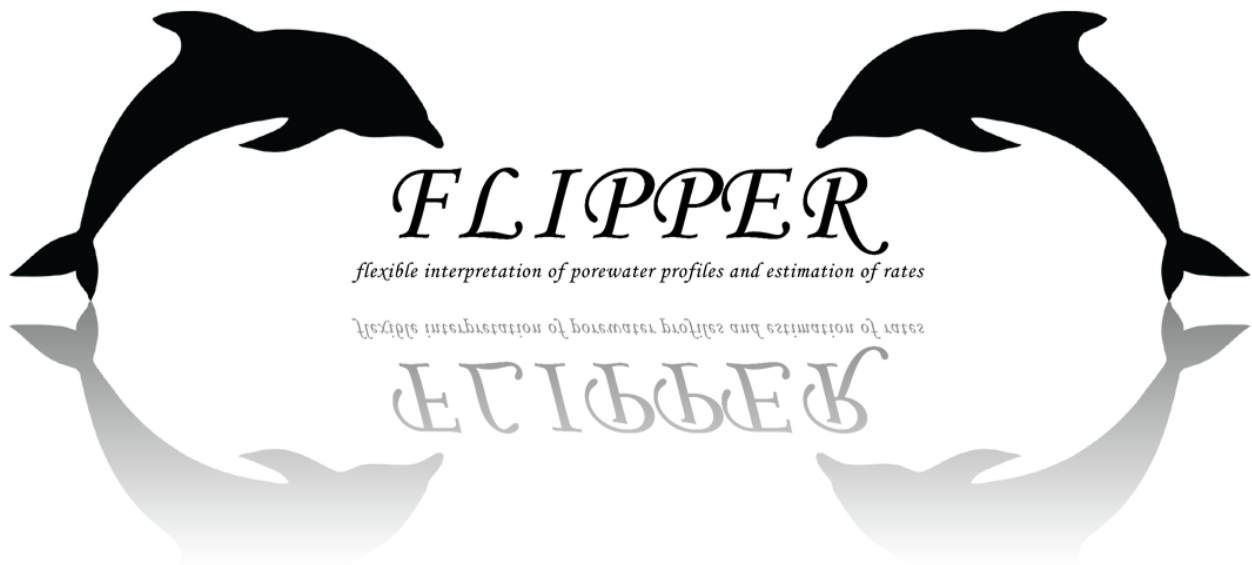


FLIPPER: Flexible Interpretation of Porewater Profiles and Estimation of Rates



Introduction

FLIPPER (short for: FLeXible Interpretation of Porewater Profiles and Estimation of Rates) has been created by Laurine Burdorf and myself to provide a flexible way to analyze porewater data in sediments, continuing from the initial work of (Berg, Risgaard-Petersen, and Rysgaard 1998). The goal is to be able to extract fluxes in/out of the sediment, and net production or consumption rates from a vertical porewater profile. FLIPPER provides three different ways to analyze porewater data:

- calculation of a diffusive flux
- fitting of a simple diagenetic profile (cfr (Berg, Risgaard-Petersen, and Rysgaard 1998))
- calculation of the first and second derivatives of a concentration profile

All three of these methods are combined in a simple R-function *FLIPPER_func*. In this manual, we will go through all three approaches and show how they can be calculated using FLIPPER.

Preparatory steps

Preparing your R environment

The first step is to source the two main functions

```
source("user interface function.R")
source("FLIPPER_plotfunction.R")
```

```
## Loading required package: marelac
```

```
## Loading required package: shape
```

```
## Loading required package: signal
##
## Attaching package: 'signal'
## The following objects are masked from 'package:stats':
##
##   filter, poly
## Loading required package: fractaldim
## Loading required package: abind
## Loading required package: tcltk
## Loading required package: ReacTran
## Loading required package: rootSolve
## Loading required package: deSolve
## Loading required package: FME
## Loading required package: coda
## Loading required package: wavelets
```

The FLIPPER function

We can take a quick first look at the syntax of the FLIPPER function (but we will go in more depth later)

```
FLIPPER.func(input,species,por.cte=NA,E.cte=NULL,tort.dep=1,method=NULL,env.parms=NULL,full.output=FALSE,
             # optional input for gradient
             gradient.parms=NULL
             # optional input for discrete
             discrete.parms=NULL,
             # optional input for continuous
             continuous.parms=NULL,
             )
```

We can see that FLIPPER requires **input** (your data) and **species** (which element are you analyzing), and can optionally be supplied with porosity (**por.cte**), electrical field (**E.cte**), a choice of tortuosity (**tort.dep**, more on that later), which method (**method**), a set of environmental parameters (**env.parms**), whether you want all possible output (**full.output**). Finally there is also the possibility to supply specific parameters for each of the three analysis methods (**gradient.parms**, **discrete.parms**, **continuous.parms**). Through a series of example analyses of porewater profiles, we will discuss the different types of input, how to use FLIPPER, and part of the theory behind FLIPPER.

Preparing a dataset

Now we want to load a dataset to work with

```
load("datasets to test/O2_testprofiles.RData")
```

This will load three different O2 profile, called *profile.A*, *profile.B* and *profile.C*. All three of them are dataframes, and have an identical structure. You can look at the top of profile.A

```
head(profile.A)
```

```
##           x      C.true      C por tort    J.true R.true      Ds
## 1 -0.245 0.2500000 0.2491410    1    1 7.651203      0 1.330373
## 2 -0.235 0.2499999 0.2519317    1    1 7.651203      0 1.330373
```

```
## 3 -0.225 0.2499997 0.2483376 1 1 7.651203 0 1.330373
## 4 -0.215 0.2499996 0.2515561 1 1 7.651203 0 1.330373
## 5 -0.205 0.2499994 0.2495664 1 1 7.651203 0 1.330373
## 6 -0.195 0.2499992 0.2512210 1 1 7.651203 0 1.330373
```

You will notice there are 8 different columns. These three oxygen profiles are artificially generated, which means we have applied a depth-dependent reaction rate, and calculated the concentration and transport profiles. Hence, the columns **C.true**, **J.true** and **R.true** are the simulated profiles. Afterwards we have added noise to the data (column **C**), and have generated an artificial oxygen profile. For the purpose of this exercise, we only have to look at columns **x** and **C**. Column **x** represents the depth (in centimeters) and column **C** the concentration (in mM). You can imagine this is data generated by analyzing oxygen in marine sediments using microsensor profiling. The units of FLIPPER are *m* (for **x**) and *mmol m⁻³* (for **C**). The first thing we have to do is to create a dataframe **input** with the right columns (and with the data in the right units), so it can be analyzed by FLIPPER.

```
input.profile <- as.data.frame(cbind(profile.A$C*1e3,profile.A$x*1e-2))
colnames(input.profile) <- c('C','x')
```

The resulting input dataframe then looks like this (negative depths are values in the watercolumn)

```
head(input.profile)
```

```
##           C           x
## 1 249.1410 -0.00245
## 2 251.9317 -0.00235
## 3 248.3376 -0.00225
## 4 251.5561 -0.00215
## 5 249.5664 -0.00205
## 6 251.2210 -0.00195
```

The next thing we have to tell FLIPPER are the environmental conditions in which we measured our porewater profiles. For our case, this would be 10.0 °C for the temperature (**TC**) and 35.0 for the salinity (**S**). So we make a list with those values.

```
env.parms <- list(TC=10.0,S=35.0)
```

And finally, we have to supply the porosity of the sediment. This can be done directly in the **input** dataframe, if we have a depth-dependent porosity profile, or as a constant value. Since the porosity will not vary much over the first 0.5 cm of the sediment, we can just supply a constant value.

```
por.cte <- 0.8
```

Now we have the essential data to analyze our porewater *O₂* profile. Let's start using FLIPPER.

The gradient method

We supply our **input** dataframe, **por.cte** value and **env.parms** list to FLIPPER, and indicate we are analyzing an oxygen profile (c("O2")) in the FLIPPER function). **tort.dep** indicates the preferred correction of the diffusion coefficient for sediment tortuosity, we will discuss this in depth later, for now we just keep the default value (1). The *plot.FLIPPER(test)* call will plot the result of the analysis.

```
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                     env.parms=env.parms,
                     method="gradient")
plot.FLIPPER(test)
```

Once we run FLIPPER via the action button below, a pop-up window will appear and ask us to select the two points between which we want to calculate the diffusive flux. To calculate a gradient, we need to fit a

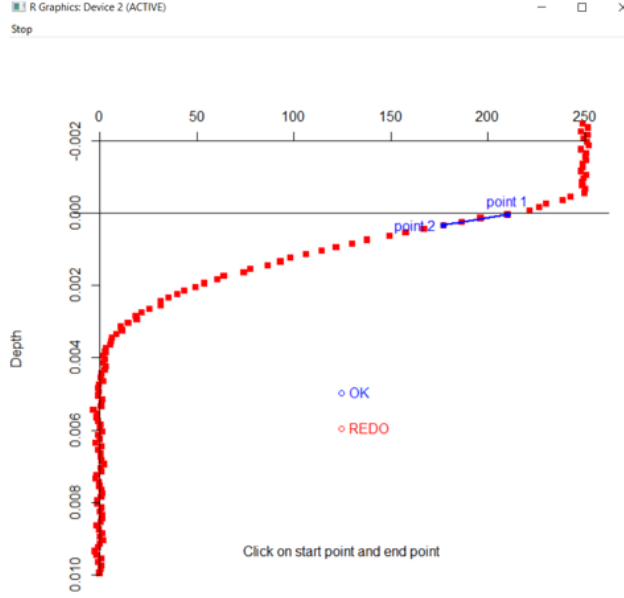


Figure 1: Selected points on gradient FLIPPER

linear model through the concentration versus depth plot. The slope of this fit then gives us the concentration gradient. To have a robust estimate of the slope, it is important to select at least 3 points (and preferably more) points. Let's first select the first four points on the concentration profile (so the four points closest to the sediment-water interface).

```
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                    env.parms=env.parms,
                    method="gradient")
plot.FLIPPER(test)
```

FLIPPER calculated for us a diffusive oxygen flux of $8.012 \text{ mmol m}^{-2} \text{ d}^{-1}$ (*R.int* on the plot, the minus indicates consumption; see below for a bit more discussion about the relation between *R.int* and fluxes). The red line indicates the linear fit through the selected points. The gradient method in FLIPPER calculates the diffusive flux following Fick's first law of diffusion (Fick 1855);

$$J_D = -\phi D_S \frac{\delta C}{\delta x}$$

from which you can see you need the porosity (ϕ), an effective diffusion coefficient (calculated by FLIPPER using the *diffcoef* function of CRAN:marelac from the supplied salinity and temperature) and the concentration gradient (calculated between the points you indicated on the plot, using the *lm* function from R). The effective diffusion coefficient in the sediment can be calculated as

$$D_S = \frac{D_0(S,T)}{\theta^2}$$

where θ^2 is the tortuosity. Tortuosity is a correction for the effective pathlength because in sediments, a dissolved compound cannot diffuse in straight way, because particles block its way (see (Boudreau 1997) for many more details). Tortuosity can be estimated from porosity. By default (if **tort.dep** = 1), the tortuosity is calculated as (Boudreau 1996)

$$\theta^2 = 1 - 2\ln(\phi)$$

Other options exist; $\theta^2 = \phi^{-1}$ (**tort.dep**=2), $\theta^2 = \phi^{-2}$ (**tort.dep**=3), $\theta^2 = 1 + 3(1 - \phi)$ (**tort.dep**=4), should you desire another tortuosity calculation.

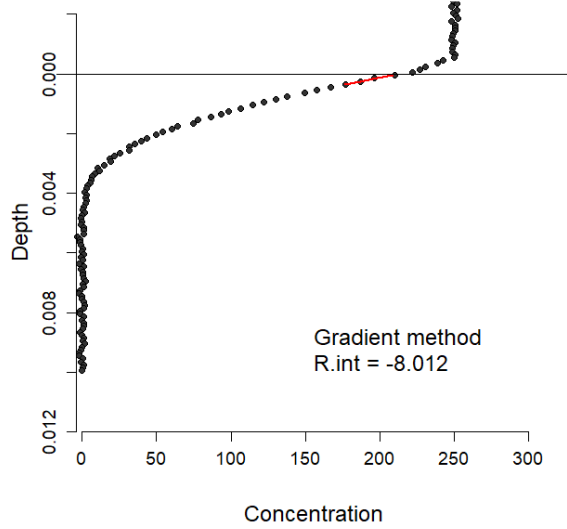


Figure 2: gradient FLIPPER output

An alternative way of calculating an oxygen flux into the sediment is by calculating the diffusive oxygen flux in the water column, through the so-called Diffusive Boundary Layer. In this layer just above the sediment-water interface, advection is not efficient, and diffusion is the dominant transport pathway for dissolved substances. For more in depth reading about the DBL and its impact on O₂ fluxes, you can refer to (Jørgensen and Des Marais 1990) and many other publications on this subject. For now, all you need to know is that the flux of O₂ through the DBL should be equal to the flux of O₂ into the sediment. We can test this by running FLIPPER again, and this time select the first point just above the SWI, and the 6th point above the SWI (where the constant concentration of O₂ starts decreasing).

```
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                    env.parms=env.parms,
                    method="gradient")
plot.FLIPPER(test)
```

If all went well, FLIPPER calculated for us a diffusive oxygen flux of $7.536 \text{ mmol m}^{-2} \text{ d}^{-1}$, which, given the uncertainty associated with the measurements, is pretty close to the oxygen flux estimated from the first sediment layer. Let's run FLIPPER again, and this time we will select a point just below the sediment-water interface, and one near the bottom of the concentration profile (just for illustrative purposes)

```
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                    env.parms=env.parms,
                    method="gradient")
plot.FLIPPER(test)
```

You will notice that the red line this time does not go through the points. This is because FLIPPER is trying to fit a linear model through all the points between the two selected by the user. You will thus have to be careful if your profile contains outliers (e.g. that can be generated by a spike in power), as FLIPPER will use the points provided by you.

