 4], which solves the ordinary differential equation

$$\dot{x} = -\nabla E_n^s(x) \quad (7)$$

using ODE45 in Matlab. The gradient is computed as a finite difference of the surrogate.

The first degree of freedom is the same bond rotation that we used in the previous section. The additional degree of freedom was the rotation of the central

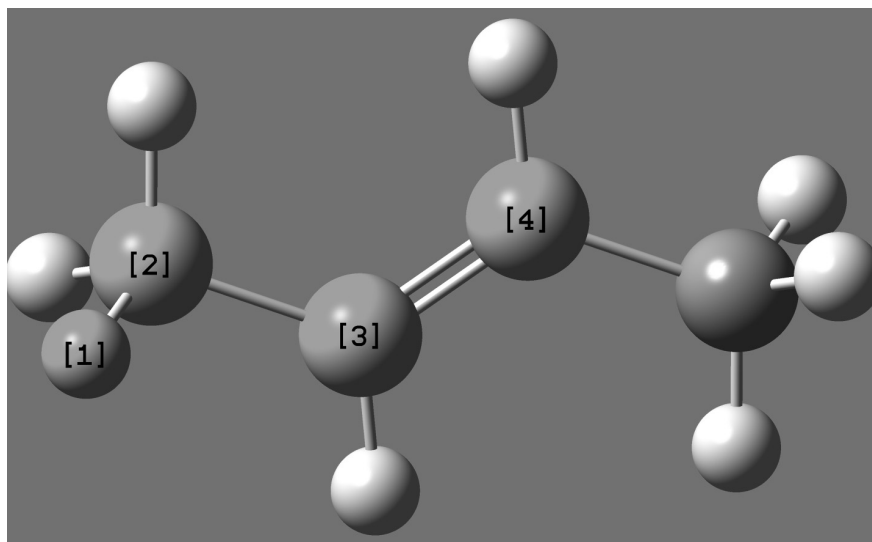


Figure 3. Our second degree of freedom is the rotation of the bond between atoms 2 and 3.

bond between atoms 2 and 3 pictured in Figure 3. We anticipate applying our algorithms to larger molecules with structures similar to 2-butene, but instead of hydrogen atoms bonded to the outer carbons they have benzene rings or other complicated structures bonded to those structures. We anticipate these rings to rotate throughout the transition.

This method successfully simulated the reaction with 2 degrees of freedom, but it is quite inefficient. Figure 4 shows the transition path overlaid on the full PES. It is clear that most of the surface went unseen by the optimization and those points were likely unnecessary for the simulation.

### 3. INCREMENTAL SURFACES

We decreased the number of points in the computation by computing small patches of the PES incrementally. Starting with an initial geometry, we computed the energy at gridpoints on a square grid that was much smaller than the full PES. Once we had a surrogate, we integrated to a local minimum or to a boundary of the surrogate. When the integrator hit a boundary, we made a new patch of the surrogate and integrated again. This process continues until a local minimum is found interior to a patch, then we emit or excite and continue the process.

