

1. Gibbs

```
Usage: gibbs [gibfile] [-O] -i gubin -o gibout
       -p prmtop -c inpcrd -r restrt
       -ref refc -x mdcrd -v mdvel -e mden
       -inf mdinfo -ms micstat
       -cm constmat -cs cnstscrt -a patnrg
```

-O Overwrite output files.

1.1. Introduction

This is a guide to *gibbs*, the AMBER module concerned with free energy calculations. This module of the AMBER suite of programs calculates the free energy difference, ΔG , between two states "0" and "1":

$$\Delta G = G_1 - G_0 \quad (1)$$

In LEaP, *State 1* is the default state and State 0 is defined by setting the perturbed atom parameters in the "Edit selected atoms" table in the xleap Unit Editor and using the saveAmberParmPert command to make the topology and coordinate files. (Note that this convention is "backwards" compared to that in *sander*.)

The free energy difference is calculated in a series of incremental steps which connect physical states 1 and 0 through a series of not-necessarily-physical intermediates. The character of the system at each of these intermediate steps is related to a parameter λ .

1.2. Free Energy Techniques Available in GIBBS

There are several techniques available in GIBBS/AMBER 4.0 for evaluating the free energy difference between two states, all based on various statistical mechanical relationships. These include:

- (1) Free Energy Perturbation (FEP) Window Growth: The free energy is calculated at discrete and uniformly spaced intervals of λ using the formulae:

$$G_{\lambda(i+1)} - G_{\lambda(i)} = -RT \ln \langle \exp[-(V_{\lambda(i+1)} - V_{\lambda(i)})/RT] \rangle_{\lambda(i)} \quad (2)$$

$$\Delta G = G_1 - G_0 = \sum_i G_{\lambda(i+1)} - G_{\lambda(i)} \quad (3)$$

where G_0 and G_1 are the free energies of states 0 and 1, respectively, $V_{\lambda(i)}$ is the potential energy function representative of state $\lambda(i)$, and $\langle \rangle_{\lambda(i)}$ means use the ensemble average of the enclosed quantity, representative of state $\lambda(i)$. The ensemble is evaluated from an MD trajectory run with $V = V_{\lambda(i)}$. The user specifies the numbers of equilibration (NSTPE or NSTMEQ) and data collection (NSTPA or NSTMUL) steps for each $\lambda(i) \rightarrow \lambda(i+1)$ "window".

- (2) Slow growth – the same as window growth, except lambda changes by a small amount at every step. Lambda changes slowly enough that it is assumed the system remains in equilibrium at every step (i.e. NSTPE=0, NSTPA=1). Thus the ensemble average in Equation (2) is replaced by its instantaneous value at each step.

- (3) Thermodynamic integration – instead of Equations (2) and (3), we use

$$G_1 - G_0 = \int_0^1 \langle \partial V / \partial \lambda \rangle_{\lambda} d\lambda \quad (4)$$

to calculate the free energy difference. In practice, the integral is approximated by a summation over discrete intervals in λ .

- (4) Dynamically Modified Windows – the equations of FEP (2 and 3) are used as described for method 1 above. But instead of using pre-chosen uniformly-spaced intervals of λ , the width ($\delta\lambda = \lambda(i+1) - \lambda(i)$) of each window is determined during the run, based on the recent value of the slope, $\partial G / \partial \lambda$, of the accumulated free energy versus λ curve. This allows the simulation to be run more "slowly" when the free energy is changing very quickly, and more "quickly" when it is not.
- (5) Dynamically Modified Thermodynamic Integration – Uses the same λ adjustment algorithm as for FEP (method 4), but the intervals in λ correspond to the points at which the integrand in Equation (4) is evaluated to approximate the integral.
- (6) Potential of Mean Force (PMF) Calculations – the user can elect to constrain any chosen set of internals (distances, angles, torsions) to a chosen lambda-dependent pathway. By selecting the appropriate option (NCORC=1), the contribution to the free energy from such constraints will be calculated. This constitutes a PMF calculation. PMF calculations can be carried out as part of either a FEP Window Growth or Dynamically Modified Windows run (1 and 3 above).

1.3. Understanding the Output

(a) *Window growth, slow growth, dynamically modified windows:* At specified intervals during the simulation, the energies calculated up to that point will be reported in the format:

```
Current Lambda = 0.850000
Last F.E. update: Lambda = 0.800000 Step = 4000 Method = F.E.P.
Accumulated "forward" quantities (Nonbond change)
  Lam+d_lam = 0.850000 F_energy = +0.64300
  ELEC = 0.000 NONB = +0.643 14NB = 0.000
  14EL = 0.000 BADH = 0.000
Accumulated "reverse" quantities (Nonbond change)
  Lam-d_lam = 0.750000 F_energy = -0.62130
  ELEC = 0.000 NONB = -0.621 14NB = 0.000
  14EL = 0.000 BADH = 0.000
```

When the free energies reported were last updated, the values of lambda and step number were as given on the second line. Note that the *current* values of λ and Step may be different, if the free energies have not yet been updated to reflect the ensemble now being generated. Also reported on the second line is the method being used to calculate free energy differences: F.E.P. is Free Energy Perturbation (standard or Dynamically Modified Windows); T.I. is Thermodynamic Integration (standard or Dynamically Modified Windows); Slow Growth is self explanatory.

Both "forward" and "reverse" accumulated free energies are reported. By default, GIBBS carries out "double-wide sampling", which means that at every value of λ we calculate the free energies both for going $\lambda \rightarrow \lambda + \delta\lambda$ and for going $\lambda \rightarrow \lambda - \delta\lambda$. The values "Lam+d_lam" and "Lam-d_lam" which are reported were the values at the last free energy update. If there were no sampling errors in our calculations, the independent sums of the "forward" and "reverse" values over the entire simulation would be the same, except for sign. Their actual difference gives us a *lower bound* on the error. By convention, the "forward" energy always corresponds to the energy for the process represented by λ increasing $0 \rightarrow 1$. Similarly, the "reverse" energy corresponds to the process represented by λ decreasing $1 \rightarrow 0$. *This is true regardless of the direction in which the actual simulation was run.*

Along with the total accumulated free energies in the "forward" and "reverse" directions, a component breakdown of the energies is given. Components listed include: ELEC (electrostatics, except 1-4's); NONB (non-bonds, except 1-4's); 14NB (1-4 nonbonds); 14EL (1-4 electrostatics) and BADH (bonds, valence angles and torsion angles). *Note that for Windows and Dynamically Modified Windows, these components are only estimates.* For slow growth and thermodynamic integration, they are exact.

If PMF calculations are performed, a sixth component will be listed, CORC. The procedure used to perform a PMF makes it difficult to separate contributions due to the constraints themselves from those due to non-bonded/electrostatic interactions. For this reason, in these cases CORC will reflect the sum total of all three types of contributions and the individual non-bonded/electrostatic contributions will be reported as 0's.

(b) *Thermodynamic integration:* The output is similar to that described above, except that, because of the integral which must be evaluated in thermodynamic integration (TI) (Equation 4), double-wide sampling is not possible. Thus, only a "forward" set of energies is reported. Again, by convention, these value have the sign appropriate for the $0 \rightarrow 1$ conversion, regardless of the direction in which the simulation was actually run.

If the calculation of individual entropy/enthalpy contributions is requested, these will also be included in the output, following the same forward/reverse conventions as above.

1.4. Defining States and Obtaining Appropriate Starting Coordinates

The default state from which to start the perturbation is usually $\lambda=1$. However, you can equilibrate at either $\lambda=1$ or $\lambda=0$ (or any arbitrary value of λ) as follows:

- Set ISLDYN (line 14) to +-2 or +-3;
- Set NRUN (line 5) to 1;
- Set NSTLIM (line 8) to the number of steps of equilibration desired;
- Set ALMDA (line 14) to the value of λ at which equilibration is to take place;
- And set NSTMEQ (line 14) to any value greater than NSTLIM.

The program is capable of handling periodic boundary conditions with the solute in a solvent bath either with constant volume or constant pressure. Additionally, it is possible to decouple the free energy into electrostatic and van der Waals contributions, if desired.

1.5. Suggested introductory references

The papers listed here emphasize the theory and experience with the AMBER programs; general reviews are available that cover many other aspects of the field [1-4]. The basic operation

and capabilities of *gibbs* are described in a series of papers by Dave Pearlman, Peter Kollman, and their co-workers [5-8]. Because it has so many options, the *gibbs* program has been used a lot for comparative studies of free energy methods [9-12]. Some recommendations for free energy calculations, with specific reference to the options in *gibbs*, are given at the end of this chapter. Thermodynamic integration calculations can also be carried out with *sander* (see that chapter).

1.6. Assigning files

GIBBS incorporates a file assignment protocol which is easy to use, and which will work on all computers. In addition, file assignments can optionally be specified using flags on the command line.

```
gibbs [gibfile] [-O] [-i PIN] [-p PPARM] [-c PINCRD]
      [-o POUT] [-r PREST] [-inf PINFO] [-ms MICSTAT]
      [-cm CONSTMAT] [-cs CNSTSCRT] [-a PATNRG]
      [-x PCOORD] [-v PVEL] [-e PEN] [-ref PREFC]
```

where PIN, PPARM, etc. are replaced by the appropriate filenames to be assigned. The meanings of the various files are given below.

If "gibfile" is present, it must be the first option given, and this file will be read to make the file assignments. In this case, any remaining flags are ignored. Otherwise, all assignments are made using command-line flags. Any flags not specified default to the given name (e.g. if -o is not specified, output would be in file POUT).

GIBBS I/O FILE ASSIGNMENTS

file	unit	purpose
INPUT:		
PIN	5	Control data for the run (described below).
PPARM	8	Topology file (created by LEaP)
PINCRD	9	Initial positions and (optionally) velocities.
PREFC	10	Reference coordinates for optional position restraints (only if NTR = 1)
OUTPUT:		
POUT	6	Formatted results and diagnostics
PREST	16	Restart coordinates and velocities. For restarts, this file should be assigned to PINCRD.

PINFO	7	Short file containing a summary of current energies. For monitoring runs which are executing.
MICSTAT	27	A concise summary of important energy information for each window/interval.
CONSTMAT	28	Contains data related to the matrix of free energy data generated. Only used when IPER>0 for one or more of the constraints/restraints defined with INTR > 0 (see line 13).
CNSTSCRT	42	Contains data required when generating a matrix of free energies corresponding to two independent sets of constraints (IPER>0 and INTR>0; see line 13).
PCOORD	12	Archived coordinate sets (if NTWX > 0)
PVEL	13	Archived velocity sets (if NTWV > 0)
PEN	15	Archived energy related data (if NTWE > 0)

1.7. Control parameters

The title (line 1) must be the first line in PIN. All remaining standard flags are entered in the namelist *&cntrl*.

TIMLIM	Time limit for the job (in seconds). Default = 999999.
IREST	Flag to restart the run.
	= 0 Normal start (default)
	= 1 Job to be restarted. The accumulated free energies, current value of lambda, and other required quantities are read from the end of the input coordinate file (PINCRD). This file should be the PREST file written by the simulation being restarted.
IBELLY	Flag for belly type dynamics.
	= 0 No belly run (allow all atoms to move; default).
	= 1 Belly run. The subgroups of atoms which are allowed to move are read as groups from file PIN. See the section on GROUP in the Appendices.

ICHDNA Option to modify the charge of end hydrogens during in vacuo simulations. Without this option, molecular dynamics calculations on nucleotides will result in bonding between the 5' and 3' hydrogens and the corresponding phosphate groups.

- = 0 no charge modification (default)
- = 1 modify charge

IPOL for inclusion of polarizabilities in the force field.

