



# FEMVib an Automated Publicly Available Web-Service for Solving the Vibrational Schrödinger Equation

Peter Zajac, Dong Xu and Andrew Cooksy

October 2012

Publication Number: CSRCR2012-03

Computational Science &  
Engineering Faculty and Students  
Research Articles

Database Powered by the  
Computational Science Research Center  
Computing Group & Visualization Lab

## COMPUTATIONAL SCIENCE & ENGINEERING



**SAN DIEGO STATE  
UNIVERSITY**

Computational Science Research Center  
College of Sciences  
5500 Campanile Drive  
San Diego, CA 92182-1245  
(619) 594-3430



# FEMVib an Automated Publicly Available Web-Service for Solving the Vibrational Schrödinger Equation

Peter Zajac

*Computational Science Research Center, 5500 Campanile Dr., San Diego, 92182-1245*

Dong Xu

*Boise State University, Department of Chemistry and Biochemistry, University Dr. 1910, Boise, Idaho 83725*

Andrew Cooksy

*San Diego State University, Department of Chemistry and Biochemistry, 5500 Campanile Dr., San Diego, 92182-1245*

---

## Abstract

A computational package has been developed to solve the bounded vibrational Schrödinger equation for up to three coupled coordinates on any given effective potential energy surface (PES) [1]. This article presents a new and improved version of the package with a fully automated point-wise PES evaluation as well as a revised routine for the computation of the Wilson G-matrix on scattered surfaces. Since the PES is given as a discrete set of points an interpolation scheme is required to automate the process of stiffness matrix and global matrix assembly within the finite element paradigm. The recent development of automatic PES generation [4] resulted in two different interpolation schemes available in the package, namely interpolating moving least squares (IMLS)[5] and Kriging interpolation [12], with parameter tuning. A significant improvement has been made to the Wilson G-matrix evaluation in terms of removing rotational discrepancies for small molecules before imposing Eckart conditions and allowing scattered surface calculations during the transformation from Cartesian to internal coordinates. Lastly, the package has been fully automated and made publicly available through the world wide web, thus enabling research groups around the world to analyze their vibrational spectra using our finite element method based approach.

*Keywords:* Vibrational Schrödinger Equation, Finite Element Method, Wilson G-matrix, PES interpolation

---

## 1. Introduction

Many different approaches have been developed over the past thirty years for solving the vibrational Schrödinger equation. One of the first attempts using perturbation theory and variational computation was introduced by Handy and co-workers [13, 14] in the late 1970s. These attempts were followed by the introduction of exact Hamiltonians by Watson for nonlinear [15] and linear [16] molecules, based on the use of regular rectilinear normal coordinates. Many important developments in theoretical molecular spectroscopy came after these groundbreaking studies which used exact kinetic energy operators in (nearly) variational computations. These investigations [17, 18, 19, 20, 21, 22, 23, 24, 25] were aimed at determining a large number of accurate rotational-vibrational energy levels of polyatomic species of increasing size and complexity. This goal has been achieved for triatomic molecules [26, 27], for which exceedingly accurate PESs can be developed. For larger systems, it has been computationally very difficult to predict the spectra, especially near the dissociation limits. Our goal when developing FEMVib package has been to

solve the vibrational Schrödinger equation (VSE) for a polyatomic molecule, starting from an arbitrary PES of up to three dimensions, without requiring the selection of specific coordinates or basis functions. The wavefunctions and energies obtained from these calculations may then be used to predict vibrational spectra involving strongly coupled and anharmonic modes. FEMVib has been tested to resolve the eigenvalues and wavefunctions of hundreds of vibrational energy states to high accuracy and precision. It may be used to calculate the complete vibrational spectra of triatomic molecules or to approximate larger systems through a “relaxed” model that allows complete coupling of up to three selected vibrational coordinates. FEMVib provides physical chemists with a general, robust and accurate computational tool for molecular vibrational analysis.

The analytical representation of potential energy surfaces for reactive systems has been a challenge in the field of quantum chemistry for the past few decades [28, 29]. With the improvement of ab-initio methods the task became even more intricate because of their high accuracy and expense of computation. The PES is a crucial step towards analysis of vibrational energies [1], chemical reaction

modeling, and transition states analysis [2, 3]. There have been several methods used for fitting the PES, in particular Shepard interpolation [30], spline interpolation [31, 34, 35], Bayesian approach [36], Voronoi-step representation introduced by Suhm [42], tessellation using Clough-Tocher interpolant [43] and Distributed Approximating Functionals [44]. One of the most recent and most successful developments of PES construction using Interpolating Moving Least Squares (IMLS) [4, 5, 6, 7, 8] utilizes two different basis set functions and compares the first derivative deviations. This is done dynamically during the PES construction, hence the choice of points is controlled by the largest value of gradient at a given location. This scheme is implemented in our FEMVib package for users who used IMLS for their PES construction. We have investigated several interpolation schemes for PESs that have been generated randomly or using regular grid expansions but without dynamic corrections - namely Kriging Interpolation, Radial Basis Function (RBF) Interpolation, Shepard Interpolation, Distributed Approximating Functionals and Multi-level B-splines. Six different functions have been used in 2D and 3D for optimization testing of the above mentioned interpolation schemes. The interpolation method that we found to be the most suitable and among the most accurate was the Kriging interpolation. It has been implemented into FEMVib as an alternative in case the PES has not been constructed using IMLS, along with a parameter tuning. We will present the tests performed when tuning the "shape" parameter for these interpolation schemes, changing the grid density and using different analytical testing surfaces.

The PES represents the potential energy term in the Hamiltonian for the VSE. Its kinetic energy term, which is of equal importance, can be expressed in various coordinate representations [45]. When coordinates other than Cartesian or Normal are used the kinetic energy term becomes very complicated. Moreover, if anharmonic and coupled vibrational motions are present the quantum mechanical treatment of higher vibrational states becomes increasingly difficult. It is therefore convenient to formulate and solve the VSE in internal coordinates, which represent bond distances, valence angles, torsions, etc. and which allow the computational treatment of anharmonic vibrations [46, 49]. The main property of the internal coordinate set is that it remains unaffected by translations or rotations of the molecule as a whole by applying the Eckart conditions [50]. Internal coordinates can describe the system from a chemically relevant point of view and extend well beyond the simple bond distances and angles. One of the major problems with internal coordinates is associated with their formulation of masses. This becomes an issue when transforming the motion from Cartesian to the internal coordinate system, which introduces the concept of reduced masses. Despite the fact that the exact formulation of reduced masses has been known for decades, little has been published in this area. The first-principle approach for the determination of reduced masses was used

in a few cases [51, 52, 53] and only recently a package has been developed [57] for the computation of reduced masses in a generalized internal coordinates set. This package has been previously implemented in the former version of FEMVib. There were two main problems with this formulation: (i) the calculation assumed structured grid and (ii) small symmetrical molecules were not rotated into the same reference frame and compared to the reference structure hence the Wilsons G-Matrix elements representing the transformation between the two coordinate sets were incorrect. We have addressed this problem by developing a general G-matrix calculator (GCAL) for up to three internal vibrational coordinates that works on scattered data sets and accounts for pre-rotation of small symmetrical structures (e.g. H<sub>2</sub>O, FHF<sup>-</sup>, etc.). GCAL has been successfully implemented and serves as a part of the FEMVib package.

The FEMVib package is implemented mostly in the C++ programming language, where the Libmesh library [59] has been used to produce sparse, structured Hamiltonian matrices. The GCAL routine is also implemented using C++ programming language, while a Perl script is used to sort out the singular values which correspond to similar molecular structures with different sets of internal coordinates. The generalized eigenvalue problem is solved using the Jacobi-Davidson algorithm [11] obtained from PySPARSE [65] an open source Python library. The entire package has been fully automated and is being offered as a web-service. The application interface was developed using OPAL Toolkit [66], a JAVA [67] based web-engine which runs under ANT[68] and TOMCAT [69]. Details regarding the installation, user interface and input parameters are given in Appendix B.

## 2. Methods

### 2.1. IMLS

Consider a scattered data set of  $N + 1$  points  $\{x_i; i = 0, 1, \dots, N\}$  with corresponding functional values  $\{f_i; i = 0, 1, \dots, N\}$ . The standard formulation of the least-square problem is given as:

$$S_x(p) = \sum_{i=0}^N [p(x_i) - f_i]^2 \quad (1)$$

where  $p(x)$  represents a polynomial of the form  $p(x) = \sum_{i=0}^m a_i x^i$  and  $S_x$  is to be minimized. Note,  $m \leq N$ .

The weighted least-square method has a similar analytical representation, except every deviation  $p(x_i) - f_i$  in equation 1 is weighted depending on the distance from the interpolated point  $|x - x_i|$ . Therefore, we minimize  $S_x$  in:

$$S_x(p) = \sum_{i=0}^N w_i [p(x_i) - f_i]^2 \quad (2)$$

where  $w_i(x)$  are positive weight functions. They behave asymptotically, where  $w_i(x) \rightarrow \infty$  as  $x \rightarrow x_i$ . Commonly



## 2.2. Kriging and RBF Interpolation

Radial basis function interpolation at a point  $x_m$  on a grid of scattered data denoted  $x_i$  (called the “enters”) has the functional form  $F(x)$  given by:

$$F(x_m) = \sum_i^N a_i \times \Phi(\|x_m - x_i\|), \quad (15)$$

where  $\|\cdot\|$  represents the Euclidean norm,  $a_i$  are the weights to be optimized and  $\Phi(\|x_m - x_i\|)$  are the radial basis functions. The value at each interpolated point on the grid is given as a weighted sum of the distance basis functions. To determine the weights in equation 15 we need to solve the linear system given in vector form:

$$\mathbb{A} \times \vec{a} = \vec{f}, \quad (16)$$

where  $f$  is the vector of functional values and  $\mathbb{A}$  is the collocation matrix with elements  $\Phi(\|x_i - x_j\|)$ ,  $1 \leq i, j, \leq N$ . For the system in equation 16 to be solvable we need the  $\mathbb{A}$  matrix to be invertible. The non-singularity of this matrix depends on the choice of  $\Phi$  and can be guaranteed for multiquadric and inverse multiquadric of RBFs [37]. The form of RBFs used in our study are well known and are listed with their analytical representation in Table 1.

Function Name	Function Form
multiquadric	$(r^2 + r_0^2)^{1/2}$
inverse multiquadric	$(r^2 + r_0^2)^{-1/2}$
thinplate	$r^2 \times \log(r/r_0)$

Table 1: Radial basis functions. Here  $r$  is the radial distance and  $r_0$  represents the “shape” parameter which is chosen by the user.

Here it becomes clear that the error of the interpolation using RBF depends on the choice of the  $r_0$  parameter and thus the choice is very important for further applications that require high accuracy. More sophisticated methods than [32, 33] for selecting the optimal value of this parameter were developed by Rippl [38] and further improved by [39], where a “cost functional” is minimized with respect to the RMS.

Ordinary Kriging interpolation is often associated with the acronym B.L.U.E. for “best linear unbiased estimator.” Ordinary Kriging is “linear” because its estimates are weighted linear combinations of the available data; it is “unbiased” because it tries to have the mean residual  $m_R$  equal to zero; and it is “best” because it aims at minimizing the variance of the errors  $\sigma_R^2$ . The RBFs are also linear and unbiased methods; however the distinguishing feature of ordinary Kriging is its aim of minimizing the error variance [12]. The goals of ordinary Kriging are in a practical sense unattainable because  $m_R$  and  $\sigma_R^2$  are not known and therefore cannot be minimized. The best that can be done is build a model or “variogram” of the data

and work with the average error and error variance for this model. Ordinary Kriging uses a probabilistic model in which the bias and the error variance can both be calculated and then weights chosen for the nearby samples that ensure that the average error for the modeled  $m_R$  is exactly zero and that the modeled error variance  $\sigma_R^2$  is minimized.