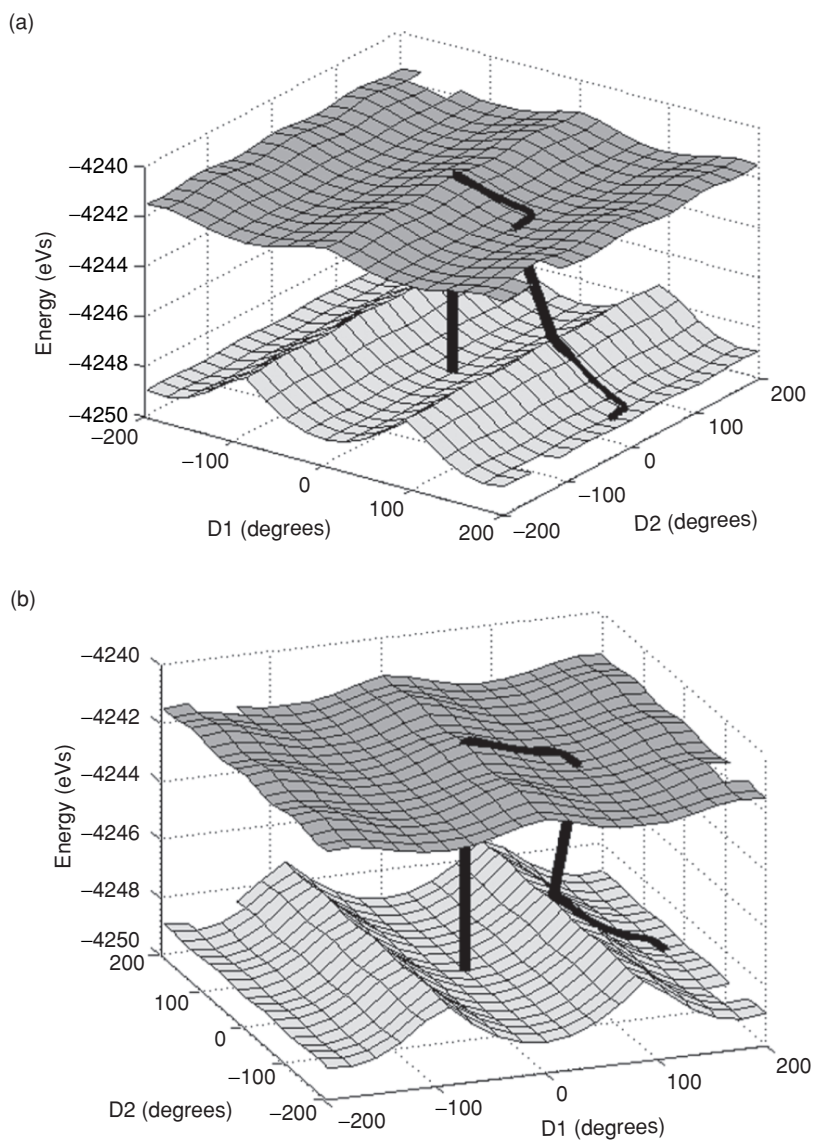


Figure 4. Successful transition path for 2-butene simulated on a full PES.

Using this method to simulate the 2-D path for 2-Butene produced the same results as the full PES. Figure 5 shows the transition path computed by incremental surface construction. The entire simulation that included computing a full PES in figure 4 lasted 3275.66 seconds while the incremental



