forward_model_explicit

October 14, 2013

1 Forward eco-physiological modelling of ΔO_{18} in tree rings

A small function to convert degrees Celsius to Kelvin

```
In [1]: def C2K(T):
            """
            conversion celsius to Kelvin
            """
            return T + 273.16
```

1.0.1 fixed parameters

Some things to play with ?

```
In [2]: gs = 0.25 # ??? What is gs ?
```

```
In [3]: leaf_width =  0.015
```

```
In [4]: PAR = 925.
```

Constants for the Craig / Gordon parameters

```
In [5]: d_source_H2O = -5.17
```

```
In [6]: fract_through_stomata = 32
```

```
In [7]: fract_through_boundary_layer = 28
```

Constants related to the Peclet effect

```
In [8]: eff_length = 0.0077
```

```
In [9]: C = 55.5 * 1000. # ??? why that ?
```

Constants for calculations of Δ cellulose and Δ leaf

```
In [10]: C_O_fract = 27
```

```
In [11]: Dcel_Dom = 9
```

```
In [12]: prop_exc = 0.45
```

```
In [13]: prop_Xylem = 0.56
```

1.0.2 Environmental inputs

```
In [14]: airtemp = 14.2666666667
         windspeed = 5.4
         rh = 84.4
         pressure = 1015.3333333333
         gs = 0.25
         leaf_width = 0.015
         PAR = 925
```

1.0.3 Energy balance calculations

```
In [15]: rs = 1. / gs

Out[15]: 4.0

In [16]: r_times_b = 3.8 * (leaf_width**0.25)*(windspeed**(-0.5))

Out[16]: 0.5722812689529017

In [17]: rb = 0.89 * r_times_b

Out[17]: 0.5093303293680825

In [18]: gr = (4*0.98*(0.000000056703)*(C2K(airtemp)**3))/(29.2)

Out[18]: 0.18075504491562489

In [19]: rBH = 1./((1./r_times_b)+gr)

Out[19]: 0.5186325082604192

In [20]: Qtot = (PAR/4.6)*2

Out[20]: 402.1739130434783

In [21]: Qabs = 0.5 * Qtot

Out[21]: 201.08695652173915
```

1.0.4 Calculating ϵ

```
In [22]: lesstemp = airtemp - 1.

Out[22]: 13.2666666667

In [56]: estemp = (6.13753 * exp(lesstemp * ((18.564 - (lesstemp/254.4)))/(lesstemp +255.57)))*100

In [57]: lesstemp_K = C2K(lesstemp)

In [58]: s = (((6.13753 * exp(airtemp * ((18.564 - (airtemp/254.4)))/(airtemp +255.57))))-estemp)/(C2K(a

Out[58]: -1513.8241096132349

In [59]: smbar = 6.13753*(((airtemp+255.7)*(18.564 - (2*airtemp/254.4)) - airtemp*(18.564 - \
                     (airtemp/254.4)))/((airtemp+255.57)**2))*(exp(airtemp*(18.564 - \
                                                     (airtemp/254.4))/(airtemp + 255.57)))

In [60]: epsilon = (smbar*44012)/(29.2*(pressure))
```

1.0.5 Calculating $\frac{EA}{EI}$

```
In [28]: ea = (rh / 100) * (6.13753 * exp(airtemp * ((18.564 - (airtemp/254.4)))/(airtemp +255.57)))
```

```
Out[28]: 13.781965981701569
```

```
In [29]: es = (6.13753 * exp(airtemp * ((18.564 - (airtemp/254.4)))/(airtemp +255.57)))
```

```
Out[29]: 16.329343580215127
```

```
In [30]: D = (((6.13753 * exp(airtemp * ((18.564 - (airtemp/254.4)))\
                             /(airtemp +255.57))))-ea)/pressure
```

```
Out[30]: 0.0025089076807422602
```

```
In [31]: temp_diff = (rBH*((Qabs*(rs+rb))-(44012*D)))/(29.2*(rs+rb+(epsilon*rBH)))
```

```
Out[31]: 2.6567603450679336
```

```
In [32]: leaf_temp = airtemp + temp_diff
```

```
Out[32]: 16.923427011767934
```

```
In [33]: ei = (6.13753 * exp(leaf_temp * ((18.564 - (leaf_temp/254.4)))\
                            /(leaf_temp +255.57)))
```

```
Out[33]: 19.360275561858991
```

```
In [34]: leaf_temp_K = C2K(leaf_temp)
```

```
Out[34]: 290.08342701176798
```

```
In [35]: ea_ei = ea / ei
```

```
Out[35]: 0.71186827572087574
```

1.0.6 Calculating transpiration

```
In [36]: transpiration = (epsilon * rBH * Qabs / 44012. + D) \
         / (rs + rb + epsilon * rBH)
```

```
Out[36]: 0.0011702780786241177
```

1.0.7 Craig / Gordon parameters

```
In [37]: d_water_vapour = d_source_H2O + -1*(2.644-3.206*(1000/C2K(airtemp))+\
                                            1.534*(1000000/(C2K(airtemp)*C2K(airtemp))))
```

```
Out[37]: -15.228112002230901
```

```
In [38]: ek = ((fract_through_stomata*1/gs)+(fract_through_boundary_layer*rb))/((1/gs)+rb)
```

```
In [39]: e_star = 2.644-3.206*((10**3)/leaf_temp_K)+1.534*((10**6)/(leaf_temp_K**2))
```

```
In [40]: dv = ((d_water_vapour/1000.)*(1+(d_source_H2O/1000))+(d_source_H2O/1000.))*1000.
```

```
In [41]: dv = ((d_water_vapour/1000)*(1+(d_source_H2O/1000))+(d_source_H2O/1000))*1000
```

```
In [42]: de = ek+e_star+((d_water_vapour-ek)*ea_ei)
```

1.0.8 Estimating the Peclet effect

```
In [43]: C = 55.5*1000 ## ?????
```

```
In [44]: D_Peclet = 0.000000119*(exp(-(637/(leaf_temp_K-137))))
```

```
In [45]: p_Peclet = (transpiration*eff_length)/(C*D_Peclet)
```

```
In [46]: DL = (de*(1-exp(-1*p_Peclet)))/p_Peclet
```

```
In [47]: dL = ((DL/1000)*(1+(d_source_H2O/1000))+(d_source_H2O/1000))*1000
```

1.0.9 Calculating Δ cellulose and Δ leaf

```
In [48]: D_sucrose = DL + C_O_fract
```

```
In [49]: D_cellulose = (DL*(1-(prop_exc*prop_Xylem)))+C_O_fract
```

```
In [50]: D_leaf = D_cellulose - Dcel_Dom
```

```
In [51]: d_sucrose = ((D_sucrose/1000)*(1+(d_source_H2O/1000))+(d_source_H2O/1000))*1000
```

```
In [52]: d_leaf = ((D_leaf/1000)*(1+(d_source_H2O/1000))+(d_source_H2O/1000))*1000
```

1.0.10 OUTPUT = ΔO_{18} in tree-rings cellulose

```
In [53]: OUTPUT = ((D_cellulose/1000)*(1+(d_source_H2O/1000))+(d_source_H2O/1000))*1000
```

```
In [54]: OUTPUT
```

```
Out[54]: 27.441235386168799
```

```
In [54]:
```