**Gromacs - Feature #1849**

**expanded ensemble -- Adaptive Integration Method**

11/03/2015 12:10 AM - Christopher Mirabzadeh

<table>
<thead>
<tr>
<th>Status:</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Christopher Mirabzadeh</td>
</tr>
<tr>
<td>Category:</td>
<td>analysis tools</td>
</tr>
<tr>
<td>Target version:</td>
<td>5.x</td>
</tr>
</tbody>
</table>

**Description**

I am attempting to include the Adaptive Integration Method (DOI: [http://dx.doi.org/10.1103/PhysRevE.69.056704](http://dx.doi.org/10.1103/PhysRevE.69.056704)) to the expanded ensemble functions. I have included the files as of version 5.0.4 that I have edited. I'm at the point where my code seems to be working. I would like some feedback on the internal terms that I've chosen to use based on the calculations needed to be made. See my questions below.

Edits I have made thus far:

- Names.c, Enums.h -- Added "aim" as an mdp option in the lmc move names.
- State.h -- added definitions to the df_history_t structure
- Expanded.c -- added AIMChooseNewLambda() method
- Typedefs.c -- edited this to init the arrays I created

These are the mdp options I have created that make aim selectable, along with expanded ensemble options:

- Lmc-move = aim
- Nstdhdl = 1
- Nstexpanded = 1

For my AIMChooseNewLambda() method I have borrowed heavily from the ChooseNewLambda() method available in expanded.c.

**Algorithm:**

1. Store the current potential energy -- dfhist->store_fepot[fep_state] = enerd->term[F_EPOT]
2. Randomly choose a direction +/- lambda -- using metropolis sampler as found in ChooseNewLambda() method
3. fep_state is the current/old configuration of the system
4. lamtrial is the new configuration of the system
5. Get the energy difference between fep_state and lamtrial de = U(lamtrial) - U(fep_state)
6. Trapezoidal rule df = integral from lamtrial to fep_state
   \[ df = 0.5 \times \frac{(dfhist->store_fepot[lamtrial] + dfhist->store_fepot[fep_state]) \times (lamtrial - fep_state)}{nlim} \]
7. If \( \exp(-beta \times (de - df)) \) is greater than random(0,1), then accept and update count
8. Update fep_state
9. Calculate running average
   \[ \text{delta} = \text{enerd->term}[F_DVDL] - dfhist->davg[fep_state]; \]
10. Update Free energy estimates
    \[ dfhist->davg[fep_state] += \text{delta} \times dfhist->aim_at_lam[fep_state]; \]

**Questions:**

- Am I using the correct term, enerd->term[F_POT], to store the potential energy of the system at fep_state?
- Is there another term that has the potential energy difference between lambda states?
Am I using the correct derivative, `ener->term[F_DVDL]`, for the derivative of the potential energy between lambda states? I have already asked this question before. I'm just looking for additional confirmation. What if someone chooses vdw-lambdas instead of fep-lambdas? Will the code need to use a different term for the derivative?

Thanks

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>expanded.c</td>
<td>50.9 KB</td>
<td>11/02/2015</td>
<td>Christopher Mirabzadeh</td>
</tr>
<tr>
<td>state.h</td>
<td>14.1 KB</td>
<td>11/02/2015</td>
<td>Christopher Mirabzadeh</td>
</tr>
<tr>
<td>names.c</td>
<td>8.08 KB</td>
<td>11/02/2015</td>
<td>Christopher Mirabzadeh</td>
</tr>
<tr>
<td>typedefs.c</td>
<td>26.6 KB</td>
<td>11/02/2015</td>
<td>Christopher Mirabzadeh</td>
</tr>
<tr>
<td>enums.h</td>
<td>13.7 KB</td>
<td>11/02/2015</td>
<td>Christopher Mirabzadeh</td>
</tr>
</tbody>
</table>