- = 0 non polar calc (no polarizabilities read from "prmtop"; default).
- = 1 turn on polarization calculation.

I3BOD For 3-body terms with a polarization calc.

- = 0 No 3-body terms to be defined. Default.
- = 1 Read and use 3-body interaction definitions (see card 18). 3-Body terms only have an effect when polarization is turned on (IPOL=1).

NTX Option to read the initial coordinates and velocities.

Options 1-3 are used when no set of starting velocities is available (e.g. when starting from a set of minimized coordinates). Options 4-5 are used when: 1) a starting set of velocities is available (e.g. after MD equilibration or on an MD RESTART); and 2) The coordinates/velocities were generated with MD run either without periodic boundary conditions, or with constant VOLUME periodic boundary conditions. (Box dimensions, if any, are taken from the PARM file). Options 6,7 are used when both a starting set of velocities are available and the coordinates/velocities were generated with MD run using constant PRESSURE periodic boundary conditions. Note: box dimensions only appear in coordinate files written (as PREST) after simulations using periodic boundary conditions (constant volume or constant pressure).

- = 1 X is read; no velocity information read (Amber format); default
- = 2 X is read; no velocity information read (unformatted)
- = 4 X and V are read (unformatted)
- = 5 X and V are read (Amber format)
- = 6 X, V and BOX are read (unformatted)
- = 7 X, V and BOX are read (formatted)

NTXO Option to write the final coordinates and velocities.

- = 0 X, V and BOX are written to file 'PREST' (unformatted)
- = 1 X, V and BOX are written to file 'PREST' (Amber format); default.
- IG The seed for the random number generator. The MD starting velocity is dependent on the random number generator seed. The generator works most effectively when the seed is large and an odd or a prime number (e.g. 71277, the default).
- TEMPI Initial temperature, default is 0.0. If $TEMPI > 1.0e-06$, the velocities are taken from a maxwellian distribution with TEMPI (K). Choosing a low initial temperature (e.g. 10K) allows the calculation to reach the equilibrium conditions with the residual forces in the system during the initial steps. TEMPI is ignored if $NTX > 3$.
- HEAT If $ABS(HEAT) \geq 1.0E-06$, all the velocities are multiplied by HEAT. Default is 0.0.
- NTB Flag for periodic boundary conditions. If $NTB \text{ .EQ. } 0$ then the boundary conditions are NOT applied. The periodic box may be rectangular or monoclinic depending on the value of BETA.
- = 0 no periodicity is applied; default.
- = 1 constant volume
- = 2 constant pressure.
- IFTRES Flag to remove the nonbonded cutoff from the solute.
- = 0 ALL solute - solute nonbonded interactions are calculated, and the boundary conditions are not applied to the solute. For simulations of highly charged solutes in a water bath, it can be useful to calculate ALL solute - solute nonbonded interactions in order to reduce electrostatic problems. Note that this option is intended for small solutes, and will generate many more nonbonded pairs than the normal method if the solute is large. This option is useful for DNA and counterions. Note: if counterions are added in edit, then they are considered part of the solute.
- = 1 Nonbondeds are evaluated normally; default.
- Note: IFTRES will only have an effect when periodic boundary conditions are employed ($NTB > 0$). When $NTB=0$, $IFTRES=1$ behavior (normal nonbond generation) always occurs.
- BOXX(1..3) Lengths of the edges of the periodic box. If $IBXRD > 0$, then the values specified here will be used. Otherwise, the values specified here are ignored and the values in the PARM output file (if $NTX < 7$) or the values in PINCRD (if $NTX \geq 7$) will be used.
- BETA Angle between the x- and z- axes of the box in degrees. The y- axis is assumed to be orthogonal to the other axes. ($0 < BETA < 180$). The information given for BOX(1..3) above applies to BETA as well. Non-orthogonal systems do not currently work correctly. Therefore, if $IBXRD > 1$, BETA must be set to 90.0, which is the default.
- IBXRD If $IBXRD > 0$, then the values of BOX(1..3) and BETA specified here will be used. Otherwise, the values in the PPARM or PINCRD file will be used (see

above).

- NRUN Number of MD-runs of NSTLIM steps to be performed; default is 1. Since the restart coordinates are written only at the end of each run, it is sometimes desirable to break a long run into a series of shorter steps. If NRUN is set > 1 , one should ensure that the number of equilibration+data_collection steps (if performing windows/TI) divides evenly into NSTLIM (line 8). The number of picoseconds of molecular dynamics is equal to the product of NRUN X NSTLIM X DT.
- NTT Switch for temperature scaling. Note that several of the temperature coupling options available here are new to version 4 of GIBBS. Several of these are rather ad-hoc, and may not result in a thermodynamically relevant ensemble. (They may be useful when using MD strictly to sample conformational space). For free energy calculations, it is recommended you stick with NTT = 0 (constant energy), NTT = 1 (constant temperature) or NTT = 5 (constant temperature, separate solute/solvent temperature scaling).
- < 0 Re-assign random velocities whenever the current temperature deviates by more than DTEMP from DTEMP0 (target temperature), and every ABS(NTT) steps. Velocities are assigned in a Maxwellian distribution. By default, velocities are reset for all atoms. If NSEL > 0 (see below), NSEL atoms are selected at random each time a velocity reassignment is to take place, and only those atoms have their velocities reassigned. (Be sure to set DTEMP0 to a very large value if you wish to disable its action with this option).
- Note that the procedure which assigns velocities makes the assignments as if all particles possessed three independent degrees of translational freedom. If SHAKE is used, this will not strictly be the case, and the effective temperature immediately after velocity assignment will be higher than the target temperature. As velocity contributions along the constrained directions are dissipated, the temperature will rapidly adjust towards the target.
- = 0 Classical dynamics. Never rescale/reassign velocities after the start. [The total energy (kinetic + potential) is conserved; same as in older versions of GIBBS.]
- = 1 Constant temperature, using the Berendsen coupling algorithm. A single scaling factor for velocities is used (same as in older versions of GIBBS). This is the default.
- = 2 Constant temperature, using the Berendsen coupling algorithm. But only consider the solute temperature in determining the velocity scaling on each step. Could result in solvent atoms having very high temperature, and not generally recommended.
- = 3 Constant temperature, using Berendsen algorithm. But only rescale when temperature deviates from TEMP0 by more than DTEMP. Single scaling factor.
- = 4 When temperature deviates from TEMP0 by more than DTEMP, do one quick scale of the velocities to bring them back to TEMP0. Otherwise, do not scale.

- = 5 Constant temperature, using the Berendsen coupling algorithm, and with separate solute/solvent velocity scaling factors. This option is recommended as a replacement for NTT=1, and can help alleviate the "cold solute/hot solvent" problem.
- TEMP0 Reference temperature at which the system is to be kept if NTT not = 0. Default is 298.
- DTEMP The deviation allowed in the constant temperature MD-runs (read but ignored if NTT=0,1,2 or 5). Default is 10.
- TAUTP Temperature relaxation time when NTT .gt. 0. This is a damping factor which prevents abrupt changes in the system, if the temperature exceed specified deviations. Generally, values for TAUTP should be in the range of 0.1-0.4. Smaller values of TAUTP result in "tighter" coupling. Default is 0.1.
- TAUTS If NTT=5, then TAUTP is the temperature relaxation time for the solute, while TAUTS is the relaxation time for the solvent. If is specified as 0.0, TAUTS is set equal to TAUTP. Generally, TAUTS should be in the range of 0.1-0.4, with smaller values resulting in "tighter" coupling. If NTT.NE.5, TAUTS is read but ignored. Default is 0.1.
- ISOLVP Only used if NTT = 2 or 5 (sep. solute/solvent temp coupling)
- = 0 default solvent atom pointer is used. If periodic boundary conditions are being used, this is the last solute atom. Otherwise, it will be the last atom of the system (which results in no separate solute/solvent coupling). Note that counterions are by default considered part of the `_solute_`.
- > 0 Gives the number of the last atom to be considered part of the "solute". ISOLVP should generally be specified if NTT = 5 and NTB = 0. ISOLVP only affects temperature scaling.
- NSEL Only used if NTT < 0 (random velocity reassignments)
- = 0 When velocity reassignment takes place, velocities for all atoms are reassigned (default).
- > 0 When velocity reassignment takes place, NSEL atoms are randomly selected, and only the velocities for those atoms are reassigned.
- DTUSE The value of `d_TEMP` used in approximating the temperature derivatives by finite differences. DTUSE is only used when individual enthalpy/entropy values are being calculated (ISANDE = 1, line 12). DTUSE should generally be <= 1.0 (larger values often cause floating overflows/ underflows). Default is 1.0.
- NTP Flag for constant pressure dynamics. This option MUST be set to 1 or 2 when the MD calculation is done with constant pressure periodic boundary conditions (NTB=2, line 4).
- = 0 Classical dynamics without any Pressure Monitoring (default)
- = 1 MD with isotropic position scaling
- = 2 MD with anisotropic diagonal (x-,y-,z-) position scaling
- NPSCAL Flag for the type of scaling in case of constant pressure run.

- = 0 Uniform coordinate Scaling (default)
- = 1 Sub molecules Center of mass Scaling
- PRES0 Reference pressure at which the system is maintained (when NTP > 0) in units of bars, where 1 bar ~ 1 atm. Default = 1.0.
- COMP Inverse compressibility of the system when NTP > 0. The unit is in 1.0E-06/bar (the default value of 44.6 is recommended).
- TAUP Pressure relaxation time when NTP .gt. 0 The recommended value is between 0.1 and 1.0 ps. Default is 0.4.
- NDFMIN Number of degrees of freedom that will be subtracted from the total number of degrees of freedom to account for center of mass removal, belly runs, etc. (This will be a value between 0 and 6). By default (if NDFMIN.GE.0), this value will be set automatically. For nearly all simulations, you should accept the default calculated when NDFMIN = 0. If you set NDFMIN<0, then ABS(NDFMIN) additional degrees of freedom will be subtracted *in addition to* the number calculated automatically. This option is provided so that you can account for systems containing extended linear moieties that reduce the true number of degrees of freedom from that which would be calculated by a simple 3N-6 determination. For example, if you used a linear triatomic molecule for your solvent, you would need to set NDFMIN = -(number of solvent molecules).
- NTCM Flag for the removal of translational and rotational motion from the initial velocities. NOTE: this flag is automatically set to 0 if belly option is used.
- = 0 The translational and rotational motion about the center of mass is not removed (default)
- = 1 The above motion is removed and NTCM is reset to 0. If velocities are being periodically reassigned according to a Boltzmann distribution (NTT < 0) and NTCM = 1, then center of mass motion will be removed after each reassignment.
- NSCM After NSCM steps the above motion will be removed again if NTB .EQ. 0. This flag should be set to -1 if the belly option is used. This results in NSCM .EQ. 90 000 000 steps. Default is -1.
- ISVAT Residue-based periodic imaging flag ISVAT is ignored when periodic boundary conditions are not used.
- = 1 Residue-based periodic boundary conditions are used (default). For each residue, imaging is determined based on the position of the atom in the residue which is closest to the residue's initial center of mass. Both solute and solvent atoms are imaged on a residue basis. Each atom of any solute or solvent residue "sees" the same image of any interacting residue.
- = 2 Same as 1, except that for each atom of the `_solute_`, different whole-residue images on interacting residues may be used. Can be useful when a solute residue is fairly long in one or more dimensions. The code required to implement ISVAT=2 does not vectorize, and may result in a substantial hit to performance on vector machines. For this reason, ISVAT=1 should be used except where ISVAT=2 is clearly

