

9/24/12

Things to do:

- consider more directly sampling the anharmonicity
  - can we sample  $\langle E - E_{H_0} \rangle$  instead of  $\langle E \rangle - \langle E_{H_0} \rangle$ ?
- Reconsider truncated HO H<sub>2</sub>O tinker case seems to be getting stuck in  $-\infty$
- ✓ Run using Daniels outputs
- ✓ Directly include tinker code in PEAC
- ✓ Rework the math on HO propagator using harmonic modes ( $\Delta V$  should be diagonal, not  $H_0$ )
  - ↳ remember, harmonic modes are just new coordinates that uncouple  $V_{ab}$ , they are not  $E$ -basis!

Code clean up to do:

- Move update & (old) slice V to potential/?
- Sort out  $P \leftrightarrow V$  relationship possibly:
  - create obj. physics containing  $V \& P$
  - call estimate from  $V$ , not  $P$  so you can have all the information in one object
- Create new object "Estimator" to allow for more flexibility

9/26/12

## PIMC

Cartesian  $|x\rangle$   $\epsilon$ -basis for  $\Delta V$   
 norm modes  $|n\rangle$  ( $\epsilon$ -basis for  $H_0$ ) not true

$$Q = \text{Tr } e^{-\beta \hat{H}} = \int \dots \int dx_1 \dots dx_p \langle x_1 | e^{-\frac{\beta}{P} \hat{H}} | x_2 \rangle \langle x_2 | e^{-\frac{\beta}{P} \hat{H}} | x_3 \rangle \dots \langle x_p | e^{-\frac{\beta}{P} \hat{H}} | x_1 \rangle$$

$$\begin{aligned} \langle x_1 | e^{-\frac{\beta}{P} (H_0 + \Delta V)} | x_2 \rangle &= \langle x_1 | e^{-\frac{\beta}{P} H_0} e^{-\frac{\beta}{P} \Delta V} + O\left(\frac{\beta^2 \Delta V}{P}\right) | x_2 \rangle \\ &= \int dx' \langle x_1 | e^{-\frac{\beta}{P} H_0} | x' \rangle \langle x' | e^{-\frac{\beta}{P} \Delta V} | x_2 \rangle \\ &= \langle x_1 | e^{-\frac{\beta}{P} H_0} | x_2 \rangle e^{-\frac{\beta}{P} \Delta V(x_2)} \end{aligned}$$

$$Q = \int \dots \int dx_1 \dots dx_p \prod_{i=1}^P P_0(x_i, x_{i+1}, \frac{\beta}{P}) e^{-\frac{\beta}{P} \Delta V(x_i)}$$

$$\langle E \rangle = -\frac{\partial}{\partial \beta} \ln Q = \frac{1}{Q} \int \dots \int dx_1 \dots dx_p \left[ \sum_{i=1}^P -\frac{\partial}{\partial \beta} P_0(x_i, x_{i+1}, \frac{\beta}{P}) + \frac{\Delta V(x_i)}{P} \right] \prod_{i=1}^P P_0(x_i, x_{i+1}, \frac{\beta}{P}) e^{-\frac{\beta}{P} \Delta V(x_i)}$$

## Free Particle Propagator:

$$H_0 = \hat{T}, \quad \Delta V = V$$

$$P_0(x_1, x_2, \frac{\beta}{P}) = \left( \frac{2\pi \hbar^2 \beta}{mP} \right)^{-\frac{1}{2}} e^{-\frac{mP(x_1 - x_2)^2}{2\beta \hbar^2}}$$

$$-\frac{\partial}{\partial \beta} \ln P_0(x_1, x_{i+1}, \frac{\beta}{P}) = \frac{d}{2\beta} - \frac{mP(x_1 - x_{i+1})^2}{2\beta^2 \hbar^2}$$

$$E_{\text{est}} = \sum_i \frac{1}{P} \left( \frac{d}{2\beta} - \frac{mP(x_1 - x_{i+1})^2}{2\beta^2 \hbar^2} \right) + \frac{V(x_i)}{P} = \frac{1}{P} \sum_i \left( \frac{d}{2\beta} - \frac{m \|x_i - x_{i+1}\|^2}{2\beta^2 \hbar^2} \right) + \langle V \rangle$$

Levy flights (collective,  $H_0$  MC moves)

For restraining first of  $\Delta$  bonds:

$$\langle x_{i+1} \rangle = \frac{\Delta x_i + x_{i+1}}{\Delta + 1}$$

$$\sigma^2 = \frac{1}{1 + \frac{1}{\Delta}} \frac{\beta}{P m}$$

Work: enlarge to view

$$\begin{aligned} P_0(x_1, x_2, \theta) &= \langle x_1 | e^{-\frac{p^2}{2m} \frac{1}{\theta} p^2} | x_2 \rangle \\ &= \int dp \langle x_1 | p \rangle \langle p | e^{-\frac{p^2}{2m} \frac{1}{\theta} p^2} | p \rangle \langle p | x_2 \rangle \\ &= \int dp \langle x_1 | p \rangle \langle p | x_2 \rangle e^{-\frac{p^2}{2m} \frac{1}{\theta} p^2} \\ &= \int dp \left( \frac{2\pi m}{2\pi \hbar^2} \right)^{\frac{1}{2}} e^{\frac{ipx_1}{\hbar} - \frac{p^2}{2m} \frac{1}{\theta} p^2} \left( \frac{2\pi m}{2\pi \hbar^2} \right)^{\frac{1}{2}} e^{-\frac{ipx_2}{\hbar} - \frac{p^2}{2m} \frac{1}{\theta} p^2} \\ &= \left( \frac{2\pi m}{2\pi \hbar^2} \right)^{\frac{1}{2}} e^{-\frac{(x_1 - x_2)^2}{2m \frac{1}{\theta}}} \end{aligned}$$



$$\Delta = 2$$

$$Q = \int \cdots \int dx_1 \cdots dx_p \prod_{i=1}^p P_0(x_i, x_{i+1}, \frac{E}{P}) e^{-\frac{E}{P} \Delta V(x_i)}$$

$$\langle E \rangle = -\frac{\partial}{\partial \beta} \ln Q = \frac{1}{Q} \int \cdots \int dx_i dx_p \left[ \sum_{i=1}^p \frac{-\partial}{\partial \beta} P_0(x_i, x_{i+1}, \frac{E}{P}) + \frac{\Delta V(x_i)}{P} \right] \prod_{i=1}^p P_0(x_i, x_{i+1}, \frac{E}{P}) e^{-\frac{E}{P} \Delta V(x_i)}$$

SHO Propagator:

$$\hat{H}_0 = \hat{T} + \hat{V}_{HO}$$

$$V_{HO} = \frac{1}{2} \sum_i m_i x_i^2$$

Normal modes

$$\gamma = \frac{\beta \hbar \omega}{P}$$

$$P_0(x_i, x_{i+1}, \frac{E}{P}) = \left( \frac{m \omega}{2 \pi \hbar \sinh(\gamma)} \right)^{1/2} e^{-\frac{E}{P} \frac{m \omega^2 [(x_i^2 + x_{i+1}^2) \cosh(\gamma) - 2 x_i x_{i+1}]}{2 \gamma \sinh \gamma}}$$

$$= \left( \frac{m \omega}{2 \pi \hbar^2 \coth(\gamma)} \right)^{1/2} e^{-\frac{E}{P} \left[ \frac{m \omega^2}{2 \gamma} (\tanh \frac{\gamma}{2} (x_i^2 + x_{i+1}^2) + \coth \gamma (x_i - x_{i+1})^2) \right]}$$

$$\frac{\partial \ln P(x_i, x_{i+1}, \frac{E}{P})}{\partial \beta} = -\frac{1}{2} \frac{\hbar \omega}{P \tanh \gamma} - \frac{m \omega^2}{2 \pi} \operatorname{sech}^2(\frac{\gamma}{2}) \frac{\hbar \omega}{2 P} (x_i^2 + x_{i+1}^2) + \frac{m \omega^2}{2 P} \coth(\gamma) \operatorname{csch}(\gamma) (x_i - x_{i+1})^2$$

$$E_{est} = \frac{1}{P} \sum_i^P \sum_j^N \frac{1}{2} \hbar \omega_j \coth(\gamma_j) + \frac{1}{2} m_j \omega_j^2 \left[ \operatorname{sech}^4(\gamma_j) \frac{x_j^{(i)2} + x_j^{(i+1)2}}{2} - \coth(\gamma_j) \operatorname{csch}(\gamma_j) (x_j^{(i)} - x_j^{(i+1)})^2 \right] + V(x_j^{(i)}) - \frac{1}{2} m_j \omega_j^2 x_j^{(i)2}$$

Levy flight Gaussian moves

Gaussian distribution for first of 1 particles:

$$\langle x_{i+1} \rangle = \frac{x_i \operatorname{csch}(\gamma) + x_{i+\Delta+1} \operatorname{csch}(\Delta \cdot \gamma)}{\coth(\gamma) + \coth(\Delta \cdot \gamma)}$$

$$\sigma^2 = \frac{1}{m \omega} \frac{1}{\coth(\gamma) + \coth(\Delta \cdot \gamma)}$$

10/15/12

Thinking about estimator for  $E_{\text{ZPE}} - E_{\text{HO}}$

Defining estimator,  $\mathcal{E}$ :

$$\text{IF } Q = \int \cdots \int dx_1 \cdots dx_p K(x_1, \dots, x_p)$$

$$\langle E \rangle = \frac{\int \cdots \int dx_1 \cdots dx_p E(x_1, \dots, x_p) K(x_1, \dots, x_p)}{Q}$$

Based on  $E = -\frac{\beta}{2p} \ln Q$

$$\mathcal{E} = \frac{1}{p} \sum_i^p \sum_j^N \frac{1}{2} \tau \omega_j \coth(Y_j) + \frac{1}{2} m_j \omega_j^2 \left[ \operatorname{sech}^2(Y_j) \frac{x_j^{(i)} + x_j^{(i+1)}}{2} - \coth(Y_j) \operatorname{csch}(Y_j) (x_j^{(i)} - x_j^{(i+1)})^2 \right] + V(x_j^{(i)}) - \frac{1}{2} m_j \omega_j^2 x_j^{(i)} \Delta V(x_j^{(i)})$$

$$\text{What we really want is: } \langle \hat{H} \rangle_H - \langle \hat{H}_{\text{HO}} \rangle_{\text{HO}} = \underbrace{\Delta \text{ZPE}}$$

$$\langle \hat{H} \rangle_H = \frac{1}{Q_H} \int \cdots \int dx_1 \cdots dx_p E(x_1, \dots, x_p) \prod_i P(x_i, x_{i+1}, \beta/p) e^{-\beta p \Delta V(x_i)}$$

$$\langle \hat{H} \rangle_{\text{HO}} = \frac{1}{Q_{\text{HO}}} \int \cdots \int dx_1 \cdots dx_p E_{\text{HO}}(x_1, \dots, x_p) \prod_i P(x_i, x_{i+1}, \beta/p) \frac{\prod_i e^{-\beta p \Delta V(x_i)} \frac{Q_H}{Q_{\text{HO}}}}{\prod_i e^{-\beta p \Delta V(x_i)}} \frac{1}{\prod_i e^{-\beta p \Delta V(x_i)}} e^{\beta p \sum_i \Delta V(x_i)}$$

$$\star \langle \hat{H} \rangle_H - \langle \hat{H}_{\text{HO}} \rangle_{\text{HO}} = \frac{1}{Q_H} \int \cdots \int dx_1 \cdots dx_p \left[ E(x_1, \dots, x_p) - E_{\text{HO}}(x_1, \dots, x_p) \frac{Q_H}{Q_{\text{HO}}} \frac{\prod_i e^{-\beta p \Delta V(x_i)}}{\prod_i e^{-\beta p \Delta V(x_i)}} \right] K_H(x_1, \dots, x_p)$$

$$\frac{1}{p} \sum_i \Delta V(x_i) + \sum_i E_H(x_i, \dots, x_p) \left[ 1 - \frac{Q_H}{Q_{\text{HO}}} \frac{\prod_i e^{-\beta p \Delta V(x_i)}}{\prod_i e^{-\beta p \Delta V(x_i)}} \right]$$

$$Q_H = \frac{\int \cdots \int dx_1 \cdots dx_p \prod_i P(x_i, x_{i+1}, \beta/p) e^{-\beta p \Delta V(x_i)}}{\int \cdots \int dx_1 \cdots dx_p \prod_i P(x_i, x_{i+1}, \beta/p)} = \langle e^{-\beta p \sum_i \Delta V(x_i)} \rangle_{\text{HO}}$$

$$= \langle 1 - \frac{\beta}{p} (\sum_i V(x_i) - V_{\text{HO}}(x_i)) + \dots \rangle_{\text{HO}}$$

$$= 1 - \frac{\beta}{p} \langle V(x_i) \rangle_{\text{HO}} - \frac{\beta}{p} \langle \sum_i \frac{1}{2} m_i \omega_i^2 x_i^2 \rangle_{\text{HO}} + \frac{1}{4} k_B \coth(\frac{\beta \hbar \omega}{2})$$

$\langle \sum_i V(x_i) \rangle_{\text{HO}}$ ? We sample this exactly if we were to accept every Levy flight move.  
 Can we store  $\langle \hat{V} \rangle$  at each move regardless of acceptance criteria and use this for  $\langle \Delta V \rangle$ ?  
 We could probably use  $e^{\beta p \Delta V}$  w/o Taylor expansion...

