

# Winding Journey Down (in Gibbs energy)

Jeffrey M. Dick

June 16, 2012

## 1 Introduction

`wjd()` appeared in CHNOSZ version 0.9-8. It implements the steepest descent algorithm for Gibbs energy minimization described by White, Johnson and Dantzig, 1958. This vignette shows some examples of its usage.

The arguments of `wjd()` have default values to simulate the example problem given by White et al., 1958. In their words, “The example taken is the determination of the composition of the gases arising from the combustion of a stoichiometric mixture of hydrazine and oxygen at 3500 °K and a pressure of 750 psi ( $\ln P_{\text{atm}} = 3.932$ ).” There are ten species included in this example. The value of temperature is not explicitly considered by the algorithm, or in the arguments to `wjd()`. Instead, the standard Gibbs energies of the species, at the given temperature, are provided (in dimensionless form, i.e.  $\Delta G^\circ / RT$ ). Note that White et al., 1958 use the terminology “free energy” and the notation  $F^\circ$ . The term “Gibbs energy” and corresponding notation is used here.

The user is encouraged to read the help page for `wjd()` in the CHNOSZ documentation for details of the implementation that are not covered in this vignette. See also Nordstrom et al. [1979] for a review of alternative algorithms, especially for aqueous systems, which are not the focus of the functions described here.

## 2 Is it winding?

The “winding” in the title refers to the observation that the abundances of species during a Gibbs energy minimization often do not all change in a monotonic fashion. This is because at each iteration a new direction of steepest descent is calculated; this direction is a vector of changes of mole numbers of all species in the system and is subject to mass balance constraints as well as the steepest-descent criterion.

Let’s run `wjd()` with its default settings and save the output.

```
> w <- wjd()
```

What are the most abundant species, and how many iterations were taken?

```
> # the order of species abundance
> oX <- order(w$X, decreasing=TRUE)
> # the stoichiometries of the two most abundant species
> w$A[head(oX,2),]
```

```
      H N O
[1,] 2 0 1
[2,] 0 2 0
```

```
> # the number of iterations
> niter <- length(w$lambda)
> niter
```

```
[1] 6
```

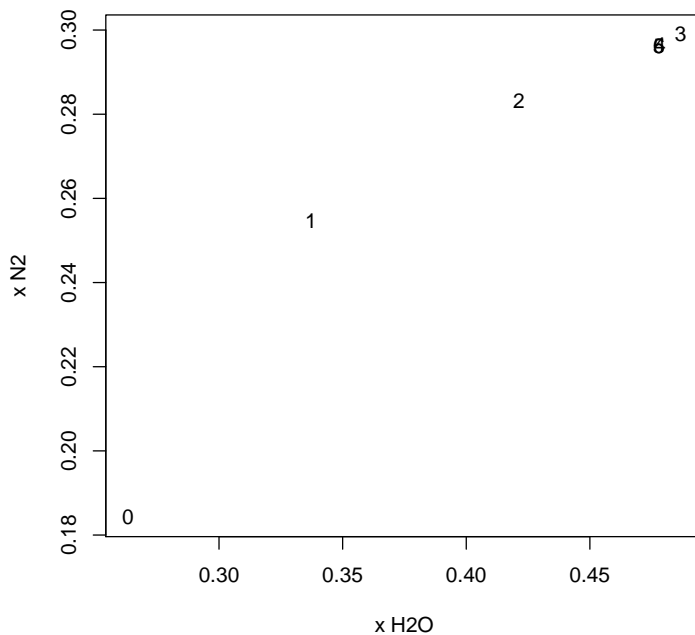
We find that H<sub>2</sub>O and N<sub>2</sub> are the most abundant species, after 6 iterations.