At a given point, assuming it is one that does not correspond to the data, the unknown functional value will be estimated as a weighted linear combination of the known values:

$$F(x_m) = \sum_{j=1}^N a_j \times f(x_j), \quad (17)$$

where  $f(x_j)$  are functional values at each grid point. For any point at which we attempt to estimate the unknown value; our model is a stationary random function that consists of several random variables, one for the values at the known grid points and one for the unknown value we are trying to estimate. If the random function is denoted as  $V(x_j)$  then the estimation model error is also a random variable and is given by [12]:

$$R(x_m) = \sum_{j=1}^N a_j \times V(x_j) - V(x_m). \quad (18)$$

Here  $V(x_m)$  is the random function at the point which value we are trying to estimate. The modeled error variance is

$$\sigma_R^2 = \sigma^2 + \sum_{i=1}^N \sum_{j=1}^N a_i a_j C_{ij} - 2 \sum_{i=1}^N a_i C_{i0}, \quad (19)$$

where  $\sigma^2$  is a random function model variance parameter,  $a_j$  represent the weights, and  $C_{ij}$  are all the covariances between every two random variables. The minimization of  $\sigma_R^2$  is accomplished by setting  $N$  partial derivatives to 0. However, as was mentioned before, the ordinary Kriging is unbiased. Therefore the only set of weights that can be chosen are those that satisfy  $\sum_{j=0}^N a_j = 1$ . This introduces a constraint into the minimization. To convert a constrained minimization into unconstrained, the Lagrange parameter [40] is introduced, which changes equation 19 to

$$\sigma_R^2 = \sigma^2 + \sum_{i=1}^N \sum_{j=1}^N a_i a_j C_{ij} - 2 \sum_{i=1}^N a_i C_{i0} + 2\mu \left( \sum_{i=1}^N a_i - 1 \right), \quad (20)$$

where  $\mu$  is the Lagrange parameter. Taking the  $N$  partial derivatives with respect to the weights  $a_j$  and setting them to zero will result in the following linear system:

$$\begin{bmatrix} C_{11} & \cdots & C_{1N} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C_{N1} & \cdots & C_{NN} & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ \vdots \\ a_N \\ \mu \end{bmatrix} = \begin{bmatrix} C_{10} \\ \vdots \\ C_{N0} \\ 1 \end{bmatrix} \quad (21)$$

or in vector form:

$$\mathbb{C} \times \vec{a} = \vec{D} \quad (22)$$

Solving this linear system by inverting the  $\mathbb{C}$  matrix gives us the weights  $a_j$  which are used to estimate our functional value equation 17. The covariance function or variogram used in this paper has the simple exponential form:

$$v(r) = \alpha \times r^{r_0}, \quad (23)$$

where  $r$  is the magnitude of the distance between any two points on the grid,  $\alpha$  is estimated from the data and  $r_0$  is the parameter chosen by the user. In our case  $r_0$  again represents the unknown parameter we changed for every grid density.

### 2.3. Wilson G-Matrix

PES represents the potential energy term  $V$  in the nuclear Hamiltonian:

$$\hat{H} = \hat{T} + V \quad (24)$$

where  $\hat{T}$  represents the kinetic energy operator. In contrast to  $V$ , the kinetic energy has a general analytical representation, which Cartesian form:

$$\hat{T}_x = -\frac{\hbar^2}{2} \sum_{i=1}^{3N} \frac{1}{m_i} \left[ \frac{\partial^2}{\partial x_i^2} \right] \quad (25)$$

where  $m_i$  is the mass of  $i$ -th atom,  $x, y$  and  $z$  are Cartesian coordinates, and  $N$  is the number of atoms in the molecule, can be re-written into [55]:

$$\hat{T}_q = -\frac{\hbar^2}{2} \sum_{r=1}^M \sum_{s=1}^M \left[ g^{1/4} \frac{\partial}{\partial q_r} \left[ g^{-1/2} \cdot G^{rs} \frac{\partial}{\partial q_s} [g^{1/4}] \right] \right] \quad (26)$$

where  $q_r$  and  $q_s$  are internal coordinates,  $G^{rs}$  are the elements of the Wilson G-matrix [45], and  $g$  is its determinant.  $M$ , the number of vibrational degrees of freedom, is equal to  $3N - 6$  for non-linear and  $3N - 5$  for linear molecule.

The Wilson G-matrix represents the transformation between Cartesian and internal coordinates. It is defined as:

$$G^{rs} = \sum_{i=1}^{3N} \frac{1}{m_i} \frac{\partial q_r}{\partial x_i} \frac{\partial q_s}{\partial x_i} \quad (27)$$

where  $\partial q_i / \partial x_j$  represents the partial derivative of  $i$ -th internal with respect to  $j$ -th Cartesian coordinate. Properties and formation of G-matrix can be found in [47].

Equation 26 can be simplified as:

$$\hat{T}_q \approx -\frac{\hbar^2}{2} \sum_{r=1}^M \sum_{s=1}^M \frac{\partial}{\partial q_r} \left[ G^{rs} \frac{\partial}{\partial q_s} \right] \quad (28)$$

where the determinant ( $g$ ) is neglected, because of its small internal coordinate dependence [56].

The nuclear Hamiltonian 24 in its most general form

still contains vibrational, rotational and translational motions of nuclei. To extract only the vibrational motions, Eckart conditions [50] need to be applied to all geometries, corresponding to a particular set of internal coordinates along the point-wise PES scan. This has to be done before the transformation from Cartesian to internal coordinates. The Eckart conditions are:

$$\sum_{\alpha=1}^N m_{\alpha} (\mathbf{d}_{\alpha} \cdot \mathbf{r}_{\alpha}^0) = 0 \quad (29)$$

and

$$\sum_{\alpha=1}^N m_{\alpha} (\mathbf{d}_{\alpha} \times \mathbf{r}_{\alpha}^0) = 0 \quad (30)$$

where  $\mathbf{r}_{\alpha}^0$  denotes the position vector of the  $\alpha$ -th atom in the reference structure, and  $\mathbf{d}_{\alpha}$  is the displacement vector of the same atom from its position in the reference structure. These conditions are not easily applied and the process involves multiple numerical steps.

First, the optimized geometries are placed into the the same plane in case they are planar. A ‘‘pre-rotation’’ routine will test their alignment with the reference structure, by rotating the molecule 180 degree along each of the three Cartesian axis. This is necessary because some molecular structures can be rotated more than one way to satisfy condition 30. Condition 29 is satisfied by setting the origin of the coordinate system into the center of mass of the molecule. Numerical routine is used to satisfy condition 30. For all of our tested systems the error for the numerical treatment of condition 30 was always smaller than  $10^{-6}$ .

After condition 30 has been applied a ‘‘post-rotational’’ modification is applied, where the molecule is again rotated 180 degrees along each Cartesian axis and compared to the reference structure. The reason for the ‘‘post-rotation’’ is that for certain molecular structures, the numerical routine might rotate the molecule 180 degrees away from the expected structure, despite condition 30 being satisfied.

Once Eckart conditions are successfully applied, the G-matrix elements may be calculated. In most cases the functional dependence of internal coordinates on Cartesian coordinates, poses a problem. The change in the internal coordinate variations are limited by Eckart conditions, resulting in 0 derivatives  $\partial q_i / \partial x_j$ .

The solution to this problem as discussed in [57], is to calculate the inverse of the G-matrix instead, hence exchange the role of dependent and independent variables [46].

$$G_{rs} = \sum_{i=1}^{3N} m_i \frac{\partial x_i}{\partial q_r} \frac{\partial x_i}{\partial q_s} \quad (31)$$

In our program every set of Cartesian coordinates (e.g.  $x_1$ ), corresponding to the same atom for every structure, is considered as a  $d$ -dimensional surface where  $q_i^k$  represents the  $i$ -th element along  $k$ -th axis;  $k = 1, \dots, d$ . We use SD-IMLS interpolation at every point on the surface and

using its basis set coefficient estimate the derivatives. This approach is general even for scattered PESs, for which a 1D-cubic spline interpolation [57] is not sufficient. This approach is also superior to [54], where contributions from all atoms are considered in the G-matrix calculation. Once we have calculated the  $G_{rs}$  we simply invert it to obtain  $G^{rs}$ .

Calculation of the inverse G-matrix elements leads directly to the kinetic energy operator:

$$\hat{T}_q = -\frac{\hbar^2}{2} \sum_{r=1}^M \sum_{s=1}^M \left[ \sum_{i=1}^{3N} m_i \frac{\partial x_i}{\partial q_r} \frac{\partial x_i}{\partial q_s} \right]^{-1} \dot{q}_r \dot{q}_s \quad (32)$$

We can re-write equation 32 in a simpler form as

$$\hat{T}_q = -\frac{\hbar^2}{2} \sum_{r=1}^M \sum_{s=1}^M G^{rs} \dot{q}_r \dot{q}_s \quad (33)$$

Following this process further in the vibrational analysis we construct the secular equation [57] for the vibrational Hamiltonian using equation 28. The formed VSE can be solved using an appropriate numerical scheme.

### 2.3.1. GCAL Program Structure

GCAL program has been written in C++ programming language. The steps behind the algorithm are outlined below:

1. Set all the structures into the same plane, if the molecule is planar.
2. "Pre-rotate" all the structures along each axis by 180 degrees and compare to reference structure. Retain the structure closest to reference.
3. Apply first Eckart condition: set the origin of the Cartesian coordinate system to the center of mass for each structure.
4. Apply second Eckart condition: rotate each structure until condition 30 is satisfied. To achieve this, a Variable Metric method in multiple dimensions is used [58].
5. "Post-rotate" all the structures to account for geometry inversions.
6. Cycle through all the Cartesian coordinate sets corresponding to components along  $x$ ,  $y$  and  $z$  axis for each atom and construct surfaces, where internal coordinates are independent variables and Cartesian are dependent.
7. Interpolate the surfaces at all points, that represent internal coordinates, using SD-IMLS and consequently estimate the derivatives.
8. Using these derivatives construct the Inverse G-matrices.
9. Invert the Inverse G-matrices.

## 3. Results

### 3.1. Interpolation Methods

Six different surfaces have been used for the testing of interpolation methods. Their analytical representation is

Function Name	2D Form	3D Form
De Jong [60]	$x^2 + 2y^2$	$x^2 + 2y^2 + 3z^2$
Rosenbrock [62]	$(1 + x^2)^2 + 100 * (y - x^2)^2$	not tested
Rastrigin [63]	$20 + (x^2 - 10\cos(\pi * x/4)) + (y^2 - 10\cos(\pi * y/4))$	$20 + (x^2 - 10\cos(\pi * x/4)) + (y^2 - 10\cos(\pi * y/4)) + (z^2 - 10\cos(\pi * z/4))$
Griewank [61]	$(x^2 + y^2)/4000 - \cos(x)\cos(y/\sqrt{2}) + 1$	not tested
Michalewicz [64]	not tested	$-\sin(x)\sin^2(x^2/\pi) - \sin(y)\sin^2(y^2/\pi) - \sin(z)\sin^2(z^2/\pi)$
"Double Well"	not tested	$(x + 4)(x + 1)(x - 3)(x - 6) + y^4 - 60y^2 + z^2$

Table 2: Testing function names and analytical forms.

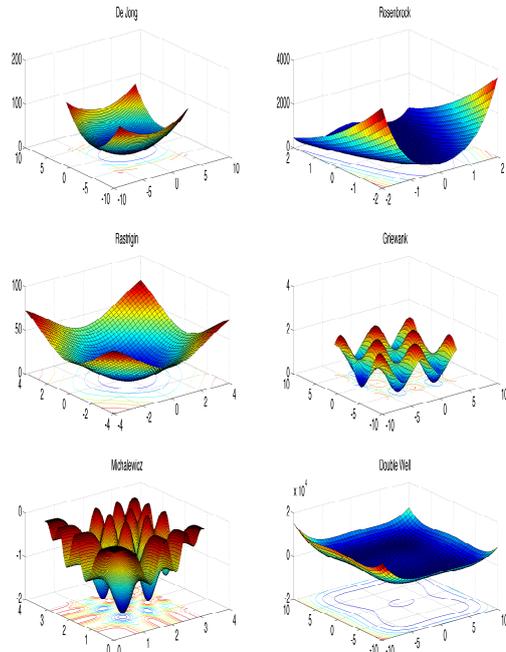


Figure 1: Visualized testing functions. Domain range for every dimension: De Jong  $(-6, 6)$ , Rosenbrock  $(-2, 2)$ , Rastrigin  $(-6, 6)$ , Griewank  $(-4, 4)$ , Michalewicz  $(0, \pi)$ , "Double Well"  $(-9, 9)$ .

given in table 2. There are three main aspects we have focused on during the testing: "shape" parameter ( $r$ ) sensitivity, accuracy and grid density dependence. Testing functions are visualized along  $x$  and  $y$  coordinates in figure 1, along with their domain boundaries. The errors were estimated on a  $20 \times 20$  regularly spaced grid for 2D and a  $10 \times 10 \times 10$  regularly spaced grid in 3D. The fi-

nal error for a test function, interpolation method, grid density and “shape” parameter was taken as the average over all the errors produced at each grid point. Twenty different grid densities have been used in 2D, ten in 3D. FEMVib is aimed to be a general solver without imposing any conditions on the surfaces density or shape, therefore each of the three aspects has been evaluated individually and based on these results an interpolation scheme has been selected for implementation.

### 3.1.1. Parameter Dependence

Inverse multiquadric and multiquadric RBFs have a strong parameter dependence. As we increase the value of  $r$  the error decreases for all the tested functions in 2D until an optimum value is reached. Once  $r$  is larger than the optimal value the average error increases and the error evaluation becomes very unstable figure 2, producing oscillations.