Let's now have a quick look at the output provided by FLIPPER, when it is run for the gradient method alone.

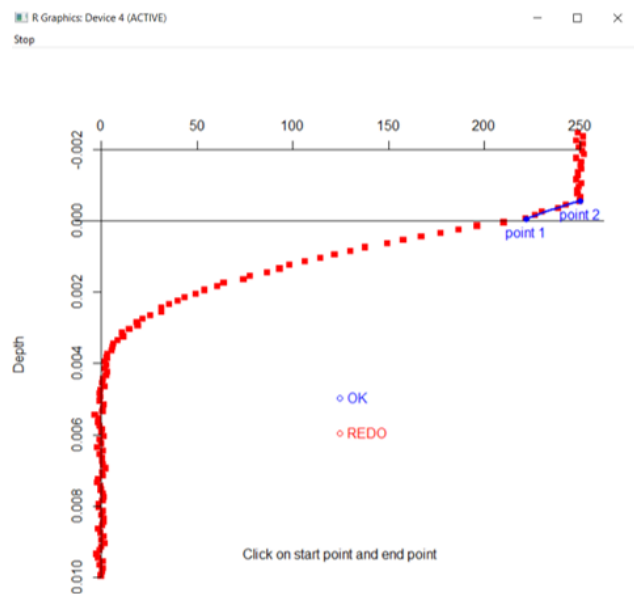


Figure 3: Selected points on gradient FLIPPER

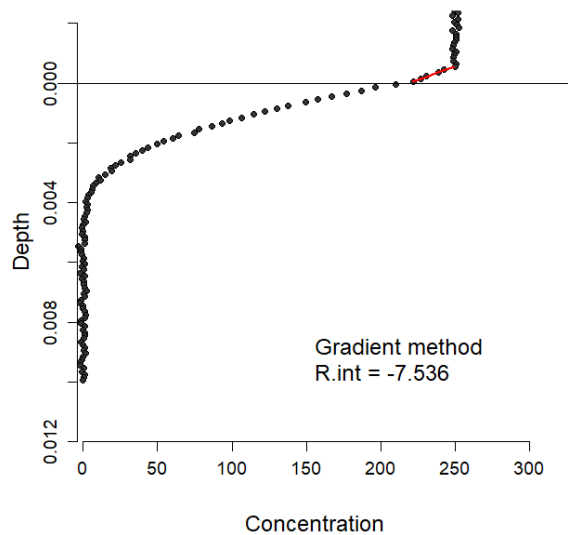


Figure 4: gradient FLIPPER output

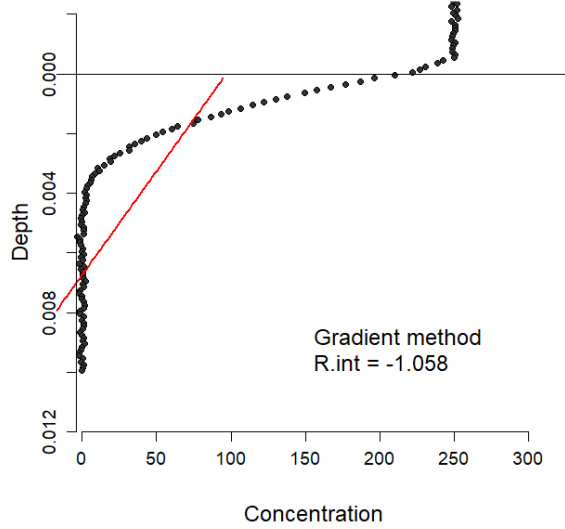


Figure 5: gradient FLIPPER output

```
print(output.example$gradient)
```

```
## $method
## [1] "gradient"
##
## $input
##          C          x por      tort
## 1  249.14098524 -0.00245 1.0 1.000000
## 2  251.93165932 -0.00235 1.0 1.000000
## 3  248.33756886 -0.00225 1.0 1.000000
## 4  251.55607813 -0.00215 1.0 1.000000
## 5  249.56641861 -0.00205 1.0 1.000000
## 6  251.22097397 -0.00195 1.0 1.000000
## 7  252.15812981 -0.00185 1.0 1.000000
## 8  248.32441400 -0.00175 1.0 1.000000
## 9  250.77935387 -0.00165 1.0 1.000000
## 10 250.28642787 -0.00155 1.0 1.000000
## 11 250.82129515 -0.00145 1.0 1.000000
## 12 248.97772029 -0.00135 1.0 1.000000
## 13 248.92227847 -0.00125 1.0 1.000000
## 14 248.16707359 -0.00115 1.0 1.000000
## 15 250.81231275 -0.00105 1.0 1.000000
## 16 249.49813237 -0.00095 1.0 1.000000
## 17 248.46877682 -0.00085 1.0 1.000000
## 18 248.85857712 -0.00075 1.0 1.000000
## 19 250.23357803 -0.00065 1.0 1.000000
## 20 250.06813313 -0.00055 1.0 1.000000
## 21 242.71439433 -0.00045 1.0 1.000000
## 22 238.81376115 -0.00035 1.0 1.000000
## 23 230.38514550 -0.00025 1.0 1.000000
```

```

## 24 226.66390836 -0.00015 1.0 1.000000
## 25 221.73139345 -0.00005 1.0 1.000000
## 26 210.21775269 0.00005 0.8 1.446287
## 27 196.37905798 0.00015 0.8 1.446287
## 28 186.61281200 0.00025 0.8 1.446287
## 29 177.18220881 0.00035 0.8 1.446287
## 30 167.21812426 0.00045 0.8 1.446287
## 31 157.77106794 0.00055 0.8 1.446287
## 32 149.40530858 0.00065 0.8 1.446287
## 33 137.63418693 0.00075 0.8 1.446287
## 34 130.34462143 0.00085 0.8 1.446287
## 35 121.88820672 0.00095 0.8 1.446287
## 36 114.46055069 0.00105 0.8 1.446287
## 37 106.57200305 0.00115 0.8 1.446287
## 38 98.14350965 0.00125 0.8 1.446287
## 39 93.15457339 0.00135 0.8 1.446287
## 40 86.72033940 0.00145 0.8 1.446287
## 41 77.59090717 0.00155 0.8 1.446287
## 42 74.35113395 0.00165 0.8 1.446287
## 43 63.96192903 0.00175 0.8 1.446287
## 44 60.51876573 0.00185 0.8 1.446287
## 45 53.88879413 0.00195 0.8 1.446287
## 46 49.45415907 0.00205 0.8 1.446287
## 47 43.68849807 0.00215 0.8 1.446287
## 48 39.91470896 0.00225 0.8 1.446287
## 49 35.37330012 0.00235 0.8 1.446287
## 50 31.34843510 0.00245 0.8 1.446287
## 51 31.35117386 0.00255 0.8 1.446287
## 52 25.63587774 0.00265 0.8 1.446287
## 53 21.45928539 0.00275 0.8 1.446287
## 54 18.66336269 0.00285 0.8 1.446287
## 55 19.13983200 0.00295 0.8 1.446287
## 56 14.68609745 0.00305 0.8 1.446287
## 57 10.70712245 0.00315 0.8 1.446287
## 58 11.47715125 0.00325 0.8 1.446287
## 59 8.51546536 0.00335 0.8 1.446287
## 60 6.51960105 0.00345 0.8 1.446287
## 61 6.03092634 0.00355 0.8 1.446287
## 62 5.36889671 0.00365 0.8 1.446287
## 63 3.27552485 0.00375 0.8 1.446287
## 64 2.97360801 0.00385 0.8 1.446287
## 65 1.63744474 0.00395 0.8 1.446287
## 66 2.68728486 0.00405 0.8 1.446287
## 67 2.04358321 0.00415 0.8 1.446287
## 68 3.23761586 0.00425 0.8 1.446287
## 69 2.55618957 0.00435 0.8 1.446287
## 70 0.84687597 0.00445 0.8 1.446287
## 71 0.68003614 0.00455 0.8 1.446287
## 72 1.77098300 0.00465 0.8 1.446287
## 73 -0.44801608 0.00475 0.8 1.446287
## 74 -0.74019664 0.00485 0.8 1.446287
## 75 -0.20820845 0.00495 0.8 1.446287
## 76 -0.84855083 0.00505 0.8 1.446287
## 77 1.30175475 0.00515 0.8 1.446287

```



```

## 78    0.88177169  0.00525 0.8 1.446287
## 79    0.76798229  0.00535 0.8 1.446287
## 80   -3.29641326  0.00545 0.8 1.446287
## 81   -1.41442202  0.00555 0.8 1.446287
## 82   -1.61533851  0.00565 0.8 1.446287
## 83   -0.76923618  0.00575 0.8 1.446287
## 84    0.28936531  0.00585 0.8 1.446287
## 85    0.02913935  0.00595 0.8 1.446287
## 86    1.31302951  0.00605 0.8 1.446287
## 87   -0.68492843  0.00615 0.8 1.446287
## 88    0.18431539  0.00625 0.8 1.446287
## 89   -2.22586623  0.00635 0.8 1.446287
## 90    0.88992907  0.00645 0.8 1.446287
## 91   -0.97916797  0.00655 0.8 1.446287
## 92    0.54293288  0.00665 0.8 1.446287
## 93    0.42419159  0.00675 0.8 1.446287
## 94    1.07216447  0.00685 0.8 1.446287
## 95    2.22449855  0.00695 0.8 1.446287
## 96    0.68213957  0.00705 0.8 1.446287
## 97    0.98976682  0.00715 0.8 1.446287
## 98   -1.55661928  0.00725 0.8 1.446287
## 99   -2.04850486  0.00735 0.8 1.446287
## 100  -0.40856510  0.00745 0.8 1.446287
## 101  -0.16777120  0.00755 0.8 1.446287
## 102    1.06186150  0.00765 0.8 1.446287
## 103    1.51612451  0.00775 0.8 1.446287
## 104    0.77186306  0.00785 0.8 1.446287
## 105   -1.41924805  0.00795 0.8 1.446287
## 106   -1.19090841  0.00805 0.8 1.446287
## 107    0.86978160  0.00815 0.8 1.446287
## 108   -0.10482927  0.00825 0.8 1.446287
## 109    1.33405187  0.00835 0.8 1.446287
## 110    1.21573244  0.00845 0.8 1.446287
## 111    0.58860990  0.00855 0.8 1.446287
## 112   -1.66255929  0.00865 0.8 1.446287
## 113   -0.30192756  0.00875 0.8 1.446287
## 114    1.26487946  0.00885 0.8 1.446287
## 115    0.13801946  0.00895 0.8 1.446287
## 116    1.75117179  0.00905 0.8 1.446287
## 117    0.06955914  0.00915 0.8 1.446287
## 118   -0.89293739  0.00925 0.8 1.446287
## 119   -2.36748426  0.00935 0.8 1.446287
## 120   -1.79141490  0.00945 0.8 1.446287
## 121    0.73518538  0.00955 0.8 1.446287
## 122   -0.71282464  0.00965 0.8 1.446287
## 123    0.74609846  0.00975 0.8 1.446287
## 124    0.37603973  0.00985 0.8 1.446287
## 125   -0.12243665  0.00995 0.8 1.446287
##
## $output
## $output$J.dif.up
## [1] 8.01177
##
## $output$J.adv.up

```

```

## [1] 0
##
## $output$R.int
## [1] -8.01177
##
## $output$fit
## $output$fit$x
## [1] 0.00005 0.00015 0.00025 0.00035
##
## $output$fit$C
## [1] 208.9289 198.0416 187.1543 176.2670
##
## $output$fit$slope
## [1] -108872.9
##
## $output$fit$intercept
## [1] 214.3725
##
## $output$fit$n
## [1] 4
##
##
##
## $parms
## $parms$env.parms
## $parms$env.parms$TC
## [1] 10
##
## $parms$env.parms$S
## [1] 35
##
## $parms$env.parms$P
## [1] 1.013
##
## $parms$env.parms$z
## [1] 0
##
## $parms$env.parms$Dmol
## [1] 0.0001330373
##
##
## $parms$gradient.parms
## $parms$gradient.parms$x.limits
## [1] NA

```

You will see that the output generated contains the selected method and the original input provided by the user. But two columns have been added to the profile; **por**(ϕ) and **tort** (θ^2). These two columns are calculated automatically from the porosity values provided (either as constant or within the **input** dataframe) - as described above. The output also contains a separate **output** list, which provides you with the diffusive flux where you selected the points (**J.dif.up**), the advective flux at the same location (**J.adv.up**), the integrated reaction rate (**R.int**), and some information related to the linear fit (which is generated by the *lm* function in R). Finally, FLIPPER also returns the environmental parameters (**env.parms**) and specific gradient parameters (**gradient.parms**). In case you did not provide any specific input, these lists will contain the default parameters. The diffusion coefficient (if not provided explicitly) will be calculated using the

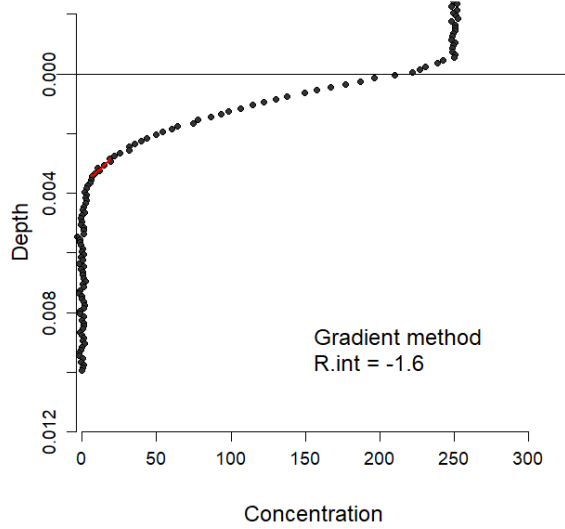


Figure 6: gradient FLIPPER output

diffcoeff function of CRAN:marelac.