Let's track their mole fractions through the iterations. Write a function that returns the mole fractions of these two species, after a specified number of iterations.

```
> iterfun <- function(imax) {  
+   w <- wjd(imax=imax)  
+   x1 <- w$X[oX[1]]/sum(w$X)  
+   x2 <- w$X[oX[2]]/sum(w$X)  
+   return(list(x1=x1, x2=x2))  
+ }
```

Then apply the function over the different numbers of iterations, from 0 (initial conditions) to 6, and plot the values.

```
> sa <- sapply(0:niter, iterfun)  
> plot(t(sa), pch=as.character(0:niter), xlab=paste("x",f1), ylab=paste("x",f2))
```



A bit of winding: the mole fractions of these two species increase up to the third iteration, but at the fourth (and less so for the fifth and sixth), their mole fractions decrease. This behavior is consistent with a decrease in Gibbs energy at *every* iteration; that can be verified by inspecting the values in the 6-iteration result:

```
> diff(w$F.Y)  
[1] -9.001450e-01 -1.690993e-01 -2.702223e-02 -1.348404e-03 -9.581509e-05  
[6] -3.417140e-06
```

The decrease in Gibbs energy becomes smaller with every iteration, as an equilibrium distribution of species is approached.

### 3 Is it near the bottom? Testing for equilibrium

#### 3.1 Change in Gibbs energy of the system becomes small

Equilibrium is synonymous with a global Gibbs energy minimum. Becoming convinced that the output from `wjd()` represents a near-equilibrium condition can be difficult. One type of observation that can be helpful is the amount of change during the iterations of the algorithm. As equilibrium is approached, it makes sense that the fractional decreases in Gibbs energy would become smaller and smaller. This is used as a stopping condition in the current implementation – if the fractional change, relative to the total energy of the system, reaches the value of `Gfrac` given in the arguments the iterations stop. Here are the fractional changes with each iteration:

```
> diff(w$F.Y)/w$F.Y[1:6]
[1] 1.929006e-02 3.555210e-03 5.661133e-04 2.823296e-05 2.006125e-06
[6] 7.154610e-08
```

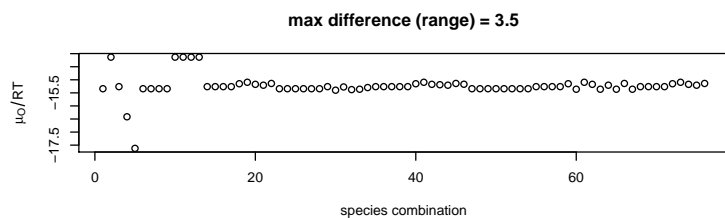
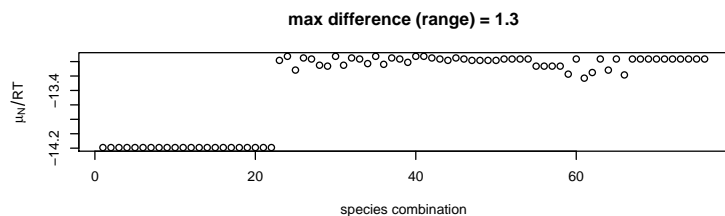
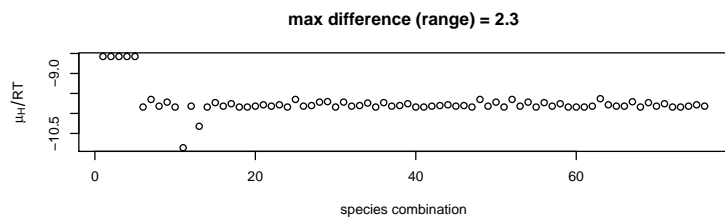
Only the last value is below the default value of `Gfrac` in `wjd()`.

#### 3.2 Chemical potentials of the elements (from different species representing them) become more uniform

Equilibrium is also synonymous with uniformity of chemical potentials ( $\mu$ ) of the thermodynamic components throughout the system. For the default system in `wjd()`, and perhaps other systems of interest, the elements themselves are an allowable definition of the components. Often, the number and compositions of the species are such that multiple combinations of species satisfying certain stoichiometric conditions (i.e., eligible to be a basis set) can be used to compute the chemical potentials of the elements. At equilibrium, these different combinations of species would all yield the same chemical potentials of the elements.