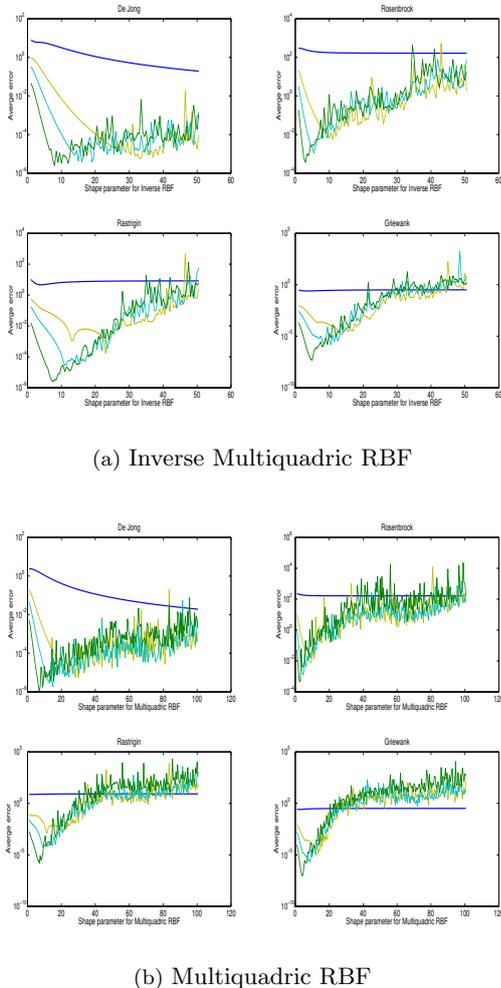
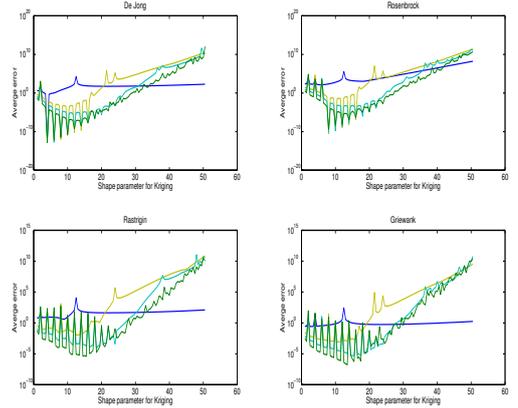
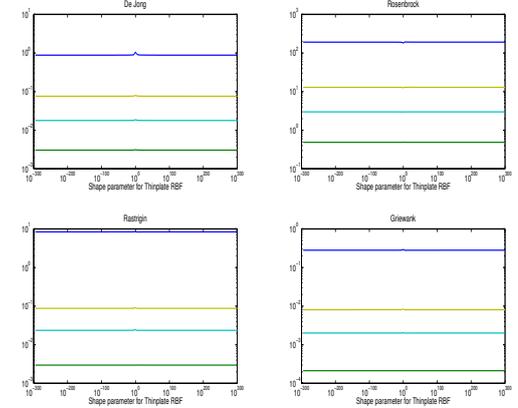


Figure 2: Average error vs “shape” parameter  $r$  for various grid densities. Blue line  $N = 16$  points, yellow line  $N = 81$  points, light blue line  $N = 225$  points, green line  $N = 484$  points.

As we increase the grid density the optimum value of  $r$  becomes smaller, except for the lowest grid density Rastrigin and Griewank functions evaluated using multiquadric RBF, where the average error stays constant for all parameter values tested. Despite the similarities in trends of the parameter functional forms for each testing function, no correlation with the grid density can be found. In other words, the optimum value of the “shape” parameter has a different behavior for different surfaces and does not relate to the average spacing on the grid.



(a) Kriging



(b) Thin-plate RBF

Figure 3: Average error vs “shape” parameter  $r$  for various grid densities. Blue line  $N = 16$  points, yellow line  $N = 81$  points, light blue line  $N = 225$  points, green line  $N = 484$  points.

Similar behavior for the “shape” parameter was observed for De Jong and Double Well in three dimensions. Both of these functions are polynomials and have a simple analytical formulation. For Rastrigin in 3D, two sharp minima are observed when inverse multiquadric interpolation is used and a single sharp minimum for multiquadric interpolation. The “shape” parameter dependence for Michalewicz is very different for inverse multiquadric and multi-

quadratic interpolation compared to other testing functions Appendix A. The optimum  $r$  value is close to 0 and as it increases the error starts growing. For high density grids the collocation matrix becomes degenerate and high oscillations can be observed. On the other hand for very small grid densities the error remains constant throughout the parameter space.

Kriging interpolation has the highest parameter sensitivity figure 3 a. The recommended range for the “shape” parameter is for  $r \in (1, 2)$  [48], but we have covered a much wider spectrum. For values  $r > 2$  the interpolation matrix is likely to become singular or degenerate, hence giving very high error. On the other hand for certain higher values of  $r$  the variogram will be very close to the actual testing function and the error would become extremely low. Such behavior is observed for all the tested functions, but higher amplitude oscillations are observed for De Jong and Rosenbrock, because of their simple analytical form in two dimensions. A similar effect is observed for De Jong and Double Well in three dimensions. The smallest error for all functions is obtained for  $r \in (1, 20)$ , yet even on this relatively short interval the error goes through very large oscillations that would have a major impact on the interpolation. For the Michalewicz testing function, used in 3D testing, the error oscillations are much smaller in the range  $r \in (1, 10)$ , compared to Griewank and Rastrigin in 2D and vanish for very low grid densities. Kriging interpolation is, despite its variogram adaptability to certain functional forms, the most sensitive with respect to parameter changes.

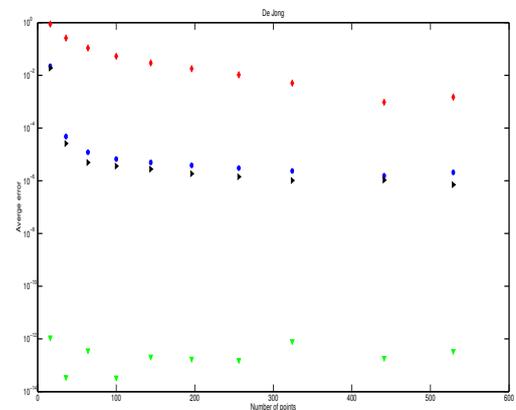
The thin-plate RBF is the most stable interpolation with respect to parameter changes figure 3 b. Because the “shape” parameter appears inside the logarithm of the shape function, we have scanned a very broad parameter space  $r \in (10^{-300}, 10^{300})$ . Minor changes in error estimation were observed for  $r$  close to 1. This, almost negligible, parameter sensitivity has put thin-plate RBF interpolation way ahead of other tested methods. Same behavior was observed for three dimensional testing cases where a small error deviation occurs close to  $r = 1$  and a very stable interpolation is obtained for the rest of the tested parameter space. Results for the three dimensional testing can be found in Appendix A.

### 3.1.2. Accuracy

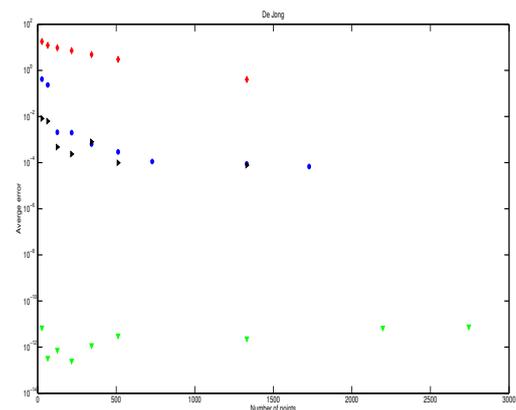
Results for accuracy are divided based on the testing function, which has the highest impact on a given tested interpolation scheme. Since the testing functions have been evaluated with each interpolation routine for different values of the “shape” parameter, our results show only the lowest error along the parameter space. This means that each result for accuracy involves parameter tuning for each function. The error estimation minima are plotted for different grid densities.

The first analyzed function is De Jong, for which the tests were performed in both two and three dimensions. The least accurate scheme for De Jong is the thin-plate

RBF. For the two dimensional case thin-plate improved the error three orders of magnitude over a 513 point range, starting at 16 points - lowest density for 2D. The second and third most accurate interpolation schemes were multiquadric and inverse multiquadric respectively. Multiquadric outperformed inverse multiquadric at each grid density in two dimensions and at most of the low grid densities in three dimensions. Their accuracy was almost identical for higher grid densities in three dimensions. Both of these interpolation schemes followed the same trend in two dimensions while significant differences could be observed in three, where the inverse multiquadric had a much higher error for lower grid densities than multiquadric and improved 4 orders of magnitude when 1700 point limit was reached. Results for two and three dimensional accuracy tests for De Jong are given in figure 4.



(a) De Jong 2D

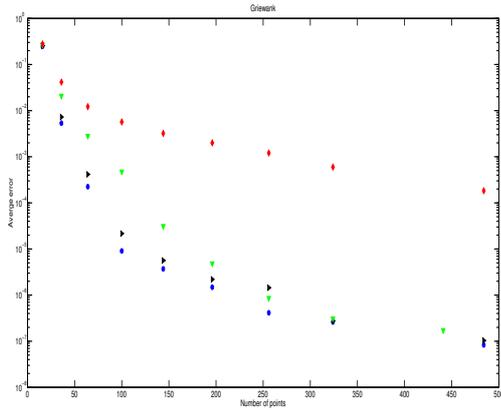


(b) De Jong 3D

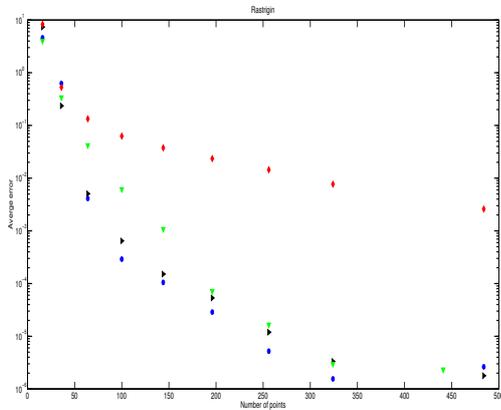
Figure 4: Minimum error for De Jong at various grid densities. Red diamonds represent thin-plate RBF, blue circles inverse multiquadric RBF, black right-triangles multiquadric RBF and green down-triangles Kriging.

Kriging interpolation is the most accurate. This is due to the variogram, which maps the spatial dependence with

the same order function as De Jong for certain values of the “shape” parameter. In this case the interpolation function becomes very close to its analytical representation. Hence the best interpolation scheme for such smooth and simple testing function as De Jong is the Kriging. Rosenbrock valley, tested only in two dimensions, yielded very similar results to De Jong in 2D with respect to accuracy, where thin-plate was the least accurate, multiquadric and inverse multiquadric were second with negligible differences between each other and Kriging came out as the best interpolation scheme for this type of surface. Results for Rosenbrock valley can be found in Appendix A.



(a) Griewank 2D

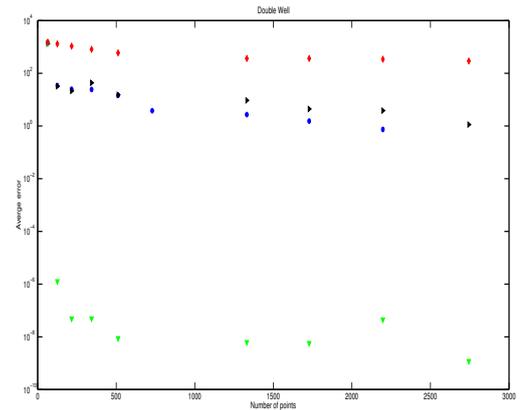


(b) Rastrigin 2D

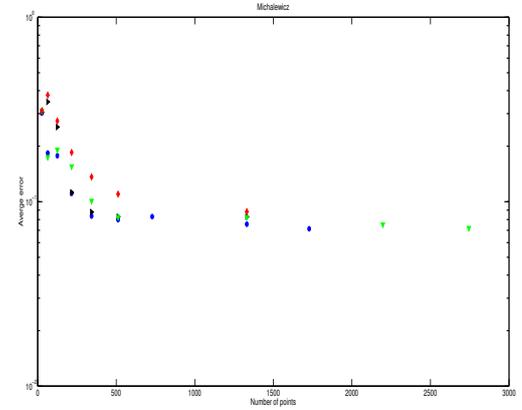
Figure 5: Minimum error for Griewank and Rastrigin testing functions in 2D at various grid densities. Red diamonds represent thin-plate RBF, blue circles inverse multi-quadric RBF, black right-triangles multi-quadric RBF and green down-triangles Kriging.

Different trend was observed for Griewank testing function as opposed to De Jong. Thin-plate RBF was the least accurate method. Because the analytical formulation of Griewank contains sine and cosine functions, Kriging’s variogram does not map the spatial dependence as well as

for simple polynomial functions, which makes it the third best interpolation scheme for this type of surface. However for high density grids Kriging interpolation was comparable to inverse multi-quadric and multi-quadric with high accuracy. Inverse multi-quadric and multi-quadric RBFs were the most accurate figure 5a, especially at low grid densities. Same trend as for Griewank was observed for Rastrigin, which was tested in two and three dimensions. Thin-plate was the least accurate, Kriging third followed by inverse multi-quadric and multi-quadric with the highest accuracy figure 5b.



