Lanczos Bound-state Calculation Code

Joshua T. Cantin
(Dated: July 15, 2013)

I. INTRODUCTION

This document contains both documentation concerning the Lanczos Bound-state Calculation Code as well as information about the underlying theory.

II. THEORY

It should be noted that it is possible that one could define a combined basis $|\gamma\rangle$ that represents $|\gamma\rangle = |\alpha\beta\rangle = |\alpha\rangle\,|\beta\rangle$ such that $\gamma[\alpha, \beta]$ in a similar manner to that done with $n[l, m]$ for the $|n\rangle = |lm\rangle$ Tesseral Harmonic basis.

We also may want to test out a specialized code (either in terms of type of rotor or the spin isomers) versus a generalized code (use the symmetrizing operator to select out the even and odd states?) and see which is computationally more efficient.

The Colbert-Miller Formulae will be used for the translational basis and the Wigner functions could possibly be used for the Asymmetric Top ($e^{im\phi} d^J_{MK}(\theta) e^{ik\chi}$).

A. Hamiltonian

It should be noted that one may be able to define a symmetrizing operator that will allow one to select out the ortho, para or spinless states from the Hamiltonian when calculating the Krylov Subspace. The operator would act as follows

$$\begin{aligned} w &= \hat{S}\hat{H}v \\ w_{para} &= \hat{S}_{even}\hat{H}v \\ w_{ortho} &= \hat{S}_{odd}\hat{H}v \\ w_{spinless} &= \hat{S}_{all}\hat{H}v \end{aligned}$$

1. Linear Rotor

The Hamiltonian for a Cartesian grid for the centre of mass position and θ and ϕ for the orientation is as follows:

$$\widehat{H}_{zxy\theta\phi} = \widehat{T}_x + \widehat{T}_y + \widehat{T}_z + \widehat{T}_{rot} + \widehat{V}_{xyz\theta\phi} \tag{1}$$

Now, the discrete variable representation (DVR) presented by Colbert and Miller in their 1992 paper will be used for the Cartesian grid. Specifically, the DVR for a variable that varies from $-\infty$ to ∞ will be used, but truncated to the box size. This DVR will behave as if the system is suspended in empty space, rather than having an infinite potential at the boundaries (i.e. particle in a box), which a DVR designed for a finite grid would represent. The grid points are as follows:

$$x_i = i\Delta x, \text{ where } i = 0, \pm 1, \pm 2, ..., \pm\infty \tag{2}$$

The corresponding basis functions are:

$$\langle x|x_i\rangle = \frac{\sin\left(\frac{\pi(x-x_i)}{\Delta x}\right)}{\pi(x-x_i)} \tag{3}$$

The grid is set such that $\langle x|x_i\rangle$ is zero for all $x_i \neq x$. Thus, it appears that while interpolation between the grid points is possible, the value at each of the grid points is independent of all of the other grid points; I believe that this simplifies the equations and the subsequent diagonalization of the Hamiltonian. The kinetic energy operator is then expressed as

$$\langle x_i|\widehat{T}_x|x_{i'}\rangle = \widehat{T}_{x_{i,i'}} = \frac{\hbar^2}{2m\Delta x^2}(-1)^{i-i'} \begin{Bmatrix} \frac{\pi^2}{3}, & i = i' \\ \frac{2}{(i-i')^2}, & i \neq i' \end{Bmatrix} \tag{4}$$

where m is the total mass of the linear rotor and Δx is the grid spacing. Note that $\widehat{T}_{x_{i,i'}}$ is diagonal in all the other grids used, whether $|yz\theta\phi\rangle$ or $|yzlm\rangle$ and that $\widehat{T}_{y_{i,i'}}$ and $\widehat{T}_{z_{i,i'}}$ are of the same form as $\widehat{T}_{x_{i,i'}}$.

For the rotation, the finite basis representation (FBR) of $|lm\rangle$ (the Tesseral Harmonics, a.k.a. the real form of the Spherical harmonics) is to be used instead of θ and ϕ. This then makes

$$\widehat{H}_{zxylm} = \widehat{T}_x + \widehat{T}_y + \widehat{T}_z + \widehat{T}_{rot} + \widehat{V}_{xyzlm} \tag{5}$$

The rotational kinetic energy is thus

$$\widehat{T}_{rot} = B\hat{l}^2 \tag{6}$$

where B is the rotational constant defined as

$$B = \frac{\hbar^2}{2I} \tag{7}$$

with I as the moment of inertia. The terms of $\widehat{T}_{rot}$ are

$$\langle lm|\widehat{T}_{rot}|l'm'\rangle = Bl(l+1)\delta_{ll'}\delta_{mm'} \tag{8}$$

where $\delta_{ii'}$ is the Kronecker delta, showing that $\widehat{T}_{rot}$ is diagonal in both l and m.