Now let's go back to our oxygen profile. Using FLIPPER we calculated an oxygen flux of $\sim 8 \text{ mmol m}^{-2} \text{ d}^{-1}$. What if we now calculate the oxygen flux deeper down in the sediment, for example around 0.3 cm (I will let you choose which points to take)

```
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                    env.parms=env.parms,
                    method="gradient")
plot.FLIPPER(test)
```

The oxygen flux is now much lower (likely around $2 \text{ mmol m}^{-2} \text{ d}^{-1}$). If $8 \text{ mmol m}^{-2} \text{ d}^{-1}$ comes in at the top of the sediment, and only $2 \text{ mmol m}^{-2} \text{ d}^{-1}$ leaves the sediment at 0.3 cm depth, we have 'lost' $\sim 6 \text{ mmol m}^{-2} \text{ d}^{-1}$ in those 3 cm. This 'loss' is what has been consumed by microorganisms (and macro/meio organisms, should they be present in your sediment core), we can thus say that the integrated oxygen consumption in the first 0.3 cm of the sediment is $\sim 6 \text{ mmol m}^{-2} \text{ d}^{-1}$. Should you calculate the oxygen flux at the bottom of your profile (where the O₂ concentration is constant and zero), you would get a flux of $0 \text{ mmol m}^{-2} \text{ d}^{-1}$ (you can try if you don't believe me, simply press 'Run FLIPPER' again above). That means that in the whole sediment column, the total integrated oxygen consumption rate (**R.int**) is (assuming advective transport can be neglected, which it can in unbioturbated sediments)

$$R_{int} = J_{dif}^{bottom} - J_{dif}^{top}$$

and since our diffusive flux at the bottom of our domain is zero,

$$R_{int} = -J_{dif}^{top}$$

which immediately explains why **R.int** in the gradient method is equal to **J.dif.up** (note that this is only valid for species that are completely consumed in the sediment, like - generally - oxygen is). Should you use the gradient function of FLIPPER to calculate gradients at different location of a profile, **R.int** does not mean anything anymore. We could now theoretically calculate a diffusive flux at different locations in our porewater profile, and this would allow us to delineate different consumption zones. If we want to know the production/consumption rate of a compound at a certain location in the sediment (and not the integrated

rate), we should take the derivative of the flux, or

$$R(x) = \frac{\delta J}{\delta x}$$

which can be approximated as

$$R(x) = \frac{\Delta J}{\Delta x} = \frac{J_{dif}^{bottom} - J_{dif}^{top}}{x_{bottom} - x_{top}}$$

Now, instead of doing this manually for the whole profile, we could also set up a mass balance equation for a dissolved compound in porewater, and fit that to our concentration profile. This is exactly what the discrete method in FLIPPER does, so now it is time to move to the next step.

The discrete method (diagenetic model fitting)

The discrete method is in essence the original method of the ‘PROFILE’ program of (Berg, Risgaard-Petersen, and Rysgaard 1998), but with some small amendments and extension. FLIPPER will fit the steady state mass balance equation for a porewater solute (Boudreau 1997, F. J. R. Meysman, Boudreau, and Middelburg (2005))

$$\frac{\delta}{\delta x} \left[\phi D_s \frac{\delta C}{\delta x} - \phi v C_i \right] + \phi \alpha (C_0 - C) + R = 0$$

where ϕ is the porosity, x is the depth coordinate, C is the pore-water concentration, C_0 is the bottom-water concentration, α is the bio-irrigation coefficient, D_s is the effective diffusion coefficient of the solute (i.e. corrected for porosity) and v is the advective velocity of the porewater. The original PROFILE did not account for advection v , but did include bio-mixing (which we here ignore, as diffusion is much faster). FLIPPER also has the possibility to take into account ionic drift due to an electrical field (which will be discussed in more detail below). We supply all the terms in the equation above to FLIPPER (these are essentially the same parameters as we supplied above for calculating the diffusive flux, aside from the irrigation term α), aside from the production profile R . The goal is then to find R , so that the modelled C fits our data. The exact procedure to do this is explained in detail in (Berg, Risgaard-Petersen, and Rysgaard 1998). In short, FLIPPER (as PROFILE before him) will divide the sediment in discrete zones with a constant R , it will start at 1 and continue to increase the number of zones (2, 3, 4, ..). For each number of zones, FLIPPER will calculate the model cost (as the sum of squared residuals between the modelled concentration profile and supplied data) and evaluate whether increasing the number of zones is statistically improving the model-data fit (it will this for all zone increases, so 1->2, 2->3, 3->4, ...). Once the number of zones is selected, it will check whether combining adjacent zones significantly decreases the model-data fit, and continue lumping zones together until the fit significantly worsens.

The discrete method in FLIPPER can be called as follows

```
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                    env.parms=env.parms,
                    discrete.parms=discrete.parms,
                    method="discrete")

plot.FLIPPER(test)
```

We now have an extra (optional) list we can supply **discrete.parms**, which contains parameters specifically for the discrete method. The default parameters for the discrete method are the following

```
L.down <- max(input[(input$x>=0.0),"x"])
x.up    <- min(input[(input$x>=0.0),"x"])
irr     <- 0
irr.att <- 0.03
N       <- 200
i.end   <- min(12,length(unique(input$x[!is.na(input$C)]))-2)
initial.zones <- NULL
```

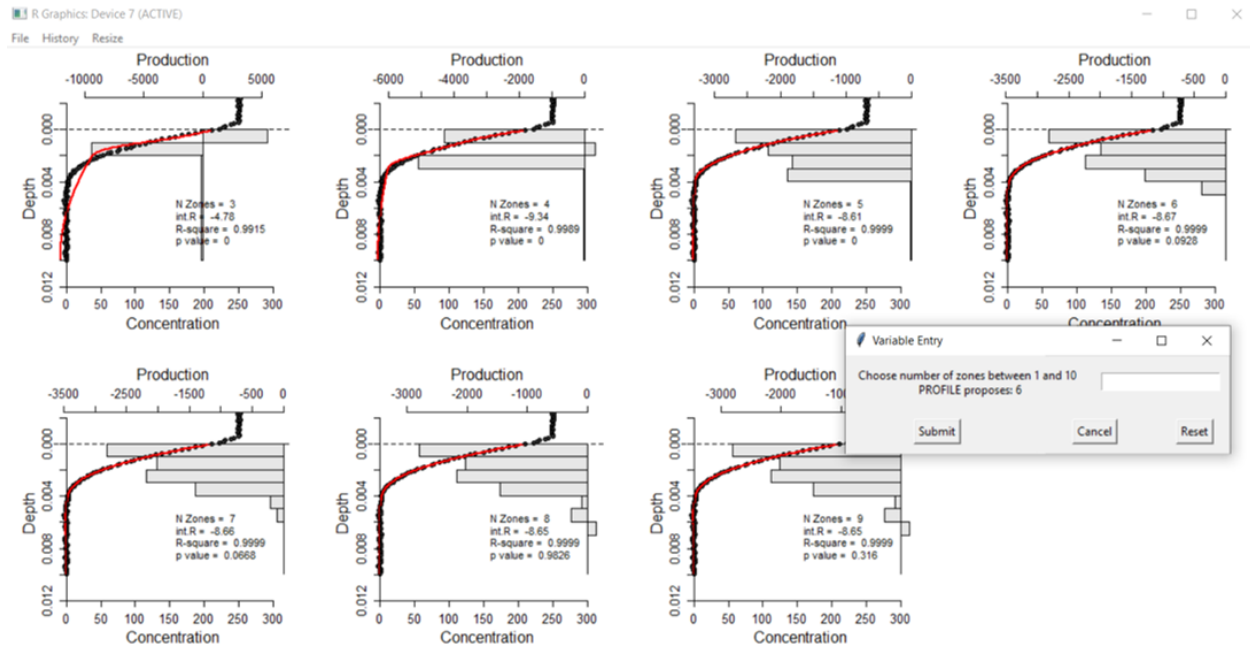


Figure 7: discrete FLIPPER popup

```
prob      <- 0.01
UBC       <- "conc.up"
LBC       <- "no.flux"
flux.up   <- NULL
flux.down <- NULL
```

But we do not need to worry about those just yet. Let's first analyze our oxygen profile with the default parameters. Just create an empty list (note that you can also just not supply a **discrete.parms** list) and run FLIPPER. A pop-up window will appear, for now just supply the value FLIPPER proposes.

```
discrete.parms <- list()
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                     env.parms=env.parms,
                     discrete.parms=discrete.parms,
                     method="discrete")

plot.FLIPPER(test)
```

This immediately illustrates how the discrete method functions, as you can see the discrete production zones (negative production indicates consumption). The red line again shows the model fit to the data. If all went well, you should see FLIPPER now calculated an integrated production rate of $-8.669 \text{ mmol m}^{-2} \text{ d}^{-1}$, which is slightly higher than the gradient approach predicted, but very similar.

Let's now have a quick look at the output provided by FLIPPER, when it is run for the discrete method alone.

```
## $method
## [1] "discrete"
##
## $input
##           C           x por      tort v
## 1  249.14098524 -0.00245 1.0 1.000000 0
```

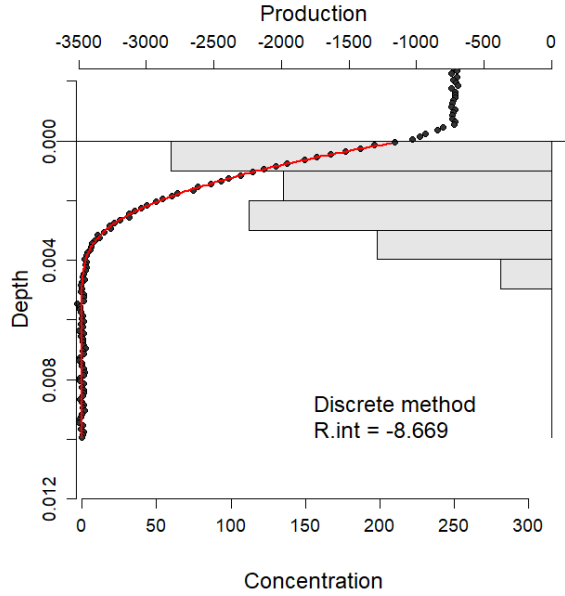


Figure 8: discrete FLIPPER output

```
## 2 251.93165932 -0.00235 1.0 1.000000 0
## 3 248.33756886 -0.00225 1.0 1.000000 0
## 4 251.55607813 -0.00215 1.0 1.000000 0
## 5 249.56641861 -0.00205 1.0 1.000000 0
## 6 251.22097397 -0.00195 1.0 1.000000 0
## 7 252.15812981 -0.00185 1.0 1.000000 0
## 8 248.32441400 -0.00175 1.0 1.000000 0
## 9 250.77935387 -0.00165 1.0 1.000000 0
## 10 250.28642787 -0.00155 1.0 1.000000 0
## 11 250.82129515 -0.00145 1.0 1.000000 0
## 12 248.97772029 -0.00135 1.0 1.000000 0
## 13 248.92227847 -0.00125 1.0 1.000000 0
## 14 248.16707359 -0.00115 1.0 1.000000 0
## 15 250.81231275 -0.00105 1.0 1.000000 0
## 16 249.49813237 -0.00095 1.0 1.000000 0
## 17 248.46877682 -0.00085 1.0 1.000000 0
## 18 248.85857712 -0.00075 1.0 1.000000 0
## 19 250.23357803 -0.00065 1.0 1.000000 0
## 20 250.06813313 -0.00055 1.0 1.000000 0
## 21 242.71439433 -0.00045 1.0 1.000000 0
## 22 238.81376115 -0.00035 1.0 1.000000 0
## 23 230.38514550 -0.00025 1.0 1.000000 0
## 24 226.66390836 -0.00015 1.0 1.000000 0
## 25 221.73139345 -0.00005 1.0 1.000000 0
## 26 210.21775269 0.00005 0.8 1.446287 0
## 27 196.37905798 0.00015 0.8 1.446287 0
## 28 186.61281200 0.00025 0.8 1.446287 0
## 29 177.18220881 0.00035 0.8 1.446287 0
## 30 167.21812426 0.00045 0.8 1.446287 0
## 31 157.77106794 0.00055 0.8 1.446287 0
## 32 149.40530858 0.00065 0.8 1.446287 0
```

