Evaluation routine for automated parameterization

See: Per–Ola Norrby, Tommy Liljefors, J. Comput. Chem. ***1998****, 1146.*

ParEval [–c][–f][–v][–z][–x*list*|–r*list*]

[–a|–d*list*|–m*list*|–s|–t*list*] *results*

list

A list of values. This can *either* be given as a comma–separated list on the command line, in which case there must be no spaces in the list or between option and list, *or* as the name of a file containing the values in white–space separated free format. If a file name is given, it must be separated from the option by a space, and must not begin with a hyphen (–).

results

The tabulator–separated file containing all force field results. The file consists of tabulator–delimited rows with the following format: “*scale reference* $calc_0$ $calc_1$ … ”. The columns contain, in order, scaling factors, reference values, and force field results for different parameter sets. The default mode of operation is to accumulate and output the merit function for all force fields. The scale factors can optionally be preceded by another column of data identification labels, which are ignored.

–a | –d*list* | –m*list* | –s | –t*list*

For each row of the *results* file, the derivatives are calculated using forward differentiation (unless –c is also specified). The derivatives ($calc_n$–$calc_0$) are calculated, and the design matrix **A** and the residual vector **b** are accumulated. When –a is given, **A** and **b** are simply output. For the remaining options, **A** is modified using the values in *list,* the resulting matrix **A’** is inverted, and solve for parameter modifications $\mathbf{x}=(\mathbf{A'})^{-1}\mathbf{b}$. For –d, treat the values in *list* as dampening factors t o be added to the diagonal elements of **A**. When –m is specified, the diagonal elements are multiplied by the factor (1+), us ing *list* as Levenberg–Marquardt f actors. Solution by SVD is invoked by –s or –t, with all singular values below a threshold dropped before backsubstitution. With –s, all singular values are used as threshold values in turn, whereas with –t, the threshold values are found in *list.*

–c

Use central differentiation. The derivatives are 0.5*($calc_{2n-1}$–$calc_{2n}$). The first and second absolute derivatives of the merit function are also calculated. Parameter modifications are estimated from one–dimensional Newton–Raphson.

–f

Output is formatted to a more narrow column format, instead of the default line format.

–x*list*

Use a cutoff to skip output of trial solutions for which the total radius of the parameter change (the square root of the sum of squares of all changes) falls outside the limits given in *list*. Note that the parameter modifications Δp_j are given in units of the numerical differentiation step, making the radius dimensionless. The default is to output all solutions where 0.01<radius<100. If two values are given, the smaller is taken as the lower limit. If only one value is given, the inverse is used as the second limit. If no *list* is given, the cutoff function is turned off.

–r*list*

The *list* should only contain one value, the thrust radius. When the total radius of the parameter change exceeds , pa rameter modifications are scaled to make the final radius equal to (s ee –x). The default is to use a cutoff instead. If no *list* is given, cutoff and scaling is turned off.

–v

Verbose mode, generate diagnostic output.

–z

Analysis mode, outputs a list of errors and squared errors, for sorting.

Parameter modification routine for automated parameterization

See: Per-Ola Norrby, Tommy Liljefors, J. Comput. Chem. ***1998****, 1146.*

ParMod { -n *parlist* | -p *parlist* | -m *parlist* | -d[c][*num*] *parlist* |
    -s *solfile parlist parlist* [-l*list*] }
    [-i *file*] [-r *root*] [-o *file*] [-*num*] [-v] [-z]

Normally, output is sent to standard output unless -o is specified. There are two exceptions: -m and -s each generate multiple output files. For these, output file names are generated by incrementing a numerical extension and adding it to a root file name (see -r and -*num*). The input force field file name is by default “mm3.fld”, but can be changed by -i.

parlist

The name of a file containing parameter specification, one parameter per line. A parameter specification must obey the following format: “*row position value [const] [%format] ...* ”. *Row* and *position* must be integers, whereas *value* and the optional positive *const* (see -d) are floating point values. Additional text on the row, or rows that cannot be interpreted, are treated as comments. All parameters are constrained to be positive, unless *position* is negative. *Row* and the absolute value of *position* are used to identify the parameter to be modified within the force field file. If *position* is 1-3, it corresponds to one of the three standard positions within a Macromodel force field file, otherwise gives explicitly the starting column for the parameter. The standard output format %9.4f is assumed (9 positions, 4 decimals), unless a *format* is explicitely given. If *format* is not a valid C format for one floating point value, an error will occur. When *parlist* is used as a template for a new parameter list, only valid input is copied, that is, comments are lost.

-n *parlist*

A sorted parameter list with the current values from the input force field is sent to normal output, using *parlist* as a template. The output list can be used for parameter tethering.

-d[c][*num*] *parlist*

Generates and outputs a differentiated, sorted parameter list from *parlist*. If *const* is present in parlist, it is used as the increment. If the optional c is given, two parameter rows are output for each input parameter, first an increment and then a decrement (for use with central differentiation). If two comma-separated numbers $_1$ and $_2$ are input as *num,* the increment is calculated as max(param* $_1$, $_2$) (overriding *const*). If only one number is given, it is used both for $_1$ and $_2$. In the absence of *const* and *num*, $_1$= $_2$=0.01.

-p *parlist*

Outputs a force field file using all parameter values found in *parlist*.