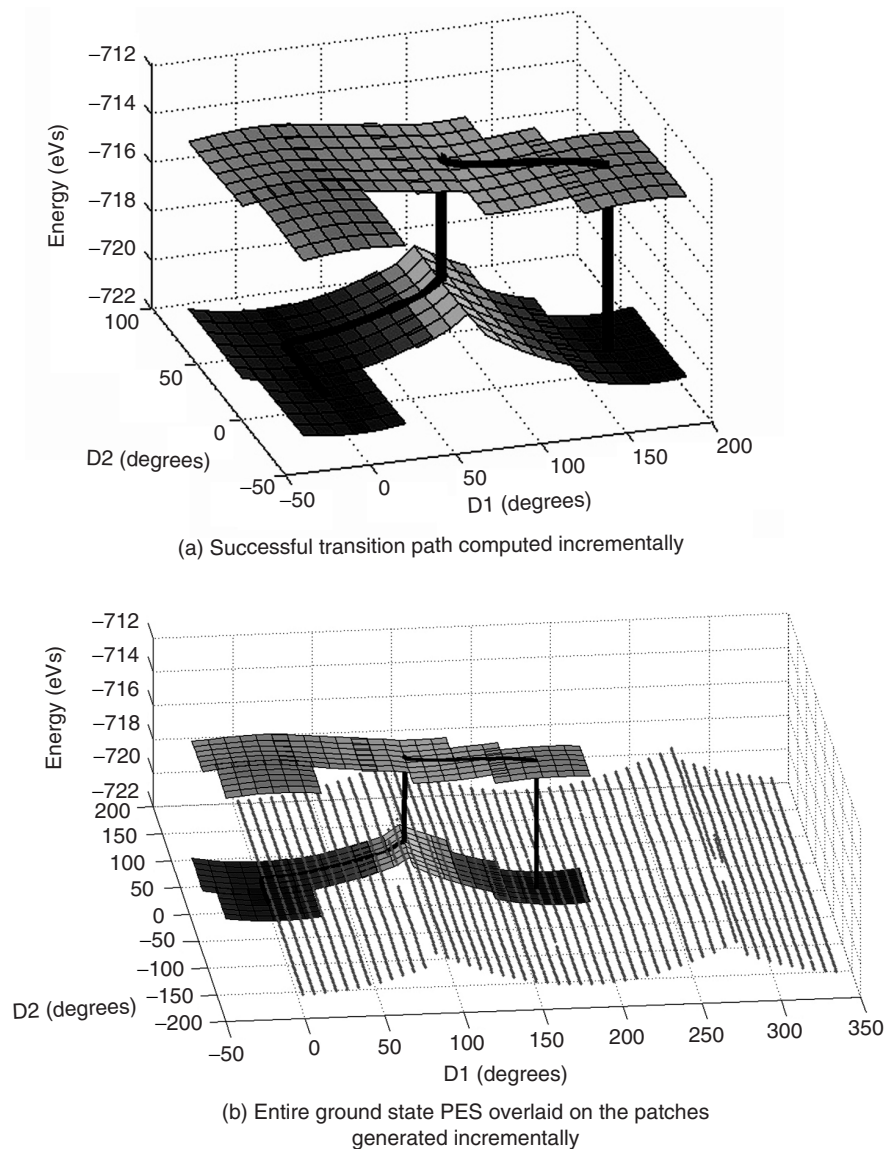


Figure 5. Incremental surface simulation in 2 degrees of freedom.

surfaces required only 1404.93 seconds. The speedup is limited by the continuation scheme for drawing the patch described in [6]. The biggest drawback to this method comes when we plan to increase the degrees of freedom. Accurate cubic splines require dense grids and when we increase the

degrees of freedom, the number of gridpoints for the spline will grow exponentially.

#### 4. SPARSE INTERPOLATION

The algorithm proposed by Smolyak in 1963 [9] is commonly used in high dimensional numerical integration [1, 8, 11, 10] for the following reasons:

- The number of nodes grows polynomially with the degrees of freedom.
- The interpolation exhibits polynomial exactness meaning that for every  $\kappa$  there is a formula so that the interpolation for all polynomials of degree  $\kappa$  or less is the exact polynomial.
- The nodes are spread throughout space in an unbiased way.
- The formulas for determining the gridpoints are easily calculable.

Smolyak's algorithm starts with a set of one dimensional interpolation formulas

$$U_m(x) = \sum_{j=1}^m f(x_j) l_j(x). \quad (8)$$

Where  $l_{j(x)}$  is the  $j^{\text{th}}$  Lagrange interpolation polynomial

$$l_j(x) = \prod_{i=1, i \neq j}^m \frac{x - x_i}{x_j - x_i} \quad (9)$$

Smolyak selects the nodes,  $\{X_m\}$ , so that they are nested. This means that an interpolation of index  $m$  will contain every node from the prior interpolation

$$\{X_{m-1}\} \subset \{X_m\}. \quad (10)$$

The common choice for nested nodes are the extrema of the Chebyshev polynomials

$$X_j^m = \left\{ -\cos \frac{\pi(j-1)}{2^{m-1}} \right\} \quad j = 1, \dots, 2^{m-1} + 1 \quad (11)$$