## 33	137.63418693	0.00075	0.8	1.446287	0
## 34	130.34462143	0.00085	0.8	1.446287	0
## 35	121.88820672	0.00095	0.8	1.446287	0
## 36	114.46055069	0.00105	0.8	1.446287	0
## 37	106.57200305	0.00115	0.8	1.446287	0
## 38	98.14350965	0.00125	0.8	1.446287	0
## 39	93.15457339	0.00135	0.8	1.446287	0
## 40	86.72033940	0.00145	0.8	1.446287	0
## 41	77.59090717	0.00155	0.8	1.446287	0
## 42	74.35113395	0.00165	0.8	1.446287	0
## 43	63.96192903	0.00175	0.8	1.446287	0
## 44	60.51876573	0.00185	0.8	1.446287	0
## 45	53.88879413	0.00195	0.8	1.446287	0
## 46	49.45415907	0.00205	0.8	1.446287	0
## 47	43.68849807	0.00215	0.8	1.446287	0
## 48	39.91470896	0.00225	0.8	1.446287	0
## 49	35.37330012	0.00235	0.8	1.446287	0
## 50	31.34843510	0.00245	0.8	1.446287	0
## 51	31.35117386	0.00255	0.8	1.446287	0
## 52	25.63587774	0.00265	0.8	1.446287	0
## 53	21.45928539	0.00275	0.8	1.446287	0
## 54	18.66336269	0.00285	0.8	1.446287	0
## 55	19.13983200	0.00295	0.8	1.446287	0
## 56	14.68609745	0.00305	0.8	1.446287	0
## 57	10.70712245	0.00315	0.8	1.446287	0
## 58	11.47715125	0.00325	0.8	1.446287	0
## 59	8.51546536	0.00335	0.8	1.446287	0
## 60	6.51960105	0.00345	0.8	1.446287	0
## 61	6.03092634	0.00355	0.8	1.446287	0
## 62	5.36889671	0.00365	0.8	1.446287	0
## 63	3.27552485	0.00375	0.8	1.446287	0
## 64	2.97360801	0.00385	0.8	1.446287	0
## 65	1.63744474	0.00395	0.8	1.446287	0
## 66	2.68728486	0.00405	0.8	1.446287	0
## 67	2.04358321	0.00415	0.8	1.446287	0
## 68	3.23761586	0.00425	0.8	1.446287	0
## 69	2.55618957	0.00435	0.8	1.446287	0
## 70	0.84687597	0.00445	0.8	1.446287	0
## 71	0.68003614	0.00455	0.8	1.446287	0
## 72	1.77098300	0.00465	0.8	1.446287	0
## 73	-0.44801608	0.00475	0.8	1.446287	0
## 74	-0.74019664	0.00485	0.8	1.446287	0
## 75	-0.20820845	0.00495	0.8	1.446287	0
## 76	-0.84855083	0.00505	0.8	1.446287	0
## 77	1.30175475	0.00515	0.8	1.446287	0
## 78	0.88177169	0.00525	0.8	1.446287	0
## 79	0.76798229	0.00535	0.8	1.446287	0
## 80	-3.29641326	0.00545	0.8	1.446287	0
## 81	-1.41442202	0.00555	0.8	1.446287	0
## 82	-1.61533851	0.00565	0.8	1.446287	0
## 83	-0.76923618	0.00575	0.8	1.446287	0
## 84	0.28936531	0.00585	0.8	1.446287	0
## 85	0.02913935	0.00595	0.8	1.446287	0
## 86	1.31302951	0.00605	0.8	1.446287	0

```

## 87 -0.68492843 0.00615 0.8 1.446287 0
## 88 0.18431539 0.00625 0.8 1.446287 0
## 89 -2.22586623 0.00635 0.8 1.446287 0
## 90 0.88992907 0.00645 0.8 1.446287 0
## 91 -0.97916797 0.00655 0.8 1.446287 0
## 92 0.54293288 0.00665 0.8 1.446287 0
## 93 0.42419159 0.00675 0.8 1.446287 0
## 94 1.07216447 0.00685 0.8 1.446287 0
## 95 2.22449855 0.00695 0.8 1.446287 0
## 96 0.68213957 0.00705 0.8 1.446287 0
## 97 0.98976682 0.00715 0.8 1.446287 0
## 98 -1.55661928 0.00725 0.8 1.446287 0
## 99 -2.04850486 0.00735 0.8 1.446287 0
## 100 -0.40856510 0.00745 0.8 1.446287 0
## 101 -0.16777120 0.00755 0.8 1.446287 0
## 102 1.06186150 0.00765 0.8 1.446287 0
## 103 1.51612451 0.00775 0.8 1.446287 0
## 104 0.77186306 0.00785 0.8 1.446287 0
## 105 -1.41924805 0.00795 0.8 1.446287 0
## 106 -1.19090841 0.00805 0.8 1.446287 0
## 107 0.86978160 0.00815 0.8 1.446287 0
## 108 -0.10482927 0.00825 0.8 1.446287 0
## 109 1.33405187 0.00835 0.8 1.446287 0
## 110 1.21573244 0.00845 0.8 1.446287 0
## 111 0.58860990 0.00855 0.8 1.446287 0
## 112 -1.66255929 0.00865 0.8 1.446287 0
## 113 -0.30192756 0.00875 0.8 1.446287 0
## 114 1.26487946 0.00885 0.8 1.446287 0
## 115 0.13801946 0.00895 0.8 1.446287 0
## 116 1.75117179 0.00905 0.8 1.446287 0
## 117 0.06955914 0.00915 0.8 1.446287 0
## 118 -0.89293739 0.00925 0.8 1.446287 0
## 119 -2.36748426 0.00935 0.8 1.446287 0
## 120 -1.79141490 0.00945 0.8 1.446287 0
## 121 0.73518538 0.00955 0.8 1.446287 0
## 122 -0.71282464 0.00965 0.8 1.446287 0
## 123 0.74609846 0.00975 0.8 1.446287 0
## 124 0.37603973 0.00985 0.8 1.446287 0
## 125 -0.12243665 0.00995 0.8 1.446287 0
##
## $output
## $output$J.dif.up
## [1] 8.485942
##
## $output$J.adv.up
## [1] 0
##
## $output$J.dif.down
## [1] 0
##
## $output$J.adv.down
## [1] 0
##
## $output$R.vol

```



```

##      depth      Prod
## 1 0.000000 -2816.5948268
## 2 0.000995 -1985.4737315
## 3 0.001990 -2240.8264601
## 4 0.002985 -1290.9642224
## 5 0.003980 -376.3198435
## 6 0.004975  -0.4572424
##
## $output$R.int
## [1] -8.668903
##
## $output$fit
##      x      C
## 1 0.00007475 2.073637e+02
## 2 0.00012425 2.017493e+02
## 3 0.00017375 1.962287e+02
## 4 0.00022325 1.908019e+02
## 5 0.00027275 1.854688e+02
## 6 0.00032225 1.802296e+02
## 7 0.00037175 1.750841e+02
## 8 0.00042125 1.700325e+02
## 9 0.00047075 1.650746e+02
## 10 0.00052025 1.602104e+02
## 11 0.00056975 1.554401e+02
## 12 0.00061925 1.507636e+02
## 13 0.00066875 1.461808e+02
## 14 0.00071825 1.416918e+02
## 15 0.00076775 1.372966e+02
## 16 0.00081725 1.329952e+02
## 17 0.00086675 1.287876e+02
## 18 0.00091625 1.246737e+02
## 19 0.00096575 1.206537e+02
## 20 0.00101525 1.167274e+02
## 21 0.00106475 1.128672e+02
## 22 0.00111425 1.090732e+02
## 23 0.00116375 1.053452e+02
## 24 0.00121325 1.016834e+02
## 25 0.00126275 9.808764e+01
## 26 0.00131225 9.455802e+01
## 27 0.00136175 9.109451e+01
## 28 0.00141125 8.769711e+01
## 29 0.00146075 8.436582e+01
## 30 0.00151025 8.110063e+01
## 31 0.00155975 7.790156e+01
## 32 0.00160925 7.476860e+01
## 33 0.00165875 7.170175e+01
## 34 0.00170825 6.870100e+01
## 35 0.00175775 6.576637e+01
## 36 0.00180725 6.289785e+01
## 37 0.00185675 6.009543e+01
## 38 0.00190625 5.735913e+01
## 39 0.00195575 5.468894e+01
## 40 0.00200525 5.208485e+01
## 41 0.00205475 4.955538e+01

```

## 42	0.00210425	4.710052e+01
## 43	0.00215375	4.472027e+01
## 44	0.00220325	4.241464e+01
## 45	0.00225275	4.018362e+01
## 46	0.00230225	3.802721e+01
## 47	0.00235175	3.594541e+01
## 48	0.00240125	3.393822e+01
## 49	0.00245075	3.200565e+01
## 50	0.00250025	3.014768e+01
## 51	0.00254975	2.836433e+01
## 52	0.00259925	2.665560e+01
## 53	0.00264875	2.502147e+01
## 54	0.00269825	2.346196e+01
## 55	0.00274775	2.197706e+01
## 56	0.00279725	2.056677e+01
## 57	0.00284675	1.923109e+01
## 58	0.00289625	1.797003e+01
## 59	0.00294575	1.678358e+01
## 60	0.00299525	1.567174e+01
## 61	0.00304475	1.460288e+01
## 62	0.00309425	1.357701e+01
## 63	0.00314375	1.259413e+01
## 64	0.00319325	1.165423e+01
## 65	0.00324275	1.075731e+01
## 66	0.00329225	9.903379e+00
## 67	0.00334175	9.092433e+00
## 68	0.00339125	8.324472e+00
## 69	0.00344075	7.599496e+00
## 70	0.00349025	6.917505e+00
## 71	0.00353975	6.278499e+00
## 72	0.00358925	5.682478e+00
## 73	0.00363875	5.129442e+00
## 74	0.00368825	4.619390e+00
## 75	0.00373775	4.152324e+00
## 76	0.00378725	3.728242e+00
## 77	0.00383675	3.347145e+00
## 78	0.00388625	3.009033e+00
## 79	0.00393575	2.713906e+00
## 80	0.00398525	2.461764e+00
## 81	0.00403475	2.222152e+00
## 82	0.00408425	1.995071e+00
## 83	0.00413375	1.780519e+00
## 84	0.00418325	1.578498e+00
## 85	0.00423275	1.389007e+00
## 86	0.00428225	1.212046e+00
## 87	0.00433175	1.047615e+00
## 88	0.00438125	8.957151e-01
## 89	0.00443075	7.563450e-01
## 90	0.00448025	6.295051e-01
## 91	0.00452975	5.151954e-01
## 92	0.00457925	4.134159e-01
## 93	0.00462875	3.241667e-01
## 94	0.00467825	2.474476e-01
## 95	0.00472775	1.832588e-01