-m *parlist*

For each parameter value in *parlist*, one new force field file is generated where only this parameter has been substituted. Output by default goes to "mm3.*n*", where *n* starts at 1 and is incremented by 1 for each parameter. The root file name is changed by -r, whereas a new starting value for *n* can be input as -*num*.

-s *solfile parlist1 parlist2*

Solfile contains a number of trial solutions generated by ParEval, *parlist1* are the initial parameters, and *parlist2* are the incremented values for numerical differentiation. Trial solution rows in *solfile* are identified as those containing the text sequences "max=" and "min=". The subsequent numbers are read, and one force field file is generated for each row. Output by default goes to "par.*n*", where *n* starts at 1 and is incremented by 1 for each parameter. The root file name is changed by -r, whereas a new starting value for *n* can be input as -*num*. It is an error if the number of parameters in the input parameter lists do not correspond to each other and to the number of values found in the trial solution.

-l*list*

List is a comma-separated list of scale factors (no spaces). For each solution generated by –s, scale the step by the values in *list* and generate one force field for each value. The total number of force fields generated is the number of values in list times the number of trial solutions in *solfile* (see -s).

-i *file*

Use *file* as the input force field (default: "mm3.fld").

-o *file*

Send normal output to *file* instead of to standard output.

-r *root*

Use *root* as the file name root when generating output file names for –m and –s. Default is "mm3." for –m, "par." for –s (notice the period). See -*num* for the extension used.

-*num*

Use the integer *num* as the initial extension when generating output file names for –m and –s. The extension is incremented by 1 for each new output file. Default: 1.

-v

Verbose mode, generate diagnostic output.

-z

Zero unallowed negative parameters instead of deleating the file.

Utility routine for automated parameterization

See: Per–Ola Norrby, Tommy Liljefors, J. Comput. Chem. ***1998****, 1146.*

ParUtil {–b *file*|–g *file*|–a *file*|–j *file*|–l *file*} [–s*list*] [–f*list*] [–w[*list*]] [–2] [–r*num*] [–t*num*] [–o *outfile*] [–v] [–4 *file1*] [–e[{r|g|s}] *file*] [–p *file1 file2*] [–i *label*]

The utility will Boltzmann average conformational energies or convert energy derivatives between Gaussian94, Spartan, Eigensystems, and the parameterization format. Normal operation is to read input from *file* and send results to standard output. An input file name is required, but can be "–" (for standard input). Numerical options are given as *num* (one number) or *list* (a comma–separated list of numbers). There must be no spaces between the option and *num* or *list,* or within *list.*

–b *file*

The file is assumed to contain groups of steric energies for Boltzmann averaging. Initial white space on rows is always ignored. Groups are separated by rows starting with "***". Within a group, two rows are used for each conformer, one for an identifier and the subsequent row for the energy of that conformer (there can be multiple columns with energies from different force fields). In the identifier row, only the first character is significant. All conformers with the same identifier character will be combined in the denominator, the remaining will go to the numerator. Output is one relative free energy for each group of input conformers.

–g *file*

Convert Cartesian energy derivatives in a formatted Gaussian94 checkpoint file to a file with one column of scale factors and one column of derivatives for use as reference values in the automated parameterization, mass–weighted if used with option –w.

–a *file*

As –g, but reads a Spartan archive file. Implies –2.

–j *file*

As –g, but reads a Jaguar restart–file (*.01.in). Implies –2.

–l *file*

As –g, but reads the archive entry of a Gaussian94 log-file.

–i *label*

Causes a column with labels (*label* and cart. coord.) to be added to the param. file.

–f*list*

The *list* must contain two comma–separated floating point values (used in double precision) for converting first and second energy derivatives from Gaussian94 units (Hartree and Bohr) to force field units (kJ/mol and Å in MacroModel). Used by –g and –d. The default values are: “4961.475515,9375.829222”.

–s*list*

The *list* must contain two comma–separated scale factors. The values are output in column one, separated by tabs from the first or second energy derivatives, respectively.

–w*list*

Use mass–weighting and input extra atomic weights. The *list* must contain comma–separated pairs of atom numbers (integer, between 0 and 200) and atomic weights (floating point). Many masses are known.

–2

Calculate second derivatives only. Two values must still be input if –f or –s are specified.

–r*num*

New value for the gas constant (if input to –b is not in kJ/mol). Default: R=8.3143*10^{-3}.

–t*num*

New temperature for Boltzmann averaging (in K). Default: T=298.15.

–o *outfile*

Send normal output to *outfile* instead of to standard output.

–v

Verbose mode, generate diagnostic output.

–e

Two possible modes: if a single –e is used as an option, the eigenvalues and eigenvectors of the input Hessian (from –a or –g) are output, together with the reduced masses if mass–weighted (–w). If the option instead is –er, –eg, or –es, the eigensystem is read and output as a Hessian, in parameterization, Gaussian94, or Spartan format, respectively.

–4 *file*

The *file* should contain rows with two integers and a floating point value. The integers are interpreted as a pair of atoms for which the scale factor should be multiplied by the floating point value. This can be used to increase the scaling weight for atoms in a 1–4 relationship.

–p *file1 file2*

Paste the files, like Unix paste, but does not crash on long rows (can, however, misbehave if input tab-separated columns are wider than 1000 characters.