We also define a tensor product between one dimensional interpolation formulas

$$\bigotimes_{i=1}^d U_{m_i}(\vec{x}) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \cdots \sum_{i_d=1}^{m_d} f(\vec{x}) \prod_{j=1}^d l_{i_j}(x_j). \quad (12)$$

Smolyak's formula uses a linear combination of tensors of one dimensional interpolations. For a degree  $k$  interpolation in  $d$  dimensions the formula is

$$A(k + d, d)(x) = \quad (13)$$

$$\sum_{k+1 \leq |z| \leq k+d} (-1)^{k+d-|z|} \cdot \binom{d-1}{k+d-|z|} \cdot \otimes_{q=1}^d U^{z_q}. \quad (14)$$

Where  $z$  is a multi-index containing the degrees of the one dimensional interpolations. It is clear why the sets must be nested since the tensor products of one dimensional interpolations contain interpolations of differing degree. If these contained different sets of points, the grid would not be nearly as sparse. An estimate for the number of gridpoints for the interpolation in dimension  $d$  of degree  $k$  is

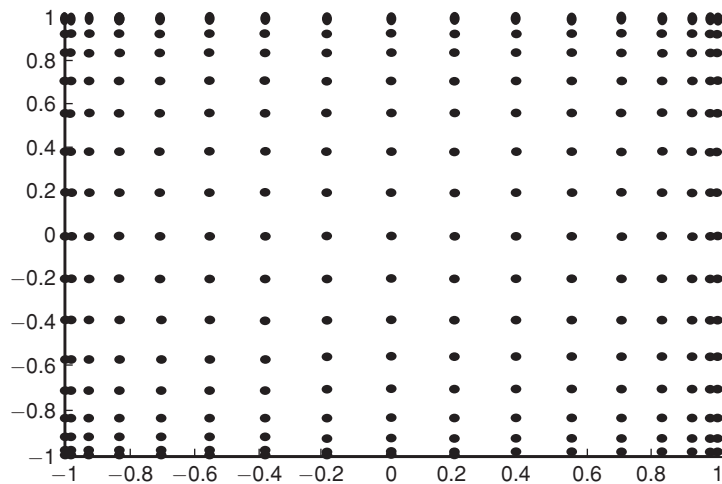
$$\dim(A(k + d, d)) \frac{2^k}{k!} d^k. \quad (15)$$

Instead of using Smolyak's algorithm we could use a full tensor product interpolation of the same nodes. For instance, a one dimensional cubic interpolation requires 4 points. We could use a tensor grid of those 4 points in  $d$  dimensions to build an interpolation which would require  $4^d$  gridpoints. Table 1

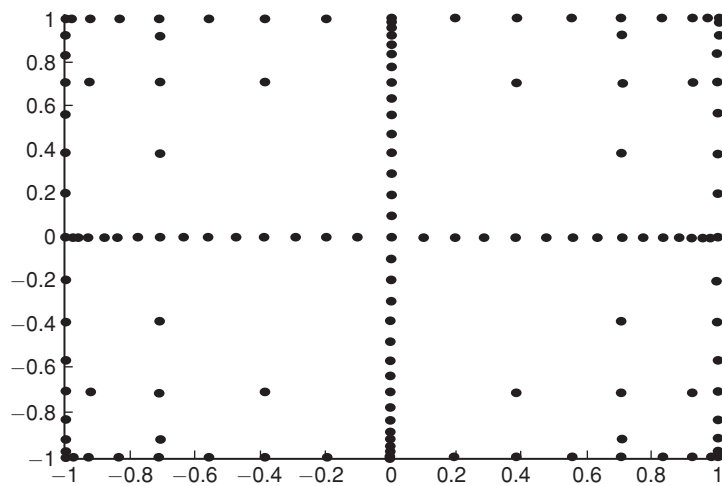
Table 1. Cubic Interpolation Points by Dimension and Method.