Now, the potential $\widehat{V}$ is calculated and stored in the $|xyz\theta\phi\rangle$ basis and must be converted to the $|xyzlm\rangle$ basis. First of all, the terms of $\widehat{V}$ in the $|xyz\theta\phi\rangle$ basis is

$$\langle xyz\theta\phi|\widehat{V}_{xyz\theta\phi}|x'y'z'\theta'\phi'\rangle = V(x,y,z,\theta,\phi)\delta_{xyz\theta\phi,x'y'z'\theta'\phi'} \tag{9}$$

To convert from $|xyz\theta\phi\rangle$ to $|xyzlm\rangle$, one can find the elements of $\widehat{V}_{xyz\theta\phi}$ in $|xyzlm\rangle$

$$\langle xyzlm|\widehat{V}_{xyzlm}|x'y'z'l'm'\rangle = \langle xyzlm|\widehat{V}_{xyz\theta\phi}|x'y'z'l'm'\rangle \tag{10}$$

$$= \langle lm|\widehat{V}_{\theta\phi}(x,y,z)|l'm'\rangle \tag{11}$$

$$= \int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} \langle lm|\theta\phi\rangle \langle\theta\phi|\widehat{V}_{\theta\phi}(x,y,z)|\theta'\phi'\rangle \langle\theta'\phi'|l'm'\rangle \,\partial\phi\sin\theta\partial\theta\partial\phi'\sin\theta'\partial\theta' \tag{12}$$

$$= \int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} \widetilde{Y}_{lm}(\theta,\phi)V(\theta,\phi;x,y,z)\widetilde{Y}_{l'm'}(\theta,\phi)\partial\phi\sin\theta\partial\theta \tag{13}$$

$$\begin{aligned}= \int_{\theta=0}^{\pi}\int_{p=-1}^{1} \frac{1}{\sqrt{1-p^2}}[&\widetilde{Y}_{lm}(\theta,\cos^{-1}p)V(\theta,\cos^{-1}p;x,y,z)\widetilde{Y}_{l'm'}(\theta,\cos^{-1}p) \\ &+ \widetilde{Y}_{lm}(\theta,2\pi-\cos^{-1}p)V(\theta,2\pi-\cos^{-1}p;x,y,z)\widetilde{Y}_{l'm'}(\theta,2\pi-\cos^{-1}p)]\partial p\sin\theta\partial\theta\end{aligned} \tag{14}$$

$$\begin{aligned}= \int_{\cos\theta=-1}^{1}\int_{p=-1}^{1} \frac{1}{\sqrt{1-p^2}}[&\widetilde{Y}_{lm}(\theta,\cos^{-1}p)V(\theta,\cos^{-1}p;x,y,z)\widetilde{Y}_{l'm'}(\theta,\cos^{-1}p) \\ &+ \widetilde{Y}_{lm}(\theta,2\pi-\cos^{-1}p)V(\theta,2\pi-\cos^{-1}p;x,y,z)\widetilde{Y}_{l'm'}(\theta,2\pi-\cos^{-1}p)]\partial p\partial(\cos\theta)\end{aligned} \tag{15}$$

$$\begin{aligned}= \int_{q=-1}^{1}\int_{p=-1}^{1} \frac{1}{\sqrt{1-p^2}}[&\widetilde{Y}_{lm}(\cos^{-1}q,\cos^{-1}p)V(\cos^{-1}q,\cos^{-1}p;x,y,z)\widetilde{Y}_{l'm'}(\cos^{-1}q,\cos^{-1}p) \\ &+ \widetilde{Y}_{lm}(\cos^{-1}q,2\pi-\cos^{-1}p)V(\cos^{-1}q,2\pi-\cos^{-1}p;x,y,z)\widetilde{Y}_{l'm'}(\cos^{-1}q,2\pi-\cos^{-1}p)]\partial p\partial q\end{aligned} \tag{16}$$

$$\begin{aligned}\approx \int_{q=-1}^{1}\sum_{\beta=1}^{n_\beta} w_\beta^{GC}[&\widetilde{Y}_{lm}(\cos^{-1}q,\cos^{-1}p_\beta)V(\cos^{-1}q,\cos^{-1}p_\beta;x,y,z)\widetilde{Y}_{l'm'}(\cos^{-1}q,\cos^{-1}p_\beta) \\ &+ \widetilde{Y}_{lm}(\cos^{-1}q,2\pi-\cos^{-1}p_\beta)V(\cos^{-1}q,2\pi-\cos^{-1}p_\beta;x,y,z)\widetilde{Y}_{l'm'}(\cos^{-1}q,2\pi-\cos^{-1}p_\beta)]\partial q\end{aligned} \tag{17}$$

$$\begin{aligned}\approx \sum_{\alpha=1}^{n_\alpha} w_\alpha^{GL}\sum_{\beta=1}^{n_\beta} w_\beta^{GC}[&\widetilde{Y}_{lm}(\cos^{-1}q_\alpha,\cos^{-1}p_\beta)V(\cos^{-1}q_\alpha,\cos^{-1}p_\beta;x,y,z)\widetilde{Y}_{l'm'}(\cos^{-1}q_\alpha,\cos^{-1}p_\beta) \\ &+ \widetilde{Y}_{lm}(\cos^{-1}q_\alpha,2\pi-\cos^{-1}p_\beta)V(\cos^{-1}q_\alpha,2\pi-\cos^{-1}p_\beta;x,y,z)\widetilde{Y}_{l'm'}(\cos^{-1}q_\alpha,2\pi-\cos^{-1}p_\beta)]\end{aligned} \tag{18}$$

$$= \tag{19}$$

where $\widetilde{Y}_{lm}(\theta,\phi)$ are the Tesseral harmonics, w_α^{GL} and q_α are the Gauss-Legendre quadrature weights and abscissae, and w_β^{GC} and p_β are the Gauss-Chebyshev quadrature weights and abscissae. NOTE: THE ABOVE THREE OR FOUR EQUATIONS HAVE DEFINITE PROBLEMS