10/16/12

## Deriving A2PE estimator

$$\langle \hat{H} \rangle_{\hat{H}} - \langle \hat{H}_{HO} \rangle_{HO} = \frac{1}{Q_{HO}} \int \dots \int dx_1 \dots dx_p \left[ \frac{1}{P} \sum_i \Delta V(x_i) + \mathcal{E}_{HO}(x_1, \dots, x_p) \left[ 1 - \frac{Q_H}{Q_{HO}} \prod_{i=1}^P e^{\beta \sum_i \Delta V(x_i)} \right] \right] K_H(x_1, \dots, x_p)$$

$$= \langle \Delta V \rangle_{\hat{H}} + \int \int dx_1 \dots dx_p \mathcal{E}_{HO}(x_1, \dots, x_p) \left( 1 - \frac{Q_H}{Q_{HO}} e^{\beta \sum_i \Delta V(x_i)} \right) \frac{K_H(x_1, \dots, x_p)}{Q_H}$$

$$\frac{Q_H}{Q_{HO}} = \frac{\int \int dx_1 \dots dx_p e^{-\beta \sum_i \Delta V(x_i)} K_{HO}(x_1, \dots, x_p)}{Q_{HO}} \neq \langle e^{-\beta \sum_i \Delta V} \rangle_{HO} \times \left[ \int \int dx_1 \dots dx_p e^{-\beta \sum_i \Delta V(x_i)} \frac{K(x_1, \dots, x_p)}{Q} \right]^{-1}$$

Estimator for  $F(V) = \frac{\sum_i F(V(x_i))}{P} \neq F(\frac{\sum_i V(x_i)}{P})$

$$\begin{aligned} \langle F(V) \rangle &= \int \int dx_1 \dots dx_p e^{-\beta \sum_i V(x_i)} \frac{F(V(x_1)) \dots F(V(x_p))}{Q} \frac{1}{Q} \\ &= \int \int dx_1 \dots dx_p F(V(x_1)) \dots F(V(x_p)) \frac{1}{Q} \text{ by reordering variables} \\ &= \int \int dx_1 \dots dx_p \frac{\sum_i F(V(x_i))}{P} \frac{e^{-\beta \sum_i V(x_i)}}{Q} \frac{1}{Q} \\ &\neq \int \int dx_1 \dots dx_p F(\frac{\sum_i V(x_i)}{P}) \frac{1}{Q} \end{aligned}$$

- ① For the Morse potential,  $\langle \Delta V \rangle < 0$  so we have a stable calculation if we use the second equation for  $\frac{Q_H}{Q_{HO}}$  and estimate this term along w/ A2E using estimator:  $e^{-\beta \sum_i \Delta V(x_i)}$ , stirring this increasingly accurate  $\frac{Q_H}{Q_{HO}}$  & apply to A2E estimator

- ② Another possibility is to use the HO sampling & the following A2E form & just do everything in the HO ensemble

$$\begin{aligned} \langle \hat{H} \rangle - \langle \hat{H}_{HO} \rangle_{HO} &= \frac{1}{Q_{HO}} \int \int dx_1 \dots dx_p \left[ \sum_i \left( \frac{Q_{HO}}{Q} e^{-\beta \sum_i \Delta V(x_i)} - \mathcal{E}_{HO}(x_i) \right) K_{HO}(x_i) \right] \\ &= \frac{1}{Q_{HO}} \int \int dx_1 \dots dx_p \left[ \sum_i \Delta V(x_i) \frac{Q_{HO}}{Q} e^{-\beta \sum_i \Delta V(x_i)} + \mathcal{E}_{HO}(x_i) \left( \frac{Q_{HO}}{Q} e^{-\beta \sum_i \Delta V(x_i)} - 1 \right) \right] K_{HO}(x_i) \end{aligned}$$

$$\frac{Q_{HO}}{Q} = \left[ e^{-\beta \sum_i \Delta V(x_i)} \right]_{HO}$$

where  $[ \cdot ]_{HO}$  indicates sampling over HO ensemble

After talking w/ Todd, it seems clear that evaluating  $\frac{Q_{HO}}{Q}$  independently from  $e^{-\beta \sum_i \Delta V}$  will be no good. Free energy calculations are too difficult.

Maybe there is a way to calculate  $\frac{Q_{HO}}{Q} e^{-\beta \sum_i \Delta V(x_i)}$ ?

10/23/12

$$\begin{aligned} \frac{e^{-\beta \sum_i \Delta V(x_i)}}{\left[ e^{-\beta \sum_i \Delta V(x_i)} \right]} &\approx \frac{1 + \beta \sum_i \frac{\Delta V(x_i)}{P} - \frac{\beta^2}{2} \sum_i \frac{\Delta V(x_i)}{P} \frac{\Delta V(x_i)}{P} + \dots}{\left[ 1 + \beta \sum_i \frac{\Delta V(x_i)}{P} - \frac{\beta^2}{2} \sum_i \frac{\Delta V(x_i)}{P} \frac{\Delta V(x_i)}{P} + \dots \right]_{HO}} \rightarrow \text{this requires sampling at each sampling point!} \\ &= \left[ \frac{1}{e^{\beta \left( \sum_i \Delta V(x_i) - \sum_j \Delta V(y_j) \right)}} \right]_{HO} \approx \frac{1}{\left[ 1 + \beta \left( \sum_i \frac{\Delta V(x_i)}{P} - \sum_j \frac{\Delta V(y_j)}{P} \right) + \frac{\beta^2}{2} \left( \sum_i \frac{\Delta V(x_i)}{P} - \sum_j \frac{\Delta V(y_j)}{P} \right)^2 + \dots \right]_{HO}} \\ &\approx \frac{1}{1 + \beta \langle \Delta V \rangle - \beta \sum_j \frac{\Delta V(y_j)}{P}} \end{aligned}$$

10/24/12

Second attempt at AnE estimator

$$\begin{aligned}\langle H + \Delta V \rangle_{H+\Delta V} - \langle H \rangle_H &= -\frac{\partial}{\partial \beta} \ln Q_{H+\Delta V} + \frac{\partial}{\partial \beta} \ln Q_H \\ &= -\frac{\partial}{\partial \beta} \ln \frac{Q_{H+\Delta V}}{Q_H}\end{aligned}$$

$$\frac{Q_{H+\Delta V}}{Q_H} = \left( \frac{\int \cdots \int dx_1 \cdots dx_p e^{\beta \left( \frac{\sum \Delta V(x_i)}{P} \right)} K_{H+\Delta V}(x_1, \dots, x_p)}{\int \int dx_1 \cdots dx_p K_{H+\Delta V}(x_1, \dots, x_p)} \right)^{-1}$$

$$\begin{aligned}\frac{\partial}{\partial \beta} \ln \frac{Q_H}{Q_{H+\Delta V}} &= \frac{Q_{H+\Delta V}}{Q_H} \frac{\partial}{\partial \beta} \int \int d\bar{x} e^{\beta \left( \frac{\sum \Delta V(x_i)}{P} \right)} \frac{K_{H+\Delta V}}{Q_{H+\Delta V}} \\ &= \frac{Q_{H+\Delta V}}{Q_H} \int \cdots \int dx_1 \cdots dx_p \sum_i \frac{\Delta V(x_i)}{P} e^{\beta \left( \frac{\sum \Delta V(x_i)}{P} \right)} \frac{K_{H+\Delta V}}{Q_{H+\Delta V}} + e^{\beta \left( \frac{\sum \Delta V(x_i)}{P} \right)} \frac{\partial}{\partial \beta} \frac{K_{H+\Delta V}}{Q_{H+\Delta V}}\end{aligned}$$

$$\frac{\partial}{\partial \beta} \frac{\prod_i P_o(x_i, x_{i+1}, \frac{f}{P}) e^{-P \Delta V(x_i)}}{\int \int d\bar{x} \prod_i P_o(x_i, x_{i+1}, \frac{f}{P}) e^{-P \Delta V(x_i)}}$$

$$= \left\{ \underbrace{\left[ \sum_j \frac{\frac{\partial}{\partial f} P_o(x_j, x_{j+1}, \frac{f}{P})}{P_o(x_j, x_{j+1}, \frac{f}{P})} - \frac{\Delta V(x_j)}{P} \right] K(\bar{x}) Q - K \left( - \int \int dy \Sigma(y) K(y) \right)}_{-\Sigma(\bar{x})} \right\} \frac{1}{Q^2}$$

$$= \underbrace{-\Sigma(\bar{x}) K(\bar{x})}_{Q} - \frac{K(\bar{x})}{Q} \underbrace{\int \int dy \Sigma(y) \frac{K(y)}{Q}}_{\langle \hat{H} + \hat{V} \rangle_{\hat{H} + \hat{V}}}$$

$$\begin{aligned}\frac{\partial}{\partial \beta} \ln \frac{Q_H}{Q_{H+\Delta V}} &= \frac{Q_{H+\Delta V}}{Q_H} \int \cdots \int dx_1 \cdots dx_p \sum_i \frac{\Delta V(x_i)}{P} e^{\beta \left( \frac{\sum \Delta V(x_i)}{P} \right)} \frac{K(x)}{Q} - e^{\beta \left( \frac{\sum \Delta V(x_i)}{P} \right)} \Sigma(\bar{x}) \frac{K(\bar{x})}{Q} - e^{\beta \left( \frac{\sum \Delta V(x_i)}{P} \right)} \langle \hat{H} + \hat{V} \rangle \frac{K(\bar{x})}{Q} \\ &= \int \cdots \int d\bar{x} \sum_i \frac{\Delta V(x_i)}{P} \frac{K_H(\bar{x})}{Q_H} - \Sigma(\bar{x}) \frac{K_H(\bar{x})}{Q_H} - \langle \hat{H} + \hat{V} \rangle \frac{K_H(\bar{x})}{Q_H} \\ &= [\langle H + V \rangle_{H+\Delta V} - \langle H \rangle_H] \quad \text{sign error somewhere}\end{aligned}$$

This got us nowhere  $\approx$

10/25/12

Need to consider implications of using a classical potential to estimate  $E_{\text{an}}$

We zero the energy at equilibrium fixing the absolute difference problem.

~~Consider directly sampling  $E_{\text{HO}}$  directly to give  $\langle \tilde{H}_{\text{HO}} \rangle_{\text{HO}}$~~  ← this doesn't make sense

Want  $\langle \tilde{T} + V_0(x_0) + \frac{1}{2}x^2 \rangle$  approximate this for classical potential using MP2 harmonic modes

11/10/12

Martin makes a good point that we should just directly use the harmonic modes from the AMEOBA force field. Vibrate.f appears to do get normal modes and frequencies. This will allow the code to be a self-contained anharmonic correction code independent of how HO ZPE calculated

One concern is that we will not be at the equilibrium geom. wrt AMEOBA

↳ one solution is that this method will fail if the diffuse is too large anyway

Note: promising first results w/ water. Shama interested in use for zeolites.

Martin suggested thinking about diffusion MC. Talk w/ Eric N. about this.

10/25/12

Calling fitter directly from my PJMC code:

General idea

- Make fitter \*.o files & libfitter.a
- Create wrapper (\*.l) file for functions I want to call
- In make file link (-l) to libfitter.a

# Diffusion MC

Cyrus Umrigar

$$\sum_j \frac{\phi}{\phi_j} P_{ij} \phi_j \psi_j = \sum_i \phi_i \psi_i$$

$$e^{-\tau(H,E)} \leftarrow \text{drift tells you if you have right } E$$

$$e^{-\tau H} \rightarrow \frac{n}{\pi} e^{-\frac{\pi}{n} \nabla^2 - \frac{\pi}{n} V}$$

↳ calculate drift potential from short time diffusion

How to calculate drift velocity?

2/5/13

Got better data for anharmonicity in H<sub>2</sub>O dimer from CCSD(T) + matches way better than VCI 4 CCSD numbers from qchem

Want to compare various force fields and hopefully find E<sub>ank</sub> to be relatively invariant..  
need other parameter files

3/18/14

Want to get something finished here.

Options:

- Deal w/ divergence of HO propagator
- Better estimator for anharmonicity i.e.  $\langle E_{H_0 + \Delta V} \rangle_{H_0 + \Delta V} - \langle E_{H_0} \rangle_{H_0}$ 
  - idea: is there some form of re-weighting that would accomplish this?

4/18/14

What would an estimator to  $\langle H_{HO} \rangle_{HO}$  be in the full ensemble?

Aha! Pat pointed out before that this is just umbrella sampling!

Umbrella sampling:

$$\tilde{\pi}(x) = \frac{w(x) K(x)}{\int w(x) K(x) dx} \quad \text{so we've changed the probability/ensemble}$$

$$\text{then } \langle A \rangle = \frac{\langle A_w \rangle_{\tilde{\pi}}}{\langle 1/w \rangle_{\tilde{\pi}}}$$

$$\text{For us, } K_{\tilde{\pi}} = e^{-\frac{E}{kT} \Delta V} K_{HO}$$

Notation Note: using  $\{A\}$  to mean quantum expectation value &  $\langle A \rangle$  as ensemble average so  $\{\hat{H}\} = \langle \hat{E} \rangle$

$$\text{so, } E_{\text{anh.}} = \{H\} - \{H_{HO}\}_{HO} = \left\langle \varepsilon_{HO} + \frac{1}{kT} \sum_j \Delta V(x_j^{(i)}) \right\rangle - \frac{\left\langle \varepsilon_{HO} e^{\frac{E}{kT} \Delta V} \right\rangle}{\left\langle e^{\frac{E}{kT} \Delta V} \right\rangle}$$

Hmm... this doesn't give us a smaller value to estimate but does give us a target that is moving in the same noise at least. The problem is just if sampling the exponential will be awful and whether  $\langle \varepsilon_{HO}/w \rangle \frac{1}{\langle 1/w \rangle}$  converges similarly to  $\langle \varepsilon \rangle$ .