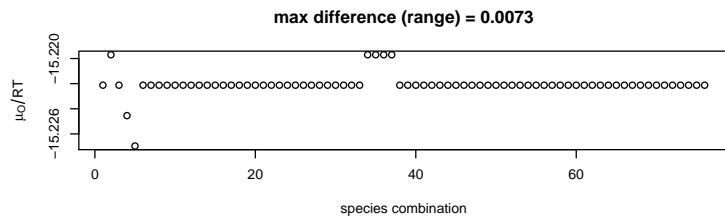
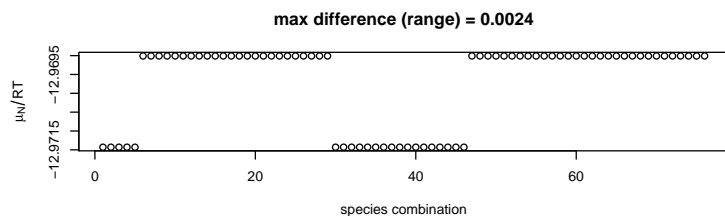
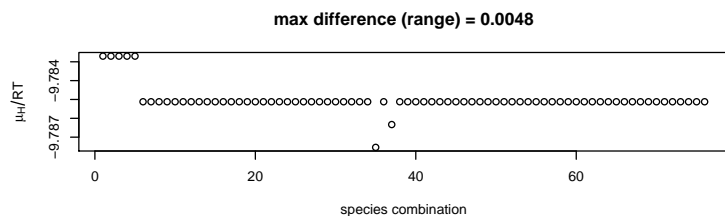
There is a supporting function, `element.potentials()`, that computes the chemical potentials of the elements using different eligible combinations of species. It has an option to plot the results. Let's first look at a plot showing the results after 3 iterations instead of 6.

```
> w3 <- wjd(imax=3)
> ep3 <- element.potentials(w3, plot.it=TRUE)
```



Here's the plot for the default settings of `wjd()`:

```
> ep <- element.potentials(w, plot.it=TRUE)
```



That one shows considerably less deviation than the first plot. The differences don't become zero, but perhaps they are small enough to accept as an operational solution. There is a function, `is.near.equil()`, that provides a logical type result based on the range of chemical potentials of the elements calculated from

the different eligible species combinations. It says “TRUE” if the maximum difference for any element is below  $\mu/RT = 0.01$ , but this limit can be changed through the `tol` argument to the function. With the default setting, `is.near.equil()` does give a different answer for the 3- and 6-iteration results:

```
> is.near.equil(w3)

[1] FALSE

> is.near.equil(w)

[1] TRUE
```

### 3.3 Different initial solutions yield similar results

Although it seems to be the case in the examples shown so far, increasing the number of iterations does not necessarily bring one closer to true equilibrium, since being trapped in a local energy minimum is a possibility. Sometimes local minima can be shown to exist by starting with different mole numbers of species, that still yield the same bulk composition (mole numbers of elements). `guess()` exists to facilitate this sort of investigation. Its input is the stoichiometric matrix of species in the system (the default refers to the same system as in `wjd()`) and the bulk composition; it outputs one set or a series of sets of mole numbers of species that satisfy the bulk composition constraints. In general, this is an underdetermined problem, so there are usually an infinite number of possible solutions. For the methods used in the function to generate the guesses, refer to the help page and the comments in the function code itself. For now let's compare the guess made by `guess()` with the default species composition in `wjd()` (that is taken from White et al., 1958).

```
> as.list(args(wjd))$Y

c(0.1, 0.35, 0.5, 0.1, 0.35, 0.1, 0.1, 0.1, 0.1, 0.1)

> Y1 <- guess()
> Y1

[1] 0.012 0.002 0.990 0.992 0.002 0.002 0.002 0.002 0.002 0.002
```

Now let's run the minimization using the first guess, count the number of iterations, and test it for equilibrium and also make a plot showing the chemical potentials of the elements.