B. Basis States

1. Rotor Bases

a. Linear Rotor For the linear rotor, the tesseral spherical harmonics will be used as the basis states and are defined as follows (taken from `http://en.wikipedia.org/wiki/Spherical_harmonics`):

$$Y_{lm}(\theta,\phi) = \begin{cases} \frac{1}{\sqrt{2}}\left(Y_l^m(\theta,\phi) + (-1)^m Y_l^{-m}(\theta,\phi)\right) & \text{if } m>0 \\ Y_l^0(\theta,\phi) & \text{if } m=0 \\ \frac{1}{i\sqrt{2}}\left(Y_l^{-m}(\theta,\phi) - (-1)^m Y_l^m(\theta,\phi)\right) & \text{if } m<0 \end{cases} \tag{20}$$

$$= \begin{cases} \sqrt{2}N_{(l,m)}P_l^m(\cos\theta)\cos m\phi & \text{if } m>0 \\ Y_l^0(\theta,\phi) & \text{if } m=0 \\ \sqrt{2}N_{(l,|m|)}P_l^{|m|}(\cos\theta)\sin|m|\phi & \text{if } m<0 \end{cases} \tag{21}$$

where $N_{(l,m)}$ is a normalization constant and is

$$N_{(l,m)} \equiv \frac{1}{\sqrt{2\pi}}\sqrt{\frac{(2l+1)}{2}\frac{(l-m)!}{(l+m)!}} \tag{22}$$

Note that we incorporate the Condon-Shortley phase in $N_{(l,m)}$, so that

$$N_{(l,m)} \equiv (-1)^m \frac{1}{\sqrt{2\pi}}\sqrt{\frac{(2l+1)}{2}\frac{(l-m)!}{(l+m)!}} \tag{23}$$

$Y_l^m(\theta,\phi)$ are the Laplace Spherical Harmonics and are

$$Y_l^m(\theta,\phi) = N_{(l,m)}P_l^m(\cos\theta)\,e^{im\phi} \tag{24}$$

P_l^m are the associated Legendre polynomials of non-negative m, defined as

$$P_l^m(x) = (-1)^m\left(1-x^2\right)^{\frac{m}{2}}\frac{d^m}{dx^m}\left(P_l(x)\right) \tag{25}$$

where $P_l(x)$ are the Legendre Polynomials, expressed using Rodrigues' formula:

$$P_l(x) = \frac{1}{2^l l!}\frac{d^l}{dx^l}[\left(x^2-1\right)^l] \tag{26}$$

In practise, the associated Legendre Polynomials ($P_l^m(x)$) are calculated by first calculating

$$P_l^l(x) = (2l-1)\,P_{l-1}^{l-1}(x)\sqrt{1-x^2}$$

for $l=1$ to $l=l_{max}$, where $P_0^0=1$.

Then, the associated Legendre Polynomials are calculated for $m=0$ to $m=l_{max}-1$ and for $l=1$ to $l=l_{max}$ as follows

$$P_m^{l+1}(x) = \frac{\left[(2l+1)\,xP_m^l(x) - (l+m)\,P_m^{l-1}(x)\right]}{l-m+1} \tag{27}$$

where $P_m^{l-1}(x)=0$ if $l-1<m$.

1. Legendre Polynomial Recursion Relations From Abramowitz and Stegun, pg.334, the Legendre Polynomials can be generated using the following recurrence relations, where

$$\{\nu|\nu\epsilon\mathbb{C}\}$$
$$\{\mu|\mu\epsilon\mathbb{C}\}$$
$$\{n|n\geq 0, n\epsilon\mathbb{Z}\}$$
$$\{m|m\geq 0, m\epsilon\mathbb{Z}\}$$
$$\{z|z\epsilon\mathbb{C}\}$$
$$\{x|x\epsilon\mathbb{R}\}$$

If the degree (ν) is varying,
8.5.3:

$$(\nu-\mu+1)\,P^{\mu}_{\nu+1}(z) = (2\nu+1)zP^{\mu}_{\nu}(z) - (\nu+\mu)P^{\mu}_{\nu-1}(z) \tag{28}$$

8.5.4:

$$(z^2-1)\frac{dP^{\mu}_{\nu}(z)}{dz} = \nu z P^{\mu}_{\nu} - (\nu+\mu)P^{\mu}_{\nu-1}(z) \tag{29}$$

If the order (μ) is varying,
8.5.1:

$$P^{\mu+1}_{\nu}(z) = \sqrt{z^2-1}[(\nu-\mu)zP^{\mu}_{\nu}(z) - (\nu+\mu)P^{\mu}_{\nu-1}(z)] \tag{30}$$

8.5.2:

$$(z^2-1)\frac{dP^{\mu}_{\nu}(z)}{dz} = (\nu+\mu)(\nu-\mu+1)\sqrt{z^2-1}P^{\mu-1}_{\nu}(z) - \mu z P^{\mu}_{\nu}(z) \tag{31}$$

If the order (μ) and the degree (ν) are varying,
8.5.5:

$$P^{\mu}_{\nu+1}(z) = P^{\mu}_{\nu-1}(z) + (2\nu+1)\sqrt{z^2-1}P^{\mu-1}_{\nu}(z) \tag{32}$$