(a) Double Well 3D

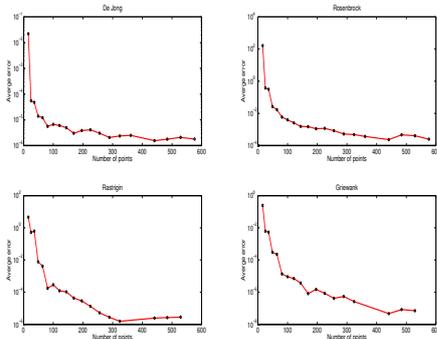


(b) Michalewicz 3D

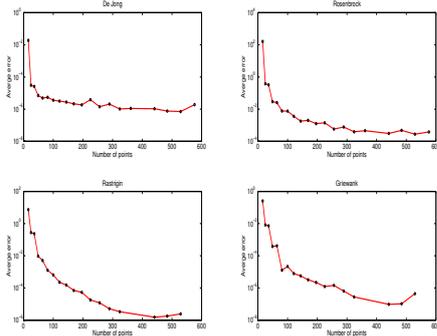
Figure 6: Minimum error for Double Well and Michalewicz testing functions in 3D at various grid densities. Red diamonds represent thin-plate RBF, blue circles inverse multi-quadric RBF, black right-triangles multi-quadric RBF and green down-triangles Kriging.

The last two of the tested functions were Michalewicz and Double Well, both in three dimensions. Scenario for Double Well, figure 6a, was similar to De Jong and Rosenbrock. Because the function has a polynomial character the variogram mapped the spatial dependence well, thus making the error for Kriging interpolation ten orders of

magnitude lower than the second best interpolation. Second and third best were inverse multiquadric and multiquadric respectively, followed by thin-plate RBF. Double Well is a fourth order polynomial along  $x$  and  $y$  and second order along  $z$  with very steep walls, which had an impact on the value of error that may have appeared too large compared to other surfaces. Michalewicz, figure 6b, was the most interesting surface out of all in terms of accuracy, where despite its sine and cosine analytical formulation, all of the interpolation schemes performed roughly the same starting from certain grid density;  $N = 700$  points on the surface. Before this point limit is reached the least accurate scheme is thin-plate RBF followed by multiquadric RBF. The second most accurate is Kriging, which is in contrast with the previous testing where sine or cosine like functions were better interpolated using multiquadric or inverse multiquadric RBFs. The most accurate scheme was the inverse multiquadric RBF. Again all of these observations are valid up to a certain grid density, from which the error was roughly the same for all the tested interpolation routines.



(a) Inverse Multiquadric RBF



(b) Multiquadric RBF

Figure 7: Average error vs grid density for inverse multiquadric and multiquadric RBFs. For each tested surface in 2D.

### 3.1.3. Grid Density Dependence

Grid density dependence determines the rate of convergence of each interpolation scheme based on the number of regularly distributed points along the surface. All the graphs in this section were evaluated at the optimum “shape” parameter value. Each surface has different domain boundaries which has a strong influence on the error as well as different analytical form, which may lead to significantly different absolute error estimates for different surfaces.

Two dimensional density dependence is depicted in figure 7, figure 8 and figure 9. For inverse multiquadric the most significant error decrease is within the first 200 points added to the surface figure 7. For De Jong the error dropped three orders of magnitude, for Rosenbrock more than four orders of magnitude change was achieved in this interval.

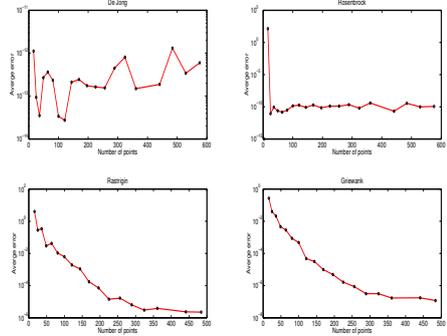


Figure 8: Average error vs grid density for Kriging. For each tested surface in 2D.

Compared to De Jong in three dimensions, both inverse multiquadric and multiquadric needed 500 points to achieve three orders of magnitude error improvement. The decrease for Griewank was almost six orders and Rastrigin more than five orders of magnitude in two dimensions. After the first 200 points the decrease in error slow down, following an exponential behavior. Multiquadric RBF followed the same pattern as inverse multiquadric figure 7b .

Kriging interpolation, as discussed earlier, does a mapping of the region using the variogram. For De Jong this led to an average error equal  $10^{-13}$  starting with 16 points along the surface. Roughly the same accuracy was achieved using 64 points in three dimensions. For Rosenbrock the variogram needed 25 points to bring the average error ten orders of magnitude to  $10^{-10}$ . Once the region was mapped for these two functions no error improvement with additional points was observed. In contrast to De Jong, after adding 513 points, Kriging improved only five orders of magnitude for Rastrigin and six orders of magnitude for Griewank figure 8. The overall performance of Kriging on all the surfaces is excellent, when the only drawback that remains is the parameter sensitivity.

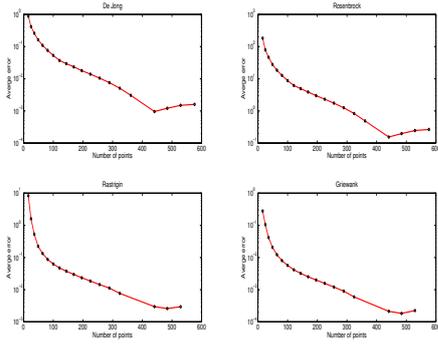


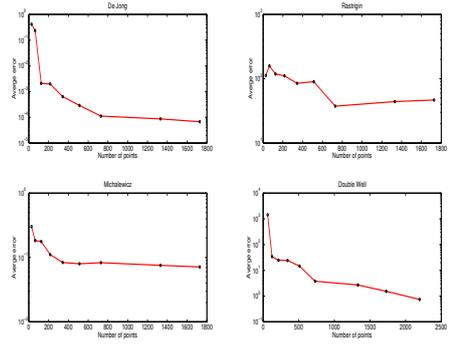
Figure 9: Average error vs grid density for thin-plate RBF. For each tested surface in 2D.

Thin-plate improved only two orders of magnitude after 200 points were added to the De Jong surface in 2D. After 441 points the error dropped three orders of magnitude from the initial value. Similar results were obtained for Rosenbrock, where the only difference was the absolute value of the estimated error. For both of the tested surfaces the error slightly increased passed 450 points. The error improved almost five orders of magnitude for Rastrigin in 2D, when 513 points were added and three order for Griewank over the same amount of points added to the surface in 2D figure 9.

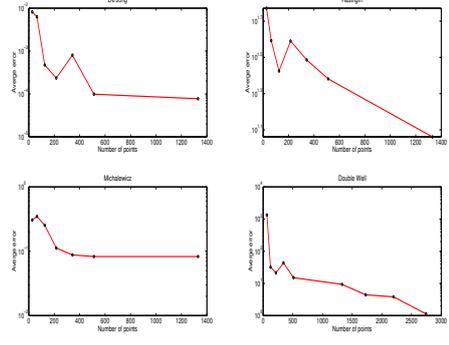
The rate of convergence for three dimensional surfaces was, as expected, much lower. For inverse multiquadric in 3D almost 600 points were needed for De Jong to lower the error three orders of magnitude. Error for Double Well decreased three orders over a 2700 point interval. Error changes with increasing number of points were very low for Rastrigin and Michalewicz in 3D. Error for these two functions did not improve more than an order of magnitude over a 1300 point range figure 10a. The rate of convergence for multiquadric on all four three dimensional surfaces was the same as for inverse multiquadric over 1600 point interval. The only difference was an increase in the error around 200 to 300 point range for De Jong, Michalewicz and Double Well figure 10b.

Kriging interpolation yielded similar results for De Jong in 3D as in 2D. Space mapping took less than 64 points after which the error for the surface did not improve with additional points. Michalewicz and Rastrigin testing surfaces in 3D where hard to interpolate and the error did not converge more than order of magnitude as it was with inverse and multiquadric. In addition the error did not converge continuously for Rastrigin but rather oscillatory. This surprising behavior is depicted in figure 11a. Error for Double Well converged, as expected, ten orders of magnitude, once the region was mapped. 150 points sufficed for the mapping and no significant change in error was observed after this limit.

Despite its low accuracy the thin-plate RBF converged consistently with increasing number of points at each sur-



(a) InverseMultiquadric RBF



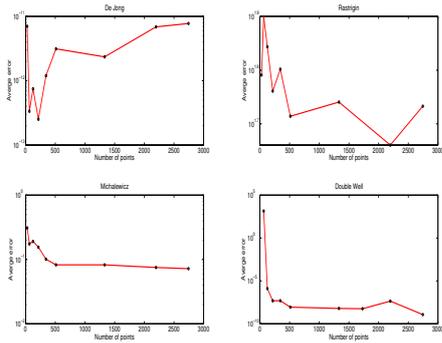
(b) Multiquadric RBF

Figure 10: Average error vs grid density for inverse multi-quadric and multiquadric RBF. For each tested surface in 3D.

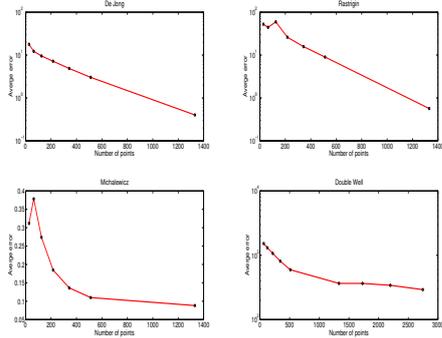
face. For De Jong the error improved more than an order of magnitude over a 1300 point range and two orders for Rastrigin. A surprisingly low error decrease was observed for Double Well, where the error did not change more than one order compared to at least three orders for all other interpolation methods figure 11b.

To find the best interpolation scheme for our purposes we had to compare all three aspects: “shape” parameter sensitivity, grid density dependence and accuracy; and decide on a winner. The highest parameter sensitivity is found to be for the Kriging interpolation. The multi-quadric and inverse multiquadric are second with respect to parameter sensitivity. Thin-plate is by far the least sensitive and hence most suitable interpolation based on parameter sensitivity.

The fastest converging interpolation scheme for polynomial type surfaces is Kriging interpolation. For more complicated analytical forms that include various powers of sine and cosine functions, multiquadric and inverse multiquadric RBFs have a higher convergence rate than Kriging for low density grids. Thin-plate RBF has the lowest convergence rate out of all the tested interpolation schemes. This makes thin-plate RBF the least desirable interpolation scheme with respect to the rate of conver-



(a) Kriging



(b) Thin-plate RBF

Figure 11: Average error vs grid density for Kriging and thin-plate RBF. For each tested surface in 3D.

gence. Despite Kriging converging slower for low density grids, for higher density grids it reached the convergence rate of both inverse multiquadric and multiquadric and converged extremely fast on the polynomial surfaces. We therefore rate it as the overall fastest converging interpolation scheme. The most accurate interpolation scheme is again Kriging, followed by inverse multiquadric and multiquadric RBFs. The least accurate is thin-plate RBF. In two out of the three tested aspects Kriging has proven to be the best, when the only drawback is its parameter sensitivity. We have therefore decided to implement it as our main interpolation method.

The parameter sensitivity for Kriging is a crucial aspect in terms of accuracy for FEMVib. This issue is addressed by a simple parameter tuning before the actual interpolation of the PES takes place. Initially 38 regularly distributed points along the parameter interval  $r \in (1, 20)$  are evaluated. After a minimum is found, 8 additional evaluations are made around the minimum taking 0.2 size steps in the parameter space. The parameter value, where the error reaches its minimum, is used for the actual interpolation. The only disadvantage associated with the tuning is computational time. Since the surface does not have an analytical representation, we used the “leave-one-out” method, used by Rippa [38] to calculate the error. This

method is used for small sets where a further partitioning would lead to distorted error estimates. In the leave-one-out method, one iterates through all  $N$  data points and at each iteration step leaves out one point for error estimation. The interpolation scheme is initialized without this point and its functional value is used as error estimate. After cycling through all the points on the grid the final error prediction is given as the average of these errors.

### 3.2. Wilson G-Matrix Calculator - GCAL

The results are divided into two and three dimensional examples, and further into structured and scattered grids. One dimensional vibrational analysis does not require Wilsons G-Matrix, due to the absence of coupling. All the studied structures were chosen so that their G-Matrix elements could be calculated analytically and hence compared to the results from our GCAL program. Tabulated G-Matrix elements for a 3 atom system are given in table 3, notation used in the table is based on the system depicted in figure 12 [45]. Note, only six elements are given in table 3 instead of 9, because of G-Matrix symmetry. In the results we use two different error measures: the first is the error on the grid which we get by comparing the absolute values at each point, the second is a percentage error. This percentage error represents the average error on the grid points weighted by the actual expected value, multiplied by 100%. Also, note that G-Matrix elements have units  $amu^{-1}$ , which is the inverse of atomic mass unit. The units will be skipped throughout this section. All the surfaces were calculated using Molpro [70].