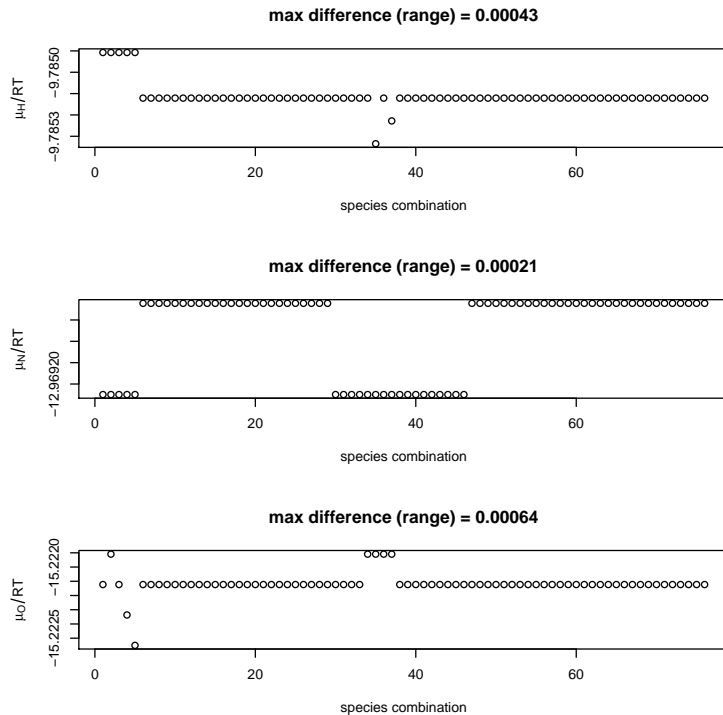
```
> wY1 <- wjd(Y=Y1)
> niterY1 <- length(wY1$lambda)
> niterY1

[1] 9

> is.near.equil(wY1)

[1] TRUE

> epY1 <- element.potentials(wY1, plot.it=TRUE)
```



Whoa, those differences are a lot smaller than the chemical potentials of the elements resulting from the default initial solution of mole numbers of species! Also note that there were 3 more iterations than using the default solution in `wjd()`.

Not all guesses lead to the same conclusion. Let's make a simple test for closeness to equilibrium of all of the guesses available from `guess()` for the default set of species and bulk composition. First generate all of the guesses. Note that `guess()` returns NA for compositionally eligible combinations of species that have some negative mole numbers, so we filter those out.

```
> Ys <- guess(iguess=NULL)
> # total number of species combinations
> length(Ys)

[1] 76

> # species combinations that have all-positive mole numbers
> iYs <- which(!is.na(Ys))
> nguess <- length(iYs)
> nguess

[1] 39
```

Now let's run the minimization using each of those guesses, and test if each one `is.near.equil()`:

```
> sapply(iYs,function(i) is.near.equil(wjd(Y=Ys[[i]])))

[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Looks like they are. What if we're really picky though and want to make sure the chemical potentials of the elements are very well-defined? We can decrease the tolerance of `is.near.equil()`:

```
> sapply(iYs,function(i) is.near.equil(wjd(Y=Ys[[i]]),tol=0.0001))
```

```
[1] FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
[13] FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE FALSE FALSE TRUE
[25] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE
[37] TRUE FALSE FALSE
```

Well some of the tests failed. But maybe this is because we have stopped iterating too soon. Let's iterate until a we have smaller change in Gibbs energy of the system:

```
> sapply(iYs, function(i) {
+   is.near.equil(wjd(Y=Ys[[i]], Gfrac=1e-9), tol=0.0001)
+ })

[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[25] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[37] TRUE TRUE TRUE
```

We're back to passing all of the equilibrium tests based on uniformity of chemical potentials of the elements. What is the difference between the resulting species abundances for the first and last ones, relative to the species abundances in the first?