It should also be noted that (A&S pg.333 8.4.1 and 8.4.3)

$$P^{0}_{0}(z) = 1 \tag{33}$$
$$P^{0}_{1}(z) = z \tag{34}$$

and (A&S pg.333 8.2.1)

$$P^{\mu}_{-\nu-1}(z) = P^{\mu}_{\nu}(z) \tag{35}$$

For the system used here, μ can be set to m, ν to n, and z to x, resulting in:
If the degree (n) is varying,

$$(n-m+1)\,P^{m}_{n+1}(x) = (2n+1)xP^{m}_{n}(x) - (n+m)P^{m}_{n-1}(x) \tag{36}$$

$$(x^2-1)\frac{dP^{m}_{n}(x)}{dx} = nxP^{m}_{n} - (n+m)P^{m}_{n-1}(x) \tag{37}$$

and Bonnet's Recursion Formula if $m=0$ (from `http://en.wikipedia.org/wiki/Legendre_polynomials`)

$$(n+1)\,P_{n+1}(x) = (2n+1)xP_{n}(x) - nP_{n-1}(x) \tag{38}$$

from this, the corresponding formula for the derivative is found to be (from `http://en.wikipedia.org/wiki/Legendre_polynomials`)

$$(x^2-1)\frac{dP_n(x)}{dx} = nxP_n - nP_{n-1}(x) \tag{39}$$

If the order (m) is varying,

$$P_n^{m+1}(x) = \sqrt{x^2-1}[(n-m)xP_n^m(x) - (n+m)P_{n-1}^m(x)] \tag{40}$$

$$(x^2-1)\frac{dP_n^m(x)}{dx} = (n+m)(n-m+1)\sqrt{x^2-1}P_n^{m-1}(x) - mxP_n^m(x) \tag{41}$$

If the order (m) and the degree (n) are varying,

$$P_{n+1}^m(x) = P_{n-1}^m(x) + (2n+1)\sqrt{x^2-1}P_n^{m-1}(x) \tag{42}$$

It should also be noted that (A&S pg.333 8.4.1 and 8.4.3)

$$P_0^0(x) = 1 \tag{43}$$
$$P_1^0(x) = x \tag{44}$$

2. Orthonormality Test of the associated Legendre Polynomials A numerical test of the associated Legendre Polynomials tested can be performed using their orthonormality property. The normalized associated Legendre Polynomials are defined as:

$$\widetilde{P}_l^m(x) = \sqrt{\frac{(2l+1)}{2}\frac{(l-m)!}{(l+m)!}}P_l^m(x) \tag{45}$$

where

$$\{n|n \geq 0, n\epsilon\mathbb{Z}\}$$
$$\{m|m \geq 0, m\epsilon\mathbb{Z}\}$$
$$\{x|x\epsilon\mathbb{R}\}$$

and $P_l^m(x)$ are the non-normalized associated Legendre polynomials:

$$P_l^m(x) = (-1)^m\left(1-x^2\right)^{\frac{m}{2}}\frac{d^m}{dx^m}\left(P_l(x)\right) \tag{46}$$

where $P_l(x)$ are the Legendre Polynomials, expressed using Rodrigues' formula:

$$P_l(x) = \frac{1}{2^l l!}\frac{d^l}{dx^l}[\left(x^2-1\right)^l] \tag{47}$$

Now, orthonormality states that, for fixed m,

$$\int_{-1}^{1}\widetilde{P}_l^m(x)\widetilde{P}_{l'}^m(x)dx = \delta_{ll'} \tag{48}$$

if $0 \leq m \leq l$ (taken from PN and `http://en.wikipedia.org/wiki/Associated_Legendre_polynomials`). Now, based on the Gauss-Legendre Quadrature

$$\int_{-1}^{1}\widetilde{P}_l^m(x)\widetilde{P}_{l'}^m(x)dx = \sum_{\alpha=1}^{n_\alpha} w_\alpha\widetilde{P}_l^m(x_\alpha)\widetilde{P}_{l'}^m(x_\alpha) = \delta_{ll'} \tag{49}$$

where the condition $l \leq 2n-1$ is required for the relation to be exact (i.e. $R_n = 0$), as shown. The definitions of x_α and w_α can be found in section II B 1 a 3. If one defines a matrix,

$$L^m_{l\alpha} = \widetilde{P}^m_l(x_\alpha)\sqrt{w_\alpha} \tag{50}$$

then

$$\sum_{\alpha=1}^{n_\alpha} w_\alpha \widetilde{P}^m_l(x_\alpha)\widetilde{P}^m_{l'}(x_\alpha) = \sum_{\alpha=1}^{n} L^m_{l\alpha}(L^m_{l'\alpha})^\intercal = \delta_{ll'} \tag{51}$$

Also, for fixed l, (from `http://en.wikipedia.org/wiki/Associated_Legendre_polynomials`)

$$\int_{-1}^{1} \frac{\widetilde{P}^m_l(x)\widetilde{P}^{m'}_l(x)}{1-x^2}dx = \begin{cases} 0 & \text{if } m \neq n \\ \frac{(l+m)!}{m(l-m)!} & \text{if } m = n \neq 0 \\ \infty & \text{if } m = n = 0 \end{cases} \tag{52}$$

Thus,

$$L^{n\intercal}L^m = L^n L^{m\intercal} = \mathbb{1} \tag{53}$$

Though I think the Chebyshev Quadrature may be needed, given the form of the integral.

Now, the $|lm\rangle$ is not a direct product of $|l\rangle$ and $|m\rangle$ (i.e. $|lm\rangle \neq |l\rangle\,|m\rangle$) as l and m are not independent quantum numbers, but $|m| \leq l$, with the restriction that $l \geq 0$. This makes storing the $|lm\rangle$ basis in a matrix more difficult. However, this issue is at least partially alleviated by defining a new quantum number $n = l^2 + l + m$, with $0 \leq n \leq (l^2_{max} + 2l_{max}) = (l_{max}+1)^2 - 1$.