Dimension	Sparse Grid Points	Tensor Grid Points
2	29	16
3	69	64
4	137	256
5	241	1024
6	389	4096
7	589	16384
8	849	65536
9	1177	262144
10	1581	1048576

compares the number of Smolyak points for cubic exactness to the number of tensor points for cubic exactness. Figure 6 shows the 5th degree Smolyak points along with the tensor grid that would result from the same set of one dimensional points.



(a) Gridpoints resulting from tensor of chebyshev nodes



(b) Sparse gridpoints resulting from the same chebyshev nodes

Figure 6. Tensor grid vs. Sparse grid.

## 5. SPARSE INCREMENTAL SURFACES

We have shown in previous work that computing the potential energy at a given point on the PES requires a nearby optimized structure as an initial input due to the fact that energy computations are constrained optimizations [?]. Smolyak's formula doesn't translate into a continuation scheme nearly as easily as square grids so we chose to abandon continuation on the incremental surfaces. This will limit the size of the grids we use for the surrogates.

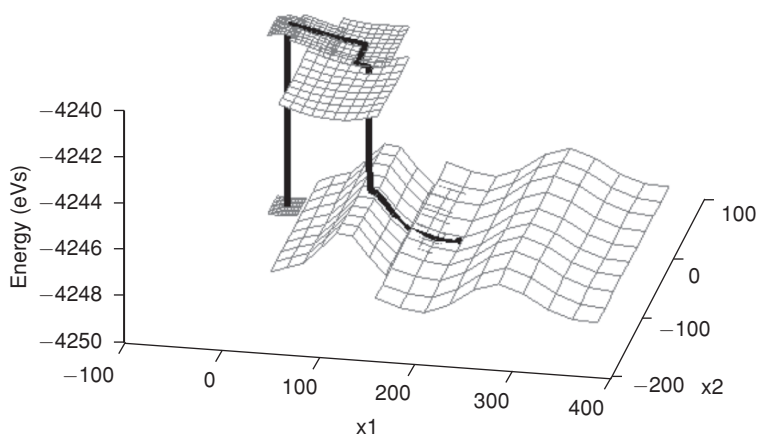
We borrow ideas from trust region optimization algorithms [3] to adapt the length of the patch. We choose a parameter,  $\sigma$ , which is a multiplier for growing and shrinking the size of the patch. If the value of the reduction of the energy predicted by the surrogate at the final point of integration on the patch,  $pred$ , is close to the actual reduction of energy,  $ared$ , then we multiply the patch length by  $\sigma$ . Similarly if the two metrics are not close at all, then we divide the patch length by  $\sigma$ . We will also shrink the patch if the number of iterations for each gridpoint to converge grows too large, likewise if the energy does not converge at all. If  $F(x_{cur}, H)$  is the function that performs continuous steepest descent on the current surrogate, this adapting patch algorithm is

```

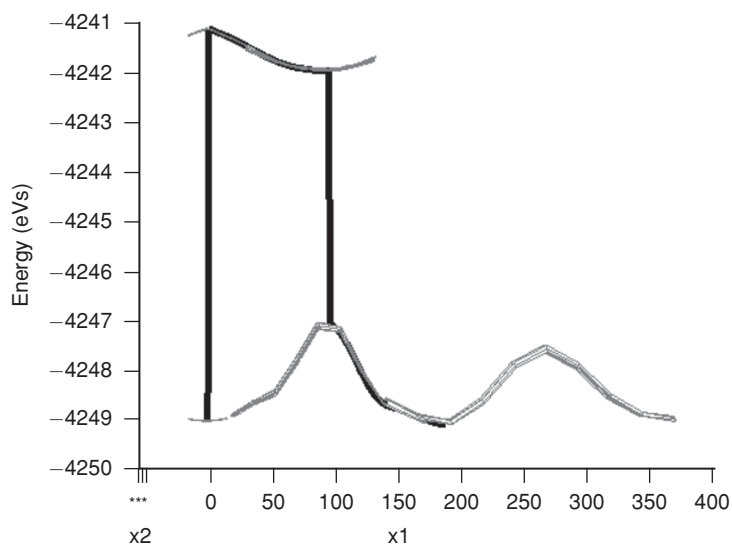
function [ $x_f$ ]=adapt_patch( $x_0, \sigma, \epsilon, h_0, \max\_iters$ )
  at_min=False, H= $h_0$ 
  while at_min=False do
    [ $at\_min, \max\_it, NaNs, x_f, \hat{E}_f$ ]=F( $x_{cur}, H$ )
    calculate E( $x_f$ )
    ared=E( $x_{cur}$ )-E( $x_f$ )
    pred=E( $x_{cur}$ )- $\hat{E}(x_f)$ 
    if  $\left( \left\| 1 - \frac{ared}{pred} \right\| \leq \epsilon(1) \right)$  and ( $\max\_it \leq \max\_iters$ ) and ( $NaNs=0$ ) then
      H= $\sigma H$ 
    else if  $\left( \left\| 1 - \frac{ared}{pred} \right\| \geq \epsilon(2) \right)$  or ( $\max\_it > \max\_iters$ ) or ( $NaNs \neq 0$ ) then
      H =  $\frac{H}{\sigma}$ 
    end if
     $x_{cur} = x_f$ 
  end while