96 0.00477725 1.316003e-01
97 0.00482675 9.247191e-02
98 0.00487625 6.587378e-02
99 0.00492575 5.180586e-02
100 0.00497525 5.026817e-02
101 0.00502475 4.874570e-02
102 0.00507425 4.723846e-02
103 0.00512375 4.574644e-02
104 0.00517325 4.426965e-02
105 0.00522275 4.280808e-02
106 0.00527225 4.136174e-02
107 0.00532175 3.993062e-02
108 0.00537125 3.851472e-02
109 0.00542075 3.711405e-02
110 0.00547025 3.572861e-02
111 0.00551975 3.435838e-02
112 0.00556925 3.300339e-02
113 0.00561875 3.166362e-02
114 0.00566825 3.033907e-02
115 0.00571775 2.902975e-02
116 0.00576725 2.773565e-02
117 0.00581675 2.645678e-02
118 0.00586625 2.519313e-02
119 0.00591575 2.394471e-02
120 0.00596525 2.271151e-02
121 0.00601475 2.149353e-02
122 0.00606425 2.029078e-02
123 0.00611375 1.910326e-02
124 0.00616325 1.793096e-02
125 0.00621275 1.677388e-02
126 0.00626225 1.563203e-02
127 0.00631175 1.450541e-02
128 0.00636125 1.339400e-02
129 0.00641075 1.229783e-02
130 0.00646025 1.121688e-02
131 0.00650975 1.015115e-02
132 0.00655925 9.100645e-03
133 0.00660875 8.065367e-03
134 0.00665825 7.045314e-03
135 0.00670775 6.040485e-03
136 0.00675725 5.050881e-03
137 0.00680675 4.076502e-03
138 0.00685625 3.117347e-03
139 0.00690575 2.173417e-03
140 0.00695525 1.244712e-03
141 0.00700475 3.312313e-04
142 0.00705425 -5.670246e-04
143 0.00710375 -1.450056e-03
144 0.00715325 -2.317862e-03
145 0.00720275 -3.170444e-03
146 0.00725225 -4.007802e-03
147 0.00730175 -4.829934e-03
148 0.00735125 -5.636842e-03
149 0.00740075 -6.428525e-03

```
## 150 0.00745025 -7.204984e-03
## 151 0.00749975 -7.966218e-03
## 152 0.00754925 -8.712227e-03
## 153 0.00759875 -9.443011e-03
## 154 0.00764825 -1.015857e-02
## 155 0.00769775 -1.085891e-02
## 156 0.00774725 -1.154402e-02
## 157 0.00779675 -1.221390e-02
## 158 0.00784625 -1.286856e-02
## 159 0.00789575 -1.350800e-02
## 160 0.00794525 -1.413221e-02
## 161 0.00799475 -1.474120e-02
## 162 0.00804425 -1.533496e-02
## 163 0.00809375 -1.591350e-02
## 164 0.00814325 -1.647681e-02
## 165 0.00819275 -1.702490e-02
## 166 0.00824225 -1.755776e-02
## 167 0.00829175 -1.807540e-02
## 168 0.00834125 -1.857782e-02
## 169 0.00839075 -1.906501e-02
## 170 0.00844025 -1.953697e-02
## 171 0.00848975 -1.999371e-02
## 172 0.00853925 -2.043523e-02
## 173 0.00858875 -2.086152e-02
## 174 0.00863825 -2.127259e-02
## 175 0.00868775 -2.166843e-02
## 176 0.00873725 -2.204904e-02
## 177 0.00878675 -2.241444e-02
## 178 0.00883625 -2.276460e-02
## 179 0.00888575 -2.309955e-02
## 180 0.00893525 -2.341926e-02
## 181 0.00898475 -2.372376e-02
## 182 0.00903425 -2.401303e-02
## 183 0.00908375 -2.428707e-02
## 184 0.00913325 -2.454589e-02
## 185 0.00918275 -2.478949e-02
## 186 0.00923225 -2.501786e-02
## 187 0.00928175 -2.523100e-02
## 188 0.00933125 -2.542892e-02
## 189 0.00938075 -2.561162e-02
## 190 0.00943025 -2.577909e-02
## 191 0.00947975 -2.593134e-02
## 192 0.00952925 -2.606836e-02
## 193 0.00957875 -2.619016e-02
## 194 0.00962825 -2.629673e-02
## 195 0.00967775 -2.638808e-02
## 196 0.00972725 -2.646420e-02
## 197 0.00977675 -2.652510e-02
## 198 0.00982625 -2.657077e-02
## 199 0.00987575 -2.660122e-02
## 200 0.00992525 -2.661645e-02
##
##
## $parms
```

```

## $parms$env.parms
## $parms$env.parms$TC
## [1] 10
##
## $parms$env.parms$S
## [1] 35
##
## $parms$env.parms$P
## [1] 1.013
##
## $parms$env.parms$z
## [1] 0
##
## $parms$env.parms$Dmol
## [1] 0.0001330373
##
##
## $parms$discrete.parms
## $parms$discrete.parms$L.down
## [1] 0.00995
##
## $parms$discrete.parms$x.up
## [1] 5e-05
##
## $parms$discrete.parms$irr
## [1] 0
##
## $parms$discrete.parms$irr.att
## [1] 0.03
##
## $parms$discrete.parms$prob
## [1] 0.01
##
## $parms$discrete.parms$UBC
## [1] "conc.up"
##
## $parms$discrete.parms$LBC
## [1] "no.flux"
##
## $parms$discrete.parms$flux.up
## NULL
##
## $parms$discrete.parms$flux.down
## NULL
##
## $parms$discrete.parms$N
## [1] 200
##
## $parms$discrete.parms$i.end
## [1] 12
##
## $parms$discrete.parms$initial.zones
## NULL
##

```

```

## $parms$discrete.parms$C.up
## [1] 210.2178
##
## $parms$discrete.parms$grid
## $x.up
## [1] 5e-05
##
## $x.down
## [1] 0.00995
##
## $x.mid
## [1] 0.00007475 0.00012425 0.00017375 0.00022325 0.00027275 0.00032225
## [7] 0.00037175 0.00042125 0.00047075 0.00052025 0.00056975 0.00061925
## [13] 0.00066875 0.00071825 0.00076775 0.00081725 0.00086675 0.00091625
## [19] 0.00096575 0.00101525 0.00106475 0.00111425 0.00116375 0.00121325
## [25] 0.00126275 0.00131225 0.00136175 0.00141125 0.00146075 0.00151025
## [31] 0.00155975 0.00160925 0.00165875 0.00170825 0.00175775 0.00180725
## [37] 0.00185675 0.00190625 0.00195575 0.00200525 0.00205475 0.00210425
## [43] 0.00215375 0.00220325 0.00225275 0.00230225 0.00235175 0.00240125
## [49] 0.00245075 0.00250025 0.00254975 0.00259925 0.00264875 0.00269825
## [55] 0.00274775 0.00279725 0.00284675 0.00289625 0.00294575 0.00299525
## [61] 0.00304475 0.00309425 0.00314375 0.00319325 0.00324275 0.00329225
## [67] 0.00334175 0.00339125 0.00344075 0.00349025 0.00353975 0.00358925
## [73] 0.00363875 0.00368825 0.00373775 0.00378725 0.00383675 0.00388625
## [79] 0.00393575 0.00398525 0.00403475 0.00408425 0.00413375 0.00418325
## [85] 0.00423275 0.00428225 0.00433175 0.00438125 0.00443075 0.00448025
## [91] 0.00452975 0.00457925 0.00462875 0.00467825 0.00472775 0.00477725
## [97] 0.00482675 0.00487625 0.00492575 0.00497525 0.00502475 0.00507425
## [103] 0.00512375 0.00517325 0.00522275 0.00527225 0.00532175 0.00537125
## [109] 0.00542075 0.00547025 0.00551975 0.00556925 0.00561875 0.00566825
## [115] 0.00571775 0.00576725 0.00581675 0.00586625 0.00591575 0.00596525
## [121] 0.00601475 0.00606425 0.00611375 0.00616325 0.00621275 0.00626225
## [127] 0.00631175 0.00636125 0.00641075 0.00646025 0.00650975 0.00655925
## [133] 0.00660875 0.00665825 0.00670775 0.00675725 0.00680675 0.00685625
## [139] 0.00690575 0.00695525 0.00700475 0.00705425 0.00710375 0.00715325
## [145] 0.00720275 0.00725225 0.00730175 0.00735125 0.00740075 0.00745025
## [151] 0.00749975 0.00754925 0.00759875 0.00764825 0.00769775 0.00774725
## [157] 0.00779675 0.00784625 0.00789575 0.00794525 0.00799475 0.00804425
## [163] 0.00809375 0.00814325 0.00819275 0.00824225 0.00829175 0.00834125
## [169] 0.00839075 0.00844025 0.00848975 0.00853925 0.00858875 0.00863825
## [175] 0.00868775 0.00873725 0.00878675 0.00883625 0.00888575 0.00893525
## [181] 0.00898475 0.00903425 0.00908375 0.00913325 0.00918275 0.00923225
## [187] 0.00928175 0.00933125 0.00938075 0.00943025 0.00947975 0.00952925
## [193] 0.00957875 0.00962825 0.00967775 0.00972725 0.00977675 0.00982625
## [199] 0.00987575 0.00992525
##
## $x.int
## [1] 0.0000500 0.0000995 0.0001490 0.0001985 0.0002480 0.0002975 0.0003470
## [8] 0.0003965 0.0004460 0.0004955 0.0005450 0.0005945 0.0006440 0.0006935
## [15] 0.0007430 0.0007925 0.0008420 0.0008915 0.0009410 0.0009905 0.0010400
## [22] 0.0010895 0.0011390 0.0011885 0.0012380 0.0012875 0.0013370 0.0013865
## [29] 0.0014360 0.0014855 0.0015350 0.0015845 0.0016340 0.0016835 0.0017330
## [36] 0.0017825 0.0018320 0.0018815 0.0019310 0.0019805 0.0020300 0.0020795
## [43] 0.0021290 0.0021785 0.0022280 0.0022775 0.0023270 0.0023765 0.0024260

```

```

## [50] 0.0024755 0.0025250 0.0025745 0.0026240 0.0026735 0.0027230 0.0027725
## [57] 0.0028220 0.0028715 0.0029210 0.0029705 0.0030200 0.0030695 0.0031190
## [64] 0.0031685 0.0032180 0.0032675 0.0033170 0.0033665 0.0034160 0.0034655
## [71] 0.0035150 0.0035645 0.0036140 0.0036635 0.0037130 0.0037625 0.0038120
## [78] 0.0038615 0.0039110 0.0039605 0.0040100 0.0040595 0.0041090 0.0041585
## [85] 0.0042080 0.0042575 0.0043070 0.0043565 0.0044060 0.0044555 0.0045050
## [92] 0.0045545 0.0046040 0.0046535 0.0047030 0.0047525 0.0048020 0.0048515
## [99] 0.0049010 0.0049505 0.0050000 0.0050495 0.0050990 0.0051485 0.0051980
## [106] 0.0052475 0.0052970 0.0053465 0.0053960 0.0054455 0.0054950 0.0055445
## [113] 0.0055940 0.0056435 0.0056930 0.0057425 0.0057920 0.0058415 0.0058910
## [120] 0.0059405 0.0059900 0.0060395 0.0060890 0.0061385 0.0061880 0.0062375
## [127] 0.0062870 0.0063365 0.0063860 0.0064355 0.0064850 0.0065345 0.0065840
## [134] 0.0066335 0.0066830 0.0067325 0.0067820 0.0068315 0.0068810 0.0069305
## [141] 0.0069800 0.0070295 0.0070790 0.0071285 0.0071780 0.0072275 0.0072770
## [148] 0.0073265 0.0073760 0.0074255 0.0074750 0.0075245 0.0075740 0.0076235
## [155] 0.0076730 0.0077225 0.0077720 0.0078215 0.0078710 0.0079205 0.0079700
## [162] 0.0080195 0.0080690 0.0081185 0.0081680 0.0082175 0.0082670 0.0083165
## [169] 0.0083660 0.0084155 0.0084650 0.0085145 0.0085640 0.0086135 0.0086630
## [176] 0.0087125 0.0087620 0.0088115 0.0088610 0.0089105 0.0089600 0.0090095
## [183] 0.0090590 0.0091085 0.0091580 0.0092075 0.0092570 0.0093065 0.0093560
## [190] 0.0094055 0.0094550 0.0095045 0.0095540 0.0096035 0.0096530 0.0097025
## [197] 0.0097520 0.0098015 0.0098510 0.0099005 0.0099500
##
## $dx
## [1] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [9] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [17] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [25] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [33] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [41] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [49] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [57] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [65] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [73] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [81] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [89] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [97] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [105] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [113] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [121] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [129] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [137] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [145] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [153] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [161] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [169] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [177] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [185] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
## [193] 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05 4.95e-05
##
## $dx.aux
## [1] 2.475e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05
## [8] 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05
## [15] 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05 4.950e-05

```

[illegible]

[illegible]

[illegible]

```
## [166] 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05
## [171] 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05
## [176] 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05
## [181] 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05
## [186] 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05
## [191] 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05
## [196] 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05 9.198538e-05
## [201] 9.198538e-05
##
## attr(,"class")
## [1] "prop.1D"
##
## $parms$discrete.parms$v
## $mid
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##
## $int
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##
## attr(,"class")
## [1] "prop.1D"
```

we still have the **method** and **input** values (the method now obviously shows “discrete”). In the **parms** section we now have **discrete.parms** instead of **gradient.parms**. Aside from the possible input values shown above, FLIPPER also return the depth grid (**grid**), porosity grid (**por.grid**), irrigation grid (**irr.grid**), effective diffusion coefficient grid (**Ds**) and advective velocity grid (**v**). These have been generated using the *setup.grid.1D* function from CRAN:ReacTran, and are just provided for information. The **output** list now gives us again the diffusive and advective fluxes at the top of the domain (**J.dif.up,J.adv.up**), the fluxes at the bottom of the domain (**J.dif.down,J.adv.down**), the volumetric production rate (**R.vol**), the integrated production rate (**R.int**) and the model fit (**fit**). The volumetric production rate is given in a table form. The depth is the depth of the beginning of the production zone (the top) and the zone ends at the depth where the layer below starts. So in the example below

```
print(output.example$discrete$output$R.vol)
```

```
##      depth      Prod
## 1 0.000000 -2816.5948268
## 2 0.000995 -1985.4737315
## 3 0.001990 -2240.8264601
## 4 0.002985 -1290.9642224
## 5 0.003980 -376.3198435
## 6 0.004975  -0.4572424
```

the first production zone (-2816.59...) runs from depth 0.0 m (the sediment-water interface) to 0.000995 m. The last zone (zone 6) runs from 0.004975 m to the end of the domain. If you want, you can compare these values to the plot above. Note that volumetric production is given in $mmol\ m^{-3}\ d^{-1}$, since it is the

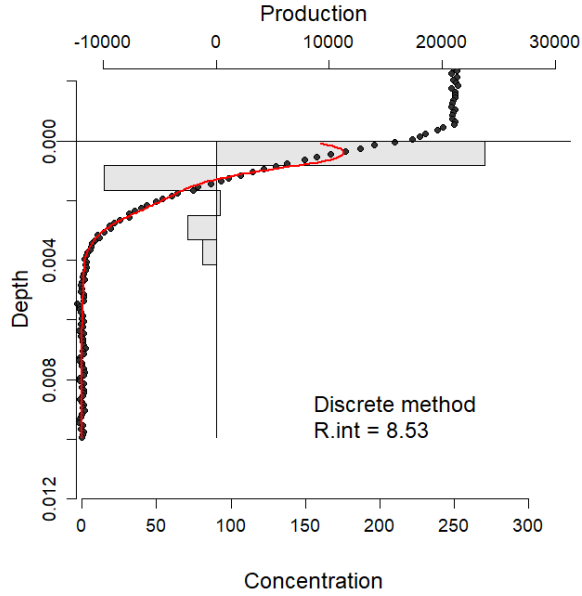


Figure 9: discrete FLIPPER output

derivative of flux to depth (see above).