3. Gauss-Legendre Quadrature From Abramowitz and Stegun pg.887, Section 25.4.29:
The quadrature is

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} w_i f(x_i) + R_n \tag{54}$$

The related polynomials for this quadrature are the Legendre Polynomials $P_n(x), P_n(1) = 1$. Also,

$$x_i = i\text{th root of } P_n(x) \tag{55}$$

$$w_i = \frac{2}{(1-x_i^2)(P'_n(x_i))^2} \tag{56}$$

$$R_n = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\xi), (-1 < \xi < 1) \tag{57}$$

It should be noted that $R_n = 0$ if $f(x)$ is a polynomial of degree $2n-1$ or less (as $f^{(2n)}(x) = 0$ in that case).

4. Gauss-Chebyshev Quadrature From Abramowitz and Stegun pg.889, Section 25.4.39:
The quadrature is

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}dx = \sum_{i=1}^{n} w_i f(x_i) + R_n \tag{58}$$

The related orthogonal polynomials are the Chebyshev Polynomials of the First Kind:

$$T_n(x), T_n(1) = \frac{1}{2^{n-1}} \tag{59}$$

Also,

$$x_i = \cos\frac{(2i-1)\pi}{2n} \tag{60}$$

$$w_i = \frac{\pi}{n} \tag{61}$$

$$R_n = \frac{\pi}{(2n)!2^{2n-1}} f^{(2n)}(\xi), (-1 < \xi < 1) \tag{62}$$

5. $|lm\rangle$ **Basis Notes** The $|lm\rangle$ basis size can be enumerated as follows:

$$\begin{aligned} Count &= \sum_{m=-l_{max}}^{l_{max}} \sum_{l=|m|}^{l_{max}} 1 \qquad (63)\\ &= \sum_{m=-l_{max}}^{l_{max}} (l_{max} - |m| + 1)\\ &= 2\sum_{m=1}^{l_{max}} (l_{max} - m + 1) + l_{max} + 1\\ &= 2l_{max}^2 - l_{max}(l_{max}+1) + 2l_{max} + l_{max} + 1\\ &= l_{max^2} - l_{max} + 3l_{max} + 1\\ &= l_{max}^2 + 2l_{max} + 1\\ &= (l_{max}+1)^2 \qquad (64) \end{aligned}$$

b. Asymmetric Top

C. Potential Energy

1. Orientational

a. Linear Rotor Given a potential energy matrix $V(\theta,\phi)$ that is diagonal in θ and ϕ, V can be expressed as a vector $\vec{V}(\theta,\phi) = \text{diag}(V)$.Now, the matrix elements of V are

$$V_{nn'}(\theta,\phi) = \int_0^{2\pi} \partial\phi \int_{-1}^{1} \partial(\cos\theta)\, Y_{lm[n]}(\theta,\phi)\, V(\theta,\phi)\, Y_{lm[n']}(\theta,\phi) \qquad (65)$$

where $lm[n]$ denotes that the quantum numbers l and m are treated as a function of n. In this case, l and m are looked up in a table given n as the index. $Y_{lm[n]}(\theta,\phi)$ is defined as in (20) and (21), but is also defined here as

$$Y_{lm[n]}(\theta_\alpha,\phi_\beta) = P_{lm}(\cos\theta_\alpha) N_m(\phi_\beta) \qquad (66)$$

where α and β are the indices of the θ and ϕ grids, respectively, such that $\theta \,\epsilon\, [0,\pi]$ and $\phi \,\epsilon\, [0,2\pi]$ so that

$$\begin{aligned} Y^{\alpha\beta}_{lm[n]} &= P^{\alpha}_{lm[n]} N^{\beta}_{m[n]} \qquad (67)\\ &= P^{\alpha}_{lm} N^{\beta}_{m} \qquad (68) \end{aligned}$$

where the $lm[n]$ is changed to lm except where there is ambiguity in the index n.

Now, within the Lanczos algorithm, a Krylov subspace is generated using an initial vector $v_{n'}$ for the entire Hamiltonian. And, since, V is a part of the Hamiltonian, the potential Krylov vector is

$$u_n = \sum_{n'} V_{nn'} v_{n'} \qquad (69)$$

Now, since both θ and ϕ are discretized, V can actually be calculated using a Gaussian Quadrature

$$V_{nn'} = \sum_{\alpha,\beta} Y^{\alpha\beta}_{lm[n]} w_\alpha^{\frac{1}{2}} w_\beta^{\frac{1}{2}} V_{\alpha\beta} w_\alpha^{\frac{1}{2}} w_\beta^{\frac{1}{2}} Y^{\alpha\beta}_{lm[n']} \qquad (70)$$

where w_i are the weights for each grid and are split apart for the purposes of symmetric transformations. Thus,

$$\begin{aligned} u_n &= \sum_{n'} V_{nn'} v_{n'} \qquad (71)\\ &= \sum_{n'} \sum_{\alpha,\beta} Y^{\alpha\beta}_{lm[n]} w_\alpha^{\frac{1}{2}} w_\beta^{\frac{1}{2}} V_{\alpha\beta} w_\alpha^{\frac{1}{2}} w_\beta^{\frac{1}{2}} Y^{\alpha\beta}_{lm[n']} v_{n'} \qquad (72)\\ &= \sum_{\alpha,\beta} Y^{\alpha\beta}_{lm[n]} w_\alpha^{\frac{1}{2}} w_\beta^{\frac{1}{2}} V_{\alpha\beta} w_\alpha^{\frac{1}{2}} w_\beta^{\frac{1}{2}} \sum_{n'} Y^{\alpha\beta}_{lm[n']} v_{n'} \qquad (73) \end{aligned}$$

2. Partial Summation

Similarly to the partial summation that can be performed for the direct product translational basis $|n_x n_y n_z\rangle$, it may be beneficial to try a partial summation method for the $|lm\rangle$ basis in order to have a computationally more efficient algorithm.