G Matrix Element	Analytical Expression
$G_{11}$	$\mu_1 + \mu_3$
$G_{12}$	$\mu_3 \cdot \cos(\phi)$
$G_{13}$	$\frac{-\mu_3 \cdot \sin(\phi)}{r_{32}}$
$G_{22}$	$\mu_2 + \mu_3$
$G_{23}$	$\frac{-\mu_3 \cdot \sin(\phi)}{r_{31}}$
$G_{33}$	$\frac{\mu_1}{r_{21}^2} + \frac{\mu_2}{r_{32}^2} + \mu_3 \left( \frac{1}{r_{31}^2} + \frac{1}{r_{32}^2} - \frac{2\cos(\phi)}{r_{31}r_{32}} \right)$

Table 3: Tabulated G-Matrix elements for a 3 atom system. Elements corresponding to system depicted in figure 12.

The first tested system is  $BeF_2$ . Two internal coordinates have been changed:  $Be - F$  stretch and  $F - Be - F$  bend. A regular structured grid has been used, where  $Be - F$  ranged from 0.8 to 1.6 Å taking 0.2 Å steps and  $F - Be - F$  ranged from 2.79 to 3.66 rad, taking 0.145 rad steps, which resulted into a  $5 \times 6 = 30$  point surface.

Results for  $BeF_2$  are given in figure 13. The average error for  $G_{11}$  is 0.032, which represents 18%. For  $G_{13}$  the average error is 0.0067, which might seem high in terms of percentage 605%, but with respect to the density of the surface, numerical precision and its low contribution to the kinetic energy term, is still very good, considering

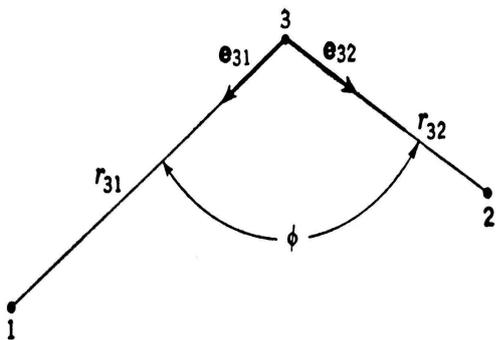


Figure 12: Nonlinear triatomic system. Taken from [45]. Where  $r_{31}$  is the distance between atoms 1 and 3,  $r_{32}$  is the distance between atoms 2 and 3 and  $\phi$  is the 1-3-2 bend angle.

the magnitude of  $G_{11}/G_{12}$ . The average error for  $G_{33}$  is 0.0304. This is 32% average error based on the expected values for each point.

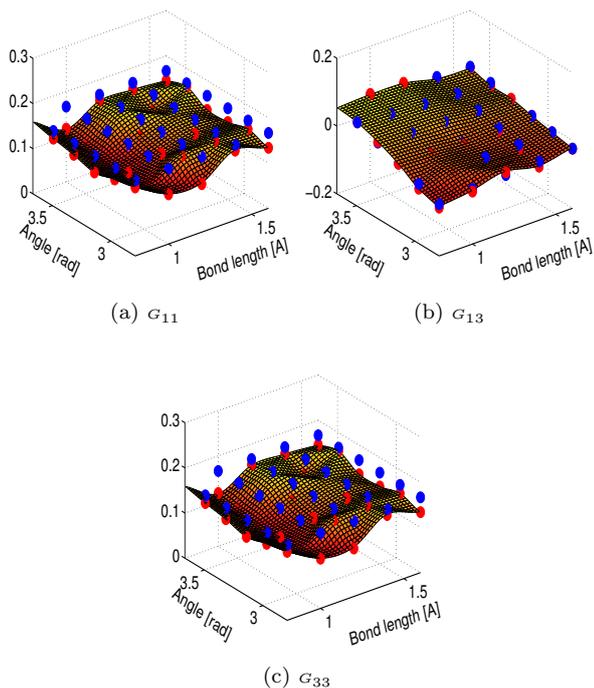


Figure 13: Calculated  $\text{BeF}_2$  G-Matrix surfaces marked with red dots and analytical values marked with blue dots for  $G_{11}$ ,  $G_{13}$  and  $G_{33}$  corresponding to  $\text{Be}-\text{F}$  stretch and  $\text{F}-\text{Be}-\text{F}$  bend.

The second tested system is  $\text{CO}_2$ . Two internal coordinates have been changed corresponding to  $\text{C}-\text{O}$  stretches. We have calculated two different surfaces, one is structured the other is scattered. The structured surface is from  $0.532\text{\AA}$  to  $1.765\text{\AA}$  along the first stretch and from  $0.583\text{\AA}$  to  $1.832\text{\AA}$  along the second stretch. The spacing

for the points is not constant along the surface. Similar boundaries applied for the scattered surface:  $0.55\text{\AA}$  to  $1.543\text{\AA}$  along the first and  $0.501\text{\AA}$  to  $1.850\text{\AA}$  along the second stretch. Surfaces for each G-Matrix are depicted in figure 14.

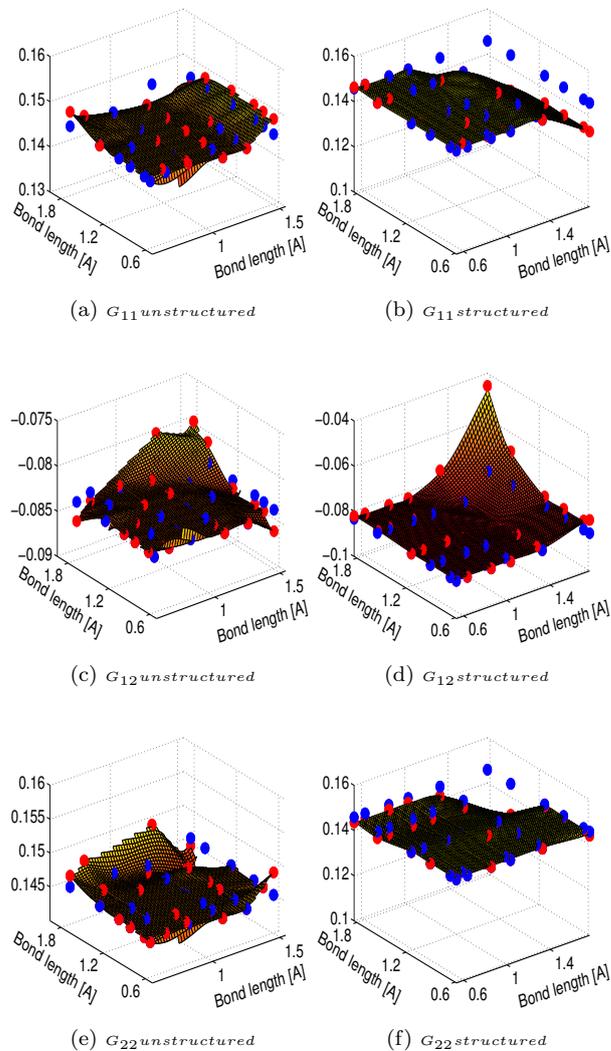


Figure 14: Calculated  $\text{CO}_2$  G-Matrix surfaces on scattered and structured grids marked with red dots and analytical values marked with blue dots for  $G_{11}$ ,  $G_{12}$  and  $G_{22}$  corresponding to  $\text{C}-\text{O}$  stretches.

Results for both types of grid are very accurate, which might be partly caused by the linearity of the molecule and hence simplicity of the calculation. For unstructured grid the average error on the grid points for  $G_{11}$  is  $7.512 \cdot 10^{-4}$  which corresponds to 0.51% error. Errors at  $G_{12}$  and  $G_{22}$  are  $4.89 \cdot 10^{-4}$  and  $4.98 \cdot 10^{-4}$  respectively. This represents 0.58% and 0.34% error. The average error for  $G_{11}$  on the structured grid is 0.00328, which is 2.25%. Average errors for  $G_{12}$  and  $G_{22}$  were 0.0024 and 0.00205, which in terms of error percentages comes out to 2.88% and 1.4%. Because the surfaces are small in terms of the number of calculated

points, the entire grid was used for the SD-IMLS derivative evaluation.

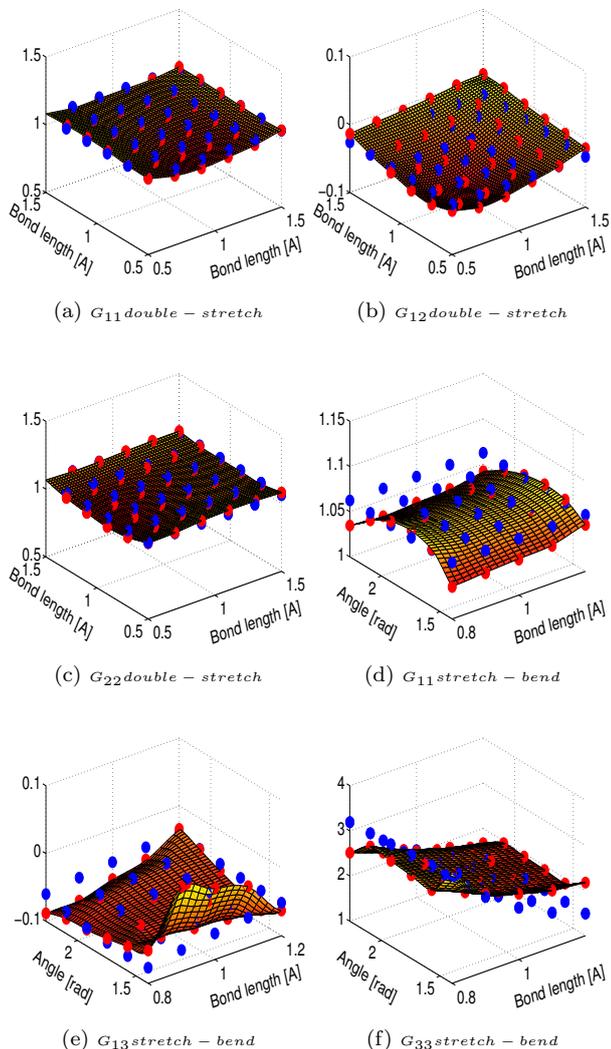


Figure 15: Calculated H<sub>2</sub>O G-Matrix surfaces on double-stretch and stretch-bend surfaces marked with red dots and analytical values marked with blue dots.

The third studied system was water H<sub>2</sub>O. When developing GCAL, water was the first system we have investigated. Because of its symmetry and nonlinearity, translation and rotation of Cartesian coordinates alone does not lead to the correct results. The first problem is the initial orientation. Molpro [70], does not align the optimized structures along the same set of axis. Therefore a “pre-rotation” has to be used, where all the structures are aligned as close as possible to the reference structure. After the second Eckart condition is applied a “post-rotational” routine aligns the structures with the reference structure once again. If this were not done some of the structures despite satisfying the Eckart conditions would not be aligned the correct way with the reference.

The first surface for H<sub>2</sub>O was calculated along the two

stretches  $H - O$ . We have used regular and structured square grid comprised of 36 points, starting at  $0.5\text{\AA}$ , and taking  $0.2\text{\AA}$  steps up to  $1.5\text{\AA}$ . The average error on the grid points for  $G_{11}$  is 0.024 which represents 2.32% error. For  $G_{12}$  the average error on the grid points was 0.015, which in terms of percentage is 59%, however with respect to its magnitude and contribution to the kinetic energy this error is negligible. Again the ration of  $G_{11} : G_{12}$  is roughly 100 : 1, which would make the 59% error for  $G_{12}$  a 0.59% absolute error for the kinetic energy. The last G-Matrix element surface  $G_{22}$  had a grid error of 0.024 corresponding to 2.32%.

The other water surface was calculated changing the stretch  $H - O$  and bend  $H - O - H$ . The stretch extended from  $0.8\text{\AA}$  to  $1.2\text{\AA}$  taking a step of  $0.1\text{\AA}$ . The bend went from 1.39 rad to 2.26 rad in 0.17 rad steps, which created a regular and structured 30 point grid. The errors on grid points for  $G_{11}$ ,  $G_{13}$  and  $G_{33}$  were 0.011, 0.017 and 0.29 respectively. In terms of percentage and contribution to the kinetic energy the first two errors were low 1.09% for  $G_{11}$  and 29.8% for  $G_{13}$ . The error for  $G_{33}$  was 15.21%, relatively high due to high absolute value of the error on the grid points. Results for both water surfaces are given in figure 15.