```

Once again we applied our method to 2-butene with the parameter  $\sigma = 1.5$ . The algorithm once again simulated the transition path which can be seen in Figure 7. At each iteration the patch size grew. This is due to the fact that 2-butene is a very simple molecule with a well-understood transition. Once we apply this algorithm



(a) First view of transition path using sparse patches



(b) Second view of transition path using sparse patches

Figure 7. Simulation of transition path for 2-butene using sparse patches in 2 degrees of freedom.

to more complex molecules we do not expect the surfaces to grow consecutively to completion. We have also applied our algorithm with a third degree of freedom to 2-butene. That additional torsion angle can be seen in Figure 8. The energy path taken by the simulation with three degrees of freedom can be seen in Figure 9, but

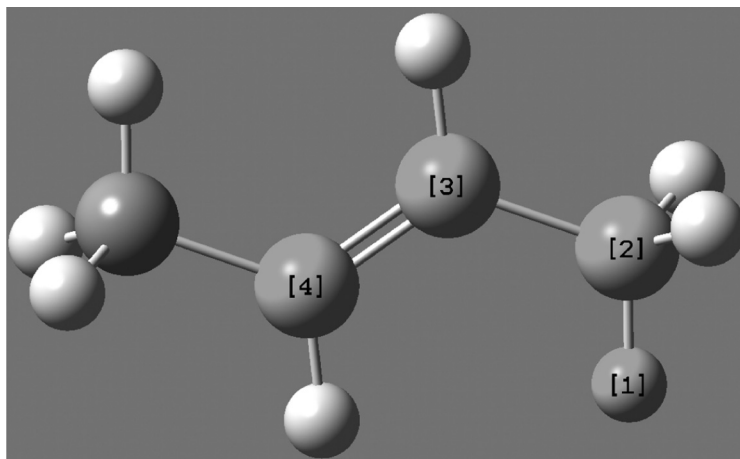


Figure 8. 3rd degree of freedom is the rotation of the bond between atoms 2 and 3.