Now, the last sum in (73) can be broken apart and rearranged as follows

$$\sum_{n'} Y^{\alpha\beta}_{lm[n']} v_{n'} = \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} P^{\alpha}_{lm} N^{\beta}_{m} v_{lm} \tag{74}$$

$$= \sum_{m=-l_{max}}^{l_{max}} N^{\beta}_{m} \sum_{l=|m|}^{l_{max}} P^{\alpha}_{lm} v_{lm} \tag{75}$$

$$= \sum_{m=-l_{max}}^{l_{max}} N^{\beta}_{m} \tilde{v}^{\alpha}_{m} \tag{76}$$

where l_{max} is the highest l incorporated in the numerical calculations.

Now, though each of these forms has the same mathematical meaning, they may require a different number of FLOPS to actually be calculated.

For (74), all of the terms are calculated at once within a double loop. The number of FLOPS for (74) is strictly, however, twice the number of terms since there are two multiplications within the equation. However, if (74) were to be calculated, P^{α}_{lm} and N^{β}_{m} would be precomputed and combined, such that $Y^{\alpha\beta}_{lm} = P^{\alpha}_{lm} N^{\beta}_{m}$. This reduces the number of FLOPS *within the on-the-fly Hv calculation* to one per term. Thus,

$$\text{Terms}\left(\sum_{n'} Y^{\alpha\beta}_{lm[n']} v_{n'}\right) = \text{Terms}\left(\sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} Y^{\alpha\beta}_{lm} v_{lm}\right) \tag{77}$$

$$= \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} 1 \text{ for each } \alpha, \beta$$

$$= \sum_{l=0}^{l_{max}} (2l+1) n_\alpha n_\beta$$

$$= n_\alpha n_\beta \sum_{l=0}^{l_{max}} (2l+1)$$

$$= n_\alpha n_\beta [\sum_{l=1}^{l_{max}} (2l+1) + (2l+1)_{l=0}]$$

$$= n_\alpha n_\beta [2\frac{l_{max}(l_{max}+1)}{2} + l_{max} + 1]$$

$$= n_\alpha n_\beta [l_{max}(l_{max}+1) + l_{max} + 1]$$

$$= n_\alpha n_\beta [l^2_{max} + l_{max} + l_{max} + 1]$$

$$= n_\alpha n_\beta (l_{max}+1)^2 \tag{78}$$

$$= \tag{79}$$

For (75), both the outer and inner sums should be treated separately, hence equation (75). Concentrating on the inner sum, the number of terms is

$$\text{Terms}(\tilde{v}^{\alpha}_{m}) = \text{Terms}\left(\sum_{l=|m|}^{l_{max}} P^{\alpha}_{lm} v_{lm}\right) \tag{80}$$

$$= n_\alpha \left(l_{max} - |m| + 1\right) \tag{81}$$

where n_α is the number of α grid points. The $+1$ term is included as the number of numbers (inclusive) from a to b is $b - a + 1$.

However, this is not complete. For when $\tilde{v}_m^\alpha$ is calculated, one must also perform the calculation for all of the m terms within the vector, since P_{lm}^α also depends on m. Thus, the number of terms is

$$\begin{aligned}
\text{Terms}(\tilde{v}_m^\alpha) &= \sum_{m=-l_{max}}^{l_{max}} \text{Terms}\left(\sum_{l=|m|}^{l_{max}} P_{lm}^\alpha v_{lm}\right) && (82)\\
&= \sum_{m=-l_{max}}^{l_{max}} n_\alpha \left(l_{max} - |m| + 1\right) && (83)\\
&= n_\alpha \sum_{m=-l_{max}}^{l_{max}} \left(l_{max} - |m| + 1\right) && (84)\\
&= n_\alpha \left(2\sum_{m=1}^{l_{max}} \left(l_{max} - m + 1\right) + \left(l_{max} - |m| + 1\right)_{m=0}\right) && (85)\\
&= n_\alpha \left(2\left(\sum_{m=1}^{l_{max}} l_{max} - \sum_{m=1}^{l_{max}} m + \sum_{m=1}^{l_{max}} 1\right) + l_{max} + 1\right) && (86)\\
&= n_\alpha \left(2\left(l_{max}^2 - \frac{l_{max}\left(l_{max}+1\right)}{2} + l_{max}\right) + l_{max} + 1\right) && (87)\\
&= n_\alpha \left(2l_{max}^2 - l_{max}^2 - l_{max} + 2l_{max} + l_{max} + 1\right) && (88)\\
&= n_\alpha \left(l_{max}^2 + 2l_{max} + 1\right) && (89)\\
&= n_\alpha \left(l_{max} + 1\right)^2 && (90)
\end{aligned}$$

Thus, there are $n_\alpha \left(l_{max}+1\right)^2$ multiplications required (i.e. FLOPS) in order to calculate $\tilde{v}_m^\alpha$. From this point, one then needs to calculate $\sum_{m=-l_{max}}^{l_{max}} N_m^\beta \tilde{v}_m^\alpha$, which has the number of terms

$$\text{Terms}\left(\sum_{m=-l_{max}}^{l_{max}} N_m^\beta \tilde{v}_m^\alpha\right) = n_\alpha n_\beta \left(2l_{max}+1\right) \tag{91}$$

where n_β is the number of grid points for the β basis. Now, similar to the reasoning that the calculation of $\tilde{v}_m^\alpha$ must include the calculation of all m terms, so here, since $\tilde{v}_m^\alpha$ is dependent on α, does $\sum_{m=-l_{max}}^{l_{max}} N_m^\beta \tilde{v}_m^\alpha$ need to include all α terms.