- required.
- = 3 No residue-based periodic imaging. Separate imaging is done for each atom-atom pair. This is the way imaging was done in versions ≤ 3 of GIBBS (and MD). In typical operation, you would NOT want to use this option. Setting ISVAT <3 allows a cutoff of as large as $\sim 1/2$ the smallest box dimension to be used. When ISVAT=3 with periodic boundary conditions, a much smaller cutoff/box ratio must be used.
- NSTLIM
- > 0 Number of MD-steps per run to be performed. NRUN such runs will be carried out.
- = -1 Continue simulation until done, or until TIMLIN is exceeded. This option is often used with dynamically modified procedures (since we don't know at the outset how many total steps will be required). This is the default.
- INIT Flag for different starting procedures. If option NTX is less than 5, INIT should be equal to 3. If option NTX is greater than or equal to 5, this option should be equal to 4.
- = 3 V(T-DT/2) is obtained by calculating force(T); default.
- = 4 Input V(T-DT/2) is used for the starting velocity
- T The time at the start (psec). Only for your own use. Not important for the simulation. Default is 0.0.
- DT The time step (psec); default is 0.001. (Note that in the special case where window growth is requested by using the unrecommended flag combination (IFTIME = 0 and ISLDYN = 0; line 14), DT is replaced by the value of DTA on line 15).
- VLIMIT Limiting velocity; default is 0.0. If .ne. 0.0, then any component of the velocity that is greater than abs(VLIMIT) will be reduced to VLIMIT (preserving the sign), and a warning message will be printed. This can be used to avoid occasional instabilities in molecular dynamics runs. VLIMIT should generally be set (if at all) to a value like 20., which is well above the most probable velocity in a Maxwell-Boltzmann distribution at room temperature. Note that although it is anticipated that use of a liberal (large) value of vlimit should not adversely affect the statistics accumulated during a free energy simulation, this has not yet been definitively demonstrated.
- IVEMAX Maximum times VLIMIT may be exceeded. If IVEMAX >0, then IVEMAX specifies the number of times the limiting velocity VLIMIT can be exceeded in a simulation. If VLIMIT is exceeded \geq IVEMAX times, the simulation will stop. If IVEMAX =0, there is no limit on the number of times VLIMIT can be exceeded. Default is 0.
- NTC Flag for SHAKE to perform bond length constraints. Constraining the bond lengths removes the highest frequency motions from the system and usually allows somewhat larger timesteps to be used.
- = 1 SHAKE is not performed (default)
- = 2 bonds involving hydrogen are constrained. No bonds which are part of the pert group are constrained.

	= 3	all bonds are constrained
TOL		Relative geometrical tolerance for bond constraints in SHAKE. Smaller values give tighter tolerances. The (default) recommended value is ≤ 0.0005 Angstrom
TOLR2		Relative geometrical tolerance for angle and torsion constraints (radians). Smaller values give tighter tolerances. The (default) recommended value is ≤ 0.0001 rad.
NCORC		Constraint energy flag.
	= 0	No constraint contributions to the free energy are calculated (default).
	= 1	The contributions to the free energy from any constraint whose equilibrium value changes with lambda will be calculated. This includes: A) Any constrained internals defined at the end of the input (see flag INTR, line 13); and B) any SHAKE-en bonds (see NTC). If NCORC=1 is specified, the program will determine which atoms of the system have positions which are dependent on the constraints, and all of these will effectively be included in the "perturbed group". This forces some time-consuming calculations. If no constraints are changing with lambda, be sure to set NCORC=0. The procedure used to perform a PMF makes it difficult to separate contributions due to the constraints themselves from those due to non-bonded/electrostatic interactions. For this reason, in these cases CORC will reflect the sum total of all three types of contributions and the individual non-bonded/ electrostatic contributions will be reported as 0's. Note: If you are using a "belly" with NCORC=1, you must ensure that all residues of the pert group are part of the moving belly, and that, additionally, any residues sharing constrained bonds with the pert group (if any) are part of the moving belly.
ISHKFL		Flag which determines what the program will do in the event of a SHAKE/internal constraint failure.
	= 0	Program halts immediately. This is what the old versions of Amber did.
	= 1	Program will write a restart file containing the coordinates before the failed call to the constraint routine (+ velocities, if applicable). The program will then halt.
	> 1	The coordinates will not be constrained on any iteration for which the constraint routine fails. If constraint failure occurs on more than ISHKFL-1 contiguous steps, the program will stop as described for ISHKFL=1. This is the default
ITIMTH		Defines which method should be used to calculate constraint free energy contributions when NCORC=1 and the Thermodynamic Integration method (IDIFRG=1) approach is being used.
	= 0	Use the Potential Forces (PF) method (default).

- = 1 Use the Constraint Forces (CF) method.
- =-1 Use the PF method, override program warnings about constraints within closed rings.

Two methods for determining the constraint free energy contributions during TI have been derived in the literature. The PF method appears to be more efficient, and so is the default. However, PF method cannot be used when any constraints of the system which are changing with lambda (and hence contribute to the free energy) are part of a closed ring. In this case, the CF method must be used. The program will flag any constraints of the perturbed group which are part of a closed ring, and will stop with a warning if TI is used with PF in such a case. If none of these constrained bonds change with lambda, you can still use the PF method, but must specify ITIMTH=-1 here to ensure you have considered whether this will be appropriate. It is suggested you NOT set ITIMTH=-1 automatically, but only after ensuring that it will be appropriate.

JFASTW

Fast water definition flag. By default, the system is searched for TIP3P waters, and special fast routines are used for these molecules. There are two types of fast routines specific to TIP3P water: 1) A faster, analytic SHAKE algorithm for 3-point water; 2) A faster routine to calculate non-bonded TIP3P-TIP3P water interactions.

In normal operation, the program defaults will be acceptable. However, in rare instances (e.g. for debugging purposes, or when the user has redefined the definition of a TIP3P water), one may wish to inhibit the use of these fast routines and/or redefine the default definition used in Amber to define TIP3P waters. This option makes this possible.

- = 0 Normal (default) operation. The default AMBER definition of TIP3P water is used, and the fast water routines are used where appropriate.
- = 1 Use the fast routines for water SHAKE and non-bonds, but redefine the names the program uses to recognize TIP3P waters. The redefinition names are provided below.
- = 2 Use the fast water routine for SHAKE. Do not use the fast water routine for non-bonds.
- = 3 Use the fast water routine for SHAKE. Do not use the fast water routine for non-bonds. Redefine the names the program uses to recognize TIP3P waters. The redefinition names are provided below (line 17).
- = 4 Do not use fast water routines for either SHAKE or non-bonds.

NTF

Flag for force evaluation. Typically set to the same value as NTC.

- = 1 complete interaction is calculated (default)
- = 2 bond interactions involving H-atoms omitted, except bonds in the perturbed group (use with NTC = 2, see above SHAKE options)
- = 3 all the bond interactions are omitted (use with NTC = 3)
- = 4 angle involving H-atoms and all bonds are omitted

- = 5 all bond and angle interactions are omitted
 - = 6 dihedrals involving H-atoms and all bonds and all angle interactions are omitted
 - = 7 all bond, angle and dihedral interactions are omitted
 - = 8 all bond, angle, dihedral and non-bonded interactions are omitted
- NTID Flag for solvent pairlist behavior.
- = 0 only the first atom of each solvent molecule is used when generating the non-bonded pairlist for a periodic system (for water, this is the oxygen). If this atom lies within the specified cutoff, the entire solvent molecule is included in the non-bonded pairlist. This can result in a substantial speedup in non-bonded pairlist generation, and is recommended when using water as the solvent. This is the default.
 - =86 all atoms in a solvent molecule are considered when generating the non-bonded pairlist for a periodic system. If any atom of the solvent molecule lies within the specified cutoff, all atoms of the solvent molecule will be included in the non-bonded list. This is the behavior of versions of AMBER ≤ 3.0 .

A value of NTID=0 is suggested for calculations using water as a solvent. For calculations using larger solvent molecules, one should carefully consider whether using only the first atom is appropriate. Regardless of the value of NTID, all atoms of the *solute* are considered when deciding whether to include a second residue in the interacting non-bonded list for the solute residue. NTID will have no affect for non-periodic systems.
- NTNB Flag for non-bonded pair list generation.
- = 0 no pair list will be generated (unlikely you would choose this).
 - = 1 pair list will be generated (default)
- NSNB After NSNB steps the non-bonded pair list will be updated. Default = 50.
- IDIEL Type of dielectric function to be used.
- = 0 distance dependent dielectric function (for in vacuo simulations of "aqueous" systems).
 - = 1 constant dielectric function (always use with explicit solvent, e.g. water); default.
- IELPER Flag to control the "electrostatic decoupling" of the perturbation energy
- = 0 Regular run; no electrostatic decoupling (default).
 - = 1 Only the electrostatic contribution to the free energy is calculated keeping the geometry and the VDW parameters pertaining to LAMBDA = 1.
 - =-1 Only the non-electrostatic (VDW, etc.) contributions to the free energy are calculated and the system changes from that characteristic of LAMBDA = 1 to 0 (or from that characteristic of LAMBDA = 0 to LAMBDA = 1 depending on the signs of IFTIME or ALMDEL).

In electrostatic decoupling, two runs have to be performed, one for electrostatic and the other for VDW etc. contributions. This is useful when a polar or charged group is being established or removed. However, the LAMBDA = 1 state must pertain to the established group, and the LAMBDA = 0 to the removal of the group. The decoupling MUST go through the following perturbation cycle: electrostatic LAMBDA = 1 \rightarrow 0 with LAMBDA(vdw) = 1, followed by van der Waals LAMBDA = 1 \rightarrow 0. If the simulation is started at LAMBDA = 0, then reverse the above procedure. In this way, charges never appear on atoms which do not possess a vdw radius which avoids very close contacts due to charge-charge attractions.

Notes: (1) Two separate runs are needed to fully carry out the decoupling calculation. (2) In the IELPER=+1 phase, any added restraints/constraints (if INTR > 0) will be fixed at the values they have when lambda=1. (They will still only be applied, however, over the ranges specified). (3) The free energy contribution from internal constraints is never calculated during the IELPER=+1 phase (it is calculated during the IELPER=-1 phase).

To summarize:

IELPER	internals/vdw	electrostatics
+1	fixed @ lambda=1 (non-pert) values	vary
-1	vary	fixed @ lambda=0 (pert) values

IMGSLT

Flag to control the Solute-Solvent interaction in the case of PB simulation

- = 0 The Boundary condition is applied to solute-solvent interactions (default)
- = 1 No Solute-Solvent imaging. Solute does not see image solvent. This assumes that the solute is centered in the periodic system, and is not free to migrate. Do not use this with mobile solutes. This option is mainly useful for large solutes.

IDSX0

Flag which controls how the mixed van der Waals parameters are calculated for atom pairs where one atom vanishes (at either lambda=1 or lambda=0). (See Ref. 6).