By default, FLIPPER sets the upper boundary condition (**UBC**) to fixed concentration (“conc.up”), and the lower boundary condition (**LBC**) to a no gradient conditions (“no.flux”). You can also set **UBC** to “flux.up”, and define a fixed flux in **flux.up**. The **LBC** can be changed to fixed concentration (“conc.down”) or fixed flux (“flux.down”), define the flux in **flux.down**. Let’s say we are convinced that there is a flux of oxygen going out of the sediment, and that it is $8.0 \text{ mmol m}^{-2} \text{ d}^{-1}$. We can fill that in in our discrete parameter list (the negative sign indicates an upward flux) and run FLIPPER again

```
discrete.parms <- list(UBC="flux.up",LBC="conc.down",flux.up=-8.0)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  discrete.parms=discrete.parms,
  method="discrete")
plot.FLIPPER(test)
```

You will see that FLIPPER tried really hard to fit the profile with an upward flux out of the sediment. While this example does not make much sense, it could help to change the boundary conditions if FLIPPER is not able to find a good fit to your concentration profile with the given boundary conditions. Sometimes you can have a profile with a clear gradient at the bottom of the domain, in that case it makes sense to change **LBC** to “C.down”. Other parameters that could be changed are **L.down** and **x.up**, which define the top and bottom of the profile that should be analyzed by FLIPPER, for example - say we only want to analyze the data down to 0.3 cm depth (Note that changing the **x.up** parameter is a bit more sensitive and might lead to FLIPPER crashing);

```
discrete.parms <- list(L.down=0.003)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  discrete.parms=discrete.parms,
  method="discrete")
plot.FLIPPER(test)
```

what happens now is that we get many more zones. The reason for this is that FLIPPER will fit 12 zones

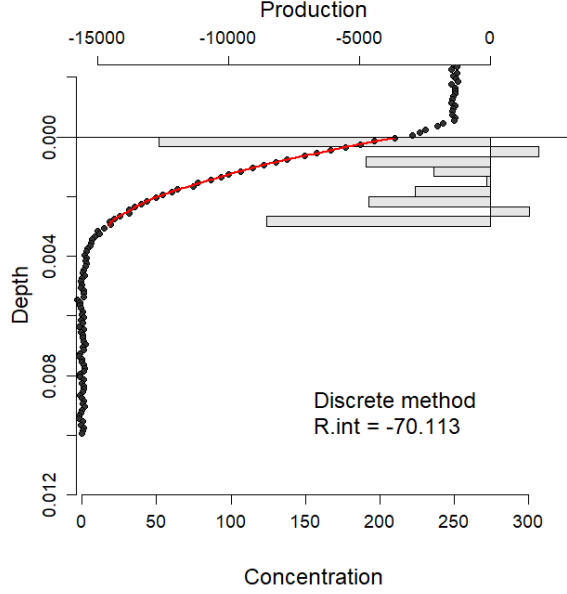


Figure 10: discrete FLIPPER output

on the 0.0 - 0.3 cm depth interval instead of the 0.0 - 1.0 cm depth interval. Because there are more zones, FLIPPER will be fitting much shorter segments of the concentration profile, which might lead to more noise in your production profile (as you can see - for example - from the production zone at depth). You are able to manipulate the number of zones FLIPPER uses via the **i.end** and **initial.zones** parameters. **i.end** tells FLIPPER the maximum number of zones he needs to test initially, whereas **initial.zones** tells FLIPPER to immediately start the lumping process from a given number of zones. For example;

```
discrete.parms <- list(i.end=3)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  discrete.parms=discrete.parms,
  method="discrete")
plot.FLIPPER(test)
```

and

```
discrete.parms <- list(initial.zones=5)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  discrete.parms=discrete.parms,
  method="discrete")
plot.FLIPPER(test)
```

The parameters related to irrigation **irr** and **irr.att** will be discussed below. The parameters **prob** tells the confidence interval at which to reject the null hypothesis. While we do not recommend to change this value, increasing it will generally lead to a higher number of zones. Parameter **N** is the number of depth layers the diagenetic model should take into account, and should not be changed.

We can now take our analysis a step further. As discussed above, the discrete analysis tries to find a solution to the steady-state mass-balance equation

$$\frac{\delta}{\delta x} \left[\phi D_s \frac{\delta C}{\delta x} - \phi v C \right] + \phi \alpha (C_0 - C) + R = 0$$

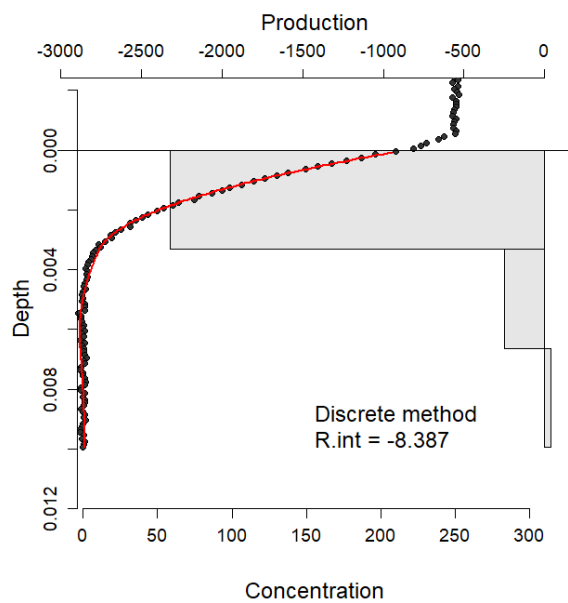


Figure 11: discrete FLIPPER output

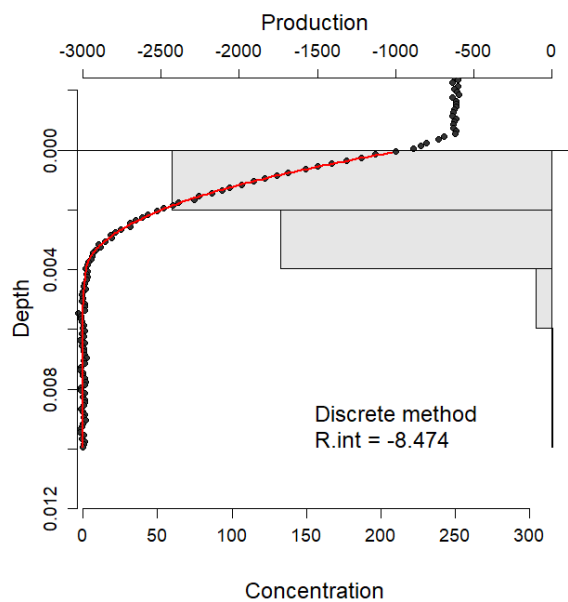


Figure 12: discrete FLIPPER output

If we ignore irrigation for now (this is not straight forward to quantify in marine sediments, and needs special consideration - we refer to (Thibault de Chanvalon et al. 2017) for a more detailed discussion), we know that transport of a dissolved species in porewater can be calculated as

$$J = -\phi D_s \frac{\delta C}{\delta x} + \phi v C$$

and if we assume steady-state, production can be calculated as

$$R = -\frac{\delta}{\delta x} \left[\phi D_s \frac{\delta C}{\delta x} - \phi v C \right]$$

or, if we work out the differential equation

$$R = -\phi D_s \frac{\delta^2 C}{\delta x^2} - D_s \frac{\delta \phi}{\delta x} \frac{\delta C}{\delta x} - \phi \frac{\delta D_s}{\delta x} \frac{\delta C}{\delta x} + \phi v \frac{\delta C}{\delta x} + v C \frac{\delta \phi}{\delta x} + \phi C \frac{\delta v}{\delta x} = -\phi D_s \frac{\delta^2 C}{\delta x^2} - \left(D_s \frac{\delta \phi}{\delta x} + \phi \frac{\delta D_s}{\delta x} - \phi v \right) \frac{\delta C}{\delta x} + \left(v \frac{\delta \phi}{\delta x} + \phi \frac{\delta v}{\delta x} \right) C$$

For our oxygen profile, we can assume that the porosity is constant with depth (no change over the first 0.5 cm), and so $\frac{\delta \phi}{\delta x}$ is zero. For a dissolved compound, diffusional transport is much faster than advective transport, and so we can ignore v and $\frac{\delta v}{\delta x}$. Finally, we can assume that salinity and temperature is constant in the first 0.5 cm, and since porosity is also constant, $\frac{\delta D_s}{\delta x}$ is zero. So our equation for R simplifies to

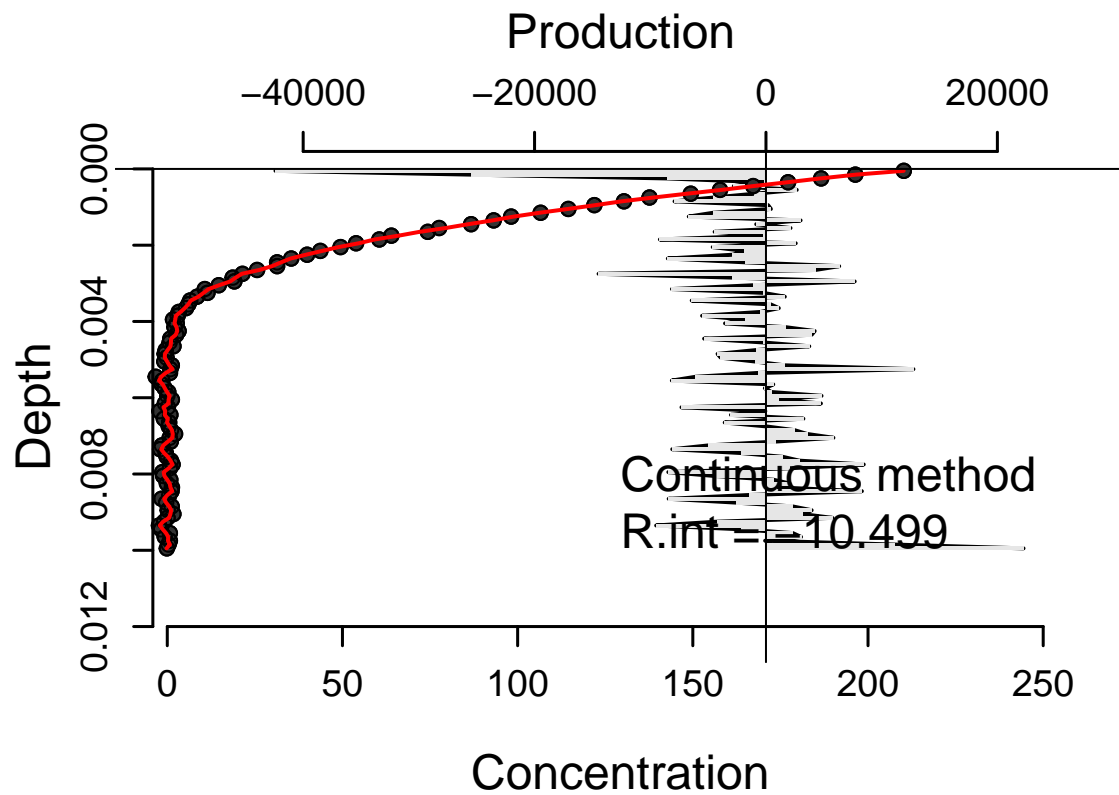
$$R = -\phi D_s \frac{\delta^2 C}{\delta x^2}$$

which shows that if we can find the first and second derivatives of the concentration profile, we can directly calculate transport and production. It is this approach that is used in the continuous FLIPPER method.