This leaves the total number of terms as

$$\begin{aligned}
\text{Terms}\left(\sum_{n'} Y_{lm[n']}^{\alpha\beta} v_{n'}\right) &= \text{Terms}\left(\sum_{m=-l_{max}}^{l_{max}} N_m^\beta \tilde{v}_m^\alpha\right) + \text{Terms}(\tilde{v}_m^\alpha) && (92)\\
&= n_\alpha \left(l_{max}+1\right)^2 + n_\alpha n_\beta \left(2l_{max}+1\right) && (93)\\
&= n_\alpha [\left(l_{max}+1\right)^2 + n_\beta \left(2l_{max}+1\right)] && (94)
\end{aligned}$$

Here, it can be seen that though the α grid must be iterated through for the whole calculation, the β grid need only be evaluated for the $2l_{max}+1$ terms. This may give a potential computational cost savings. Especially so, since β is a large grid relative to the $|lm\rangle$ basis.

1. FLOPS for each section of the Quadrature During the entire process of calculating Vv, the FLOPS, using partial summation, can be summarized as shown in table I.

We should double check whether steps 4 and 5 can be reordered to reduce the number of FLOPS, though, at this point, I have a feeling not.

TABLE I. FLOP Count for Potential Quadrature

Step #	# FLOPS	Index Change for v_{lm}
1	$n_\alpha(l_{max}+1)^2$	$lm \to \alpha m$
2	$n_\alpha n_\beta(2l_{max}+1)$	$\alpha m \to \alpha\beta$
3	$n_\alpha n_\beta$	$V_{\alpha\beta} \cdot v_{\alpha\beta}$
4	$n_\alpha n_\beta(2l_{max}+1)$	$\alpha\beta \to \alpha m$
5	$n_\alpha(l_{max}+1)^2$	$\alpha m \to lm$

D. Code Theory

For a linear array in a direct product basis, such as the eigenstates, the index of the basis, p can be found as

$$p = n_i n_j n_k n + n_i n_j k + n_i j + i \tag{95}$$
$$= n_i(n_j n_k n + n_j k + j) + i \tag{96}$$
$$= n_i(n_j(n_k n + k) + j) + i \tag{97}$$
$$= ((n n_k + k)n_j + j)n_i + i \tag{98}$$

where i is the inner most index and varies most rapidly as p increases, while n is the outer most index and varies most slowly. This would be useful for the v or u vectors, where each is represented in the basis $|xyzn\rangle = |xyzlm\rangle$, where $n[l,m]$ according to the ordering of l and m as determined by $\sum_{m=-l_{max}}^{l_{max}} \sum_{l=|m|}^{l_{max}} 1$. The mapping of n to l,m and vice versa is given by the 1D and 2D arrays generated by the function *genIndices_lm()*.

III. CODE

The following contains documentation and notes about the code.

A. General Notes

If the potential changes:

- In basisFcns.cpp (which contains Hv_prep and Hv), if it is to be changed for another potential, look in the Hv_prep function in the potential calculation section for things to to change.
- Look in ulm_calc, loop 3, to find something else to change.

If memory becomes an issue, these are areas that one can modify:

- One can reuse the Legendre Polynomials for both the $\phi = 2\pi - \arccos(\cos\phi)$ and $\phi = \arccos(\cos\phi)$ cases, since the Legendre Polynomials do not depend on ϕ and the program currently stores the polynomials twice them in Hv_prep.

If one would like to increase the amount of parallelization, the reduction clause would be very useful in certain for loops. See `http://msdn.microsoft.com/en-us/library/88b1k8y5%28v=vs.80%29.aspx` and `http://www.viva64.com/en/a/0054/#ID0EBCBM`.

It should be noted that the most time-critical point in the code seems to be the *Vv_5D_oneCompositeIndex* function in *basisFcns.cpp* for the Hv calculation. This is stated as when the loop within this function was parallelized with a collapse(3) clause, the execution time dropped by nearly 50%. For example, from 0.885s (wall time) to 0.453s.

Parallelization left the output unchanged (for the entire vector $u = Hv$, about 49000 values) from at least June 26th, 2013 at 3:26pm (this is the oldest full output found) until July 8th, 2013 at 12:22pm. Also, the values do not change between runs, indicating no random values likely generated by memory access or initialization problems.

With the following inputs: # #Make sure that there is no space before the equals sign and only one after; you will get an error or weird results otherwise nx= 10 x_max(nm)= 1.0 ny= 10 y_max(nm)= 1.0 nz= 10 z_max(nm)= 1.0 l_max= 6 thetaPoints= 10 phiPoints= 10 pointPotential_nx= 10 pointPotential_x_max(nm)= 1.0 pointPotential_ny= 10 pointPotential_y_max(nm)= 1.0 pointPotential_nz= 10 pointPotential_z_max(nm)= 1.0 totalMass(g/mol)= 2.0 momentOfInertia(g/mol.nm$\hat{2}$)= 8.0 geometryFilename= sI_unitCelltest_CM_EA_ZYZ.xyz

Table II has the execution time information for running Hv on a Mac OS X 10.6.8 computer with 3.06GHz Intel Core i3 processor and 4GB 1333MHz DDR3 RAM. The processor has two cores and four total threads. The data is for a single run only. The executable is named *BasisTestMultipleHv* and the input file *inputFile.txt*. The program was run with the command "time ./BasisTestMultipleHv inputFile.txt 100", for example (the actual number of Hv's would have been 101 for an input of 100). From the input file, the total basis size can be seen to be $n\_x * n\_y * n\_z * (l\_max + 1)^2 = 10 * 10 * 10 * (6 + 1)^2 = 1000 * 49 = 49000$.