4/24/14

Perhaps a better approach than umbrella sampling could be thermodynamic integration

Write  $V_1 \geq V_0 = V_{H_0} + V_1 = V_{H_0} + \Delta V$  + interpolating continuously from  $\lambda=0$  to 1

$$\text{ex: } V_\lambda = V_{H_0} + \lambda \Delta V$$

$$\text{Then } E_{\text{inh}} = \langle H \rangle - \langle H_{H_0} \rangle_{H_0} = \int_0^1 d\lambda \frac{\partial}{\partial \lambda} \langle H_\lambda \rangle_\lambda$$

So, what is an expression for  $\frac{\partial}{\partial \lambda} \langle H_\lambda \rangle_\lambda$ ?

$$\langle H_\lambda \rangle_\lambda = \frac{\sum_\lambda k_\lambda}{\sum k_\lambda} = -\frac{1}{\beta} \ln \sum_\lambda k_\lambda, \quad \frac{\partial}{\partial \lambda} \langle H_\lambda \rangle_\lambda = \frac{-\sum_\lambda k_\lambda \sum_\lambda k_\lambda}{(\sum k_\lambda)^2} + \frac{\sum_\lambda \frac{\partial}{\partial \lambda} k_\lambda + \sum_\lambda \frac{\partial}{\partial \lambda} k_\lambda}{\sum k_\lambda}$$

$$K_\lambda(x_i, x_{i+1}, \beta) = \left( \frac{m\omega \sqrt{\gamma} \tanh \frac{\gamma}{2} (x_i^2 + x_{i+1}^2) + \cosh \gamma (x_i - x_{i+1})^2}{2\pi \hbar^2 \beta \sinh(\gamma)} \right)^{1/2} e^{-\frac{\beta}{2} \left[ \frac{m\omega^2}{2\gamma} (\tanh \frac{\gamma}{2} (x_i^2 + x_{i+1}^2) + \cosh \gamma (x_i - x_{i+1})^2) \right] - \frac{k_B T}{\hbar} \Delta V(x_i)}$$

$$\frac{\partial}{\partial \lambda} k_\lambda = -\frac{k_B T}{\hbar} \Delta V K_\lambda \quad \frac{\partial}{\partial \lambda} \sum_\lambda = \frac{1}{\hbar} \Delta V$$



$$\frac{\partial}{\partial \lambda} \langle H_\lambda \rangle_\lambda = -\langle -\frac{k_B T}{\hbar} \Delta V \rangle_\lambda \langle \sum_\lambda \rangle_\lambda + \langle \frac{\Delta V}{\hbar} \rangle_\lambda + \langle -\frac{k_B T}{\hbar} \Delta V \sum_\lambda \rangle_\lambda$$

Hmm... realizing that usually you do this for Free energies & it gets you out of full free energy calcs but in this case still need multiple expectation values  
However, all smaller thanks to  $\Delta V$  showing up so maybe faster convergence ...

Or, what if we calculate the free energy instead?  $F_\lambda = -\frac{\ln Q_\lambda}{k_B T}$

$$F_1 - F_0 = \int_0^1 \frac{\partial F_\lambda}{\partial \lambda} d\lambda = \int_0^1 d\lambda \frac{\sum_\lambda \frac{\partial}{\partial \lambda} k_\lambda}{\sum_\lambda k_\lambda} = \int_0^1 d\lambda \frac{\sum_\lambda \frac{\Delta V}{\hbar} k_\lambda}{Q_\lambda} = \int_0^1 d\lambda \langle \frac{\Delta V}{\hbar} \rangle_\lambda$$

~~Weird initial tests seem to show  $\langle \Delta V \rangle_\lambda$  gets bigger as  $\lambda$  goes from 1 to  $\frac{1}{2}$  in Morse potential!~~  
Never mind.

5/22/14

Okay, caught my double bug finally... I had forgotten to multiply old PofEnergy by  $\lambda$ ! Hmm...  $\langle \lambda \Delta V \rangle$  still gives competitive results on tests...

$\langle \Delta V \rangle(\lambda)$  is almost perfectly linear in Morse potential!!

At one point considered using  $H_\lambda = H_0 + \lambda^2 \Delta V$  w/  $\Delta F = \langle 2\lambda \Delta V \rangle_{\lambda^2}$   
but since  $\langle \Delta V \rangle_{\lambda^2}$  linear, only need a single point!

6/4/14

Results are looking good!

- For  $H_2$  Morse, converges 2 times as fast as  $\langle E \rangle - E_{H0}$  wrt P
- Much more stable for  $H_2O$ . Convergence wrt P better & needs fewer samples!
- Same for  $H_2O$  dimer

Now to consider  $\Delta F_{\text{ab initio}}$ :

This is what I'm already calculating

$$\Delta F_{\text{ab init}} = F_{\text{exact}} - F_{H0} \approx \underbrace{(F_{\text{DFT}} - F_{FF})}_{\Delta F_{\text{ab init}}} + \underbrace{(F_{FF} - F_{H0})}_{\text{This is what I'm already calculating}}$$

$$\Delta F_{\text{ai}} = \sum_{\text{nuc}} \left\langle U_{\text{DFT}} - U_{\text{FF}} \right\rangle$$

$$\approx \langle \Delta U \rangle_{\text{FF}} - \frac{1}{2} \langle (\Delta U - \langle \Delta U \rangle_{\text{FF}})^2 \rangle$$

$$\approx \langle \Delta U \rangle_{\text{FF}}$$

Important to remember that  $U_{\text{DFT}}$  is the nuclear potential =  $E_{\text{ee}} + V_{\text{NN}}$ !

How can  $U_{\text{FF}}$  and  $V_{\text{NN}}$  be related to each other? I guess we still need to make some old fashioned T.

How to sample this? Could use PIMC but maybe easier semiclassical approach?

6/11/14

It seems that  $\langle \delta(\Delta V)^2 \rangle$  doesn't increase w/ P!

This could be a very important advantage, but how do I prove it?

$$\langle \delta V \rangle = \frac{\int \Delta V K}{Q}$$

$$\langle \delta(\Delta V)^2 \rangle = \langle \Delta V \rangle^2 + \langle \Delta V^2 \rangle - 2\langle \Delta V \rangle^2 = \langle \Delta V^2 \rangle - \langle \Delta V \rangle^2 = \frac{Q \int \Delta V^2 K - \int \Delta V K}{Q^2}$$

↳ See paper Herman 1982 for kinda complicated algebra

6/19/14

First lets just test  $\Delta_{\text{abinit}}$  by generating energy + geometries + run qchem independently + average by hand.

→ at storefreq, print V + geometry of specific bead

V.txt formatted as "# 1E V(hartree)"

geoms saved as #.in w/ \$molecule section

How to implement final version?

- Maybe run Test System the same way but have estimator that returns  $V_{\text{abinit}} - V_{\text{MM}}$   
↳ should all Systems have a  $\Delta V_{\text{abinit}}$  estimator?
- Better idea: just modify V-estimator scaling weight by lambda if running w/ delta QChem

If dabinit, don't run TI since we're approximating  $\langle \rangle_\lambda$  as  $\langle \rangle_{\lambda=1}$

For EstimateV return  $V_{\text{MM}} - V_{\text{abinit}}$  of one random bead

- For now print  $V_{\text{MM}}$  + save xyz coordinate

6/23/14

Thinking more carefully about what value we want for estimator:

$$\Delta F_{\text{anhar}} = F_{\text{qm}} - F_{\text{HO:QM}} = \underbrace{(F_{\text{QM}} - F_{\text{MM}})}_{\Delta_{\text{ab init}}} + \underbrace{(F_{\text{MM}} - F_{\text{HO:MM}})}_{\Delta_{\text{HO}}} + \underbrace{(F_{\text{HO:MM}} - F_{\text{HO:QM}})}_{\Delta_{\text{HO:QM}}}$$

V-Tinker subtracts off  $E_{\text{eqib:MM}}$  when calculating the potential

$$\text{Final, } \Delta F_{\text{anhar}} = \left( F_{\text{QM}} - E_{\text{QM}}^{\text{eff}} - \underbrace{F_{\text{MM}} + E_{\text{MM}}^{\text{eff}}}_{\tilde{F}_{\text{MM}}} \right) + \left( F_{\text{MM}} - E_{\text{MM}}^{\text{eff}} - \underbrace{F_{\text{MM}}^{\text{HO}} + E_{\text{MM}}^{\text{eff}}}_{\tilde{F}_{\text{MM}}^{\text{HO}}} \right) + \left( F_{\text{MM}}^{\text{HO}} - E_{\text{MM}}^{\text{eff}} - \underbrace{F_{\text{QM}}^{\text{HO}} + E_{\text{QM}}^{\text{eff}}}_{\tilde{F}_{\text{QM}}^{\text{HO}}} \right)$$

$\tilde{F}_{\text{MM}}$  are what is returned by V-Tinker, so running freq. at equib w/ QM will give

$$F_{\text{QM}}^{\text{HO}} + E_{\text{QM}}^{\text{eff}}$$

$$\Delta F_{\text{anhar}} = \underbrace{\int d\lambda \langle \tilde{V}_{\text{MM}} - \tilde{V}_{\text{MM}}^{\text{HO}} \rangle}_{\text{thermodynamic integration}} + \underbrace{\langle V_{\text{QM}} - \tilde{V}_{\text{MM}} \rangle}_{\text{force field QM correction}}_{\text{QM}} - E_{\text{QM}}^{\text{eff}} + \underbrace{(\tilde{F}_{\text{MM}}^{\text{HO}} - \tilde{F}_{\text{QM}}^{\text{HO}})}_{\text{absolute energy shift}}$$

Hmm... sampling  $V_{\text{QM}} - V_{\text{MM}}$  in the MM vs.  $\lambda$  ensemble is only accurate if  $\Delta V$  is small

$$F_{\text{MM}}^{\text{HO}} - F_{\text{QM}}^{\text{HO}} = \int d\lambda \langle V_{\text{MM}}^{\text{HO}} - V_{\text{QM}}^{\text{HO}} \rangle^{\text{HO}}$$

$$\text{want } \frac{(\Delta V_{\text{QM}} - \Delta V_{\text{MM}}) K_{\lambda}}{Q_{\lambda}}$$

$$H_{11} = H_{\text{QM}}, \quad H_{01} = H_{\text{QM}}^{\text{HO}}$$

$$H_{10} = H_{\text{MM}}, \quad H_{00} = H_{\text{MM}}^{\text{HO}}$$

$$H_{\lambda\sigma} = (\sigma H_{\text{QM}}^{\text{HO}} + (1-\sigma) H_{\text{MM}}^{\text{HO}}) + \lambda (\sigma \Delta V_{\text{QM}} + (1-\sigma) \Delta V_{\text{MM}})$$

$$\Delta V_{\text{QM}} = H_{11} - H_{01}, \quad \Delta V_{\text{MM}} = H_{10} - H_{00}$$

$$\left\langle \frac{\partial H_{\lambda\sigma}}{\partial \lambda} \right\rangle = \langle \sigma \Delta V_{\text{QM}} + (1-\sigma) \Delta V_{\text{MM}} \rangle_{\lambda} = \frac{\partial F}{\partial \lambda}$$

$$\left\langle \frac{\partial H_{\lambda\sigma}}{\partial \sigma} \right\rangle = \langle \Delta V_{\text{QM}} - \Delta V_{\text{MM}} \rangle \neq \frac{\partial F}{\partial \sigma}$$

$$F_x = -\frac{1}{\beta} \ln \int e^{-\beta H_x} \quad \frac{\partial F}{\partial x} = \frac{\int \frac{\partial H_x}{\partial x} e^{-\beta H_x}}{Q_x} = \left\langle \frac{\partial H_x}{\partial x} \right\rangle_x$$

but maybe approx



$$\frac{\partial^2 F_{\lambda\sigma}}{\partial \lambda \partial \sigma} = \frac{\int \frac{\partial H_{\lambda\sigma}}{\partial \lambda} k_{\lambda\sigma}}{Q_{\lambda\sigma}} + -\beta \frac{\int \frac{\partial H_{\lambda\sigma}}{\partial \lambda} \frac{\partial H_{\lambda\sigma}}{\partial \sigma} k_{\lambda\sigma}}{Q_{\lambda\sigma}} + \beta \frac{\int \frac{\partial H_{\lambda\sigma}}{\partial \sigma} k_{\lambda\sigma}}{Q_{\lambda\sigma}} \frac{\int \frac{\partial H_{\lambda\sigma}}{\partial \lambda} k_{\lambda\sigma}}{Q_{\lambda\sigma}}$$

$$= \langle \Delta V_{QM} - \Delta V_{MM} \rangle_{\lambda\sigma} - \beta \langle (\sigma \Delta V_{QM} + (1-\sigma) \Delta V_{MM}) (V_{QM}^{HO} + \lambda \Delta V_{QM} - V_{MM}^{HO} - \lambda \Delta V_{MM}) \rangle_{\lambda\sigma} \\ + \beta \langle \sigma \Delta V_{QM} + (1-\sigma) \Delta V_{MM} \rangle_{\lambda\sigma} \langle V_{QM}^{HO} + \lambda \Delta V_{QM} - V_{MM}^{HO} - \lambda \Delta V_{MM} \rangle_{\lambda\sigma}$$

$$\int_0' d\lambda \int_0' d\sigma \frac{\partial^2 F_{\lambda\sigma}}{\partial \lambda \partial \sigma} = \int_0' d\lambda \frac{\partial^2 F_{\lambda\lambda}}{\partial \lambda^2} - \frac{\partial^2 F_{\lambda 0}}{\partial \lambda^2} = F_{11} - F_{01} - F_{10} + F_{00}$$

$$\Delta E_{anh} = F_{11} - F_{01} = F_{11} - F_{10} + F_{10} - F_{00} + F_{00} - F_{01} = \underbrace{F_{10} - F_{00}}_{\text{Sd}\lambda \langle \Delta V_{MM} \rangle} + \int \int d\lambda d\sigma \frac{\partial^2 F_{\lambda\sigma}}{\partial \lambda \partial \sigma}$$

$$= \underline{\int d\lambda \langle \Delta V_{MM} \rangle} + \int \int d\lambda d\sigma \langle \Delta V_{QM} - \Delta V_{MM} \rangle_{\lambda\sigma} - \beta \int \int d\lambda d\sigma \langle S(\sigma) L(\lambda) \rangle_{\lambda\sigma} - \langle S(\sigma) \rangle_{\lambda\sigma} \langle L(\lambda) \rangle_{\lambda\sigma}$$

$$\int \int d\lambda d\sigma \langle S(\sigma) L(\lambda) \rangle = \int \int d\lambda d\sigma \langle (\sigma \Delta V_{QM} + (1-\sigma) \Delta V_{MM}) (V_{QM}^{HO} + \lambda \Delta V_{QM} - V_{MM}^{HO} - \lambda \Delta V_{MM}) \rangle_{\lambda\sigma}$$

Assume changes wrt  $\sigma$  small in averages then  $\int_0' d\sigma \langle \sigma C \rangle_{\sigma} \approx \frac{1}{2} \langle C \rangle$

so we have:  $\approx \int d\lambda \langle \frac{1}{2} (\Delta V_{QM} + \Delta V_{MM}) L(\lambda) \rangle$

$$\int \int d\lambda d\sigma \langle S(\sigma) \rangle_{\lambda\sigma} \langle L(\lambda) \rangle_{\lambda\sigma} \approx \int d\lambda \langle \frac{1}{2} (\Delta V_{QM} + \Delta V_{MM}) \rangle_{\lambda} \langle L(\lambda) \rangle_{\lambda}$$

Isn't this second term a correlation function between  $S(\sigma)$  &  $L(\lambda)$ ?