- = 0 $r^*(\text{state where one atom vanishes}) = r^*(\text{non-vanishing atom})$ (This is the way AMBER has done this in the past) Default.
- > 0 $r^*(\text{lambda})$ will be calculated so that $r^*(\text{state where one atom vanishes}) = \text{IDSX0}/1000 r^*(\text{state where both atoms exist}) = r^*(A) + r^*(B)$
- = -1 results in $r^*(\text{state where one atom vanished}) = 0.0$

- ITRSLU During a periodic boundary conditions simulation, controls whether SOLUTE molecules which exit the primary image box will be translated back into the central box. SOLVENT molecules which exit the central image box are always translated back into the box. A molecule is considered to have floated out of the central box if the first atom of the molecule exits the box.
- = 1 Both SOLUTE and SOLVENT molecules which exit the primary image box will be translated back into the box. The system will be translated every 500 steps so that the center of geometry of the solute is centered in the primary image box. (Default; recommended for most systems).
 - = 2 Same as 1, except that the system as a whole is not periodically translated to keep the solute centered in the primary image box.
 - = 0 Only SOLVENT molecules will be translated back into the primary image box. SOLUTE molecules are not translated.
- IOLEPS Controls how parameter mixing is performed for non-bonded interactions.
- = 0 Mixing of epsilon (well-depth) van der Waals parameters done as

$$\epsilon(\lambda) = \lambda * \epsilon(\text{mixed}, \lambda = 1) + (1 - \lambda) * \epsilon(\text{mixed}, \lambda = 0)$$
 Mixing of electrostatic interactions done as

$$q_1 q_2(\lambda) = \lambda * q_1 q_2(\lambda = 1) + (1 - \lambda) * q_1 q_2(\lambda = 0)$$
 This is the default
 - = 1 Mixing of epsilon done as

$$\epsilon(\lambda) = \sqrt{(\epsilon_i(\lambda)\epsilon_j(\lambda))}$$
 Mixing of electrostatics done as

$$q_1 q_2(\lambda) = q_1(\lambda)q_2(\lambda)$$

Setting IOLEPS=1 forces mixing to be done as in older versions (e.g. 3.0, 3A) of AMBER. The "new" mixing scheme (IOLEPS=0) has several advantages, including A) a finite derivative for van der Waals interactions involving an atom which "disappears" at one end point; and B) Interaction between pairs of atoms where one/both atoms "disappear" at both end points never contribute to the energy. [One side- benefit of this is that it allows duplicate topologies; thus one can perform perturbations using the "CHARMM" methodology, if desired]. Note that if IDIFRG = 1 (thermodynamic integration), the epsilon parameters are always mixed as described for IOLEPS = 0.
- INTPRT Determines which energies contribute to the calculation of the free energy change.
- = 0 No intra-perturbed group energies are accumulated (Default; same as pre-4.0 versions of AMBER)
 - = 1 intra-pert. group non-bond energies accumulated as well (but no 1-4's).
 - = 2 intra-pert. group non-bond energies accumulated (including 1-4's).

- = 3 intra-pert group internal energies accumulated (bonds, angles, torsions)
- = 4 intra-pert group non-bond and internal energies accumulated
- = 5 intra-pert group non-bond, 1-4, and internal energies accumulated

Note: If any PMF contributions are being calculated (NCORC = 1, line 9), *all* intra-perturbed group non-bonded contributions will be calculated if INTPRT = 1,2,4 or 5 (when NCORC=1, 1-4's are not broken out separately).

- ITIP By default (ITIP=0), GIBBS assumes that if you are running a periodic boundary conditions (PBC) simulation with solvent, the solvent is TIPNP water. A special characteristic of this solvent model is that there are no h-bond (10-12) interactions between any pair of solvent molecules. A potential speedup is thus obtained by skipping all such h-bond interactions. If you choose to use a solvent model where there should be h-bond (10-12) interactions calculated between pairs of solvent molecules, set ITIP to any value other than 0. Note that in either case, all 10-12 interactions between solvent and solute molecules will still be determined normally.
- CUT The primary cutoff distance for the non-bonded pairs. Default = 8.0.
- SCNB The scale factor for 1-4 vdw interactions; if (SCNB .EQ. 0.0) then SCNB = 2.0, which is the default.
- SCEE The scale factor for 1-4 electrostatic interactions There is no namelist default, since the 1991 and previous force fields used 2.0, while the 1994 force field uses 1.2.
- DIELC Dielectric constant for the electrostatic interactions; if (DIELC .LE. 0.0) then DIELC = 1.0. Default is 1.0.
- CUT2ND An (optional) secondary cutoff. If CUT2ND > 0.0, then at every nonbonded update (every NSNB steps), the energies and forces due to interactions in the range CUT < R_{ij} <= CUT2ND will be determined. These energies and forces will be added to the non-bonded interactions within CUT distance at every timestep. The idea is that long-range interactions change more slowly than short range interactions, and thus this dual cutoff method allows one to include longer-range information at only a moderate additional cost. Default is 0.0.
- CUTPRT An (optional) alternative cutoff to be used for interactions with the perturbed group. If CUTPRT and CUT2ND are both defined, interactions in the range CUTPRT < R_{ij} <= CUT2ND will constitute the secondary cutoff range for interactions with the perturbed group. Default is 0.0.
- NTPR Flag for printing energy related quantities. for every NTPR steps these quantities will be output. Default is 100.
- NTWX Flag for packing the coordinates. For every NTWX steps the coordinates will be dumped through file 'PCOORD' in format (10F8.3). If NTWX=-1 (default), no dumping will be performed.
- NTWV For every NTWV steps the velocities will be written in file 'PVEL' in format (8F8.4). If NTWV=-1 (default), no dumping will be performed.

NTWE	Every NTWE steps energy info is written in file 'PEN' in formatted form. If NTWE=-1 (default), no dumping will be performed.
NTWXM	After NTWXM steps the NTWX switch will be inactive. Default is 999999.
NTWVM	After NTWVM steps the NTWV switch will be inactive. Default is 999999.
NTWEM	After NTWEM steps the NTWE switch will be inactive. Default is 999999.
IOUTFM	Flag for format of velocity and coordinate sets = 0 Formatted (default) = 1 Binary
ISANDE	Flag to output enthalpies and entropies, as well as free energies. Note that these quantities are typically an order of magnitude or more less precise than free energy values, and will be much more sensitive than free energies to the completeness of the ensemble statistics collected. See the discussion following the input description for more information. Setting ISANDE = 1 will also force the printing of the integrand quantity $\langle \partial V / \partial \lambda \rangle$ when Thermodynamic Integration is being performed (see the IDIFRG flag, 14.6). This can be useful if the user wishes to apply an alternative integration algorithm. Default is 0.
IPERAT	Request that free energy components or derivatives be calculated. Note that free energy components can be determined during any standard free energy simulation. Free energy derivatives can only be calculated in a special simulation where lambda does not change. = 0 No free energy components or derivatives will be calculated (default). = 1 Report free energy components. Components will be reported in file PATNRG on a per-atom basis. = 2 Report free energy components. Components will be reported in file PATNRG on a per-residue basis. = 3 Report free energy components. Components will be reported in file PATNRG on a per-molecule basis. = 4 Calculate/report free energy components or derivatives (depending on the flag ICMPDR). Values will be reported in file PATNRG for the atoms/groups defined at the end of input using GROUP input. For free energy components, free energies will be logged as defined by the GROUP definition, subject to the condition that only those atoms which are part of the perturbed group or which move with an added CONstraint will ultimately be included. All atoms not explicitly included in a group will be put in a final single group. For free energy derivatives, derivatives will be logged only for those atoms included in a group definition. Any atom of the system may be designated as part of any group (but each atom will be a member of at most one group). Typically, you will place individual atoms in their own groups when calculating derivatives.
IATCMP	If free energy components are being reported, by default only the total free energy per atom/residue/molecule/ group is reported. By setting IATCMP > 0, one can force the components to be broken down into electrostatic, non-

bonded and internal contributions. IATCMP has no affect when free energy derivatives are being calculated.

= 0 Do not break free energy components into contributions (default).

= 1 break free energy componenets into contributions.

NTATDP

Free energy components/derivatives will only be reported every NTATDP steps. Note that if free energy components are being logged, a free energy report will occur at a particular multiple of NTATDP steps only if the free energy accumulators have been updated since the last report. For free energy derivatives, energies will be reported every NTATDP steps in all cases. If NTATDP = -1, it is set to NTPR; default is 0.

ICMPDR

= 0 no free energy derivatives (default).

= 1 If IPERAT=4, log the free energy derivatives with respect to charge and the non-bonded parameters epsilon and r*. If the contributions of constraints to the free energy are being calculated (NCORC = 1), then derivatives with respect to constraints in the perturbed group (and added constraints) will also be calculated.

Free energy derivatives can only be calculated for lambda = 0 or lambda = 1. It is sufficient to define a "null" perturbed group in PARM if you simply wish to determine the non-bonded free energy derivatives of specified atoms.

NCMPDR

If free energy derivatives are being calculated (IPERAT=4 and ICMPDR=1), NCMPDR gives the number of steps of effective "equilibration." After the first NCMPDR steps, the accumulators for the free energy derivatives are cleared and reset. Free energy derivatives reported from this point forward will only reflect averaging since the accumulators were cleared. Some people prefer to use a post-processing program to analyze free energy derivatives. Such programs can usually "remove" a given initial portion of the free energy derivative information from subsequent totals. In such a case, you may wish to set NCMPDR=0 here (no "equilibration" phase), and pick the amount of data to discard in the post-processing program.

NTWPRT

Coordinate/velocity archive limit flag. This flag can be used to decrease the size of the coordinate / velocity archive files, by only including that portion of the system of greatest interest. (E.g. one can print only the solute and not the solvent, if so desired).

= 0 Coord/velocity archives will include all atoms of the system (default).

< 0 Coord/velocity archives will include only the solute atoms.

> 0 Coord/velocity archives will include only atoms 1->NTWPRT.

NTR

Flag for restraining specified atoms.

= 0 Classical MD (default)

=

MD with restraint of specified atoms

- NTRX Flag for reading the cartesian coordinates for restraint from unit PREFC. Note: the program expects coordinates for all atoms from which a subset is selected by the GROUP input which follows.
- = 0 binary form
 - = 1 formatted form (default)
- TAUR The relaxation time for restraint. Default is 1.0 ps.
- INTR
- = 0 No additional internal restraints or constraints will be read (default).
 - > 0 Additional internal restraints/constraints will be read following the normal input. Storage will be allocated for a maximum of INTR added restraints/constraints. These restraints/constraints can be used for e.g. a PMF calculation.
- IBIGM To calculate the free energy contributions of a constraint (if NCORC=1), the free energy at $\lambda \pm d_\lambda$ is evaluated by shifting the value of the constraint to its value at $\lambda \pm d_\lambda$. This change in the value of the constraint can be effected either by performing half of the shift at each end/side of the internal, or by performing the entire shift at one end.
- = 0 Half of the shift is performed at each end of the internal.
 - = 1 The entire shift occurs at the end/side of the internal which results in fewer atoms being moved. This is the default.
- The number of atoms whose positions change with shifting the constraint affects how quickly the calculation can be performed. Setting IBIGM = 1 can significantly speed up some calculations (e.g. when rotating a ring about a constrained torsion which joins it to a protein), and IBIGM should typically be set to 1 for *in vacuo* simulations. In all cases, GIBBS determines which interatomic nonbonded distances depend on constraint values, and only these are recalculated when NCORC=1.
- ISFTRP Causes the 6-12/10-12 functions used for non-bonded interactions to be replaced by "soft repulsion" terms of the form
- $$RWELL * (r^2 - r^{*2})^2$$
- where r^* is the optimal interaction distance between a pair of atoms, calculated from their respective van der Waals radii. This function is sometimes useful in structure refinement, but should *not* typically be used in free energy calculations. Atoms in the perturbed group are always treated by normal (6-12 or 10-12) non-bonded forces, regardless of the value of ISFTRP.
- = 0 regular 6-12/10-12's. No soft repulsion. Default.
 - = 1 replace 6-12's by soft repulsion.
 - = 2 replace 10-12's by soft repulsion, as well.
- RWELL Force constant (in kcal/mol) used for soft repulsion interactions. Default is 5.0.
- IFTIME Mutation flag. If ISLDYN=0, then if IFTIME = 0 (default) a standard Window Free Energy Perturbation will be carried out. The perturbation will start

at $\lambda = \text{ALMDA}$, and proceed in equally spaced intervals of $\Delta(\lambda) = \text{ALMDEL}$ until 1 ($\text{ALMDEL} > 0$) or 0 ($\text{ALMDEL} < 0$) is reached. At each value of λ , NSTPE steps of equilibration and NSTPA steps of data collection (see line 15) will be performed, and energy evaluated using Equation 2. If $\text{IFTIME} = \pm 1$ A "Slow Growth" perturbation will be carried out. The simulation will start at $\lambda = \text{ALMDA}$, and will be run in either the 0->1 direction ($\text{IFTIME} = +1$) or 1->0 direction ($\text{IFTIME} = -1$). CTIMT gives the number of psec of dynamics which would be used to perform the complete change 0->1 (or 1->0). The actual length of the simulation will depend on the starting value ALMDA. **NOTE** IFTIME is included for backwards compatibility with input files created for previous versions (< 4) of AMBER. However, it is strongly recommended that you use the ISLDYN flag to specify the type of simulation desired. If ISLDYN.NE.0, IFTIME is ignored.