TABLE II. Execution Time for Hv and Hv Prep

Hv Counts	Wall Time	CPU Time	System Time
1	0m0.272s	0m1.010s	0m0.008s
2	0m0.453s	0m1.726s	0m0.010s
3	0m0.638s	0m2.464s	0m0.011s
4	0m0.817s	0m3.147s	0m0.013s
5	0m0.996s	0m3.865s	0m0.014s
11	0m2.124s	0m8.065s	0m0.027s
21	0m3.867s	0m15.033s	0m0.046s
101	0m18.449s	1m11.321s	0m0.167s
1001	3m2.104s	11m44.949s	0m1.607s

From the above data, it can be see that Hv_prep does not take a lot of time to run. Figure 1 shows the data up to a count of 101 (1001 was excluded for clarity). The data is clearly linear in nature and it can be surmised that one Hv run takes 0.182s and Hv_Prep takes 0.090s to run.

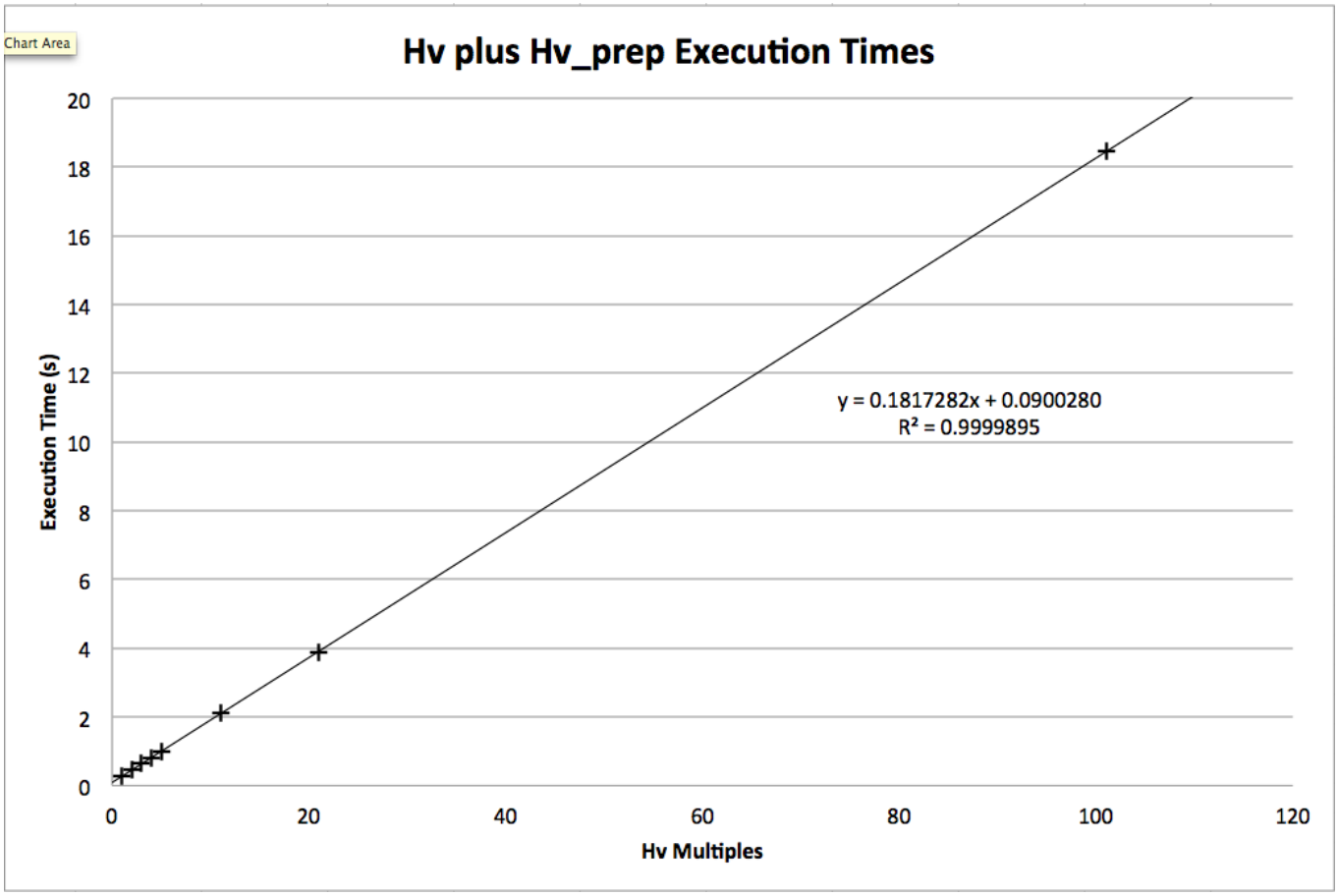

FIG. 1. Hv Plus Hv_Prep Wall Time