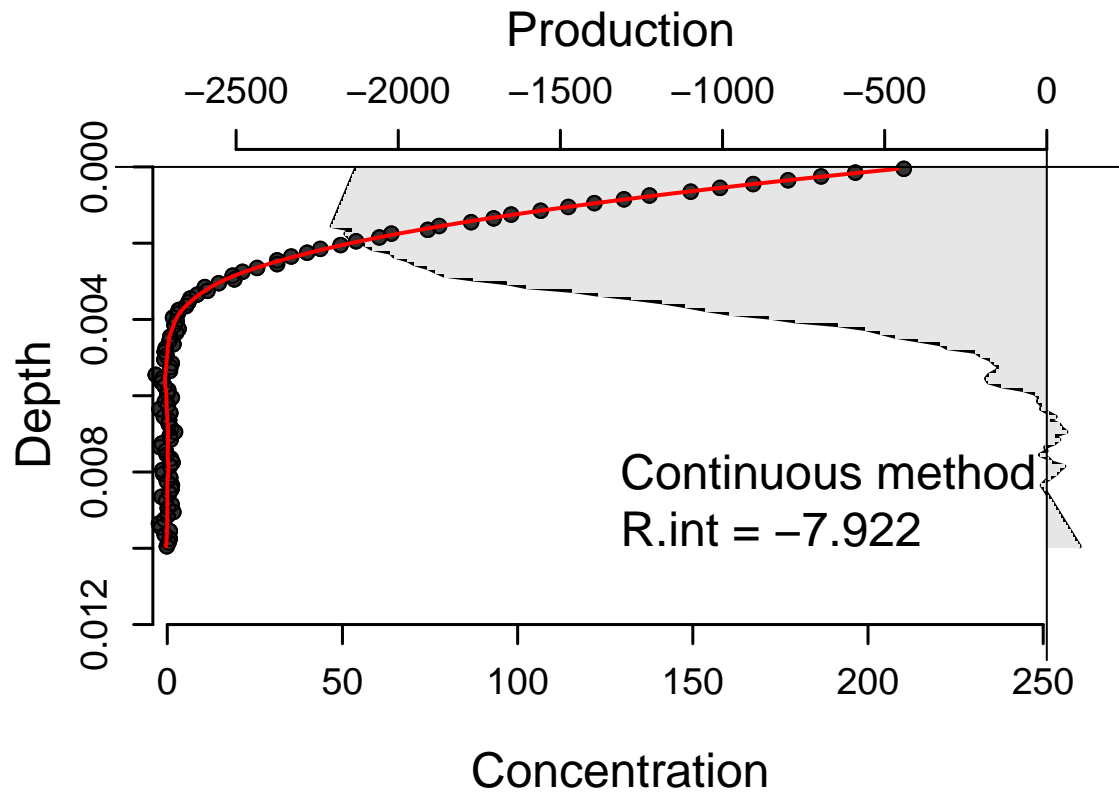
The continuous method (first and second derivative calculation)

The continuous method will derive the first and second derivative of a given concentration profile, and use that to calculate transport and production. One issue with experimental data is noise. We saw before that a higher number of zones can lead to fluctuations between positive and negative production zones when running the discrete method. One way of dealing with noise is smoothing your data. In the continuous method, we pass a window over the concentration profile, and fit a polynomial through it, and then calculate the first and second derivative of the polynomial at each point. The filter employed is a Savitzky-Golay filter, which is used a lot in analysis of spectra (Savitzky and Golay 1964). The critical point for this analysis is the filter window. If we choose too narrow a filter window (defined by parameter **n**, a window will be created of **n** points before and after the point of interest, so a total of **2n+1**), we will capture too much noise. If we choose too large a window, we will smooth too much. You can see the effect of the smoothing window on your output by changing the **n** below. Remember that **n** indicates the number of points left and right of the center point, so the actual window is **2n+1**.

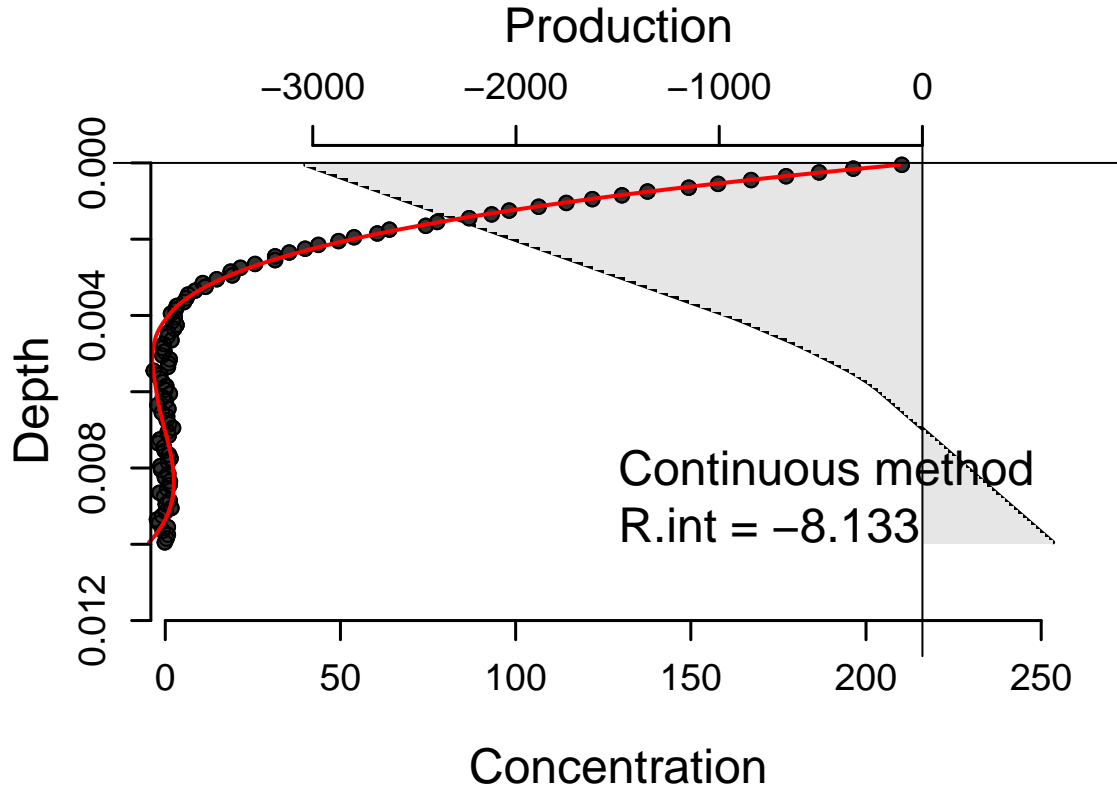
```
continuous.parms <- list(n.C=2,n.J=2,n.R=2)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
                    env.parms=env.parms,
                    continuous.parms=continuous.parms,
                    method="continuous")
plot.FLIPPER(test)
```



```
continuous.parms <- list(n.C=15,n.J=15,n.R=15)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  continuous.parms=continuous.parms,
  method="continuous")
plot.FLIPPER(test)
```

```
continuous.parms <- list(n.C=40,n.J=40,n.R=40)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  continuous.parms=continuous.parms,
  method="continuous")
plot.FLIPPER(test)
```



You will see that when the filter window is too low, we capture every twist and turn in the profiles and there is a lot of noise in the production profile. On the other hand, if we smooth too much, we are missing important features. For example, the sharp decrease in gradient at 0.4 cm is missed because the smoothing window takes into account too many of the higher concentration points. It is thus important to find the right smoothing window. You will also notice that the production profile predicted by the continuous method in FLIPPER is very similar as for the discrete method, which is what we would like (otherwise the diagenetic theory would be having issues :o)).

When you run FLIPPER interactive (the default mode), you will be shown three different pop-up windows where you will have to click on the 'break-point', this is where the curve stops decreasing and levels off (preferably at a value of 1). This break-point should be the optimal filter window. You will notice that (in general) the optimal filter window becomes higher for the higher order derivatives - so for C it is around 3 or 4, for the flux J (the first derivative $\frac{\delta C}{\delta x}$) it is around 6, and for the production profile R (the second derivative $\frac{\delta^2 C}{\delta x^2}$), it is around 15. The need to smooth more with higher order derivatives is simply a consequence of the increasing noise level (the uncertainty becomes progressively larger). When you select these points, you will notice that this way of selecting the breakpoints is not water-tight; there are still a lot of noise squiggles in the concentration profile. Another question that is not straight-forward to answer is whether you need to use the same filter window for C , $\frac{\delta C}{\delta x}$ and $\frac{\delta^2 C}{\delta x^2}$. You could argue that you should, since you want to calculate the derivatives of the same concentration profile, and if you change smoothing windows, the plotted model fit will not link to the plotted transport and production rates. On the other hand, higher order derivative introduce more noise, so it makes sense you have to increase the smoothing window for these derivatives, and you don't want to oversmooth the lower order derivatives. While it is important to be aware of these issues, choose different smoothing windows for J and R will not significantly affect your results. By default, FLIPPER sets the filter window uniformly, so all n values are set to the highest selected n (which will generally be the one selected for R). FLIPPER can also automatically select a filter size, but this function is a bit sensitive and has the tendency to crash (so this needs a bit more development). For now, we will not discuss the other options for the continuous method, as the default settings are the best ones to use.

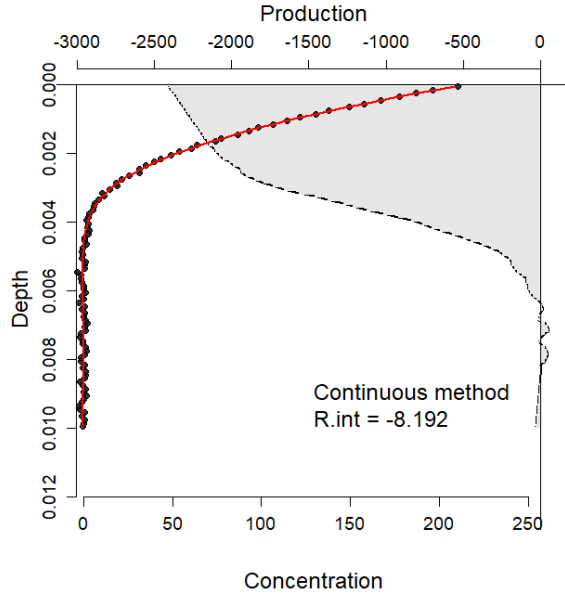


Figure 13: continuous FLIPPER output

```
continuous.parms <- list(optimal.window.size="interactive",n.uniform=FALSE)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  continuous.parms=continuous.parms,
  method="continuous")
plot.FLIPPER(test)
```

As a final point we will look at the output of the continuous method,

```
print(output.example$continuous)
```

```
## $method
## [1] "continuous"
##
## $input
## $input$user.input
##          C          x por      tort      Ds v dpor_dx dv_dx dDs_dx
## 1  249.14098524 -0.00245 1.0 1.000000 1.330373e-04 0      0      0      0
## 2  251.93165932 -0.00235 1.0 1.000000 1.330373e-04 0      0      0      0
## 3  248.33756886 -0.00225 1.0 1.000000 1.330373e-04 0      0      0      0
## 4  251.55607813 -0.00215 1.0 1.000000 1.330373e-04 0      0      0      0
## 5  249.56641861 -0.00205 1.0 1.000000 1.330373e-04 0      0      0      0
## 6  251.22097397 -0.00195 1.0 1.000000 1.330373e-04 0      0      0      0
## 7  252.15812981 -0.00185 1.0 1.000000 1.330373e-04 0      0      0      0
## 8  248.32441400 -0.00175 1.0 1.000000 1.330373e-04 0      0      0      0
## 9  250.77935387 -0.00165 1.0 1.000000 1.330373e-04 0      0      0      0
## 10 250.28642787 -0.00155 1.0 1.000000 1.330373e-04 0      0      0      0
## 11 250.82129515 -0.00145 1.0 1.000000 1.330373e-04 0      0      0      0
## 12 248.97772029 -0.00135 1.0 1.000000 1.330373e-04 0      0      0      0
## 13 248.92227847 -0.00125 1.0 1.000000 1.330373e-04 0      0      0      0
## 14 248.16707359 -0.00115 1.0 1.000000 1.330373e-04 0      0      0      0
```