- CTIMT The total length of the MD simulation (in psec) to be carried out in performing a slow growth simulation which transforms state $\lambda = 0$ into $\lambda = 1$ (or vice-versa). Note that this variable does not control the number of steps which will actually be run. For example, if CTIMT = 10psec, ALMDA = 0.0, ISLDYN = +1, and $\text{NRUN} * \text{NSTLIM} * \text{DT} = 5\text{psec}$, only half of the desired simulation would be carried out. The remainder would have to be carried out by a restart. CTIMT is only used when ISLDYN = ± 1 or ($\text{IFTIME} = \pm 1$ and ISLDYN = 0). Default is 0.0.
- ALMDA The starting value of λ for this simulation. The value can be on the inclusive interval 0.0-> 1.0. Default is 1.0.
- ALMDA = 1 corresponds to the "initial" state and ALMDA = 0 corresponds to the "final" state. Intermediate "states" are defined by a linear combination of the parameters representative of ($\lambda = 0$) and ($\lambda = 1$). For restart simulations (IREST=1, line 2), ALMDA is read directly from the restart file, and the value specified here is ignored.
- ALMDEL For Standard (fixed width) Window and TI simulations, $\text{ABS}(\text{ALMDEL})$ gives the width of each window or integration interval. If double-wide sampling is used with Window Growth (default), at each value of λ , the free energies to both +ALMDEL and -ALMDEL are evaluated. This results in "double wide sampling" (see the introductory text). If ($\text{IFTIME} = 0$ and ISLDYN=0), the sign of ALMDEL determines the direction of the change. If ISLDYN= ± 3 , the sign of ISLDYN determines the direction of the change. ALMDEL should be chosen so that the free energy change over any interval is not too large. It has been suggested (somewhat arbitrarily) that as a rule the free energy change/window should not exceed 2RT. ALMDEL is only used when ISLDYN = ± 3 or ($\text{IFTIME} = 0$ and ISLDYN = 0). Default is 0.1.
- ISLDYN Free Energy Method flag. It is recommended that you use this flag exclusively, and ignore IFTIME. Default is -3.
- = ± 1 Perform a Slow Growth simulation. The simulation will be started at ALMDA, and CTIMT psec will be required to complete the conversion to the end (0 or 1). If ISLDYN = +1, the simulation will be carried out in the direction 0-> 1. If ISLDYN = -1, the simulation will be carried out in the direction 1-> 0.

- = ±2 Perform a Dynamically Modified Window simulation. The simulation will be started at ALMDA and progress either in the direction 0-> 1 (if ISLDYN = +2) or 1-> 0 (if ISLDYN = -2). The numbers of equilibration and data collection steps performed at each window are given by NSTMEQ and NSTMUL (on this line). If IDIFRG = 0, the energy will be evaluated at each interval using Equation 2 (FEP). If IDIFRG = 1, thermodynamic integration will be carried out using Equation (4).
- = ±3 Perform a "standard" Window Growth simulation (with fixed width lambda intervals). The perturbation will start at lambda = ALMDA, and proceed in equally spaced intervals of delta(lambda) = abs(ALMDEL) until 1 (ISLDYN > 0) or 0 (ISLDYN < 0) is reached. At each value of lambda, NSTMEQ steps of equilibration and NSTMUL steps of data collection (see this line) will be performed. If IDIFRG = 0, the energy will be evaluated at each interval using Equation 2 (FEP). If IDIFRG = 1, thermodynamic integration will be carried out, using Equation (4).
- IDIFRG Thermodynamic integration flag.
- = 0 No thermodynamic integration (default).
- = 1 If windows or dynamically modified windows have been specified, the energy will be calculated using thermodynamic integration (TI) (Equation 4). The integrand will be evaluated at the endpoints of each "window", and the integral will be approximated using the trapezoidal rule (see the discussion following the input description). In addition to the integrated free energy, if ISANDE is set = 1 (see flag 12.10), the value of $\langle \partial V / \partial \lambda \rangle$ will be output at every energy update, so a different integration algorithm can be applied by the user, if desired. If slow growth has been requested, setting IDIFRG=1 has the effect of performing the slow growth summation using the non-averaging equivalent of the TI equation (4), rather than the FEP equation (2).
- NSTMEQ number of steps of equilibration to be used for each window if ISLDYN = ±2 or ±3. (Note that if windows are instead requested using the flag combination IFTIME = 0 and ISLDYN = 0, NSTPE is used). Default is 2.
- NSTMUL number of steps of data collection to be used for each window if ISLDYN = ±2 or ±3. (Note that if windows are instead requested using the flag combination IFTIME = 0 and ISLDYN = 0, NSTPA is used). Default is 2.
- NDMPMC Every NDMPMC windows, statistics will be dumped to the statistics file (MICSTAT). The statistics file contains a condensed format record of the free energy for each window interval. The MICSTAT file is not written with slow growth, or if NDMPMC is set < 0. By default NDMPMC=100. NDMPMC cannot exceed 100.
- IDWIDE Allows double-wide sampling to be turned off with FEP.
- = 0 Double-wide sampling performed when FEP windows are being calculated (default).

= 1 Double-wide sampling turned off when FEP windows are being calculated.

Double wide sampling means at each value we calculate the free energy in both the "forward" and "reverse" direction. This gives an intra-run consistency check (lower bound on the error), but requires we calculate every interval twice. The simulation can be run in roughly half the time, without the forward/reverse consistency check, by setting IDWIDE=1. The nature of thermodynamic integration (IDIFRG=1) is such that double wide sampling is never carried out. IDWIDE has no effect for such calculations.

IBNDLM By default (IBNDLM=0), $\lambda \pm d_\lambda$ is constrained to the range $0 < \lambda \pm d_\lambda < 1$. If IBNDLM=1, then $\lambda \pm d_\lambda$ can exceed the range $0 \rightarrow 1$. Useful when doing PMF-type calculations. Ignored for regular slow growth.

IAVSLP The current $dG/d\text{LAMBDA}$ slope will be approximated by a linear fit to the Accumulated G vs. LAMBDA data for the previous IAVSLP windows. Maximum value = 1000; default is 8.

IAVSLM Until IAVSLM windows have been collected, the window spacings will be fixed at ALMDL0 (line 14c). When IAVSLM windows have been collected, the slope will be calculated over all available windows, until IAVSLP windows are available. Default is 2.

```
i.e. #_windows < IAVSLM : dLAMBDA = ALMDL0
      IAVSLM <= #_windows < IAVSLP :
          dLAMBDA calculated from slope over #_windows
      #_windows >= IAVSLP :
          dLAMBDA calculated from slope over previous IAVSLP
          windows
```

If IAVSLM=-1, window widths will be fixed at ALMDL0 until IAVSLP windows are available.

ISLP Determines the direction in which the slope is calculated.

= 0 (default) use the appropriate value of ISLP (-1 or 1) to calculate the slope from energies calculated in the same direction as the simulation (recommended).

= 1 the slope is calculated from the forward (0->1) energy at each step.

=-1 the slope is calculated from the reverse (1->0) energy at each step.

= 2 the slope is calculated using the average of the redundant free energy values (from double wide sampling) over the interval in the direction opposite to the simulation, i.e. $G(\text{reverse}[\text{curr window}] - G(\text{forward}[\text{prev window}])/2$ or $G(\text{forward}[\text{curr window}] - G(\text{reverse}[\text{prev window}])/2$ for simulations run 0->1 and 1->0, respectively. This option can be useful when very few points are used to evaluate each slope (e.g. IAVSLP = 2).

= 3 the slope is calculated using the average of the forward and reverse energies at each lambda.

For best results in most cases, the slope should be calculated in the same direction as the simulation. This is the default behavior (ISLP=0). With thermodynamic integration, or when double-wide sampling is defeated, ISLP has no effect. Only options ISLP=0 or ISLP=3 should typically be used when AMXRST > 0.

- CORRSL** If the correlation coefficient for a linear fit to the previous IAVSLM windows is < CORRSL, the number of windows over which the slope is calculated will be halved (for this determination of the slope only), and the slope calculated again. This process continues until the correlation coefficient is > CORRSL. Default is 0.8.
- AMXMOV** The target free energy change per window. If M is the slope over the previous IAVSLP windows, the next value of dLAMBDA is chosen as $dLAMBDA = AMXMOV/M$. Note that when double wide sampling is defeated (IDWIDE=1) while using a window FEP technique (IDIFRG=0), the free energy change at a window is defined as the total ("forward" + "reverse") energy change. This differs from the definition when double wide sampling is used, where the free energy change at a window is approximately $1/2 * ("forward" + "reverse")$. Thus, AMXMOV should be suitably increased when IDWIDE = 1. Default is 0.1.
- IAVDEL** Number of windows over which the forward and reverse energies will be compared. If IAVDEL<0, no comparisons will be carried out. IAVDEL should always be set <0 when thermodynamic integration is used (IDIFRG = 1). Maximum value = 1000; default is -1.
- IAVDEM** The relationship between IAVDEL and IAVDEM is analogous to that between IAVSLP and IAVSLM. Default is 2.
- AMXDEL** If $< ABS (DA(for)-DA(rev)) > .GT. ABS(AMXDEL)$ then the next dLAMBDA will be scaled as $[< ABS (DA(for) - DA(rev)) > / AMXDEL] **2 * dLAMBDA$. If AMXDEL < 0, then scaling occurs in all cases. Default is 1.0.
- ALMDL0** Until enough intervals have been calculated to allow determination of dG/d_lambda and d_lambda consistent with IAVSLP and IAVSLM, an interval width of ALMDL0 will be used. Default is 0.0001.
- DLMIN** The minimum allowable window width. Default is 1.0D-6.
- DLMAX** The maximum allowable window width. Default is 0.1.
- AMXRST** If the free energy change, dG, over any window is greater than AMXRST, then the data collection phase for that window will be re-performed using a reduced value of dLAMBDA. The new value of dLAMBDA is determined as $dLAMBDA(new) = (dLAMBDA(old)/dG) * AMXMOV$. AMXRST should not be set too close to AMXMOV, or too many windows will be recalculated (which is inefficient). By default, $AMXRST=5.*ABS(AMXMOV)$.
- NORSTS** If this is a restart run, and NORSTS=1, then the restart information relating to dynamically modified windows is not read (cold start for the dynamically modified windows). NORSTS is ignored if this is not a restart run. Normally, NORSTS should be set to the default of 0.
- NTSD** The statistics relating to dynamically modified windows are written to file POUT every NTSD. If NTSD=0, then NTSD is set equal to NTPR (line 12),

and these statistics will be output every time the standard energy information is printed. Default is 0.