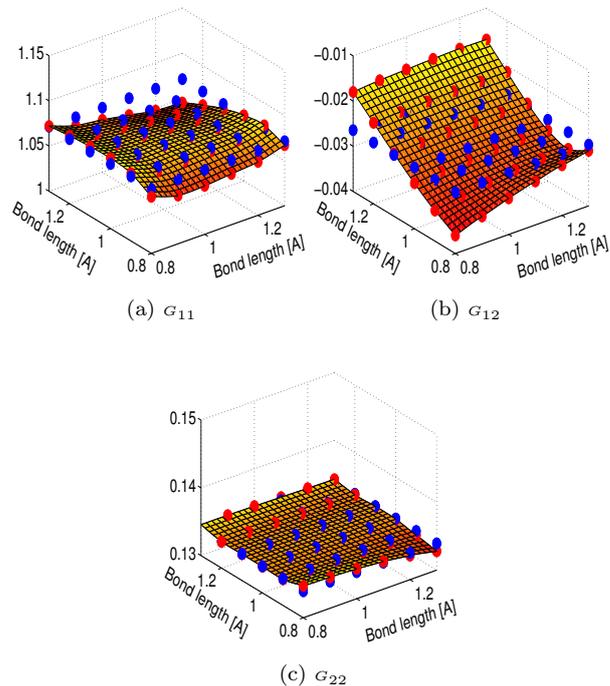


Figure 16: Calculated HNO G-Matrix surfaces marked with red dots and analytical values marked with blue dots for  $G_{11}$ ,  $G_{12}$  and  $G_{22}$  corresponding to  $H - N$  stretch and  $N - O$  stretch.

The last 2D system was HNO. Nitroxyl is a planar molecule and has no central atom symmetry. The surface was calculated along the two stretches  $H - N$  and  $N - O$ .

The grid was regular and structured composed of 36 points starting at  $0.8\text{\AA}$  taking  $0.1\text{\AA}$  steps and ending at  $1.3\text{\AA}$ . Because of the system being composed of different atoms the  $G_{11}$  and  $G_{22}$  matrices are different, which was not the case for previously studied symmetrical systems. The error for  $G_{11}$  on the grid was 0.008, which represents 0.75%. Grid error for  $G_{12}$  was 0.004 corresponding to 17.24% and for  $G_{22}$ , GCAL achieved a grid error equal to  $2.44 \cdot 10^{-4}$ , which comes up to 0.18%. As in the case of water and beryllium fluoride  $G_{12}$  for nitroxyl contributes with only 2% to the kinetic energy term compared to  $G_{11}$ , hence the overall error for the kinetic energy from this G-Matrix element would be less than 0.4% figure 16.

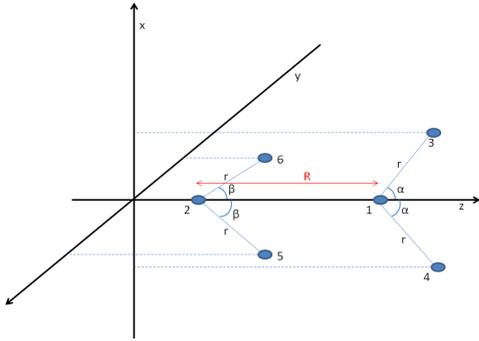


Figure 17: Artificial 3D system used for GCAL testing.

The last tested system was three dimensional artificially constructed molecule figure 17. Internal coordinates we have changed are the 5–2–6 angle  $2\beta$ , the 3–1–4 angle  $2\alpha$  and the bond distance 1–2 denoted as  $R$ . Notation  $2\alpha$  and  $2\beta$  is used to simplify the relations between Cartesian and internal coordinates. According to our model, the analytical expressions for Cartesian coordinates in terms of internals are given in table 4.

Atom	x	y	z	Mass
1	0	0	$\frac{R}{2} + \frac{1}{12}r(\cos(\beta) - \cos(\alpha))$	12
2	0	0	$-\frac{R}{2} + \frac{1}{12}r(\cos(\beta) - \cos(\alpha))$	12
3	$r \cdot \sin(\alpha)$	0	$\frac{R}{2} + \frac{1}{12}r(\cos(\beta) - \cos(\alpha)) + r \cdot \cos(\alpha)$	1
4	$-r \cdot \sin(\alpha)$	0	$\frac{R}{2} + \frac{1}{12}r(\cos(\beta) - \cos(\alpha)) + r \cdot \cos(\alpha)$	1
5	0	$r \cdot \sin(\beta)$	$\frac{R}{2} + \frac{1}{12}r(\cos(\beta) - \cos(\alpha)) + r \cdot \cos(\beta)$	1
6	0	$-r \cdot \sin(\beta)$	$\frac{R}{2} + \frac{1}{12}r(\cos(\beta) - \cos(\alpha)) + r \cdot \cos(\beta)$	1

Table 4: Analytical expressions for Cartesian coordinates in terms of internal coordinates based on figure 17.

It is simple to derive the analytical expression for the inverse G-Matrix elements.

$$\begin{aligned}
 G_{RR}^{inv} &= 7 \\
 G_{R\alpha}^{inv} &= -r \cdot \sin(\alpha) \\
 G_{R\beta}^{inv} &= r \cdot \sin(\beta) \\
 G_{\alpha\alpha}^{inv} &= 2r^2 \cdot \cos^2(\alpha) + \frac{67}{36}r^2 \cdot \sin^2(\alpha) \\
 G_{\beta\beta}^{inv} &= 2r^2 \cdot \cos^2(\beta) + \frac{67}{36}r^2 \cdot \sin^2(\beta) \\
 G_{\alpha\beta}^{inv} &= -\frac{7}{36}r^2 \cdot \sin(\alpha) \cdot \sin(\beta)
 \end{aligned}$$

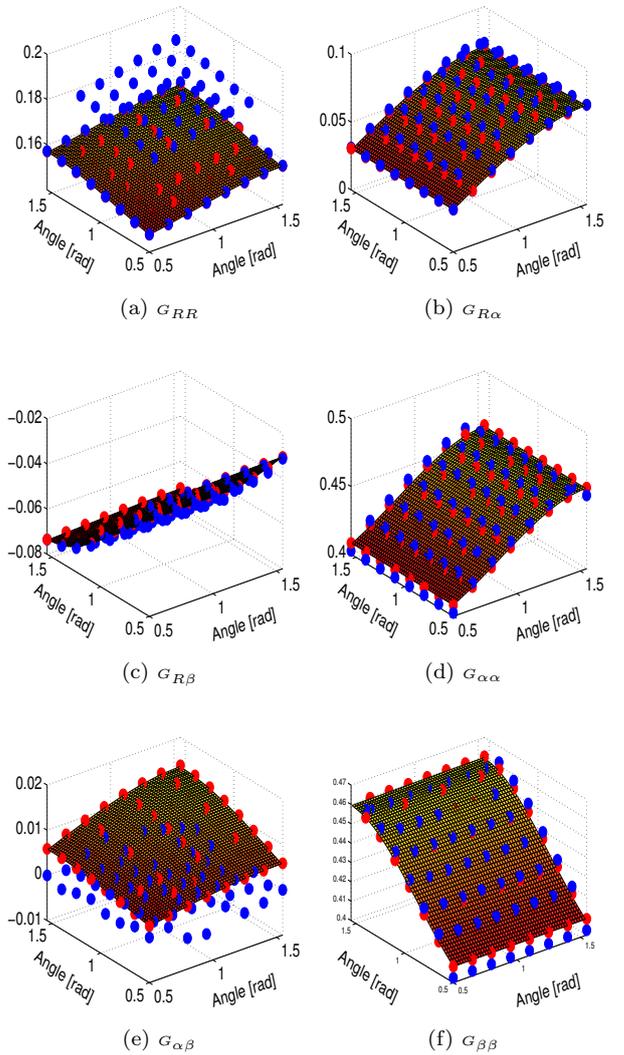


Figure 18: Calculated G-Matrix surfaces for artificial system (figure 17), for  $R = 0.5$ . Calculated values are marked with red dots; analytical values marked with blue dots.

Knowing the functional dependence of Cartesian on internal coordinates and also the inverse G-Matrix, we have used this model to test the GCAL program. The artificial

surfaces was generated along the following boundaries:  $R$  ranged from 0.5 to 1.55 while taking 8 steps of 0.15 size;  $\alpha$  and  $\beta$  had the same range and steps, however this time the values represented an angle in radians. This produced an  $8 \times 8 \times 8 = 512$  point regular structured surface. Average errors on the grid and average percent errors are given in table 5. The scenario for  $G_{\alpha\beta}$  is the same as for the two dimensional systems with small G-Matrix elements that had a relatively high error. The average ratio of  $G_{\beta\beta} : G_{\alpha\beta}$  is 0.0059, hence the contribution of  $G_{\alpha\beta}$  to the kinetic energy compared to  $G_{\beta\beta}$  is 0.5%. Therefore the 81% error will become roughly a 0.4% error on global scale considering the kinetic energy. Figure 18, depicts the G-Matrix surfaces for  $R = 0.5$ .

G Matrix Element	Grid Error	Percent Error
$G_{RR}$	0.0019	1.17 %
$G_{R\alpha}$	0.002	4.33 %
$G_{R\beta}$	0.0016	3.66 %
$G_{\alpha\alpha}$	0.0053	1.32 %
$G_{\alpha\beta}$	0.008	81.45 %
$G_{\beta\beta}$	0.0042	1.07 %

Table 5: Grid errors and percent errors for Artificial system (figure 17) at every G-Matrix surface.

*GCAL* proved to be a very accurate G-Matrix calculator on all the tested systems. Disadvantages associated with the program lie in the complexity of the Cartesian surfaces. If the surfaces are too sparse or have an overall low average values across the surface, the interpolation of the surface and consequent derivatives estimates might be inaccurate. Another problem associated with the surfaces is their filtering. After each G-Matrix surface is constructed a filtering routine has to remove the singular values, which can occur due to structural similarities for significantly different internal coordinates. These are usually represented by torsional motions, in which the structure after a 360 degree rotation can have a similar Cartesian structure but be represented by different internal coordinates.

#### 4. Conclusions

Four different interpolation schemes suitable for scattered data sets have been tested on six different surfaces, with different dimensionality, grid density and parameter tuning. Based on this study the Kriging interpolation method was chosen as the best and implemented into the *FEMVib* package. Before the PES is interpolated a parameter tuning takes place and a parameter with the lowest error on the surface is chosen for the actual interpolation. Based on the advancements in PES evaluation in the past four years the IMLS routine is also part of the *FEMVib* package, where both options are available to the user.

A novel G-Matrix calculator *GCAL* was developed and

implemented as a part of the *FEMVib* package for kinetic energy evaluation. *GCAL* removes axis inconsistencies produced by the ab-initio electronic calculations and does a pre-rotation with respect to the reference structure. This process aligns all the structures with the reference before translation or rotation are applied. Eckart conditions are applied and all the structures are numerically rotated. A post-rotational modification assures that the structures are once again aligned with the reference. This step is important for small symmetric molecules, where multiple ways of satisfying the second Eckart condition exist. The derivatives of Cartesian with respect to internal coordinates are evaluated at the grid points using the SD-IMLS. The program was tested on six different surfaces, in two and three dimensions using different types of grids. *GCAL* proved to be very accurate and general calculator for G-Matrices and further kinetic energy operator assembly.

The finite element method based solver for the Vibrational Schödinger equation, *FEMVib*, has been fully automated and made publicly available as a web service, so that groups around the world focused on theoretical spectroscopy can use it for analysis of their vibrational spectra. Input to the program consists of the PES and corresponding geometries in Cartesian coordinates. The output are the vibrational eigenstates and the corresponding wavefunctions. A detailed outline of the input options and user interfaces can be found in appendix B.

#### 5. Acknowledgements

This work was supported by National Research Foundation, Contract No. CHE-0947087. And by the Army Research Office, Contract No. 57989-CH.