If so, should they be correlated ???

$$\text{Let's try calculating this. } \beta \left\langle \frac{1}{2} (\Delta V_{QM} + \Delta V_{MM}) \left( V_{QM}^o + \frac{1}{2} \Delta V_{QM} - V_{MM}^o - \frac{1}{2} \Delta V_{MM} \right) \right\rangle \\ - \beta \frac{1}{2} \left\langle \Delta V_{QM} + \Delta V_{MM} \right\rangle \left\langle V_{QM}^o + \frac{1}{2} \Delta V_{QM} - V_{MM}^o - \frac{1}{2} \Delta V_{MM} \right\rangle$$

Gonna need to print  $\Delta V_{MM}$  &  $V_{MM}^o$

7/1/14

Figuring out cartesians to normal modes:

diagonalizing Hessian gives  $H = U^T \omega U$

$$U^T U = \mathbb{I}$$

$$U_{ijk} \cdot U_{ijk} = \delta_{ii'}$$

$$D_{ijk} q_i = x_{jk}$$

$$U_{ijk} \cdot \frac{1}{\sqrt{m_j}} = \tilde{D}_{ijk}$$

$$\text{so } \tilde{D}_{ijk} \frac{m_j}{c_i c_i} \tilde{D}_{ijk} = \tilde{D}_{ijk} \frac{\sqrt{m_j}}{c_i} \frac{\sqrt{m_j}}{c_i} \tilde{D}_{ijk} = \delta_{ii'}$$

$$D_{ijk} = \tilde{D}_{ijk} \sqrt{\sum_{j'k'} \tilde{D}_{j'k'}^2} = \tilde{D}_{ijk} c_i$$

$$\sum_{jk} \tilde{D}_{ijk}^2 = \sum_i U_{ijk} U_{ijk} \frac{1}{m_j}$$

$$c_i = \frac{1}{\sqrt{\sum_{j'k'} \tilde{D}_{j'k'}^2}} = \sqrt{\sum_{jk} U_{ijk}^2 \frac{1}{m_j}}$$

$$\sum_i c_i U_{ijk} \frac{1}{\sqrt{m_j}} q_i = x_{jk} \Rightarrow \underbrace{\sum_{jk} U_{ijk} \sqrt{m_j} \frac{1}{c_i} x_{jk}}_{\tilde{D}_{ijk} m_j \frac{1}{c_i^2} x_{jk}} = \sum_{ijk} c_i \frac{1}{c_i} U_{ijk} \sqrt{m_j} \frac{1}{m_j} \underbrace{q_i}_{\delta_{ii'}} = \sqrt{\mu_i} q_i$$

Quick fix, save  $U$  or  $c_i$  when transforming coordinates  $\star$

$$x_{jk} = \sum_i D_{ijk} q_i + x_{jk}^0 \Rightarrow x_{jk} - x_{jk}^0 = \sum_i \frac{c_i}{\sqrt{m_j}} U_{ijk} q_i$$

$$\sum_{jk} U_{ijk} \frac{\sqrt{m_j}}{c_i} \frac{1}{k} (x_{jk} - x_{jk}^0) = \sum_{ijk} \frac{c_i}{c_i} U_{ijk} U_{ijk} q_i = q_i$$

$$q_i = \sum_{jk} \tilde{D}_{ijk} \frac{m_j}{c_i^2} \frac{1}{k} (\Delta x_{jk})$$

$\star$  Just discovered that  $c_i = \mu_i$

What if we just have  $D$ ?  $D_{ijk} = U_{ijk} c_i \frac{1}{\sqrt{m_j}}$   $\underline{\underline{D}} = \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \\ 0 & 0 \end{pmatrix} \underline{\underline{U}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Don't think I can calculate  $c$  w/o  $U$ ...

Can I just find right inverse of  $D$ ? make  $n$  = collective  $jk$  index

$$\sum_i q_i D_{in} = x_n$$

$$D_{in} = L_{ij} \lambda_{jn} R_{mn}$$

$$D^{-1} = R^T (A^{-1})^T L^T \quad \sum_i D_{in} D_{nj}^{-1} = L_{ik} \lambda_{kn}^{-1} \lambda_{ml}^{-1} L_{lj} = \delta_{ij}$$

7/2/14

## Implementing V-QChem

Initialize: run (or read) equil. frequency job to get normal modes,  $\omega$ , & equil  $E$

Get Potential: given geometry, run sp & subtract off harmonic energy  
↳ for now print input & write harmonic  $E$

Instead of cart to normal mode, couldn't we just calculate potential in cartesians?

$V = \sum_i \frac{1}{2} \mu_i \omega_i^2 q_i^2$  hmm, looks like same problem of converting from  $q \rightarrow x$

- Look at Joel Bowman papers for  $H_2O$  ZPE from diffusion MC
- Look at Li Ping papers + reweighting of MC

7/29/14

Need to implement general reverse transformation of cartesians to normal modes since we need to convert from xyz into qchem modes

- linked in armadillo to get left inverse of  $D$  in  $D^{-1}q = \underline{x}$

- Still need to add the sampling of V-QChem to Sim. or TestSgs.

↳ adding it to TestSgs::EstimatorV since if this gets

hooked into qchem, it could just return  $\Delta V_{nm} - \Delta V_{qm} + \frac{1}{2}(\Delta V_{nm} + \Delta V_{qm})(L(\frac{1}{2}))$

but needs to store  $\langle \Delta V_{nm} + \Delta V_{qm} \rangle + \langle L(\frac{1}{2}) \rangle$  separately

Could sample  $\Delta V_{nm}$  frequently & at rare moments calc  $\Delta V_{qm}$  & add weighted to the running avg but would need to make sure freq. divides # runs or else it'll be off

\* Also need to figure out reweighting & make sure to do it using the  $\sigma = \lambda = \frac{1}{2}$  reference

8/1/14

Looks like reweighting is simply calculating boltzmann delta weights + normalizing them:

$$\langle \Delta V_{QM} - \Delta V_{MM} \rangle_{\sigma=1/2} = \frac{\sum_i^{\sigma=0} w_i (\Delta V_{QM} - \Delta V_{MM})}{\sum_i w_i}, \quad w_i = e^{\frac{1}{k} (\Delta V_{QM} + \Delta V_{MM}) - \frac{1}{2} \Delta V_{MM}} \quad (\frac{1}{4} \text{ from } \lambda, \sigma = \frac{1}{2})$$

$$= e^{\frac{1}{k} k_B T_p (\Delta V_{QM} - \Delta V_{MM})}$$

$$\text{If not scaled } e^{\frac{1}{k} (\Delta V_{QM} + \frac{1}{2} (\Delta V_{QM} + \frac{1}{2} \Delta V_{MM} + \Delta V_{MM} + \frac{1}{2} \Delta V_{MM}) - \frac{1}{2} \Delta V_{MM})} = \frac{1}{2} e^{[\Delta V_{QM} - \Delta V_{MM} + \frac{1}{2} (\Delta V_{QM} - \Delta V_{MM})]}$$

8/5/14

Problem: For delta AI correction we get major problems due to having different equib geometries! Made clear by plotting  $V$  for CN

Idea: What if we define MM physics from the MM equib

→ We just need a model physics for QM so we say use force field with equib

Do we also want to rescale steps?

↳ Levy flights only from MM so weight for QM should be  $V_{QM} - V_{MM}^{HO} = \Delta V_{QM} - (V_{QM}^{HO} - V_{MM}^{HO})$

\* We want to shift coordinate system to match MM/QM minima, so the question is, do we want to scale the coord. system so they have the same harmonic freq?

$$q' = \alpha q \quad V(q') = \frac{1}{2} m_i \omega_i^2 q'^2 = \frac{1}{2} m_i \alpha^2 \omega_i^2 q^2$$

If we rescale by  $\frac{\bar{\omega}_i}{\omega_i} \sqrt{\frac{m_i}{m_i}}$  we'll calculate the difference in sampling  $H_{MM}^{HO} + H_{QM}^{HO}$  in their own respective frequencies... I think.

Let's work out full coord transform:

$$V_{MM}(\bar{q}_i) = \frac{1}{2} \bar{m}_i \bar{\omega}_i^2 \bar{q}_i^2 \quad q' = \sqrt{\frac{m_i}{\bar{m}_i}} \frac{\bar{\omega}_i}{\omega_i} \bar{q}_i \rightarrow V_{MM}(q'_i) = \frac{1}{2} m_i \omega_i^2 q'^2 \quad V_{QM}(q) = \frac{1}{2} m_i \omega_i^2 q^2 \quad + \text{convert to } x \text{ using } x_{initQM}$$

So take  $\bar{q}_{MM}$  + scale, calc  $V_{QM}^{HO}$  then transform to  $x_{QM}$  using  $x_{initQM}$  + calculate energy

$$\text{by } \sqrt{\frac{m_i}{\bar{m}_i}} \frac{\bar{\omega}_i}{\omega_i}$$

8/12/14

Further code improvement idea: create estimator object & keep vector of estimators around

examples: basic E  $\rightarrow$  needs  $q_i + \Delta V$

$V \rightarrow$  needs access to  $V$

$\Delta V \rightarrow$  same

$x \rightarrow$  needs  $q_i$

$\Delta AI \rightarrow$  needs  $V_{link} + V_{qden}$

maybe member of Sim initialized w/ a system?

---

Problems to consider:

- Free energy vs. Energy  $\rightarrow$  difference w/ size of system? Temp?

- 120 k vs. 0 k?

- Using  $\beta/p$  for resampling + correcting in  $\Delta AI$ ?

Why we don't do TI

(33) Yang, W.; Cui, Q.; Min, D.; Li, H. QM/MM Alchemical Free Energy Simulations: Challenges and Recent Developments; Annual Reports in Computational Chemistry; Elsevier: 2010; Vol. 6.

---

9/9/14

Summary of recent(ish) developments:

Using optimized tinker geometries led to much lower (20%) AH2PE

$\Delta E$  correction improves this but not enough (60-80%) 

Want to try scaling coordinates but do it right this time

↳ need to reorder modes to match MM & QM by overlap

Simple alg: calculate overlap matrix, pick abs max element & assign those two nodes to each other + remove from matrix & repeat. Then reorder modes & scale by sign of overlap

$\max S_{ij} \rightarrow$  value stored in `overlap[i]`  
index stored as `swap[i] = j`

Form S

for  $i < nModes$

get abs max S index + save in swap, overlap  
zero out column + row

Permute modes, omega, & relMass by swap & scale modes by sign of overlap

9/12/14

very small  $\Delta$  between MM/QM

Implemented and seems to work well for H<sub>2</sub>O monomer, but looks just as bad for dimer

- one key difference is the monomer has 3 strong modes only + good overlap of MM + QM modes  
while the dimer has some overlaps ~ 0.7-0.8.

- Also, small differences in weak mode frequencies leads to varying  $\frac{\omega'}{\omega \sqrt{f}}$  on order of 0.5-2

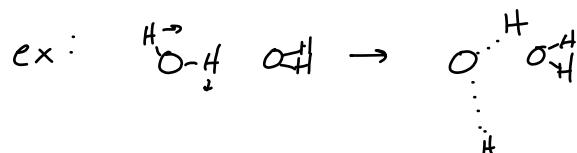
Am I maybe doing the scaling wrong? Currently taking QM  $q = q' \cdot \text{scale}$  + then converting to x from q

Should I convert from  $q'$  to x? No, calculating QM  $E_{h0} + \Delta E$

Did I forget to permute something?

11/8/14

Problem: Should be representing normal modes in internal coordinates for weak modes!



Problem: How do we pick fair, functioning internal modes??

Cases:  $\text{H}_2\text{O}$  should give  $\overset{\text{H}}{\underset{\text{H}}{\text{O}-\text{H}}}$ ,  $\text{H}_2\text{O}$  dimer maybe collective rotations

Ideas: Select nonredundant subset that are most correlated to normal mode / give most diagonal representation?

thoughts: high frequencies mostly relate to bond lengths...



we could use knowledge about connectivities

↳ pick all bonds & consider L's & dihedrals based on connectivities

↳ further consider rotations & distances between disconnected clusters

For  $\overset{\text{H}}{\underset{\text{H}}{\text{O}-\text{H}}}$  how to pick  $\text{CHOH}$  vs.  $\text{HH}$

Maybe first localize to give unsym. stretches



↳ how do bonds localize?