ALMSTP(1) Allows the values of AMXMOV, DLMIN, DLMAX, AMXRST, and NTSD to be different for different ranges in LAMBDA.

> 1 or < 0

the values defined in lines 14a-14c will remain in effect for the whole run.

> 0 and < 1

the values defined in lines 14b-14d will remain in effect for the range of LAMBDA ALMDA-> ALMSTP(1). In this case, additional line(s) are read with the values of the above variables over various ranges of LAMBDA. Each line has the format

```
AMXMOV, DLMIN, DLMAX, AMXRST, NTSD, ALMSTP(I)
```

```
FORMAT(4F14.9, I5, F14.9)
```

These lines are read until ALMSTP(I) > 1 or ALMSTP(I) < 0. Each set of values applies to the range in LAMBDA ALMSTP(I-1) -> ALMSTP(I). Note that for the last line, ALMSTP(I) must be greater than 1, or less than 0 (not equal to). This is to avoid machine precision problems. Note also that, at present, "namelist"-format input always assumes ALMSTP(1) < 0 (i.e. AMXMOV, DLMIN, etc. remain fixed over the entire run). If you wish to use the functionality described above for ALMSTP(), you must use formatted input.

NSTPE The number of steps of Equilibration before collecting the Free Energy Statistics. For each window the system is equilibrated for NSTPE steps. (When ISDYN= ± 2 or ± 3 , NSTMEQ serves the same purpose). Default is 2.

NSTPA The number of steps for data collection. The averaging is performed over this number of steps. (When ISLDYN= ± 2 or ± 3 , NSTMUL serves the same purpose). Default is 2.

DTA The time-step used for window runs specified by IFTIME=0 and ISLDYN=0. All other runs use the time-step specified on line 8. Default is 0.001

IVCAP Flag to control Cap Option. The Cap option is to solvate a spherical portion of a solute and to hold the solvent from evaporating through a half-harmonic potential.

= 0 Cap will be in effect if it is passed from the PARM file (default).

= 1 Cap will be activated except that the Cap atom pointer would be modified.

= 2 Cap will be inactivated.

NATCAP The Cap atom pointer It is the last Non-Cap atom number. If IVCAPEQ.1 then the pointer passed from the PARM file will be overwritten by this number. Default is 0.

FCAP The Force Constants for the Cap Atoms. Default is 0.0

WATNAM	The residue name the program expects for TIP3P waters. Default is "WAT".
OWTNM	The atom name program expects for the TIP3P oxygen. Default is "O ".
HWTNM1	The atom name program expects for the TIP3P 1st H. Default is "H1 ".
HWTNM2	The atom name program expects for the TIP3P 2nd H. Default is "H2 ".

The following card is read *only* if I3BOD.NE.0; (This information must be provided in the formatted form given, even if namelist format input is used above.)

```
- 18A-      1) N3B,  NION                FORMAT(2I5)
```

The number of 3body interactions to be defined, and the number of ions in the system.

Next, include N3N cards 18B to define all 3-body interactions:

```
- 18B-      1) AT1 (I)  2) AT2 (I)  3) ACON1 (I)  4) BETA31 (I)  5) GAMMA31 (I)
              6) ACON0 (I)  7) BETA30 (I)  8) GAMMA30 (I)

              FORMAT (A4, A4, 2X, 6E10.3)
```

AT1(I)	The second atom in this 3-body interaction.
AT2(I)	The third atom in this 3-body interaction.
ACON1(I)	The pre-exponential factor for this 3-body interaction for the lambda = 1 state.
BETA31(I)	The beta value for this 3-body interaction, for the lambda = 1 state.
GAMMA31(I)	The gamma value for this 3-body interaction, for the lambda = 1 state.
ACON0(I)	The pre-exponential factor for this 3-body interaction for the lambda = 0 state.
BETA30(I)	The beta value for this 3-body interaction, for the lambda = 0 state.
GAMMA30(I)	The gamma value for this 3-body interaction, for the lambda = 0 state.

```
- 19 -      IDENTIFICATION OF ATOMS WITH POSITION CONSTRAINTS
              *** ONLY IF NTR = 1 ***
```

Constraint reference atoms are obtained by first reading coordinates for the entire structure through file 'PINCRD' or 'PREFC', then specific constraint atoms are selected by group. See the section on GROUP in the Appendices for format. Does not support a namelist convention.

```
- 20 -      IDENTIFICATION OF ATOMS FOR BELLY RUN
              ***** ONLY IF IBELLY .GT. 0 *****
```

The belly atoms are loaded as groups. Consult the GROUP section in the Appendices for a description of how to define a group. The group definition immediately follows the end of the

&cntrl namelist. *The GROUP input does not support a namelist convention.*

```
- 21 -      DEFINITION OF GROUP INPUT FOR FREE ENERGY COMPONENTS
           OR DERIVATIVES ***** (ONLY IF IPERAT = 4) *****
```

For free energy components, free energies will be logged as defined by the GROUP definition, subject to the condition that only those atoms which are part of the perturbed group or which move with an added CONstraint will ultimately be included. All atoms not explicitly included in a group will be put in a final single group.

For free energy derivatives, derivatives will be logged only for those atoms included in a group definition. Any atom of the system may be designated as part of any group (but each atom will be a member of at most one group). Typically, you will place individual atoms in their own groups when calculating derivatives.

Note that in GIBBS, GROUP input supports two new features that can be helpful in defining the input for free energy components or derivatives. Both allow the creation of multiple single-atom groups:

```
ATOM -IAT1 IAT2
```

(1st atom number negative) will place each atom from IAT1 to IAT2 in its own group.

```
RES -IRES1 -IRES2
```

(both residue numbers negative) will place each atom of every residue in the range IRES1->IRES2 in a separate group. Group definition syntax is otherwise the same as described in the manual.

```
- 22 -      DEFINITIONS OF INTERNAL RESTRAINTS/CONSTRAINTS
           *** ONLY IF INTR > 0 (line 13) ***
```

Setting INTR > 0 allows the user to define here a series of internal restraints and constraints whose force constants and equilibrium values are a function of lambda. *Restraint/constraint definitions must be entered in the formatted form shown below, not in a namelist.* Restraints/constraints are read in as pairs of lines:

```
line A: IAT1, IAT2, IAT3, IAT4, IUMB, IZE, ITOR, RLMDA1, RLMDA2
        FORMAT (7(I5,1X),2F10.5)
line B: RKEQ1,REQ1,RKEQ2,REQ2,IPER,IPER2
        FORMAT (4F10.5,2I5)
```

As many restraints/constraints may be defined as are desired. A blank record signals the end of the input. This data must be entered in the formatted form shown. *It does not support a namelist convention.*

IAT1-->IAT4 The absolute atom numbers for the atoms defining the restraint. If an atom number is <0, the absolute value of the atom number is used (additional behavior for <0 values is defined when IZE=1; see below).

```
IAT3 = IAT4 = 0      : Bond restraints/constraints
IAT4 = 0             : Angle restraints/constraints
IAT1->IAT4 non-zero: Torsion restraints/constraints
```

RLMDA1

RLMDA2 The restraint/constraint will be applied only over the range in lambda (RLMDA1, RLMDA2).

RKEQ1

REQ1 The force constant in kcal/mol and equilibrium value, respectively, for the restraint/constraint at lambda = RLMDA1.

RKEQ2

REQ2 The force constant in kcal/mol and equilibrium value, respectively, for the restraint/constraint at lambda = RLMDA2. If RLMDA1=RLMDA2, the force constant and eq. value are fixed at RKEQ1 and REQ1 (RKEQ2 and REQ2 are ignored). RKEQ1 and RKEQ2 are ignored for constraints (ITOR=2). If REQ1=999. or REQ2=999., the corresponding equilibrium value is set to the current value of the internal coordinate (as determined from the input set of coordinates PINCRD). If ABS(REQ1) > 1000, the corresponding equilibrium value is set

```
REQ1 < 0:   REQ1 = current_value - [ABS(REQ1)-1000.]
REQ1 > 0:   REQ1 = current_value + [ABS(REQ1)-1000.]
```

If ABS(REQ2) > 1000, REQ2 is analogously reset.

Intermediate $K_{eq}(\lambda)$ and $R_{eq}(\lambda)$ are determined by linear interpolation between the force constants and equilibrium values at RLMDA1 and RLMDA2. No restraint/constraint is applied outside the range (RLMDA1,RLMDA2).

IZE

= 0 The restraint/constraint defined here is used in addition to other parameters corresponding to this atom sequence from parm (if any).

= 1 The restraint parameters defined here replace overlapping parameters from parm (if any) for this atom sequence. When IZE=1, any atom number IAT1->IAT4 which was specified as < 0 has a special meaning: It allows a "wildcard" match to the corresponding atom number when replacing parameters from parm. For example, the sequence -1 3 8 -14 would result in a torsional restraint which would replace parameters for all torsions centered on the bond between atoms 3 and 8. IZE is read but ignored when ITOR=2 (constraints).

IUMB

Determines the type of restraint.

= 0 The restraint is to be considered part of the molecular force field. The free energy contribution from the restraint is calculated by the standard formula (c.f. Equation 2).

= 1 The restraint is considered to be an "umbrella" term. The effects on the ensemble of the restraint are evaluated using the following function in place of Equation 2:

$$\Delta G = -RT \ln(\langle e^{-\Delta V/RT} e^{\phi/RT} \rangle_{V+\phi} / \langle e^{\phi/RT} \rangle_{V+\phi}) \quad ,$$

where ϕ is the sum of all umbrella restraint terms and ΔV is as described for Equation 2. IUMB is ignored for constraints (ITOR=2). IUMB = 1 will not work correctly with slow growth or thermodynamic integration.

ITOR Functional form/constraint flag.

= 0 If this is a torsional restraint, a potential of the form

$$K_{tor} (\tau - \tau_0)^2$$

is used. This functional form is always used for bonds and angles (ITOR = 0 has no effect for bonds/angles).

= 1 If this is a torsional restraint, a potential of the form

$$K_{tor} (1 - \cos(\tau - \tau_0))$$

is used. (ITOR = 1 has no effect for bonds/angles).

= 2 Then a constraint, rather than restraint, is applied to the corresponding internal coordinate. This is applicable to all types of internal coordinates (distances, angles, torsions). If NCORC = 1 (line 9), then an effective "potential of mean force" (PMF) contribution to the free energy will be calculated for this internal coordinate. General "holonomic" internal constraints are used, as described in Reference 7.

When ITOR = 2 (internal is being constrained), IZE is ignored, and the following occurs:

For bonds and angles, if the constrained internal matches an internal in the topology file, force constant parameters for matching internal will be set to 0.

For torsions, if the constrained internal matches an internal in the topology file, A) forces for all torsions centered on the same bond will be omitted B) The contributions to the free energy of all torsions centered on the same bond as the constraint will be calculated. This is necessary because several torsions can be centered on a central bond, and there is no fixed relative arrangement for these torsions.

IPER IPER can be used to define a simulation where two internal coordinates will be varied with two independent values of lambda. Such a simulation can be used to generate a free energy internal-internal map (sort of a free energy equivalent to a Ramachandran map) to be generated.

The output of this option is somewhat complex, and is intended for post-processing by a separate program. Any 2-D run of value will necessarily be very

compute-intensive, and a number of issues must be considered before undertaking such a simulation. This option should generally be avoided by the novice user. If you are considering performing such a simulation, you are *urged* to read Reference 8 (see above) first. For use with the IPER flag, we define:

primary lambda

the "normal" lambda; that is, the lambda used in standard GIBBS runs to describe how the system varies between the initial and final states.

secondary lambda

a second lambda, which is translated from 0->1 at each value of the "primary" lambda.