## References

- [1] D.Xu, J.Stare and A.L.Cooksy, *Solving the vibrational Schrödinger equation on an arbitrary multidimensional potential energy surface by the finite element method*, Computer Physics Communications, 180, 11, 2079-2094, 2009
- [2] A.L. Cooksy, J.K.G. Watson, C.A. Gottlieb and P. Thaddeus *The Structure of the HCCCO Radical: Rotational Spectra and Hyperfine Structure of Monosubstituted Isotopomers*, J. Chem. Phys., 101, 178-186, 1994
- [3] A.L. Cooksy, *Conjugated  $\Pi$ -Electron Systems in Reactive Molecules: Multiple Minima on  $^2A'$  Potential Surfaces of Carbon Chain Free Radicals*, J. Am. Chem. Soc., 117, 1098-1104, 1995
- [4] R. Dawes, D.L. Thompson, Y. Guo, A.F. Wagner, M. Minkoff, *Interpolating moving least-squares methods for fitting potential energy surfaces: Computing high-density potential energy surface data from low-density ab initio data points*, J. Chem. Phys. 126, 184108, 2007
- [5] G.G. Maisuradze, D.L. Thompson, A.F. Wagner, M. Minkoff, *Interpolating moving least-squares methods for fitting potential energy surfaces: Detailed analysis of one-dimensional applications*, J. Chem. Phys. 119, 19, 2003
- [6] G.G. Maisuradze, A. Kawano, D.L. Thompson, A.F. Wagner, M. Minkoff, *Interpolating moving least-squares methods for fitting potential energy surfaces: Analysis of an application to a six-dimensional system*, J. Chem. Phys. 121, 21, 2004
- [7] A. Kawano, I.V. Tokmakov, D.L. Thompson, A.F. Wagner, M. Minkoff, *Interpolating moving least-squares methods for fitting potential energy surfaces: Further improvement of efficiency via cutoff strategies*, J. Chem. Phys. 124, 054105, 2006
- [8] G.G. Maisuradze, D.L. Thompson, *Interpolating moving least-squares methods for fitting potential energy surfaces: Illustrative approaches and applications*, J. Chem. Phys. 107, 7118-7124, 2003
- [9] D.H. McLain, *Drawing contours from arbitrary data points*, Comput. J., 17, 318., 1974
- [10] M.D. Buhmann, *Radial Basis Functions: Theory and implementations*, Cambridge University Press, 2003
- [11] G.L.G.Slejpen and H.A.V.D.Vorst, *A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems*, SIAM Journal on Matrix Analysis and Applications, 17, 401, 1996
- [12] E.H. Isaaks, R.M. Srivastava *An Introduction to Applied Geostatistics*, Oxford University Press, p.278-322, 1989
- [13] J.T. Hougen, *Rotational Energy Levels of a Linear Triatomic Molecule in a  $^2\Pi$  Electronic State*, J. Chem. Phys. 36, 519, 1962
- [14] Kevin M. Dunn, James E. Boggs, and Peter Pulay, *Vibrational energy levels of methyl fluoride* J. Chem. Phys. 86, 5088, 1987
- [15] J. K. G. Watson, *Simplification of the molecular vibration-rotation hamiltonian*, Mol. Phys. 15, 479, 1968
- [16] J.K.G.Watson, *The vibration-rotation hamiltonian of linear molecules*, Mol. Phys. 19, 465, 1970
- [17] Kevin M. Dunn, James E. Boggs, and Peter Pulay, *Vibrational energy levels of methyl fluoride*, J. Chem. Phys. 86, 5088, 1987
- [18] Joon O. Jung and R. Benny Gerber, *Vibrational wave functions and spectroscopy of  $H_2O_n$ ,  $n=2,3,4,5$ : Vibrational self-consistent field with correlation corrections*, J. Chem. Phys. 105, 10332, 1996
- [19] Stuart Carter, Susan J. Culik, and Joel M. Bowman, *Vibrational self-consistent field method for many-mode systems: A new approach and application to the vibrations of CO adsorbed on Cu(100)*, J. Chem. Phys. 107, 10458, 1997
- [20] Stuart Carter and Joel M. Bowman, *The adiabatic rotation approximation for rovibrational energies of many-mode systems: Description and tests of the method*, Stuart Carter and Joel M. Bowman, J. Chem. Phys. 108, 4397, 1998
- [21] Tamar Seideman and William H. Miller, *Quantum mechanical reaction probabilities via a discrete variable representation-absorbing boundary condition Green's function*, J. Chem. Phys. 97, 2499, 1992
- [22] Matthew J. Bramley and Tucker Carrington, Jr., *A general discrete variable method to calculate vibrational energy levels of three- and four-atom molecules*, J. Chem. Phys. 99, 8519, 1993
- [23] David Luckhaus, *6D vibrational quantum dynamics: Generalized coordinate discrete variable representation and adiabatic contraction*, J. Chem. Phys. 113, 1329, 2000
- [24] David Luckhaus, *The vibrational spectrum of HONO: Fully coupled 6D direct dynamics* J. Chem. Phys. 118, 8797, 2003
- [25] Joel M. Bowman, S. Carter, X. Huang, *MULTIMODE: A code to calculate rovibrational energies of polyatomic molecules*, Int. Rev. Phys. Chem., 22, 3, 533-549, 2003
- [26] O.L.Polyansky, A.G.Csaszar, S.V.Shirin, N. F. Zobov, P. Barletta, J. Tennyson, D.W. Schwenke and P. Knowles, *High-accuracy ab Initio Rotation-Vibration transitions for water*, Science, 299, 5606, 539-542, 2003
- [27] Paolo Barletta, Sergei V. Shirin, Nikolai F. Zobov, Oleg L. Polyansky, Jonathan Tennyson, Edward F. Valeev, and Attila G. Csaszar, *CVRQD ab initio ground-state adiabatic potential energy surfaces for the water molecule*, J. Chem. Phys. 125, 204307, 2006
- [28] D.M. Hirst, *Potential Energy Surfaces*, Taylor and Francis, London, 1985
- [29] M.A. Collins, *Molecular potential-energy surfaces for chemical reaction dynamics.*, Theor. Chem. Acc., 108, 313, 2002
- [30] O. Tishchenko, D.G. Truhlar, *Gradient-based multiconfiguration Shepard interpolation for generating potential energy surfaces for polyatomic reactions*, J. Chem. Phys., 132, 084109, 2010
- [31] G.C. Schatz, *The analytical representation of electronic potential-energy surfaces*, Rev. Mod. Phys., 61, 669 1989
- [32] R.L. Hardy, *Multiquadric equations of topography and other irregular surfaces*, J. Geophys. Res., 76, 1905-1915, 1971
- [33] R. Franke, *Scattered data interpolation: Tests of some methods*, Math. Comp., 38, 181-200, 1982
- [34] J.S. Wright, J. Polowin, *Energy-conserving trajectories in bond-angle-bond-distance coordinate systems*, Chem. Phys. Let., 131, 3, 256-260, 1986
- [35] J.N.L. Connor, *Reactive molecular collision calculations*, Comp. Phys. Comm., 17, 1-2, 117-143, 1979
- [36] R.P.A. Bettens, M.A. Collins, *Learning to interpolate molecular potential energy surfaces with confidence: A Bayesian approach*, J. Chem. Phys., 111, 3, 1999
- [37] C.A. Micchelli, *Interpolation of scattered data: distance matrices and conditionally positive definite functions*, Constr. Approx. 2, 11, 1986
- [38] S. Rippa, *An algorithm for selecting a good value for the parameter  $c$  in radial basis function interpolation.*, Adv. Comput. Math., 11, 193210, 1999
- [39] C.J. Trahan, R.E. Wyatt, *Radial basis function interpolation in the quantum trajectory method: optimization of the multiquadric shape parameter*, J. Comp. Phys., 185, 27-49, 2003
- [40] C. Edwards, D. Penney *Calculus and Analytical Geometry*, Englewood Cliffs, N.J.: Prentice Hall, 1982
- [41] D. Shepard, *A two-dimensional interpolation function for irregularly-spaced data*, Proceedings of the 1968 ACM National Conference., pp. 517524, 1968
- [42] M. A. Suhm, *Reliable determination of multidimensional analytical fitting bias.*, Chem. Phys. Lett. 223, 474-480, 1994
- [43] M.R. Salazar, *A completely general methodology for fitting three-dimensional ab initio potential energy surfaces*, Chem. Phys. Let., 359, 460-465, 2002
- [44] A. Frishman, D.K. Hoffman and D.J. Kouri, *Distributed approximating functionals fit of the  $H_3$  ab initio potential energy data of Liu and Siegbahn*, J. Chem. Phys., 107, 3, 804, 1997
- [45] E.B. Willson Jr., J.C. Decius, P.C. Cross, *Molecular Vibrations: The theory of infrared and Raman vibrational spectra*, McGraw-Hill Book Company, 1955
- [46] J. Stare, G.G. Balint-Kurti, *Fourier Grid Hamiltonian Method for Solving the Vibrational Schrödinger Equation in Internal Coordinates: Theory and Test Applications*, J. Phys. Chem. A, 107, 7204-7214, 2003
- [47] E.B. Willson, *A Method of Obtaining the Expanded Secular*

- Equation for the Vibration Frequencies of a Molecule.*, J. Chem. Phys., 7, 1047, 1939
- [48] H. Wackernagel, *Multivariate Geostatistics*, 2nd edition, Berlin, Springer-Verlag, 2003
- [49] J.E. Hadder, J.H. Frederick, *Molecular Hamiltonians for Highly Constrained Model Systems*, J. Chem. Phys. 1992, 97, 3500-3520.
- [50] C. Eckart, *Some studies concerning rotating axes and polyatomic molecules*, Phys. Rev., 47, 552-558, 1935
- [51] V. Alexandrov, D.M.A.Smith, H.Rostkowska, M.J.Nowak, L.Adamowicz, W.McCharty, *Theoretical study of the O-H stretching band in 3-hydroxy-2-methyl-4-pyrone*, J. Chem. Phys., 108, 9685-9693, 1998
- [52] W.J. McCarthy, L.Lapinski, M.J.Nowak, L. Adamowicz, *Out-of-plane vibrations of NH<sub>2</sub> in 2-aminopyrimidine and formamide*, J. Chem. Phys., 108, 10116-10128, 1998
- [53] I. Matanovic, N. Doslic, *Infrared spectroscopy of the intramolecular hydrogen bond in acetylacetone: a computational approach*, J. Phys. Chem. A, 109, 4185-4194, 1995
- [54] J.H. Frederick, C. Woywod, *General formulation of the vibrational kinetic energy operator in internal bond-angle coordinates.*, J. Chem. Phys., 111, 7255-7271, 1999
- [55] B. Podolsky, *Quantum-mechanically correct form of Hamiltonian function for conservative systems.*, Phys. Rev., 32, 812-816, 1928
- [56] T.B. Malloy, *Far-infrared spectra of ring compounds: a semi-rigid model for the ring-puckering vibration in some pseudo-four-membered ring molecules.*, J. Mol. Spectrosc., 44, 504-535, 1972
- [57] J. Stare, *First-principle calculation of reduced masses in vibrational analysis using generalized internal coordinates: Some crucial aspects and examples*, J. Chem. Inf. Model., 47, 840-850, 2007
- [58] J.T. Pearson, *Variable Metric Method of Minimization*, Computer Journal, 12, 1969
- [59] B.S.Kirk, J.W.Peterson, R.H.Stogner and G.F.Carey, *A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations*, Engineering with Computers, 22, 3-4, 237-254, 2006
- [60] K. A. De Jong, *An analysis of the behavior of a glass of genetic adaptive systems*, PhD thesis, University of Michigan, 1975
- [61] A.O. Griewank, *Generalized Decent for Global Optimization.*, J. Opt. Th. Appl., 34, 11-39, 1981
- [62] H.H. Rosenbrock, *An Automatic Method for Finding the Greatest or Least Value of a Function.*, Computer J. 3, 175-184, 1960.
- [63] A. Torn, A. Zilinskas *Global optimization*, Berlin, New York, Springer-Verlag, 1989
- [64] Z. Michalewicz, *Genetic Algorithms + Data structures = Evolution programs*, Springer-Verlag, Second Edition, 1994
- [65] [pyparse.sourceforge.net/](http://pyparse.sourceforge.net/)
- [66] <http://www.nbc.net/software/opal/>
- [67] <http://www.java.com/en/>
- [68] <http://ant.apache.org/>
- [69] <http://tomcat.apache.org/>
- [70] H.-J. Werner and P. J. Knowles and G. Knizia and F. R. Manby and M. Schütz and P. Celani and T. Korona and R. Lindh and A. Mitrushenkov and G. Rauhut and K. R. Shamasundar and T. B. Adler and R. D. Amos and A. Bernhardsson and A. Berning and D. L. Cooper and M. J. O. Deegan and A. J. Dobbyn and F. Eckert and E. Goll and C. Hampel and A. Hesselmann and G. Hetzer and T. Hrenar and G. Jansen and C. Köppl and Y. Liu and A. W. Lloyd and R. A. Mata and A. J. May and S. J. McNicholas and W. Meyer and M. E. Mura and A. Nicklass and D. P. O'Neill and P. Palmieri and K. Pflüger and R. Pitzer and M. Reiher and T. Shiozaki and H. Stoll and A. J. Stone and R. Tarroni and T. Thorsteinsson and M. Wang and A. Wolf, MOLPRO, version 2010.1, a package of ab initio programs, <http://www.molpro.net>

## Appendix A.

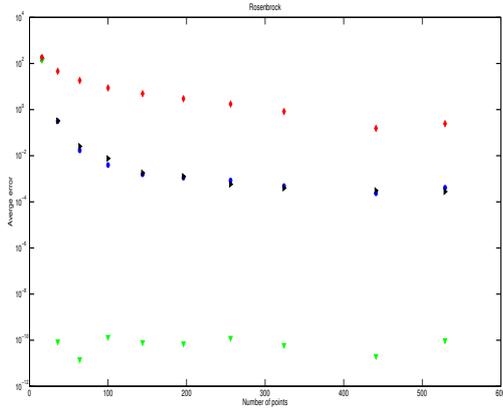


Figure A.19: Minimum error for Rosenbrock testing function in 2D at various grid densities. Red diamonds represent thin-plate RBF, blue circles inverse multiquadric RBF, black right-triangles multiquadric RBF and green down-triangles Kriging.