27 10 bonds  
18 L's

Alg. Idea:

- consider all bonds w/in some distance in possible coords list = P

Repeat:

- Start w/ highest modes and select bond most correlated to vibration modes and add to selected list = S

- Add all L's between adjacent bonds in S to P

- Do the same for torsions for all length 3 paths in S to P

until  $|S| = 3N - 6$

For clusters, maybe prefragment and include com cluster distances & rotations

In this case you'd need  $P_i$  & search paths only w/in fragment

& remove  $P_i$  when  $|S_i| = 3N_i - 6$  and  $|S_{\text{all}}| = 6(n-1)$

$$\sum_i 3N_i - 6 = 2(3N_i - 6) + (n-1)6$$

for clusters: look at low freq  $\text{SO}_4(\text{H}_2\text{O})_n$  clusters to see how to handle intercluster (distances only or angles too?)

Also, how do we choose rotation axis?

### Technical Issues to deal w/:

How to represent  $\Delta, \angle, + \curvearrowright$  in terms of cartesians?

↳ look into old E.B. Wilson maybe, this is an old problem.

Handle all the trickiness of curvilinear coord. transforms

How to determine which internal coord. "best represents" the normal mode

↳ simplistic idea: for mode  $q$  & int. coords  $b_i$   $\max \langle q | b_i \rangle$

↳ but we only want to consider at infinitesimal displacements...

↳ This inner product is probably not straight forward...

11/9/14

### Transforming to a local basis:

$x_i = i^{\text{th}}$  cartesian coordinate (i.e.  $x_1, y_1, z_1$ )

$e_i = i^{\text{th}}$  cartesian basis function  $\rightarrow r = \sum x_i e_i$

$$r = x_i e_i$$

$q_i = i^{\text{th}}$  internal coordinate (i.e.  $r_{12}, \theta_{123}, \tau_{1234}$ )

$b_i = i^{\text{th}}$  internal, local basis function (for given point  $r$ )

$b_i = \frac{\partial r}{\partial q_i} / h_i \quad h_i = |\frac{\partial r}{\partial q_i}| \quad \text{i.e. } b_i \text{ tangent to } q_i \text{ curve}$

$\underline{b}^i = \frac{\nabla q_i}{|\nabla q_i|} \quad \text{i.e. unit normal to surface of constant } q_j \neq i$

$g_{ik} \equiv b_i \cdot b_k = h_i h_k \quad \text{metric tensor}$

$$\underline{g}^{ik} \equiv b^i \cdot \underline{b}^k = (\underline{g})^{-1}_{ik}$$

$J^i_k \equiv \frac{\partial x_i}{\partial q^k} \quad \text{Jacobian} \quad (\underline{J}^{-1})_i^k = \frac{\partial q^i}{\partial x_k} \quad \times J \text{ is not invertible}$

$$\text{Note: } \underline{b}_k = \underline{e}_i \frac{\partial x_i}{\partial q^k} \quad \text{or} \quad \underline{b} = \underline{e} \underline{J}$$

$$\underline{e}_i = \underline{b}_k \frac{\partial \underline{q}^k}{\partial x_i} = (\underline{J}^{-1})_i^k \underline{b}_k$$

So what about for Hessians?

We have  $H = U^T \omega U = M^{\frac{1}{2}} D^T \omega D M^{\frac{1}{2}}$  w/ mass weighted normal modes  $D$

$$\text{Mode } i: n_i = d_i^j e_j = d_i^j \frac{\partial q^k}{\partial x_j} b^k$$

$$n_i \cdot b^j = d_i^k \frac{\partial q^j}{\partial x_k} g^{kj} = n_i^j$$

$$\text{So } n_i = n_i^j b_j \quad \underline{n} = \underline{D} \underline{J}^{-1} \underline{G}^{-1} \underline{B} \quad b_{ij} = b_i(q) \cdot e_j ?$$

11/14/14

Implementing Rollin King's object oriented internal coordinate code (opt2man)

First, need to pick a set of internal coordinates, can be done by opt2man

Then, coordUtil needs modified normalModesToCart + reverse functions

↳ this should require B matrices

11/17/14

Note: opt2man doesn't use delocalized internals... ?

Only now realized we'll have to recompute B matrices at each geom

Left singular vectors w/ non-zero s.v.s give delocalized internals

$$B = [U U'] [\Sigma_0] [V^T] \quad \tilde{B} = U \Sigma V^T \quad \begin{matrix} n_{\text{Prim}} \\ \{ \boxed{u} \} \end{matrix} \boxed{\Sigma_0} \boxed{V^T} \underbrace{\boxed{v_i}}_{n_{\text{Modes}}} \boxed{n_{\text{Cart}}} = \boxed{n_{\text{Prim}}}$$

$B^T$  gives primitives in terms of cartesians  $\boxed{n_{\text{Prim}}} \boxed{n_{\text{Cart}}}$  could be  $\boxed{n_{\text{Prim}}} \boxed{n_{\text{Cart}}}$

So to move in internal mode  $i$  by  $\alpha$  we displace cartesians by:  $\alpha B \cdot u_i$  ?

$$\boxed{n_{\text{Cart}}} \boxed{n_{\text{Prim}}} \cdot \boxed{n_{\text{Prim}}} = \boxed{n_{\text{Cart}}}$$

Let's try again:

We want  $\Delta \underline{x} = \Delta \underline{x}^i e_i$  for a change in primitive internal i

$$\underline{x}^i b_i = \underline{x}^i B_{ik}^+ e_k$$

Hmn... seems I'm not including  $M^{1/2}$  as a metric?

$$B^{-1} = M^{-1}(B^T M^{-1} B)^{-1} B^T ? \quad \text{X}$$

Looks like  $B^+$  will give angle bending but doesn't preserve C.O.M.

for that looks like we need  $M^{-1}(B^T M^{-1} B)^{-1} B^T$

Need to double check that delocalized internals diagonalize  $B^T M^{-1} B$   
rather than  $B^T B$  as in the paper

---

11/18/14

### Making sense of $M^{-1}$

The issue is that we want our displacement vectors to preserve center of mass, rather than unweighted displacement. This means our cartesian coordinates need a  $M^{-1}$  metric applied so that transformations are unitary wrt  $M^{-1}$  i.e.  $V M^{-1} V^T = \mathbb{I}$

This is why McIntosh constructs  $(B M^{-1} B^T)^{-1}$  rather than SVD of  $B$ .

However we need weighted SVD of  $B = U \Sigma V^T$  w/  $V^T M^{-1} V = \mathbb{I}$   $U^T U = \mathbb{I}$

+ can do this w/ standard SVD of  $B M^{-1/2} = U \Sigma (V^T M^{-1/2})$   $(V^T M^{-1/2})(M^{-1/2} V) = \mathbb{I}$

$$\text{Then } B^+ = M^{-1/2} (M^{-1/2} V) \Sigma^{-1} U^T = M^{-1/2} (B M^{-1/2})^+$$

To get delocalized coordinates start w/  $\text{svd}(B M^{-1/2}) = \underbrace{\begin{bmatrix} U & U' \\ 0 & 0 \end{bmatrix}}_{n_{\text{prim}}} \underbrace{\begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}}_{n_{\text{vib}}} \underbrace{\begin{bmatrix} V^T \\ V^T \cdot V \end{bmatrix}}_{n_{\text{Cart}}}$   
where  $V^T$  is mass weighted

Further consideration, making sure steps are normalized to 1

$Q = BX$  so I think rows of  $B$  should be normalized

Then we should get unit displacements in deloc. intervals using  $U^T B$ :

$$\Delta X = M^{-\frac{1}{2}}(U^T B M^{-\frac{1}{2}})^+ \Delta Q$$

11/19/14

Nope, need to normalize  $M^{\frac{1}{2}}(BM^{-\frac{1}{2}})^+$  which lists the internal displacements in terms of cartesian displacements.

So how to implement for PIMC code?

- do we calculate delocalized intervals only at the first step or recalculate throughout?

↳ For now, just at the beginning, but we have to svd  $BM^{\frac{1}{2}}$  at each step anyway

- keep set of primitives (+deloc) + for each pSlice that moves recalculate  $B$

- so if  $P=200$  + Levy flight = 50 we move 50 pSlices

so update  $B$  then move  $M^{-\frac{1}{2}}(U^T B M^{-\frac{1}{2}})^+$  for each of the 50

- this means I need to move each pSlice at once

- CoordUtil need class mat deloc +  $M^{-\frac{1}{2}}$

- CoordUtil needs normalModeToCartesian for internal coords

↳ this just probably needs addition of  $M^{-\frac{1}{2}}(U^T B M^{-\frac{1}{2}})^+$  to what is already written

11/19/14

Major issue:

Currently coords are changed absolutely but internals use local basis  
for small changes

One possibility is to take multiple steps from  $x_0$  but this sounds  
~~spec~~ inefficient...



↳ Let's implement this.

- calculate largest mode displacement for step  $\vec{s}$
  - calc minimum # steps ( $\alpha$ ) to break  $\vec{s}$  into so no modes > maxStepSize
  - $\Delta x = N \cdot B \cdot M^{-\frac{1}{2}} \cdot (BM^{\frac{1}{2}})^+ \cdot \frac{1}{\alpha} \vec{s} = \frac{1}{\alpha} N \cdot \vec{s}$   $\frac{1}{\alpha}$  times cartesian displacement
- for  $i = 1 \dots \alpha - 1$
- Make step  $\Delta x$  + calculate new  $B \equiv B_i$  X
  - $\Delta x = N \cdot B_i \cdot M^{-\frac{1}{2}} \cdot (BM^{\frac{1}{2}})^+ \cdot \frac{1}{\alpha} \vec{s}$

Note: After normalizing  $\tilde{B}^{-1}$   $B \tilde{B}^{-1}$  is diagonal but not I

→ should normalize B first + then see about  $\tilde{B}^{-1}$

↳ by row

$$Q_i = B_i^{-1} X_j \quad \tilde{B}_k^{-1} Q_i = X_k \quad \tilde{B}_k^{-1} B_i^{-1} =$$

11/24/14

So from McIntosh:  $G = BM^{-1}B^T = D\Gamma D^T + Q = M^{-1}B^T(BM^{-1}B^T)^{-1} = M^{-1}B^T B^{-1} + \omega / B^T = BM^{\frac{1}{2}}$

This just what I have. So lets go ahead + normalize  $\tilde{B}_{norm}^{-1} = \tilde{B}_{norm}^{-1} \cdot \omega_{norm}$  ~~diagonal~~  
 $\omega^{-1} \cdot B_{nn} \cdot \tilde{B}_{nn}^{-1} \omega = 1$

12/3/14

Let's work out the logic in words:

- we start with a step in normal modes
- we ultimately want to take that step in intervals
- first we represent the step in the local internal basis & take a step by converting to cartesians
- we then recalculate the local basis & using the same internal representation of the step, we convert to carts & take a steps

So first, normal mode step to carts to intervals

$$\sqrt{\boxed{c}} \boxed{\prod_{\text{cart}}^v} \|_{\text{vib}}^2 \quad B \cdot N \cdot \text{step} = \text{step}' \text{ in intervals}$$

For each step recalc  $B^+$  & apply to step'

If  $B + B^+$  are column normalised, then  $\| \text{step} \|$  will be maintained in carts

$$B^+ = M^{-1/2} (B M^{-1/2})^+ = M^{-1/2} V I$$

Originally we calculate  $\tilde{B}$  from carts to primitives.  $\sqrt{\boxed{c}}^{\text{prim}}$  transforms from primitives to linearly ind. modes

$$B = U^T \tilde{B} = \sqrt{\boxed{U}} \boxed{\tilde{B}}^c P \quad \tilde{B} = U \Sigma V^T$$

$$\tilde{B}^+ = \tilde{B}^+ U \quad \tilde{B}^+ = M^{-1/2} M^{-1/2} V^T \Sigma U^T \quad B B^+ = \underline{U^T U} \underline{\Sigma} \underline{V^T M^{-1} V} \underline{\Sigma^{-1}} \underline{U^T U} = 1$$

So it doesn't matter whether  $B^+$  defined by  $B^+ = \tilde{B}^+ U$  or  $B B^+ = 1$

$$\text{Aside: IF } N \text{ col normal} \Rightarrow \| N v \|^2 = \sum_i \sum_j N_{ij}^2 v_j^2 = \sum_j v_j^2 (\sum_i N_{ij})^2 = \sum_j v_j^2 = \| v \|^2$$

So, the carts are

$$B_{\text{new}}^+ B_{\text{Ns}} = U^T \tilde{B}_{\text{new}}^+ \tilde{B} U N_s = U^T M^{-1/2} (\tilde{B} M^{-1/2})^+ \tilde{B} U N_s =$$

12/5/14

Aha, the problem seems to be that the interfragment modes are bad

It picks the O-H-O  $\angle$  which is nearly linear + some torsion give nearly the same motion

Trying to see if principal axis works better otherwise try adding full fragment displacement + rotation

---

12/8/14

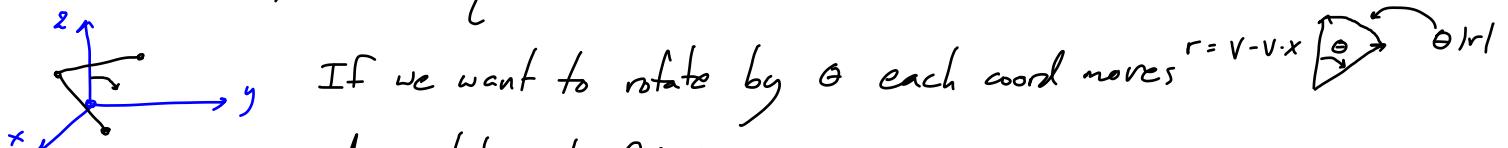
So principal axis stuff couldn't have been implemented since inertia-tensor was all kinds of wrong