- = 0 This restraint will vary with the primary lambda; i.e. the equilibrium value and force constant will be a function only of the primary lambda. This is standard behavior.
- > 0 This restraint will vary with the secondary lambda; i.e. the equilibrium value and force constant will be a function only of the secondary lambda. Lambda will be varied from 0->1 for this restraint in a series of IPER equally-spaced intervals (windows). The "secondary" lambda is not used unless one or more restraints are defined with IPER > 0.

The number of windows used for each "primary" restraint will be the same, and the number used for each "secondary" restraint will be the same. The first IPER(I) > 0 sets the number of windows used for *_all_* secondary restraints.

If secondary restraint(s) are requested, the value of IPER2 (see below) corresponding to the first value of IPER(I) > 0 defines the number of windows used for every primary restraint. Note that any dynamically modified window or slow growth flags (card 14) will be defeated in this case.

When calculating PMF-type energies (if NCORC=1), constraints will be applied in two cycles. First, dG will be calculated for +-d(internal) for only those internals for which IPER=0. Then a dG will be calculated +-d(internal) for only those internals for which IPER>0.

Any parameters (other than constraints) that vary with lambda will only change when lambda for the primary constraints changes.

If IPER > 0, window or dynamically-modified window growth must have been requested (line 14). IPER cannot be set > 0 with slow growth or with thermodynamic integration (IDIFRG > 0).

The matrix of energies from a 2-D run is contained in file CONSTMAT. A matrix can be generated with either IDWIDE = 0 or IDWIDE = 1, but it is strongly recommended that IDWIDE = 1 (no double-wide sampling) be used. In this case, five free energy difference are evaluated from each ensemble, corresponding to moves from (lam1, lam2) to (lam1, lam2+d_lam2), (lam1, lam2-d_lam2), (lam1+d_lam1, lam2), (lam1-d_lam1, lam2), (lam1-d_lam1, lam2-d_lam2). This set allows the whole free energy map to be evaluated most efficiently (see the Pearlman and Kollman reference [8] noted above).

The "secondary" lambda always changes in the "forward" direction, always starts at 0.0, and always ends at 1.0. After lambda has gone from 0->1. The primary lambda is incremented one step, the secondary lambda is reset to 0, and another cycle of secondary lambda changes occurs. At the start of each cycle of changes in the "secondary" lambda, the current coordinates are stored in file CNSTSCRT.

IPER2 If IPER > 0 for a particular restraint/constraint ("secondary" restraints defined), IPER2 gives the number of "windows" used in translating the "primary" lambda from 0 to 1. See the description of IPER above. If IPER > 0, IPER2 fixed-width windows will be used for the "primary" restraints, regardless of the behavior requested by ISLDYN, etc. (lines 14-ff).

1.8. Choices Affecting Free Energy Calculations

David A. Pearlman
(minor modifications by DAC)

The development of ever-more-powerful computers, combined with the wide dissemination of modeling packages like AMBER, puts the power to perform valuable calculations in the hands of an increasingly large number of scientists. It is tempting to say that, given the increasing sophistication of such programs, all one needs is the appropriate hardware and software to perform good experiments.

But this is not the case. As modeling programs have grown more sophisticated, they have sprouted an ever-increasing array of options—options which must be properly chosen, if worthwhile results are to be obtained. And even if the options are appropriately set, one must ensure that the program is properly suited for the chosen application. Nowhere in AMBER is this more true than the GIBBS free energy module.

Here we discuss several issues which impinge on developing an appropriate GIBBS input file, and on interpreting the results produced. One is also strongly encouraged to review the literature referenced here and in the preface to the GIBBS program.

1.8.1. What method should be used to calculate the free energy?

GIBBS offers five choices of method for calculating the free energy difference between two states. These include the general classes slow growth, free energy perturbation, and thermodynamic integration, as well as dynamically modified variants of the latter two. These were described in the introduction to GIBBS. As yet, it has not been shown conclusively what method is "best" for any particular type of problem.

- (1) Slow growth: This method to some extent has had a bad reputation, since an implicit assumption—that λ changes slowly enough that the system can be assumed to be in equilibrium at each step—does not strictly hold. The consequences of this "Hamiltonian lag" have not been fully quantified. There is recent interest in converting the original algorithms into "fast growth" techniques [13]; these approaches look very interesting but will not be covered here.
- (2) Window Growth: The equations of window growth, or Free Energy Perturbation (FEP) are exact, and, in principal, if one has the computer resources to perform sufficient sampling, one can obtain very accurate results. In practice, FEP suffers from two significant

difficulties. The first is that, in reality, we do not always sample to convergence. Unfortunately, no reliable test to prove convergence has been developed. The second problem with FEP is that Equation (2) requires that we obtain the ensemble average of a quantity which relies on the *difference* between the potential functions representative of both states $\lambda(i)$ and $\lambda(i+1)$. But the average is evaluated from the ensemble of states visited when MD is run using the potential function for state $\lambda(i)$. Thus, if states $\lambda(i)$ and $\lambda(i+1)$ are too dissimilar, it will be very difficult to obtain reliable statistics. Reducing the spacing between adjacent λ states helps circumvent this problem, but at a significant additional cost. And even then we do not have any reliable methods for assuring the problem has been avoided.

- (3) Thermodynamic Integration (TI): TI is appealing because it avoids the problem in sampling the exponential of $V(\lambda(i+1)) - V(\lambda(i))$ described for FEP above. But TI has its own problem: The driving equation of TI is an integral (Equation 4), which in practice must be calculated approximately by evaluating the integrand at discrete values of λ . Of course, TI is also susceptible to errors when a simulation is not run sufficiently long to obtain a converged value of the averaged quantity which serves as the integrand.

Note that we approximate the integral using the trapezoidal algorithm, i.e.

$$\Delta G_i = G(\lambda(i+1)) - G(\lambda(i)) = \langle \partial V / \partial \lambda \rangle_{\lambda(i+1)} - \langle \partial V / \partial \lambda \rangle_{\lambda(i)} (\lambda(i+1) - \lambda(i)) / 2 \quad (5)$$

This integration method should be reasonably accurate in most cases. But in case the user wishes to try their own integration scheme, setting ISANDE = 1 with TI will also force reporting of the values of $\langle \partial V / \partial \lambda \rangle_{\lambda(i)}$ and several other averages at every evaluation point (the other values reported relate to calculating the enthalpy/entropy, as described below).

- (4) Dynamically Modified Windows (DMW): In dynamically modified windows [5] the $\delta\lambda$ spacing between consecutive windows in FEP or TI is continually changing, to achieve a relatively constant free energy change per window. This should improve the efficiency of the calculation, by focusing proportionately more simulation time on those ranges of λ where the free energy is changing more rapidly. We have, in fact, shown that dynamically modified windows significantly improve the sampling efficiency of FEP simulations for model compounds. The biggest drawback to DMW is that, because we do not know *a priori* the exact shape of the free energy versus λ curve when we start a simulation, we cannot predict with certainty how long the simulation will take to go to completion. This caveat noted, it appears that DMW would be beneficial to most FEP and TI simulations.

1.8.2. Enthalpies and entropies

GIBBS allows the user to request that the enthalpy and entropy changes be reported in addition to the free energy (which is always reported). Two different schemes are used to calculate these quantities, depending on the free energy calculation method. Note that in either case, the enthalpy and entropy are necessarily dependent on being able to reliably extract small differences between averages of (often large) total system energies. In the case of free energy, on the other hand, we need only measure the average of a potential difference or a derivative. For this reason, enthalpy/entropy estimates are typically more than an order of magnitude less accurate than their free energy counterpart. One should be very cautious when interpreting them.

For FEP, the approximate equations given by Fleischman and Brooks [14] are used. These approximate the required temperature derivatives by a finite difference. The equations used are

derived from the FEP expression, and the sum of the resulting (enthalpy - T*entropy) will equal the reported free energy.

For TI, the enthalpy and entropy are evaluated using exact-form integral relationships given by Yu and Karplus [15] (and in other places). The (enthalpy - T*entropy) calculated by this method will not necessarily equal the reported total free energy; the difference between the two quantities can be taken as a crude indication of the reliability of the enthalpy/entropy values. The integrals are approximated by the trapezoidal rule, as described above (Equation (5)).

1.8.3. Mixing rules for vanishing atoms

By default, the optimal interaction r_{ij}^* between two atoms i and j is given by

$$r_{ij}^*(\lambda) = r_i^*(\lambda) + r_j^*(\lambda) \quad (6)$$

This is fine when neither atom "vanishes" at either λ endpoint. But now consider the case where atom i vanishes at $\lambda=0$. Then

$$r_{ij}^*(0) = r_i^*(0) + r_j^*(0) = r_j^*(0) \quad (7)$$

Thus, r_{ij}^* never gets smaller than $r_j^*(0)$. At $\lambda=0$, the mixed well depth, $\epsilon(0)$, will also be 0. But at any value of λ just slightly >0 , $\epsilon \neq 0$, and suddenly a steric "gap" between atoms i and j of r_{ij}^* will be required. This can lead to sampling inefficiencies. A better solution is to shrink $r_{ij}^*(\lambda)$ to a user-chosen small value as one of the atoms "vanishes". This is the effect of variable `IDSX0` (line 10).

1.8.4. Using Dynamically Modified Windows

The theory of DMW, and some exploratory applications, are described elsewhere [5]. A sample input for GIBBS is shown below, follow by a few important explanatory notes.

```

line
14      0 40.00000 0.00000 -0.02500 +2 0 100 100 0 0 0 0
14a     8  2  0  0.8000000  0.0100000
14b    -10 20  0.0001000
14c    1.0D-5 1.D-10 1.0D-2          0.10000000  0  0 -1.00

```

(format compressed to fit page)

Line 14

We set `ALMDEL = 0`, `ISLDYN=+2`, `IDIFRG=0`, `NSTMEQ=100`, and `NSTMUL=200`. This results in dynamically modified window FEP, with 100 steps of equilibration and 100 steps of data collection per window.

Line 14a:

On the next line, we set `IAVSLP = 8`, `IAVSLM=2`, and `CORRSL=0.8`. This means that, at most, the 8 most recently calculated (λ , accumulated_free_energy) points will be used in approximating the $\partial G/\partial \lambda$ slope. `IAVSLM=2` means that as soon as 2 points are available, we will calculate the slope from all available points, until the maximum of 8 is reached. If the best-fit line through the points fits the data with a correlation coefficient (CC) < 0.8 , then the number of points used in the current slope determination will be halved, the slope and CC will be recalculated, and the comparison against CC will be performed again. A minimum of two points are always used to calculate the slope.

AMXMOV, which is set to 0.01 here, is the target change in free energy per window we are aiming for. The $\delta\lambda$ change on the next step is calculated as

$$\delta\lambda = \frac{AMXMOV}{(\partial G/\partial\lambda)} \quad (8)$$

Note that since we don't know *a priori* what the free energy versus λ curve will look like, we do not know exactly how many steps will be required to complete the simulation. The total number of MD steps required will depend both on AMXMOV and on NSTMEQ and NSTMUL (line 14). NSTLIM can be set to -1 on line 8 to force the program to continue until the total required number of steps have been performed. Also note that the value of AMXMOV used will often depend on the magnitude of the total anticipated free energy change. For example, one would not typically want to use AMXMOV=0.01 and NSTMEQ=NSTMUL=100 if the total energy change is 50 or 100 kcal/mole, as it can be for certain electrostatic changes.

line 14b:

IAVDEL < 0, which means that the $\Delta G_{forward} - \Delta G_{reverse}$ comparisons will not be used in scaling the widths of λ windows. The viability and reliability of changes made using these types of comparisons has not yet been established.

line 14c:

ALMDL0 is set to 1.0D-5. This means that the first IAVSLM window steps (before we have enough points to calculate a slope) will be made with this small step size. This step is chosen to be small in case the energy is changing quickly in this region.