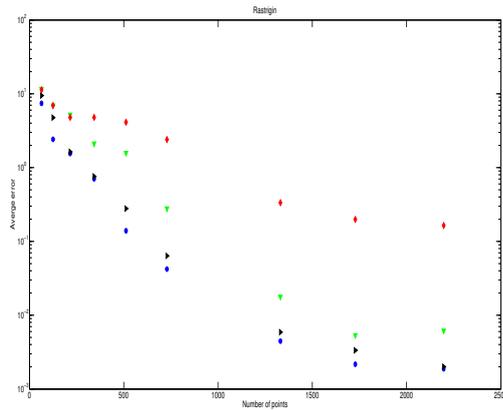


Figure A.20: Minimum error for Rastrigin testing function in 3D at various grid densities. Red diamonds represent thin-plate RBF, blue circles inverse multiquadric RBF, black right-triangles multiquadric RBF and green down-triangles Kriging.

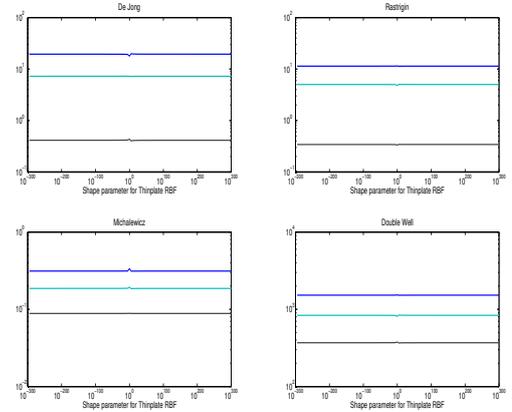


Figure A.21: Average error vs "shape" parameter  $r$  for thin-plate RBF in 3D at various grid densities. Blue line  $N = 27$  points, cyan line  $N = 216$  points, grey line  $N = 1331$  points.

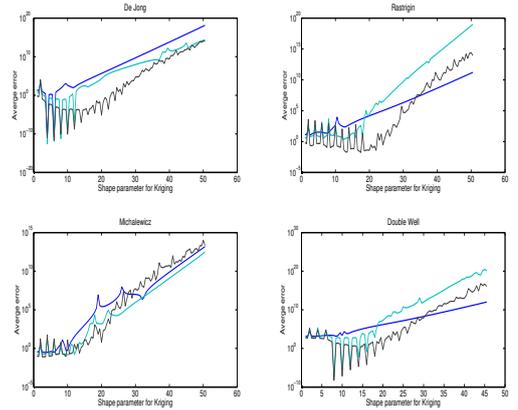


Figure A.22: Average error vs "shape" parameter  $r$  for Kriging in 3D at various grid densities. Blue line  $N = 27$  points, cyan line  $N = 216$  points, grey line  $N = 1331$  points.

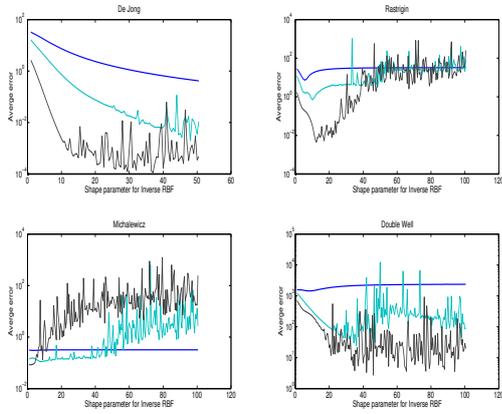


Figure A.23: Average error vs "shape" parameter  $r$  for inverse multiquadric RBF in 3D at various grid densities. Blue line  $N = 27$  points, cyan line  $N = 216$  points, grey line  $N = 1331$  points.

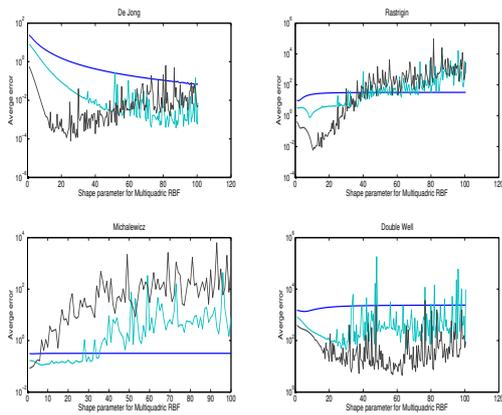


Figure A.24: Average error vs "shape" parameter  $r$  for multiquadric RBF in 3D at various grid densities. Blue line  $N = 27$  points, cyan line  $N = 216$  points, grey line  $N = 1331$  points.

## Appendix B. Web-Service

### Appendix B.1. FEMVib Dependencies

Installing FEMVib requires six main packages. The first package to be installed is *MPICH2*. *MPICH2* is a high-performance and widely portable implementation of the Message Passing Interface (MPI) standard (both MPI-1 and MPI-2). The second package is *SLEPc*, the Scalable Library for Eigenvalue Computations. *SLEPc* is a software library for the solution of large scale sparse eigenvalue problems on parallel computers. It is an extension of PETSc and can be used for either standard or generalized eigenproblems, with real or complex arithmetic. It can also be used for computing a partial SVD of a large, sparse, rectangular matrix, and to solve quadratic eigenvalue problems. The third package is *PETSc*, Portable Extensible Toolkit for Scientific Computation. *PETSc* is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations. It supports MPI, shared memory pthreads, and NVIDIA GPUs, as well as hybrid MPI-shared memory pthreads or MPI-GPU parallelism. The last package is the finite element library *LibMesh*. For FEMVib to work properly all the packages have to be defined in the environment PATH.

The last two packages are associated with the Python scripting language. The first is *ActivePython* and second is *PySparse / Numpy*. *ActivePython* is used to install binary python packages on any OS distribution, while *PySparse* and *numpy* are python libraries for fast sparse matrix operations, such as iterative solvers, preconditioners, and interfaces to efficient factorization packages.

### Appendix B.2. Web-Service

For wrapping FEMVib as a web-service we have decided to use *Opal Toolkit*. Opal exposes the scientific functionality through a generic Web services API (via a standard WSDL), manages user data, which includes creation of working directories, input and output data staging, and persistent storage for job information and metadata. It can be accessed from a multitude of languages, which in our case involves Shell script. To successfully deploy Opal, four steps need to be fulfilled. The first is to install *Java 1.5.x* or higher with its path defined in the environment. Installing *Ant 1.6,1* or higher is the second step. Ant serves as a make-like utility for compiling Java programs. Third step is installing *Apache Tomcat*. Tomcat is an open source web server, servlet container and provides a Java HTTP web server environment for Java code to run. The last step is to write an xml application specifically suited for a given package, where all the parameters and their types are defined.

### Appendix B.3. Input Parameters and UI

There are 22 input parameters:

1. *G-Matrix File*: This file contains all the optimized geometries in Cartesian coordinates along with the reference structure. The first line of the file contains three numbers: number of internal coordinates (this can be 2 or 3), number of atoms that comprise the molecule, total number of the structures (number of points calculated on the PES). The first line of the G-Matrix file is followed by the reference geometry of the molecule starting with the mass of each atom. By reference geometry or reference structure we mean the overall optimized structure of the molecule without any of the internal coordinates being fixed. Every atom has to be placed on a new line. The rest of the G-Matrix file contains sets of fixed internal coordinates followed by the Cartesian representation of the molecular geometry. Note that the order of the atoms has to stay the same as in the reference geometry.
2. *PES file*: The PES file contains only the values of internal coordinates and corresponding energies separated by space or tab. The bond lengths are given in Angstroms and angles and dihedrals in radians. Each points on the PES is on a new line.
3. *Project Name*: The project has to be given a name, which is used in association with the output files.
4. *Dimensionality*: Either 1, 2 or 3.
5. *Number of grid points along x*: Number of grid points that will define the discretization of the domain along x axis.
6. *Number of grid points along y*: Number of grid points that will define the discretization of the domain along y axis.
7. *Number of grid points along z*: Number of grid points that will define the discretization of the domain along z axis.
8. *Lower x limit*: Minimum along x axis.
9. *Upper x limit*: Maximum along x axis.
10. *Lower y limit*: Minimum along y axis.
11. *Upper y limit*: Maximum along y axis.
12. *Lower z limit*: Minimum along z axis.
13. *Upper z limit*: Maximum along z axis.
14. *Element type*: Choice of element type based on the dimensionality of the problem. The user interface is simplified by leaving the choices in a radio button format.
15. *Order of the polynomial*: Order of the polynomial used in each of the elements. First or second order can be selected. This parameter is associated with the Shape function.
16. *Shape function*: Shape function type used in each of the elements. The default values are Hermite and Lagrange.
17. *Energy units conversion*: Conversion of the kinetic energy operator to  $cm^{-1}$ . Default value is 33.71526, which corresponds to the conversion from  $\hbar^2 amu^{-1} \text{\AA}^{-2}$

18. *Number of Eigenvalues*: Number of eigenvalues to be computed.
19. *Tau*: Altitude at which the PES will be analyzed for eigenvalues. This represents the absolute value along the height of the surface since every surface is shifted so that its minimum lies at 0.
20. *Wavefunction*: Print wavefunction: yes or no.
21. *PES constructed using IMLS*: Was the PES generated dynamically using IMLS: yes or no.
22. *PES in Hartrees*: Is the PES in Hartrees: yes or no. If so the PES will be converted to  $\text{cm}^{-1}$ .

The structure of the parameter in the user interface is depicted in figure B.25.

Figure B.25: User Interface for the web-service. Visual input for above mentioned parameters.

#### Appendix B.4. Output Parameters

The output is printed in two different files. The first file contains the details about the finite element discretiza-

tion such as, cpu times, total number of elements, global and stiffness matrix assembly progress, etc. It also contains details about *PySparse* eigensolver. The number of requested eigenvalues is displayed in ascending order at the end of the file. The second file, if the option for printing the wavefunction is selected, contains the numerically reconstructed wavefunctions. These are not visualized automatically and the task is left up to the user. In case the system is three dimensional the wavefunctions will be structured along the finite element discretization of the  $z$  axis. An example of visualized 2D wavefunction, calculated on  $30 \times 30$  grid is given in figure B.26.

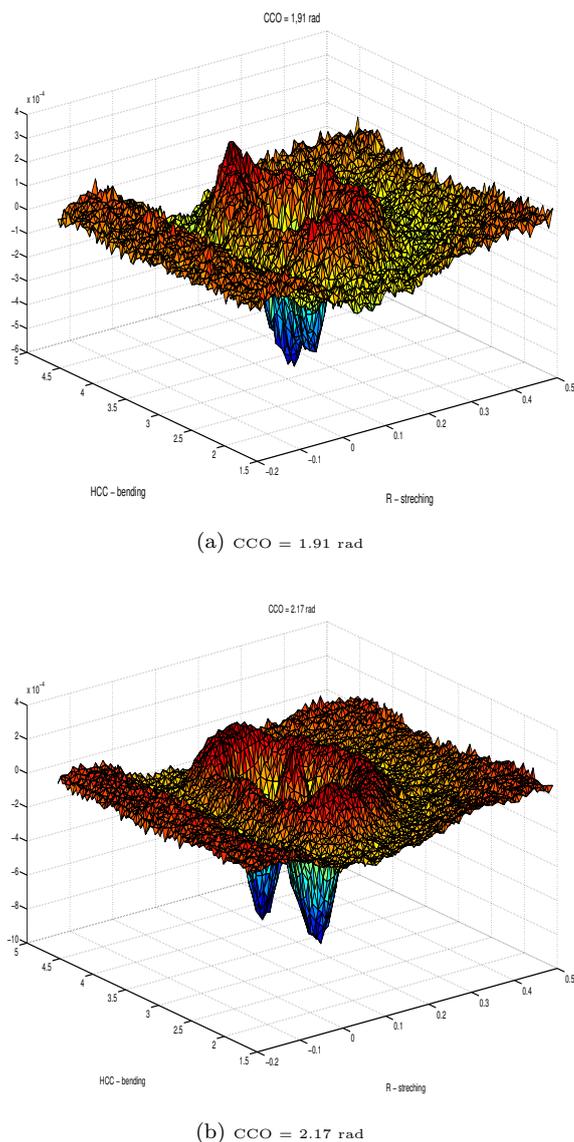


Figure B.26: Reconstructed wavefunctions for  $H - C - C$  bend and  $C - C - C$  relative stretch in  $HC_3O$ . The vibrational analysis was three dimensional, where the third vibrational coordinate was  $C - C - O$  bend.