Let's just add rotations + translations as columns to  $B$

\*Could use principal axes for rotations, but for now let's pick x,y,z axis

Translations are simple  $B_i^* = \delta_{x_i, z} \propto \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}^T$

For rotations, let's use quaternions?  $\times$



consider relative to COM so  $v_i = x_i - \text{COM}$ ,

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} v_1 \\ cv_2 - sv_3 \\ sv_2 + cv_3 \end{pmatrix} \quad \text{distance of } \theta \left| \begin{pmatrix} 0 \\ v_2 \\ v_3 \end{pmatrix} \right|$$

For n atoms distance is  $\theta \sqrt{\sum_i v_2^{(i)} + v_3^{(i)}}^2$

but linear displacement is  $\left| \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} - \begin{pmatrix} v_1 \\ cv_2 - sv_3 \\ sv_2 + cv_3 \end{pmatrix} \right| = \sqrt{\sum_i [(1-c)v_2 - sv_3]^2 + [sv_2 + (1-c)v_3]^2}$

$$= \sqrt{\sum_i (1-2c+c^2+s^2)v_2^2 + (2cs-2s+2s-2sc)v_2v_3 + (1-2c+c^2+s^2)v_3^2}$$

$$= \sqrt{2(1-c)} \sqrt{\sum_i (v_2^2 + v_3^2)} \quad \text{so difference is } (\theta - \sqrt{2-2\cos\theta}) \left| \begin{pmatrix} 0 \\ v_2 \\ v_3 \end{pmatrix} \right|$$

For normalized displacement along rotation,  $\theta = \arccos \left( 1 - \frac{1}{2|r|} \right)$

for normalized distance  $\theta = \frac{1}{2|r|} \times$

$$\text{For given } \theta \text{ row of } B_x = \begin{pmatrix} v_1 \\ cv_2 - sv_3 \\ sv_2 + cv_3 \end{pmatrix}^T - \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}^T = \begin{pmatrix} 0 \\ (1-c)v_2 - sv_3 \\ sv_2 + (1-c)v_3 \end{pmatrix}^T$$

$$\text{If } \theta = \arccos\left(\frac{2|r|^2 - 1}{2|r|^2}\right) \quad \cos \theta = \frac{2|r|^2 - 1}{2|r|^2} \quad \sin \theta = \frac{\sqrt{2|r|^2 - 1}}{2|r|^2}$$

$$1 - c = \frac{1}{2|r|^2}$$

$$\frac{1}{2|r|^2} \quad \sqrt{4|r|^2 - 1}$$

$$\text{where } r_x = V - V \cdot X$$

Actually, for translations B row should be final position  $\times$

w/ com:  $B^x = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}^T + \alpha \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}^T \quad \cancel{\| \Delta \| = \alpha \sqrt{\text{frag-mations}}}$

$$\underline{\alpha = \frac{1}{\| \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \|} = \frac{1}{\sqrt{\text{frag-mations}}}}$$

12/10/14

Problem: translations become rotation looking because of mass weighting

Idea: make translations free of change in total com

Frag A + B = not A. Want to move A in x direction while B moves counter to preserve com w/ total displacement 1

$$\Delta = \alpha \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \Delta \text{COM} = \alpha \sum_i^A 1 \cdot m_i - \beta \sum_i^B 1 \cdot m_i = 0 \Rightarrow \beta = \alpha \frac{\sum_i^A m_i}{\sum_i^B m_i} = \alpha M_B^A$$

$$\text{Now for unit step } |\Delta|^2 = 1 = \alpha^2 \left( \sum_i^A 1 + M_B^A \sum_i^B 1 \right) \quad \alpha = \sqrt{\frac{1}{\text{frag A-mation} + M_B^A \text{frag B-mation}}}$$

Still rotating, what if we take mass weighted step...  $\checkmark$

$$\alpha \begin{pmatrix} m \\ 0 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

12/10/14

Am I doing this whole conversion wrong?  
 Converting to internals & back to Cartesians is lossy.  $\tilde{B}^{-1}\tilde{B} \neq I$     $B\tilde{B}^{-1} = I$   
 but maybe fine.

$$\square \square \square \\ U \Sigma V^T$$

New issue:

Removing nonzero e-vects for deloc coords

Either fix translations/rotations so no com movement or global rotation or project them out

$$P_{x_{com}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

$$P_{x_{com}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad P_{x_{com}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

For rotations:

$$|\Delta| = \sqrt{2(1-c)} \sqrt{\sum_i (v_2^i + v_3^i)^2} = \alpha$$

$$(1 - \cos \theta) = \frac{\alpha^2}{2\|r\|} \quad \sin \theta = \sqrt{\cancel{1} + \frac{\alpha^2}{\|r\|} - \frac{\alpha^4}{4\|r\|^2}} = \frac{\alpha}{2\|r\|} \sqrt{4\|r\| - \alpha^2} = \sqrt{\frac{4\alpha^2\|r\| - \alpha^4}{4\|r\|^2}}$$

This gets us step of norm  $\alpha$  that is a rotation. We then normalize this & apply it to step. Then scale it down to take step.

$$\boxed{\frac{\alpha}{\|r\|}}$$

So what do we pick for  $\alpha/\theta$

12/13/14

Wait no, let's just do this properly & make it the tangent...

$$P_x \vec{v}_{COM} \xrightarrow{\vec{r}} \vec{\alpha} = \vec{r} \times \hat{x} = \begin{pmatrix} 0 \\ v_y \\ v_z \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ v_z \\ -v_y \end{pmatrix}$$

Success!!!

New problem: Mode 4 (interfragment translation) is messed up.

Bad delocalized coordinates remove the correct translational mode.

Notice that  $U$  has non-zero e-vals since local translations + rotations don't preserve global COM + orientation.

Idea: preserve them by projecting out global rotations + translations

$$B = B - B \sum_{i=1}^6 g_i g_i^T \quad g_1 = \begin{pmatrix} m \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} / \| \cdot \| \quad g_2 = \begin{pmatrix} 0 \\ n \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} / \| \cdot \|$$

but first need to orthogonalize.

Yay! It worked!

Also, debugged reordered atoms (fixed set\_geom in frag.cpp)

1D PES plots now look good!

Set connectivity based on input rather than length

12/15/14

Internals look good & seem to be fixing the errors in AnZPEs for H<sub>2</sub>O & dimer!!!

New problem... 10x slower as program spends 94% of time in CoordUtil  
(54% doing pinv)

Need to speed this up.

Ideas:

- increase maxStep (large limit gives carts back)
  - ↳ numerical tests on dimer show 0.02 as limit before errors become significant
- make pinv more efficient
  - ↳ already have good guess...
- use noneqib starting points
  - ↳ think about how close we are to previous geom...
- minimize the # of dof we need to change?

Alg.: Move one mode in  $\sim \frac{P}{2}$  realizations, stay or return,

Aha! I've been stupid!

Just use previous carts as guess since only one mode is changing!

Not so simple to implement:

Could have starting carts & modes stored in CoordUtil which could be set by TestSystem which would need to collect carts afterwards & store P prevCarts

New normModeToCart

← in advance, set guessCart + guessModes

set carts to guessCart (rather than initCarts)

calculate Δ in modes (part - guess modes)

take step using Δ + add to carts

← calling code needs to be updated with carts ⚡

This solution doesn't take advantage of fact that only one mode at a time changes.

12/17/14

Looks like this gets us  $\approx 3 \times$  speed up w/ same large (0.02) max Step Size + w/o resetting reference. Ideally we'd want to decrease maxStepSize to 0.005 since we need to worry about drift now...

⚡ Idea: maybe move all modes at once but shorter steps.

This actually sounds really good since we can more easily take advantage of smaller steps but not really for moving a single step

→ need to move all modes (or select #) for given range of slices

1/6/15

Let's think clearly about implementing this..

Currently pick levyNum instantiations (of P) + update one mode in each of them.

With multiple modes need to consider whether modes updated in different instantiations or all the same

→ Since we need to update only changed instantiations we should pick levyNum of them first, then pick levyModes modes & move them for each instantiation

1/7/15

Looks like using guess is not going to work out as a speed up.

It takes half the steps but need 4x the tolerance to avoid drift

Still get speed up by reducing # of instantiations updated per step

↳ to really get a sense of the improvement should consider improvement in convergence but currently not calculating this properly



Trid delta AI + got junk! Turns out it was a problem w/  
cartsToNormalModes not using internals.

Need a way to convert current cartesian displacements to internal ones

For scaled geom jobs we can just use  $V_{HO}^{MM}$  since we scale it  
to be the same.

Generally, we're passed position in MM normal modes, so can't  
we just convert to QM normal modes directly?

$3N-6$  vectors  $U_{nm} = \boxed{\quad}_{nCart}^{nModes}$   $U_{nm}$  converts to carts &  $U_{am}$  converts to modes

Aha, the problem was that normalModeToCart was using internals  
to give the proper carts but then converting from those carts,  
cartsToNormalModes would give wonky results.

Solved by simply multiplying by  $U_{am} U_{nm}$  in V\_QChem

1/12/15

Fixing DAI:

Problem: thought I fixed code by making V-QChem convert to tinker Carts  
but this also uses tinker init carts so QHO + Q full are inconsistent...

What do we actually want?

- given tinker modes convert to displacements & apply to Q initial carts  
for calculation of Q full, then QHO from  $U_q^T U_q$  on tinker modes
- for scaleGeom, can scale modes & use QChem normModesToCarts?  
X then don't need to go back
  - ↳ if internals, cartesian displacements will be different

↳ Actually I don't think this is best.

- rescale modes so models have the same harmonic basis, but then still convert to tinker cartesian displacements (fixing for initial carts)  
& get  $E_{HO}^{QM}$  from  $U_q^T U_q$

↳ this makes more sense to me since the scaling is to correct the length scale problems in sampling that arise from inaccurate MM physics but we still want to match up cartesians properly.

Hmm... maybe I'm still confused on this ...

+ mode → cart → qmode already converts from pure mode to many (+ swaps)  
so why would we want to use scaledGeom?

1/14/15

Aha! I think I've figured it out!

Need to form  $U_Q^T U_T$  to determine how t modes transform into q modes.  $U_Q^T U_T = X$ . Scale mode i by  $\frac{\mu_i \omega_i^2}{\sum_j \alpha_{ij}^2 \mu_j \omega_{jq}^2}$

This is because moving in t modes moves in non-normal quades. It is still harmonic w/  $\bar{\mu} \bar{\omega}_i^2 = \sum_j (\alpha_{ij}^2 \mu_j \omega_{jq}^2)$

---

Implemented, but still not correct scaling factor.

Issue: Normal modes are not orthogonal!

Remember, these modes are e-vecs that have been mass weighted & normalized

This is the reason we needed  $D^+ = \text{pinv}(D)$  rather than  $D^T$

Maybe we can resolve this issue by projecting out translations & rotations from  $D^+$  which could have mixed some back in while still preserving

$$D^+ D = I$$

But wait...  $\alpha$  is the correct transformation the code makes from Tinker  $\rightarrow$  QChem so must be error in my math...

Want  $V_T(x, 0, 0, \dots, 0) = V_Q(\alpha_1 x, \dots, \alpha_n x) \Rightarrow \frac{1}{2} \bar{\mu}_i \bar{\omega}_i^2 x^2 = \frac{1}{2} \left( \sum_i \mu_i \omega_i^2 \alpha_i^2 \right) x^2$

$$\Rightarrow S_i = \sqrt{\frac{\bar{\mu}_i \bar{\omega}_i^2}{\sum_i \mu_i \alpha_i^2 \omega_i^2}}$$
 ✓ This is what I had before... check for correct  $\mu$ ...?

\* Yep! But was taking  $\alpha \cdot \omega$  vs.  $\omega \cdot \alpha^T$

1/16/15

Ok, we've scaled the harmonic potential... but only for moves in a single Tinker mode. Still off for multiple modes at a time

Maybe this is good enough ... consider  $s_{ij}$  instead of  $s_i$

$$\bar{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \rightarrow \underline{x} \leq \bar{x} \quad V_T(\bar{x}) = \frac{1}{2} \sum_i \bar{\mu}_i \bar{w}_i^2 \bar{x}_i^2 \quad \left( \sum_{jk} \alpha_{ij} s_{jk} \bar{x}_k \right) \left( \sum_{lm} \alpha_{il} s_{lm} \bar{x}_m \right)$$

$$V_A(x) = \frac{1}{2} \sum_i \mu_i w_i^2 \left( \sum_{jk} \alpha_{ij} s_{jk} \bar{x}_k \right)^2$$

$$\text{Solve for } s \Rightarrow V_T(\bar{x}) = V_A(x) \quad = \frac{1}{2} \sum_{ijkl} \underbrace{\mu_i w_i^2 \alpha_{ij} \alpha_{il}}_{M_{jl}} s_{jk} s_{lm} \bar{x}_k \bar{x}_m \\ = \sum_{jklm} \bar{x}_k s_{jk} M_{jl} s_{lm} \bar{x}_m$$

- - -

$$\text{or } D = \sum_i \left( \sqrt{\mu_i} \bar{w}_i \bar{x}_i \right)^2 - \left( \sum_{jk} \sqrt{\mu_i} w_i \alpha_{ij} s_{jk} \bar{x}_k \right)^2$$

$$= \sum_i \left( \sqrt{\mu_i} \bar{w}_i \bar{x}_i - \sum_{jk} \sqrt{\mu_i} w_i \alpha_{ij} s_{jk} \bar{x}_k \right) \left( \sqrt{\mu_i} \bar{w}_i \bar{x}_i + \sum_{jk} \sqrt{\mu_i} w_i \alpha_{ij} s_{jk} \bar{x}_k \right)$$

1/20/15

~~Shoot!~~ Just realized I was missing factor of 4 in DAI  
~~spreadsheet!~~ I forgot we need weight of 2 in Gaussian quad  
 for each dimension. Fixing now ~~now~~

Dumb, dumb, dumb. Weight of 2 only for interval  $[-1, 1]$

Changed back ...