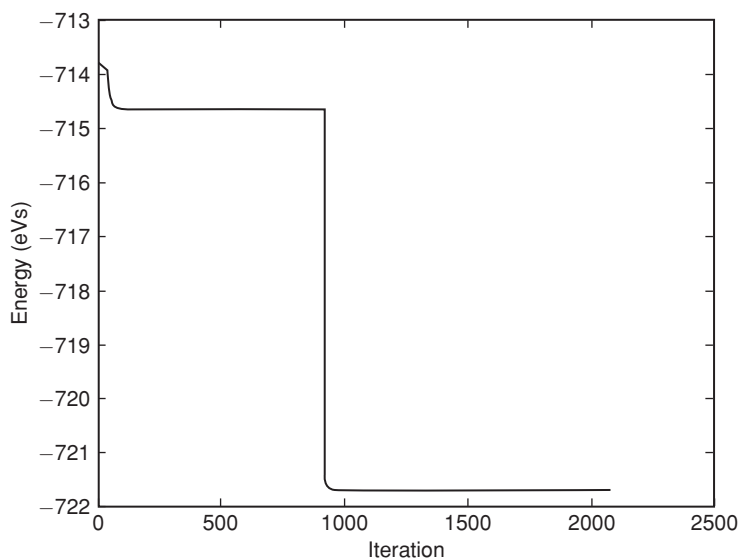


Figure 9. Energy path taken by adapting patch simulation in 3 degrees of freedom for 2-butene.

Table 2. Scalability of Sparse Incremental Surface Construction.

Degree of Interpolation	Gridpoints	Processors	Time (secs.)
2	25	13	190.96
3	69	39	194.05

we have omitted the scatter plots with an overlaid path due to the difficulty in visualization. Table 2 demonstrates that the algorithm exhibits good weak scalability.

## 6. RESOURCES

The computations in this work were performed on the North Carolina State University high performance computing cluster. The chassis has 60 quad core Xeon processors with 2GB distributed memory per core and dual gigabit ethernet interconnects. The operating system is Red Hat Linux 2.6.9. Potential energy computations are performed using Gaussian09. Python 2.5.4 performs script editing, manages parallel submissions of energy calculations, and controls the patch size. Figures and optimizations were produced in Matlab 7.8.0.347.

## 7. CONCLUSIONS

Our method of incremental surface construction decreases the compute time necessary for simulating the transition path in multiple dimensions. By using Smolyak's algorithm to determine the gridpoints at which to evaluate the potential energy, we have greatly increased the number of degrees of freedom that these simulations may incorporate. Our method of adapting the patch size at each increment will decrease the number of patches that the simulation will need to compute.

## REFERENCES

- [1] Volker Barthelmann, Erich Novak, and Klaus Ritter, *High dimensional polynomial interpolation on sparse grids*, *Advances in Computational Mathematics* **12** (2000), 273–288.
- [2] R. Courant, *Variational methods for the solution of problems of equilibrium and vibration*, *Bull. Amer. Math. Soc.* **49** (1943), 1–43.
- [3] C. T. Kelley, *Iterative methods for optimization*, SIAM, Philadelphia, Pa., 1999.
- [4] X-L Luo, C. T. Kelley, L-Z. Liao, and H-W Tam, *Combining trust region techniques and Rosenbrock methods for gradient systems*, *J. Opt. Th. Appl.* **140** (2009), 265–286.



- [5] Y. Luo, B. Gelmont, and D. Woolard, *Bio-molecular devices for terahertz frequency sensing*, Molecular and Nano Electronics: Analysis, Design, and Simulation (J.M. Seminario, ed.), 2007, pp. 55–81.
- [6] David Mokrauer, C. T. Kelley, and Alexei Bykhovski, *Efficient parallel computation of molecular potential surfaces for the study of light-induced transition dynamics in multiple coordinates*, IEEE Transactions on Nanotechnology (2010), in-press.
- [7] ———, *Parallel computation of surrogate models for potential energy surfaces*, 2010 International Symposium on Distributed Computing and Applications to Business, Engineering and Science (Los Alamitos, CA) (Quo Qingping, ed.), IEEE, 2010.
- [8] Erich Novak and Klaus Ritter, *Simple cubature formulas with high polynomial exactness*, Constr. Approx. 15 (1999), 499–522.
- [9] S. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Soviet Math. Dokl. **4** (1963), 240–243.
- [10] G. Wasilkowski and H. Wozniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, J. Complexity **11** (1995), 1–56.
- [11] Dongbin Xiu and Jan S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, Siam J. Sci. Comput. **27** (2005), 1118–1139.