```
> wYfirst <- wjd(Y=guess(iguess=1), Gfrac=1e-9)
> wYlast <- wjd(Y=guess(iguess=nguess), Gfrac=1e-9)
> (wYfirst$X - wYlast$X) / wYfirst$Y

[1] 2.399562e-10 1.440403e-09 -6.412828e-12 3.921580e-10 -9.608847e-08
[6] -6.533836e-11 -2.259359e-09 8.490986e-10 1.246336e-09 2.090106e-09
```

Based on this exercise, starting from different initial species abundances for the same bulk composition, and arriving at a similar near-equilibrium solution, we can be fairly confident that we haven't gotten stuck in a local minimum.

(In fact, for ideal reactions in closed chemical systems there is only a unique solution [Othmer, 1976]. Therefore, the algorithm should converge on the same result for any initial vector of species mole numbers with the same bulk composition.)

## 4 Running Down: Using a Thermodynamic Database

So far the examples shown here have been based on the chemical system defined by the default settings for the arguments to the functions. What if we're interested in a different system? It can be rather tedious to manually construct the stoichiometric matrices and vectors of standard Gibbs energies of the species. The function `run.wjd()` pulls the compositional and thermodynamic data for requested species from the thermodynamic database and then calls `wjd()`.

### 4.1 At a single temperature

Let's start by finding the indices in the thermodynamic database of some liquid alkanes and aromatic compounds:

```
> alkanes <- c("n-hexane", "n-heptane", "n-octane", "n-nonane")
> ialk <- info(alkanes, "liq")
> aromatics <- c("naphthalene", "anthracene", "phenanthrene", "pyrene")
> iaro <- info(aromatics, "liq")
```

Let's find the equilibrium distribution of species for a bulk composition corresponding to a single mole of each species.

```
> waa <- run.wjd(c(ialk, iaro), Y=rep(1, 8))
```

subcrt: 8 species at 298.15 K and 1 bar

```
> waa$elements
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]
C	84	84	84	84	84	84	84	84	84	84	84
H	106	106	106	106	106	106	106	106	106	106	106

```
> is.near.equil(waa)
```

```
[1] FALSE
```

The `waa$elements` shows that the bulk chemical composition in the Gibbs energy minimization was  $C_{84}H_{106}$  ( $H/C = 1.261905$ ). That FALSE means the chemical potentials of the elements calculated from different combinations of species differ by more than 0.01, the default setting of `tol` in `is.near.equil()`. We can make it TRUE by running more iterations:

```
> waa <- run.wjd(c(ialk, iaro), Y=rep(1, 8), imax=12, Gfrac=1e-14)
```

subcrt: 8 species at 298.15 K and 1 bar

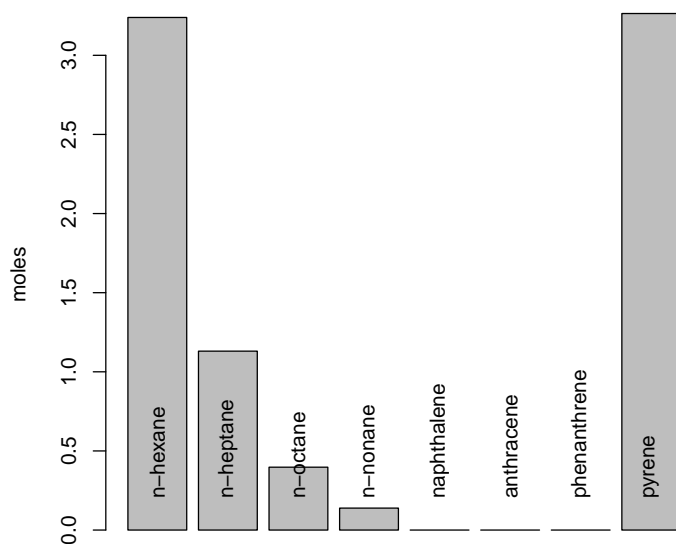
```
> is.near.equil(waa)
```

```
[1] TRUE
```

Let's plot the abundances of the species:

```
> bp <- barplot(waa$X, ylab="moles")
```

```
> text(bp, rep(0.2,8), c(alkanes, aromatics), srt=90, adj=0)
```