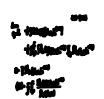
1/20/15

Should consider non MC estimation for  $\Delta A I$  sampling

$$\int \int \int d\sigma d\lambda \langle \Delta V^{QM} - \Delta V^{MM} \rangle_{\sigma \lambda} - \beta \langle SL \rangle_{\sigma \lambda} + \rho \langle S \rangle \langle L \rangle \approx \langle \Delta V^{QM} - \Delta V^{MM} \rangle_{x_1 x_2} - \beta \langle S_{x_1} L_{x_2} \rangle_{x_1 x_2} + \rho \langle S \rangle \langle L \rangle_{x_1 x_2}$$

$$\langle A(x_1, \dots, x_n) \rangle = \sum_{i=1}^n \langle A(x_i) \rangle \quad \text{since assuming } A, V, \Delta V \text{ separable}$$

confused about vector:



$$\begin{aligned} \langle A(x_n) \rangle &= \frac{\int \cdots \int \prod_{i=1}^n \prod_{t=1}^P dx_i^{(t)} A(x_n^{(t)}) P_{H_0}(x_i^{(t)}, x_i^{(t+1)}, \beta/\rho) e^{-\beta/\rho \Delta V(x_i^{(t)})}}{\int \cdots \int \prod_{i=1}^n \prod_{t=1}^P dx_i^{(t)} P_{H_0}(x_i^{(t)}, x_i^{(t+1)}, \beta/\rho) e^{-\beta/\rho \Delta V(x_i^{(t)})}} \\ &= \frac{\int \cdots \int \prod_{t=1}^P dx_n^{(t)} A(x_n^{(t)}) P_{H_0}(x_n^{(t)}, x_n^{(t+1)}) e^{-\beta/\rho \Delta V(x_n^{(t)})}}{\int \cdots \int \prod_{t=1}^P dx_n^{(t)} P_{H_0}(x_n^{(t)}, x_n^{(t+1)}) e^{-\beta/\rho \Delta V(x_n^{(t)})}} \end{aligned}$$

can't decouple these

$$\begin{aligned} Q = \text{Tr}(e^{\beta \hat{H}}) &= \int dx \langle x | e^{-\beta(H_0 + \Delta V)} | x \rangle \quad \frac{\int dx A(x) \langle x | e^{-\beta H} | x \rangle}{\int dx \langle x | e^{-\beta H} | x \rangle} \\ &= \sum_{\lambda} e^{-\beta E_{\lambda}} \quad \langle A(x) \rangle = \frac{\sum_{\lambda} \langle \lambda | A | \lambda \rangle e^{-\beta E_{\lambda}}}{\sum_{\lambda} e^{-\beta E_{\lambda}}} \end{aligned}$$

Ok, this page has been dumb...

Really, just solve for  $E$ -values +  $E$ -vectors (on grid) + use sum over states form of  $\langle \cdot \rangle$ .

Simplest is to just simply use grid on  $H$ .

I think discrete variable representation (DVR) is better since we can use hermite polynomials...

2/10/15

Have an inefficient Fourier Grid Hamiltonian coded up.

It doesn't matter since  $N_{\text{nodes}} \cdot N_{\text{grid}}^3$  irrelevant

Can calculate  $E_i + \psi_i(x_j)$   $\epsilon$ -values & vectors on grid

$$\begin{aligned} \text{Thus } \langle A(x) \rangle &= \frac{\text{Tr } A e^{-\beta H}}{\text{Tr } e^{-\beta H}} = \frac{\sum_i \langle i | A | i \rangle e^{-\beta E_i}}{\sum_i e^{-\beta E_i}} = \sum_{ij} \frac{\langle i | x_j \rangle \langle x_j | A | x_j \rangle \langle x_j | i \rangle}{Q} e^{-\beta E_i} \\ &= \sum_{ij} \frac{\psi_i^*(x_j) A(x_j) \psi_i(x_j)}{Q} e^{-\beta E_i} = \frac{\sum_i \tilde{A}_i e^{-\beta E_i}}{\sum_i e^{-\beta E_i}} \quad \text{where } \tilde{A}_i = \sum_j U_{ij} A_j U_{ji} \end{aligned}$$

For  $\langle A(x) \rangle_{\lambda_0}$  just calculate  $E_i + \psi_i(x_j)$  using  $H_{\lambda_0}$

$$H_{\gamma_2 \gamma_2} = \frac{1}{4} (H_{HO}^{am} + H_{HO}^{an} + H_{HO}^{mm} + H_{HO}^{nm})$$

To pick range of  $x_i$  to be heuristically reasonable want to properly describe first  $n$   $\epsilon$ -values where  $e^{-\beta(E_n - E_0)}$  is below some threshold.

$$\text{For } HO \quad e^{-\beta(E_n - E_0)} = e^{-\beta(n\hbar\omega)} < \delta \Rightarrow n > \frac{1}{\beta\hbar\omega} \log \frac{1}{\delta} \sim \frac{6}{\beta\hbar\omega}$$

Given  $\beta\hbar\omega$  can pick needed  $n$  & maybe use classical turning points for  $n$

$$\frac{1}{2} m v^2 x_{tp}^2 = \hbar\omega \left( n + \frac{1}{2} \right) \Rightarrow x_{tp} = \sqrt{\frac{2\hbar}{m\omega} \left( n + \frac{1}{2} \right)}$$

2/11/15

## Fourier grid algorithm for $\Delta AI$

- Calculate  $\omega_{QM}$ ,  $\omega_{MM}$ , +  $\bar{\omega} = \frac{1}{2}(\omega_{QM} + \omega_{MM})$  possibly w/ matched modes  
 $\bar{m} = \frac{1}{2}(m_{QM} + m_{MM})$

For each mode:

- Choose  $n_{max} = \frac{-\log \delta}{\beta \hbar \bar{\omega}} \sim \frac{6}{\beta \hbar \bar{\omega}}$  or min 3? +  $x_{max} = -x_{min} = \sqrt{\frac{2k}{n\bar{\omega}}(n + \frac{1}{2})}$
- Calculate  $V_{MM}^{(HO)}(x_i)$ ,  $V_{MM}(x_i)$ ,  $V_{QM}^{(HO)}(x_i)$  in PIMC code +  $V_{QM}(x_i)$  w/ Qchem

For each mode:

- Form  $T(x_i, x_j)$  from  $\bar{m}$  + grid
- Add  $\frac{1}{4}(V_{MM}^{(HO)} + V_{MM} + V_{QM}^{(HO)} + V_{QM})$  to diagonal
- calculate  $\epsilon_i + \psi_i(x_j)$
- Cumulate averages over grid points,  $\epsilon$ -values, + finally modes
  - $\tilde{A}_i = \sum_j \psi_i^*(x_j) A(x_j) \psi_i(x_j)$
  - $Q = \sum_i e^{-\beta \epsilon_i}$
  - $\langle A \rangle = \frac{1}{Q} \sum_i \tilde{A}_i e^{-\beta \epsilon_i}$

for  $V$ ,  $V^{(HO)}$  MM + QM  
and  $\frac{1}{2}(\Delta V_{QM} + \Delta V_{MM})(V_{QM}^0 + \frac{1}{2}\Delta V_{QM} - V_{MM}^0 - \frac{1}{2}\Delta V_{MM})$
- all averages  $\lambda = 5 = \frac{1}{2}$
- $\Delta AI = \langle \Delta V_{QM} - \Delta V_{MM} \rangle - \frac{1}{2} \langle (\Delta V_{QM} + \Delta V_{MM})(V_{QM}^{HO} + \frac{1}{2}\Delta V_{QM} - V_{MM}^{HO} - \frac{1}{2}\Delta V_{MM}) \rangle$   
 $+ \frac{1}{2} \langle (\Delta V_{QM} + \Delta V_{MM}) \rangle \langle (V_{QM}^{HO} + \frac{1}{2}\Delta V_{QM} - V_{MM}^{HO} - \frac{1}{2}\Delta V_{MM}) \rangle$   
 $= \bar{V}_{QM} - \bar{V}_{QM}^{HO} - \bar{V}_{MM} + \bar{V}_{MM}^{HO} - \beta \bar{S}L + \frac{\beta}{2} (\bar{V}_{QM} - \bar{V}_{QM}^{HO} + \bar{V}_{MM} - \bar{V}_{MM}^{HO}) \frac{1}{2} (\bar{V}_{QM} + \bar{V}_{QM}^{HO} - \bar{V}_{MM} - \bar{V}_{MM}^{HO})$

2/24/15

Things looking off for the grid approach.

First of all, had to modify v-QChem so that it uses same normal mode coordinates rather than converting Tinker to carts & back to QM modes

→ I just want independent 1D approximations to the PES for MM + QM

Note: distances are in internal coordinates, confirm correct displacements

3/3/15

Want to calculate variance correctly so calculate autocorrelation function + decorrelation time as discussed in Cao + Berne 1989

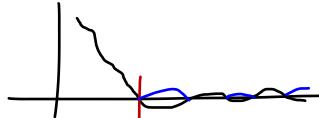
- There should be a better way to do this by taking the Fourier transform of the power spectral density (Wiener-Khinchin thm)
  - ↳ but for starters, just keep track of  $V$  (or maybe  $E$ ) over history + directly construct  $C(t)$

Created Est-AutoCorr obj. to keep past 100 values + calculate  $C(t)$

↳ Next get  $\tau$  by linear regression on  $t$  vs  $\log |C(t)|$

→ need to consider cut off for  $t$

↳ definitely if  $t < 0$  we can probably stop  
maybe not though...



how about  $N$  steps after threshold  $\epsilon$

Once we have  $\tau$  we can set the sampling + convergence steps to  $\lceil \tau \rceil$

Hmm... should probably keep closer to 1000 steps.

Also, should wait till after equilibration

$$5000 < N < 50,000$$

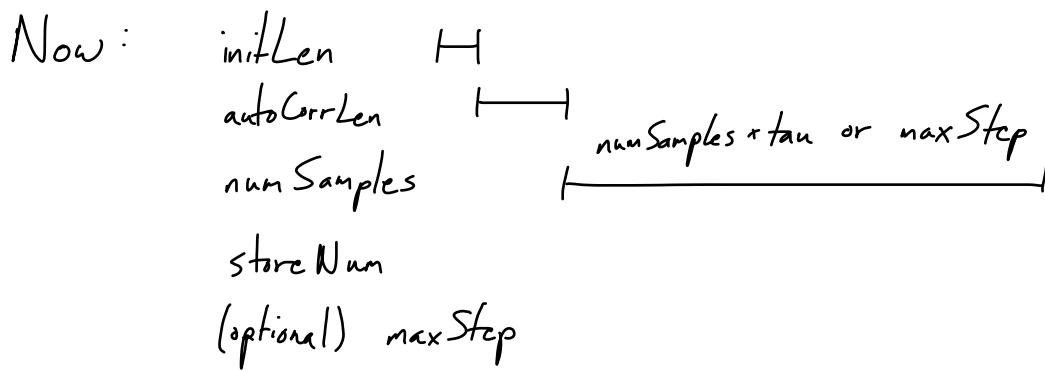
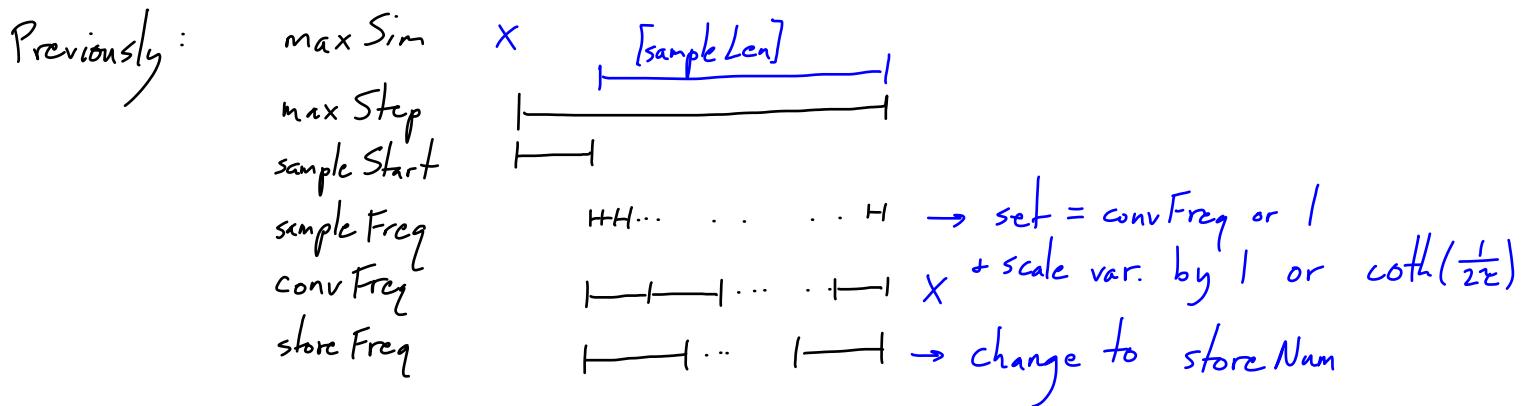
Need enough samples that decay actually starts to flatten out  $\sim 10,000$ ?

Actually more stable to sum  $\sum A(t)$  since  $\int_0^\infty e^{-\frac{t}{\tau}} dt = \tau$

Pick upper limit as  $\min(500, \text{first zero}, 5 \times \tau)$

3/10/15

## Reparameterize simulation run variables:



What about specifying error thresh:  $\text{num Samples} = \frac{\sigma^2}{\Delta^2} 2\tau$

If we want 5% error we estimate we need  $N_0 = \frac{\sigma_0^2}{(0.05 \cdot \mu_0)^2} 2\tau$

Note that we shouldn't run sim until it first reaches error thresh  
as this would bias our error

Standard Error of the Mean =  $\Delta = \frac{\sigma_{\text{error}}}{\sqrt{N}} \sqrt{2\tau}$  this is st.dev. of our estimate of  $\mu$

95% confidence interval is  $\pm 1.96 \Delta$  assuming  $N \geq 100$

Sample every step!