DLMIN is set to 1.0D-10. Typically, a value of DLMIN such as this would have no effect, since it is unlikely that the slope and AMXMOV would be such to require a step this small in the first place. At any rate, steps calculated to be smaller than DLMIN are reset to DLMIN. DLMIN can be valuable in some cases when one wishes to limit how slowly a simulation can progress.

DLMAX is set to 1.0D-2. Setting an appropriate value for DLMAX is important. If the G versus λ curve has any points of inflection, we might calculate a slope of approximately 0 at one or more points. In this case, the simple formula used to determine the next step size would indicate a very large step (as large as 1.0, the whole simulation length). This would be incorrect, as the slope could clearly turn significantly non-0 in a future range of λ . DLMAX bounds the change in such cases.

AMXRST is set to 0.10. The slope we calculate is only an approximation of the "true" instantaneous slope, and the current slope is only an estimate of the slope over the next λ interval (window). Thus, it is possible that when we calculate the actual free energy change over the next window, it will be an unacceptable amount larger than the target value. In such a case, we may want to decrease the λ step size for this window and re-evaluate the energy. AMXRST is the largest allowable energy for a step. If the energy is > AMXRST, the $\delta\lambda$ stepsize is reduced, and the energy for the window recalculated. Note that setting AMXRST too close to AMXMOV will result not only in too many windows being reevaluated (inefficient), but can also lead to biased sampling.

ALMSTP(1) is set to -1.0. If $0 < \text{ALMSTP}(1) < 1.0$, one can prescribe that the values of AMXMOV, DLMIN, DLMAX and AMXRST vary over different ranges in λ , as described in the input discussion.

1.8.5. Potential of Mean Force (PMF) calculations

It is often of interest to determine the free energy difference between two states which differ in conformation, rather than in composition. For example, one might be interested in the free energy profile for rotation about a ring in a protein. Such a profile can be determined by performing a PMF simulation. To perform such a simulation, one must be able to define conformation as a function of lambda

within the context of an otherwise free MD simulation. Fortunately, methods have been developed which allow selected internal coordinates to be constrained to chosen values, while otherwise affecting the MD trajectory only minimally. The best known of these is the SHAKE method for bond constraints. The methods of SHAKE can be extended to be generally applicable to angles and torsions. One can calculate the free energy changes that accumulate as the internal constraints are translated from those of the initial state to those of the final state. If one graphs the free energy changes as a function of the restraint target values (themselves a function of λ), one gets the free energy profile for conformational changes.

Any constraint with a target value which is itself a function of λ will contribute to the free energy as λ changes. This means that if SHAKE is used to constrain bonds of the perturbed group, and any of those bonds "grow" or "shrink" during the simulation, there will be a corresponding contribution to the free energy. In earlier work, this contribution has been overlooked, but we have shown that it must be included to reliably calculate free energies using the FEP method [6]. The contribution in such a case can be calculated simply by setting NCORC=1.

Constraints other than SHAKE-en bonds can be defined by setting INTR > 0 (line 14) and providing the definitions after the standard input (see above). Any internal coordinate can be used; Be aware, however, that any internal coordinate which is part of a closed ring will present a special set of (often tricky) considerations (see below). In typical use, no compositional (or topological) change is performed when a PMF simulation is being carried out. A GIBBS-format topology file is still required from LEaP, though. Here, one does not define any per-atom perturbations ("edit molecule" / Selection / Edit selected atoms to turn per-atom pert on/off) and does a

```
> saveamberparmpert molecule nullpert.top nullpert.crd
```

In general, PMF calculations within GIBBS may be performed with any method – FEP, TI or slow growth. Note that there is one scenario where *only* the TI (with "constraint forces") method may be used: when any constrained internals whose target values change with λ lie within a closed loop. The loop can either be part of the molecular topology, *or as a result of the added topology of the constraint(s)*. To understand why neither FEP nor TI with "potential forces" can be used in such a case, you must recognize that for these latter methods, part of the procedure for calculating constraint contributions requires that we determine which atoms of the system are affected by a rigid body translation/rotation about the constrained internals. But the requisite set of atoms is not unambiguously defined when the constraint lies within a closed loop. Fortunately, the "constraint force" implementation of TI doesn't require that we make such a determination.

It is important to note that PMF calculations are typically very compute-intensive. For FEP, Gibbs will determine which non-bonded pairs have an interatomic distance which varies with one or more constraints, and only these are re-evaluated in determining $V_{\lambda(i+1)}$. This helps reduce the amount of computer time required for a FEP simulation, although the total amount of time can remain high. The additional cpu overhead for calculating constraint energies with TI is negligible in all cases.

While we have a good error check for some torsional PMF's (the free energy values after rotating 360° should be the same), we typically have no reliable way of determining that for other simulations enough sampling has carried out to determine a converged PMF curve. Our best guard against spurious results is careful consideration of the specific problem and the inherent relaxation timescales of the surroundings.

1.8.6. Error estimates and convergence

One of the thorniest issues related to free energy calculations is estimating the error in the results. This error is often estimated in one of four ways:

- (1) Two separate free energy simulations can be run, one with λ progressing from 0 \rightarrow 1, the second with λ progressing from 1 \rightarrow 0. These two calculations should yield the same free energy value, and the difference between them (the "hysteresis") gives a lower bound on the estimate in the calculation. Errors derived in this way often underestimate the actual error.
- (2) The difference between "forward" and "backward" values for a single run. As described in the introduction, when FEP or slow growth is performed, double-wide sampling can be carried out. This ultimately results in two pseudo-independent values for the free energy, one calculated from the sum of all the $\lambda(i)\rightarrow\lambda(i+1)$ energies, and the other calculated from the sum of all the $\lambda(i)\rightarrow\lambda(i-1)$. If the results were exact, these values would be the same. In practice, they will not be, and their difference gives a crude lower bound on the inherent error. Error estimates derived in this manner tend to be even less reliable than those estimated using method (1), and are usually worthless for slow growth type runs.
- (3) Two or more simulations are run under equivalent but different conditions. For example, starting with different randomly assigned sets of velocities. The difference between the free energies provide an estimate to inherent errors. These estimates are subject to the same problems as (1) above.
- (4) A series of simulations is run which differ in the respective amounts of sampling done. For example, simulation 1 might use 100 steps of equilibration and 100 steps of data collection at each window, while simulation 2 used 200 steps of each. If the value from the shorter simulation was accurate, the value from the second simulation should be acceptably close to it. If it is not, the simulation must be run even longer to confirm convergence. This method probably provides the best insurance that convergence has been reached, but it is not definitive, and it is also the most costly.

It must be understood that none of the above methods allows a completely reliable error estimate. At best, they provide a *lower bound* on the error. A large apparent error is a good indication that the results obtained are not appropriately converged. But a low apparent error does not necessarily indicate a converged and accurate simulation.

1.8.7. Changing parameters versus dual topologies

In "standard" operation, free energy changes in GIBBS are effected by transforming the potential representative of state 1 to that representative of state 2. The topology of the system does not change. To make atoms non-interacting at one of the endpoints, they are assigned zeroed non-bond and electrostatic parameters at this endpoint.

The extended mixing rules which can be used in GIBBS (IOLEPS = 0, line 10) allow a second method to be used. One result of these new mixing rules is that if any pair of atoms "exist" only at mutually exclusive endpoints (e.g. atom i exists in state 1 but not state 2; atom j exists in state 2, but not in state 1), then effectively no non-bonded interactions are ever calculated between them. This means that, in lieu of the "standard" method which uses a single topology, we can define dual topologies, one corresponding to the $\lambda = 0$ endpoint, and the other corresponding to the $\lambda = 1$ endpoint. For the former topology, all non-bonded parameters would be defined to be 0 in the $\lambda = 1$ state. Similarly, all non-bonded parameters for the latter topology would be 0 at $\lambda = 0$. The two topologies would then never "see" each other at intermediate values of λ . Defining dual topologies can aid in performing free energy calculations where bond connectivities must change. Dual topologies is the method

incorporated into the "CHARMM" program.

On an efficiency basis, the relative merits of the two methods have not been established. Additional discussion of the two methods can be found elsewhere [6].

1. D.L. Beveridge & F.M. DiCapua. Free energy simulation via molecular simulations: Applications to chemical and biomolecular systems. *Annu. Rev. Biophys. Chem.* **18**, 431-492 (1989).
2. P. Kollman. Free energy calculations: Applications to chemical and biochemical phenomena. *Chem. Rev.* **93**, 2395-2417 (1993).
3. D.A. Pearlman & B.G. Rao. Free energy calculations: Methods and applications. In *Encyclopedia of Computational Chemistry*, P. von R. Schleyer, N.L. Allinger, T. Clark, J. Gasteiger, P.A. Kollman & H.F. Schaefer, III, Ed. Chichester: John Wiley, (1998). pp. 1036-1061.
4. T. Simonson. Free energy calculations. In *Computational Biochemistry and Biophysics*, O. Becker, A.D. MacKerell, B. Roux & M. Watanabe, Ed. New York: Marcel Dekker, (2001).
5. D.A. Pearlman & P.A. Kollman. A New Method for Carrying Out Free Energy Perturbation Calculations: Dynamically Modified Windows. *J. Chem. Phys.* **90**, 2460-2470 (1989).
6. D.A. Pearlman & P.A. Kollman. The overlooked bond-stretching contribution in free energy perturbation calculations. *J. Chem. Phys.* **94**, 4532-4545 (1991).
7. D.A. Pearlman. Determining the contributions of constraints in free energy calculations: Development, characterization, and recommendations. *J. Chem. Phys.* **98**, 8946-8957 (1993).
8. D.A. Pearlman, D.A. Case, J.W. Caldwell, W.S. Ross, T.E. Cheatham, III, S. DeBolt, D. Ferguson, G. Seibel & P. Kollman. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Comp. Phys. Commun.* **91**, 1-41 (1995).
9. D.A. Pearlman. Free energy derivatives: A new method for probing the convergence problem in free energy calculations. *J. Computat. Chem.* **15**, 105-123 (1994).
10. D.A. Pearlman. A comparison of alternative approaches to free energy calculations. *J. Phys. Chem.* **98**, 1487-1493 (1994).
11. C. Chipot, P.A. Kollman & D.A. Pearlman. Alternative approaches to potential of mean force calculations: free energy perturbation versus thermodynamics integration. Case study of some representative nonpolar interactions. *J. Comput. Chem.* **17**, 1112-1131 (1996).
12. R.J. Radmer & P.A. Kollman. Free energy calculation methods: A theoretical and empirical comparison of numerical errors and a new method for qualitative estimates of free energy changes. *J. Computat. Chem.* **18**, 902-919 (1997).
13. G. Hummer. Fast-growth thermodynamic integration: Error and efficiency analysis. *J. Chem. Phys.* **114**, 7330-7337 (2001).
14. S.H. Fleischman & C.L. Brooks, III. Thermodynamic calculations on biological systems: Solution properties of alcohols and alkanes. *J. Chem. Phys.*

15. **87**, 221-234 (1988).
H.-A. Yu & M. Karplus. A thermodynamic analysis of solvation. *J. Chem. Phys.*
89, 2366-2379 (1988).