Now let's calculate the chemical potentials of the elements (dimensionless values,  $\mu/RT$ ) ...



```
> ep <- equil.potentials(waa)
> print(ep)
```

```
      C      H
9.896363 -4.406116
```

```
> basis(c("graphite", "H2"), c("cr", "gas"))
```

```
  C H ispecies logact state
C  1 0      1937      0    cr
H2 0 2      3090      0    gas
```

```
> basis.logact(ep)
```

```
      C      H2
0.007254100 -0.006459425
```

... and the corresponding logarithms of chemical activities of a set of basis species:

```
> basis(c("graphite", "H2"), c("cr", "gas"))
```

```
  C H ispecies logact state
C  1 0      1937      0    cr
H2 0 2      3090      0    gas
```

```
> basis.logact(ep)
```

```
      C      H2
0.007254100 -0.006459425
```

## 4.2 Bulk composition instead of moles of species

run.wjd() can be given B (bulk chemical composition), instead of Y (initial numbers of moles of species). When this happens, the function tests the different initial compositions generated by guess() until one is found that is near equilibrium. Here's an example:

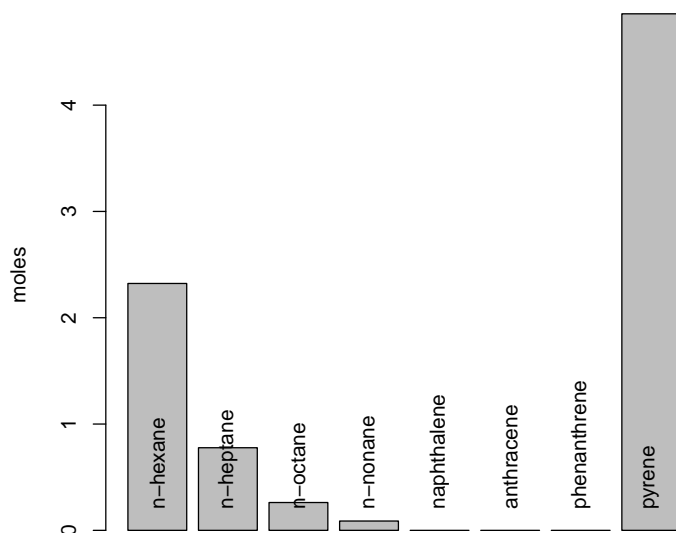
```
> waa <- run.wjd(c(ialk, iaro), B="C100H100")
```

```
subcrt: 8 species at 298.15 K and 1 bar
```

```
run.wjd: got apparently near equilibrium on initial solution 4 of 19
```

```
> bp <- barplot(waa$X, ylab="moles")
```

```
> text(bp, rep(0.2,8), c(alkanes, aromatics), srt=90, adj=0)
```



Compared with the previous example, here the lower  $H/C = 1$  defines a more oxidized system, so the increase in aromatic content is expected.

## 5 Document Information

### Revision history

- 2012-01-01 Initial version
- 2012-06-16 Running Down using alkanes and aromatics

### R session information

## References

- D. K. Nordstrom, L. N. Plummer, T. M. L. Wigley, T. J. Wolery, J. W. Ball, E. A. Jenne, R. L. Bassett, D. A. Crerar, T. M. Florence, B. Fritz, M. Hoffman, Jr. Holdren, G. R., G. M. Lafon, S. V. Mattigod, R. E. McDuff, F. Morel, M. M. Reddy, G. Sposito, and J. Thrailkill. A comparison of computerized chemical models for equilibrium calculations in aqueous systems. In E. A. Jenne, editor, *Chemical Modeling in Aqueous Systems: Speciation, Sorption, Solubility, and Kinetics*, Series 93, pages 857 – 892. American Chemical Society, 1979. URL <http://www.worldcat.org/oclc/4591313>.
- H. G. Othmer. Nonuniqueness of Equilibria In Closed Reacting Systems. *Chemical Engineering Science*, 31 (11):993–1003, 1976. doi: 10.1016/0009-2509(76)87020-0.
- W. B. White, S. M. Johnson, and G. B. Dantzig. Chemical equilibrium in complex mixtures. *J. Chem. Phys.*, 28(5):751 – 755, 1958. ISSN 0021-9606. doi: 10.1063/1.1744264.