3/24/15

$$\begin{aligned}
 \sigma_{\bar{o}}^2 &= \langle \bar{o}^2 \rangle - \langle \bar{o} \rangle^2 = \frac{1}{N^2} \sum_{i,j}^N \langle o_i o_j \rangle - \langle o_i \rangle \langle o_j \rangle \quad \text{since } \bar{o} = \frac{1}{N} \sum_i o_i \\
 &= \frac{1}{N^2} \sum_i \langle o_i^2 \rangle - \langle o_i \rangle^2 + \frac{2}{N^2} \sum_{i < j} \langle o_i o_j \rangle - \langle o_i \rangle \langle o_j \rangle \\
 &= \frac{\bar{o}}{N} + \frac{2}{N^2} \sum_{i=0}^{N-1} \sum_{t=i+1}^{N-1} \langle o_i o_t \rangle - \langle o_i \rangle \langle o_t \rangle \\
 &= \frac{\bar{o}}{N} + \frac{2}{N} \sum_{t=1}^{N-1} \frac{N-t}{N} \left( \langle o_o o_t \rangle - \langle o_o \rangle \langle o_t \rangle \right) \\
 &= \frac{\bar{o}}{N} \left( 1 + \sum_{t=1}^{N-1} 2 \frac{N-t}{N} \frac{\langle o_o o_t \rangle - \langle o_o \rangle \langle o_t \rangle}{\langle o_o^2 \rangle - \langle o_o \rangle^2} \right) = \frac{\bar{o}^2}{N} (2\tau) \\
 \text{so } \tau &= \frac{1}{2} + \sum_{t=1}^{N-1} \frac{N-t}{N} A(t)
 \end{aligned}$$

$$95\% \text{ CI given by: } \pm 1.96 \sigma \sqrt{\frac{2\tau}{N}}$$

$$\text{To get window w/in x relative error: } \mu x = 1.96 \sigma \sqrt{\frac{2\tau}{N}} \text{ so } N = \frac{2\tau / (1.96 \sigma)^2}{\mu^2 x^2}$$

$$\text{If you need to multiply x by } \alpha \text{ then } N_{new} = \frac{1}{\alpha^2} N_{old} = \left( \frac{x_{old}}{x_{new}} \right)^2 N_{old}$$

3/30/15

## Compiling on MHG

- compiling armadillo
  - load mkl
- in `include/armadillo-bits/config.hpp` uncomment `USE LAPACK`, `USE BLAS`, and importantly `BLAS LONG LONG`
- Explicitly add mkl folder to `build-aux/cmake/Modules/ARMA_FindMKL.cmake`
- install in working directory
- link to local armadillo .so file in `$MHGSW`
- set correct flags for intel compilers: `-cxx11 -mof_main`
- deleted all program files in tinker

4/21/15

Can I use enhanced same path sampling to extrapolate w/ SHO + w/ TI?

For free particle Trotter:

Note: Trotter ignores that  $Q_{FP}^{[P]} \neq Q_{FP}$

$$Q^{[P]} = \int dx e^{-\epsilon V(x)} p_0(x_i, x_{i+1}, \epsilon) = Q_{FP}^{[P]} \frac{\int dx e^{-\epsilon V(x)} p_0(x_i, x_{i+1}, \frac{\epsilon}{P})}{\int dx p_0(x_i, x_{i+1}, \frac{\epsilon}{P})} = Q_{FP}^{[P]} \langle e^{-\epsilon V(x)} \rangle_{FP}$$

But  $Q^{[m]} \neq Q_{FP} \langle e^{-\frac{\epsilon}{m} \sum_i V(x_i)} \rangle_{FP}$  since sampling depends on P!

think about  $M=1 \dots$

$$\begin{aligned} Q^{[1]} &= \int dx_1 \cdots dx_p e^{-\beta V(x_1)} p_0(x_i, x_{i+1}, \beta_p) \\ &= \int dx_1 \cdots dx_p e^{-\beta_p \beta V(x_1)} p_0 \end{aligned}$$

I think these are converging to  $Q^{[P]}$  not  $Q^{[\infty]}$ ...

4/27/15

## He Clusters

Want ~100 63-129 clusters for Grace w/ 3.1-3.6 NW distances

At 4K! Should diss. around 9 K

This temp makes sampling difficult. So need to decrease Levy snip length

Problem is we need larger snip to get larger scale motions in a reasonable time scale. I'm trying to decrease # nodes sampled to allow for larger snips

Doesn't seem to matter much actually...

The biggest deal is just getting initial high temp sampling done w/ large levy num.

4/27/15

## Rovibrational Coupling

$$H = T + V_0 + \Delta V = H_0 + \Delta V$$

Consider  $3N-6$  internal coords  $\{x_i\}$  + 3 rotational coords  $\{\theta_i\}$

$V(\underline{x}, \underline{\theta}) = V(\underline{x}, 0)$  since  $V$  is invariant to  $\underline{\theta}$  but  $T$  is not

Can we consider a rotating harmonic oscillator?

Then  $H_0$  can be made to properly sample  $\underline{\theta}$  as well as  $\underline{x}$



To do rotations properly requires keeping track of winding numbers since they're periodic.

The idea is that a path can loop around  $2\pi$  and end up at same spot as far as  $V$  is concerned, but could be far apart in imaginary time path.

- Why can't you just use  $\theta \in [-\infty, \infty]$  + mod down to  $[-\frac{L}{2}, \frac{L}{2}]$  for  $V$  calc?

Also need to mod for any eval of  $x$ .

↳ In this case is winding # just biggest multiple of  $L$  between consecutive beads?

5/1/15

## General Thoughts on PIMC for Ah ZPE

Goal: Accurate low T anharmonic vibrational energy/ZPE for molecules

Basically, calculate correction to qchem HO approx.

- Assume our  $V(x)$  is good enough approx for sampling  $\langle \Delta V \rangle$
- Assumes sampling about a single local minima
- To theoretically not diverge, need to consider a cutoff for  $x$ , but not necessary in practice for reasonable  $P$ .
- Assume TI can be truncated relatively early (we use one point!)
- We ignore rotations in current code

Can we sample using multiple harmonic wells (possibly by switching during sampling)?

$$H = T + \frac{1}{2}V_0^{\circ}(x) + \frac{1}{2}V_0'(x) + \Delta V(x)$$

$V_0^{\circ}$    $V_0'$  

$$= \frac{1}{2}(H_0^{\circ} + H_0') + \Delta V$$

$$\langle x_0 | e^{-\epsilon H} | x_1 \rangle = \langle x_0 | e^{-\frac{\epsilon}{2}(H_0^{\circ} + H_0')} e^{-\epsilon \Delta V} + O((\epsilon [H_0, \Delta V])^3) | x_1 \rangle \approx \langle x_0 | e^{-\frac{\epsilon}{2}H_0^{\circ}} e^{-\epsilon \frac{1}{2}H_0'} | x_1 \rangle e^{-\epsilon \Delta V(x_1)}$$

Maybe not  $\frac{1}{2}(V_0^{\circ} + V_0')$  but some  $r_{12}(x)V_0^{\circ}(x) + (1-r_{12}(x))V_0'(x) \dots$

or exactly  ... How does Marcus theory handle this?

5/11/15

## Group Meeting Notes:

- Intra:
- Vibrational energy (VPT2, TOSH, VCI) all have problems
  - At low T, need quantum sampling due to ZPE  $\rightarrow$  PIMC

Theory:

- Idea: use what we know from HO approach to improve PIMC sampling

- Use  $H_0$  as propagator
- Further reduce error by sampling  $\Delta r$  directly using thermodynamic integration
- Pros: reduce P, reduce error through efficient sampling + more precise sampling

Con: requires HO approx, requires low cost E, assumes dominated by one well

---

## Ideas from group meeting:

- Shouldn't SHO in cartesians converge to right result w/ large P?
- Consider ways to use mixed propagator!

↳ Problem: Doesn't TI assume  $H_0 = H_{HO}$ ?

↳ does this also cause a problem for trying multiple wells?

5/15/15

Multiple propagators!

$q_i$  is the  $i^{th}$  normal mode

$$Q = \prod_i^N \int d\vec{q}_i \langle q_i | e^{-\beta \sum_j \frac{\hat{p}_j^2}{2m_j} + \hat{V}_j} | q_i \rangle = \prod_{i=1}^N \sum_k^P \int d\vec{q}_i^k \langle q_i^k | e^{-\beta \sum_j \frac{\hat{p}_j^2}{2m_j} + \hat{V}_j} | q_i^k \rangle$$

$$= \sum_k^P \int \prod_{i=1}^N d\vec{q}_i^k \langle q_i^k | e^{-\epsilon \hat{T}} | q_i^{k+1} \rangle e^{-\epsilon V(q_i^{k+1})} \prod_{i=1}^N \int d\vec{q}_i^k \langle q_i^k | e^{-\epsilon \hat{H}_0} | q_i^{k+1} \rangle e^{-\epsilon \Delta V(q_i^{k+1})}$$

So looks good! Just affects sampling so should still be able to use TI +  $\langle \Delta V \rangle$  to get ZPE

$$\sum_i \frac{\sqrt{\mu_i}}{\sqrt{m_j}} U_{ij} q_i = x_j \quad \sum_j \frac{\sqrt{m_j}}{\sqrt{\mu_i}} U_{ji}^+ x_j = q_i$$

$$\hat{T} = \sum_i \frac{\hat{p}_i^2}{2m_i} = \sum_j \frac{1}{2m_j} \frac{\partial^2}{\partial x_j^2} \quad \frac{\partial}{\partial x_j} = \sum_i \frac{\partial}{\partial q_i} \frac{\partial q_i}{\partial x_j} = \sum_i \frac{\partial}{\partial q_i} \frac{\sqrt{m_j}}{\sqrt{\mu_i}} U_{ji}^+$$

$$T = \sum_j \frac{1}{2m_j} \sum_{ik} \frac{\partial}{\partial q_i} \frac{\partial}{\partial q_k} \frac{\sqrt{m_j}}{\sqrt{\mu_i \mu_k}} U_{ji}^+ U_{jk}^+ = \sum_{ik} \frac{1}{2m_j} \frac{1}{2m_k} \frac{1}{2\sqrt{\mu_i \mu_k}} \sum_j \cancel{U_{ij} U_{jk}^+}^{\delta_{ik}}$$

$$= \sum_i \frac{1}{2\mu_i} \frac{\partial^2}{\partial q_i^2} \quad \star$$

So we can just work in normal modes w/ reduced masses for free particle propagator!

Now consider implementation...

Rho-Mixed should mostly be structured like Rho-HO

- ↳ For Check() need to correct to  $\Delta V$  for high modes +  $V$  for low modes
- ↳ For EstimateV(), which should be est.  $\Delta V$ , need to subtract  $V_{HO}$  for all modes
- ↳ Get Levy mean + st.dev. should be split by modes as well

Implement a get  $V$  + get  $\Delta V$  function I think, otherwise need to keep around a separate Rho-HO to pass for EstimateV

At this point do we want to consider multiple P values per mode? Nah.

More implementation details:

Currently normal modes to cart is taking ~90% of time w/  $\text{intervals}$   
seems to be in  $\text{pinv}$  to calculate  $B^+$

For cartesian coords, code overhead still seems to take up most of the time

↳ this could likely be improved by just always keeping values in `arma::Vecs`  
instead of using `std::vector` but would require large rewrite

↳ notably `Particle` would have to be removed since we need a single vector of all part positions

System could just keep position as  $P \times (\dim \cdot N)$  matrix with `CoordUtil`  
keeping track of dim, mass, for normal modes + cartesians

I guess we can assume we're working in normal modes from now on.

Check if `Part` had any other info but I think it got subsumed by `CoordUtil`

5/28/15

No pen :-

I'm taking steps stupidly! Should use full step, calculate error and then step out the error iteratively.

$$x_i = x_0 + B_0^+ \Delta q_0 \quad \Delta q_i = [q(x_i) - q(x_{i-1})] - \Delta q_{i-1}$$
$$x_i = x_{i-1} + B_i^+ \Delta q_0$$

Iterate till  $\Delta q_i < \text{thresh} \sim 10^{-6}$

According to Peng et al. 1996, we need to also keep track of periodicity since  $q(\angle = 0) - q(\angle = 359) = 1$

↳ might be taken care of already in delocalized internals code.

Second possibility, if  $B^+$  way more expensive than  $B$  formation, could just keep shooting guess steps using the same  $B^+$  (maybe also store pts for extrapolation)

5/29/15

## Implementation Summary:

Simulation - High level running + sampling. Depending on System, could run PIMC, PIMD, classical MD/MC, centroid PIMC ...

System - Specializes to PIMC level. Contains particles, environment info ( $V$ , CoordUtil), and info about physics approximations ( $\rho_{HO}$ , PhysicsUtil)

$V$  - Returns  $V$

↳ implemented for Morse, Potlib ( $H_2O_2$ ), Tinker      to do: QChem

$\rho_{HO}$  - This is our  $e^{-E/H_0}$

ModifyPotential returns  $-V_0$

GetLevyMean + Sigma allow sampling from the  $\rho$  distribution

↳ allows us to accept/reject using  $e^{-\Delta E}$  only (rather than incorporating  $\langle x_i | e^{-E/H_0} | x_{ini} \rangle$ )

↳ implemented as Rho-Free, Rho-HO, + Rho-Mixed

CoordUtil - store frequency info to transfer between normal modes/carts

handles internals using some bits of optking

can return  $V_{HO}$  for given point

PhysicsUtil - basically just a big dumb struct to keep parameters around