## 15	250.81231275	-0.00105	1.0	1.000000	1.330373e-04	0	0	0
## 16	249.49813237	-0.00095	1.0	1.000000	1.330373e-04	0	0	0
## 17	248.46877682	-0.00085	1.0	1.000000	1.330373e-04	0	0	0
## 18	248.85857712	-0.00075	1.0	1.000000	1.330373e-04	0	0	0
## 19	250.23357803	-0.00065	1.0	1.000000	1.330373e-04	0	0	0
## 20	250.06813313	-0.00055	1.0	1.000000	1.330373e-04	0	0	0
## 21	242.71439433	-0.00045	1.0	1.000000	1.330373e-04	0	0	0
## 22	238.81376115	-0.00035	1.0	1.000000	1.330373e-04	0	0	0
## 23	230.38514550	-0.00025	1.0	1.000000	1.330373e-04	0	0	0
## 24	226.66390836	-0.00015	1.0	1.000000	1.330373e-04	0	0	0
## 25	221.73139345	-0.00005	1.0	1.000000	1.330373e-04	0	0	0
## 26	210.21775269	0.00005	0.8	1.446287	9.198538e-05	0	0	0
## 27	196.37905798	0.00015	0.8	1.446287	9.198538e-05	0	0	0
## 28	186.61281200	0.00025	0.8	1.446287	9.198538e-05	0	0	0
## 29	177.18220881	0.00035	0.8	1.446287	9.198538e-05	0	0	0
## 30	167.21812426	0.00045	0.8	1.446287	9.198538e-05	0	0	0
## 31	157.77106794	0.00055	0.8	1.446287	9.198538e-05	0	0	0
## 32	149.40530858	0.00065	0.8	1.446287	9.198538e-05	0	0	0
## 33	137.63418693	0.00075	0.8	1.446287	9.198538e-05	0	0	0
## 34	130.34462143	0.00085	0.8	1.446287	9.198538e-05	0	0	0
## 35	121.88820672	0.00095	0.8	1.446287	9.198538e-05	0	0	0
## 36	114.46055069	0.00105	0.8	1.446287	9.198538e-05	0	0	0
## 37	106.57200305	0.00115	0.8	1.446287	9.198538e-05	0	0	0
## 38	98.14350965	0.00125	0.8	1.446287	9.198538e-05	0	0	0
## 39	93.15457339	0.00135	0.8	1.446287	9.198538e-05	0	0	0
## 40	86.72033940	0.00145	0.8	1.446287	9.198538e-05	0	0	0
## 41	77.59090717	0.00155	0.8	1.446287	9.198538e-05	0	0	0
## 42	74.35113395	0.00165	0.8	1.446287	9.198538e-05	0	0	0
## 43	63.96192903	0.00175	0.8	1.446287	9.198538e-05	0	0	0
## 44	60.51876573	0.00185	0.8	1.446287	9.198538e-05	0	0	0
## 45	53.88879413	0.00195	0.8	1.446287	9.198538e-05	0	0	0
## 46	49.45415907	0.00205	0.8	1.446287	9.198538e-05	0	0	0
## 47	43.68849807	0.00215	0.8	1.446287	9.198538e-05	0	0	0
## 48	39.91470896	0.00225	0.8	1.446287	9.198538e-05	0	0	0
## 49	35.37330012	0.00235	0.8	1.446287	9.198538e-05	0	0	0
## 50	31.34843510	0.00245	0.8	1.446287	9.198538e-05	0	0	0
## 51	31.35117386	0.00255	0.8	1.446287	9.198538e-05	0	0	0
## 52	25.63587774	0.00265	0.8	1.446287	9.198538e-05	0	0	0
## 53	21.45928539	0.00275	0.8	1.446287	9.198538e-05	0	0	0
## 54	18.66336269	0.00285	0.8	1.446287	9.198538e-05	0	0	0
## 55	19.13983200	0.00295	0.8	1.446287	9.198538e-05	0	0	0
## 56	14.68609745	0.00305	0.8	1.446287	9.198538e-05	0	0	0
## 57	10.70712245	0.00315	0.8	1.446287	9.198538e-05	0	0	0
## 58	11.47715125	0.00325	0.8	1.446287	9.198538e-05	0	0	0
## 59	8.51546536	0.00335	0.8	1.446287	9.198538e-05	0	0	0
## 60	6.51960105	0.00345	0.8	1.446287	9.198538e-05	0	0	0
## 61	6.03092634	0.00355	0.8	1.446287	9.198538e-05	0	0	0
## 62	5.36889671	0.00365	0.8	1.446287	9.198538e-05	0	0	0
## 63	3.27552485	0.00375	0.8	1.446287	9.198538e-05	0	0	0
## 64	2.97360801	0.00385	0.8	1.446287	9.198538e-05	0	0	0
## 65	1.63744474	0.00395	0.8	1.446287	9.198538e-05	0	0	0
## 66	2.68728486	0.00405	0.8	1.446287	9.198538e-05	0	0	0
## 67	2.04358321	0.00415	0.8	1.446287	9.198538e-05	0	0	0
## 68	3.23761586	0.00425	0.8	1.446287	9.198538e-05	0	0	0

## 69	2.55618957	0.00435	0.8	1.446287	9.198538e-05	0	0	0
## 70	0.84687597	0.00445	0.8	1.446287	9.198538e-05	0	0	0
## 71	0.68003614	0.00455	0.8	1.446287	9.198538e-05	0	0	0
## 72	1.77098300	0.00465	0.8	1.446287	9.198538e-05	0	0	0
## 73	-0.44801608	0.00475	0.8	1.446287	9.198538e-05	0	0	0
## 74	-0.74019664	0.00485	0.8	1.446287	9.198538e-05	0	0	0
## 75	-0.20820845	0.00495	0.8	1.446287	9.198538e-05	0	0	0
## 76	-0.84855083	0.00505	0.8	1.446287	9.198538e-05	0	0	0
## 77	1.30175475	0.00515	0.8	1.446287	9.198538e-05	0	0	0
## 78	0.88177169	0.00525	0.8	1.446287	9.198538e-05	0	0	0
## 79	0.76798229	0.00535	0.8	1.446287	9.198538e-05	0	0	0
## 80	-3.29641326	0.00545	0.8	1.446287	9.198538e-05	0	0	0
## 81	-1.41442202	0.00555	0.8	1.446287	9.198538e-05	0	0	0
## 82	-1.61533851	0.00565	0.8	1.446287	9.198538e-05	0	0	0
## 83	-0.76923618	0.00575	0.8	1.446287	9.198538e-05	0	0	0
## 84	0.28936531	0.00585	0.8	1.446287	9.198538e-05	0	0	0
## 85	0.02913935	0.00595	0.8	1.446287	9.198538e-05	0	0	0
## 86	1.31302951	0.00605	0.8	1.446287	9.198538e-05	0	0	0
## 87	-0.68492843	0.00615	0.8	1.446287	9.198538e-05	0	0	0
## 88	0.18431539	0.00625	0.8	1.446287	9.198538e-05	0	0	0
## 89	-2.22586623	0.00635	0.8	1.446287	9.198538e-05	0	0	0
## 90	0.88992907	0.00645	0.8	1.446287	9.198538e-05	0	0	0
## 91	-0.97916797	0.00655	0.8	1.446287	9.198538e-05	0	0	0
## 92	0.54293288	0.00665	0.8	1.446287	9.198538e-05	0	0	0
## 93	0.42419159	0.00675	0.8	1.446287	9.198538e-05	0	0	0
## 94	1.07216447	0.00685	0.8	1.446287	9.198538e-05	0	0	0
## 95	2.22449855	0.00695	0.8	1.446287	9.198538e-05	0	0	0
## 96	0.68213957	0.00705	0.8	1.446287	9.198538e-05	0	0	0
## 97	0.98976682	0.00715	0.8	1.446287	9.198538e-05	0	0	0
## 98	-1.55661928	0.00725	0.8	1.446287	9.198538e-05	0	0	0
## 99	-2.04850486	0.00735	0.8	1.446287	9.198538e-05	0	0	0
## 100	-0.40856510	0.00745	0.8	1.446287	9.198538e-05	0	0	0
## 101	-0.16777120	0.00755	0.8	1.446287	9.198538e-05	0	0	0
## 102	1.06186150	0.00765	0.8	1.446287	9.198538e-05	0	0	0
## 103	1.51612451	0.00775	0.8	1.446287	9.198538e-05	0	0	0
## 104	0.77186306	0.00785	0.8	1.446287	9.198538e-05	0	0	0
## 105	-1.41924805	0.00795	0.8	1.446287	9.198538e-05	0	0	0
## 106	-1.19090841	0.00805	0.8	1.446287	9.198538e-05	0	0	0
## 107	0.86978160	0.00815	0.8	1.446287	9.198538e-05	0	0	0
## 108	-0.10482927	0.00825	0.8	1.446287	9.198538e-05	0	0	0
## 109	1.33405187	0.00835	0.8	1.446287	9.198538e-05	0	0	0
## 110	1.21573244	0.00845	0.8	1.446287	9.198538e-05	0	0	0
## 111	0.58860990	0.00855	0.8	1.446287	9.198538e-05	0	0	0
## 112	-1.66255929	0.00865	0.8	1.446287	9.198538e-05	0	0	0
## 113	-0.30192756	0.00875	0.8	1.446287	9.198538e-05	0	0	0
## 114	1.26487946	0.00885	0.8	1.446287	9.198538e-05	0	0	0
## 115	0.13801946	0.00895	0.8	1.446287	9.198538e-05	0	0	0
## 116	1.75117179	0.00905	0.8	1.446287	9.198538e-05	0	0	0
## 117	0.06955914	0.00915	0.8	1.446287	9.198538e-05	0	0	0
## 118	-0.89293739	0.00925	0.8	1.446287	9.198538e-05	0	0	0
## 119	-2.36748426	0.00935	0.8	1.446287	9.198538e-05	0	0	0
## 120	-1.79141490	0.00945	0.8	1.446287	9.198538e-05	0	0	0
## 121	0.73518538	0.00955	0.8	1.446287	9.198538e-05	0	0	0
## 122	-0.71282464	0.00965	0.8	1.446287	9.198538e-05	0	0	0

```

## 123 0.74609846 0.00975 0.8 1.446287 9.198538e-05 0 0 0
## 124 0.37603973 0.00985 0.8 1.446287 9.198538e-05 0 0 0
## 125 -0.12243665 0.00995 0.8 1.446287 9.198538e-05 0 0 0
##
## $input$continuous.input
##          C          x por          tort          Ds v dpor_dx dv_dx dDs_dx
## 26 210.21775269 0.00005 0.8 1.446287 9.198538e-05 0 0 0
## 27 196.37905798 0.00015 0.8 1.446287 9.198538e-05 0 0 0
## 28 186.61281200 0.00025 0.8 1.446287 9.198538e-05 0 0 0
## 29 177.18220881 0.00035 0.8 1.446287 9.198538e-05 0 0 0
## 30 167.21812426 0.00045 0.8 1.446287 9.198538e-05 0 0 0
## 31 157.77106794 0.00055 0.8 1.446287 9.198538e-05 0 0 0
## 32 149.40530858 0.00065 0.8 1.446287 9.198538e-05 0 0 0
## 33 137.63418693 0.00075 0.8 1.446287 9.198538e-05 0 0 0
## 34 130.34462143 0.00085 0.8 1.446287 9.198538e-05 0 0 0
## 35 121.88820672 0.00095 0.8 1.446287 9.198538e-05 0 0 0
## 36 114.46055069 0.00105 0.8 1.446287 9.198538e-05 0 0 0
## 37 106.57200305 0.00115 0.8 1.446287 9.198538e-05 0 0 0
## 38 98.14350965 0.00125 0.8 1.446287 9.198538e-05 0 0 0
## 39 93.15457339 0.00135 0.8 1.446287 9.198538e-05 0 0 0
## 40 86.72033940 0.00145 0.8 1.446287 9.198538e-05 0 0 0
## 41 77.59090717 0.00155 0.8 1.446287 9.198538e-05 0 0 0
## 42 74.35113395 0.00165 0.8 1.446287 9.198538e-05 0 0 0
## 43 63.96192903 0.00175 0.8 1.446287 9.198538e-05 0 0 0
## 44 60.51876573 0.00185 0.8 1.446287 9.198538e-05 0 0 0
## 45 53.88879413 0.00195 0.8 1.446287 9.198538e-05 0 0 0
## 46 49.45415907 0.00205 0.8 1.446287 9.198538e-05 0 0 0
## 47 43.68849807 0.00215 0.8 1.446287 9.198538e-05 0 0 0
## 48 39.91470896 0.00225 0.8 1.446287 9.198538e-05 0 0 0
## 49 35.37330012 0.00235 0.8 1.446287 9.198538e-05 0 0 0
## 50 31.34843510 0.00245 0.8 1.446287 9.198538e-05 0 0 0
## 51 31.35117386 0.00255 0.8 1.446287 9.198538e-05 0 0 0
## 52 25.63587774 0.00265 0.8 1.446287 9.198538e-05 0 0 0
## 53 21.45928539 0.00275 0.8 1.446287 9.198538e-05 0 0 0
## 54 18.66336269 0.00285 0.8 1.446287 9.198538e-05 0 0 0
## 55 19.13983200 0.00295 0.8 1.446287 9.198538e-05 0 0 0
## 56 14.68609745 0.00305 0.8 1.446287 9.198538e-05 0 0 0
## 57 10.70712245 0.00315 0.8 1.446287 9.198538e-05 0 0 0
## 58 11.47715125 0.00325 0.8 1.446287 9.198538e-05 0 0 0
## 59 8.51546536 0.00335 0.8 1.446287 9.198538e-05 0 0 0
## 60 6.51960105 0.00345 0.8 1.446287 9.198538e-05 0 0 0
## 61 6.03092634 0.00355 0.8 1.446287 9.198538e-05 0 0 0
## 62 5.36889671 0.00365 0.8 1.446287 9.198538e-05 0 0 0
## 63 3.27552485 0.00375 0.8 1.446287 9.198538e-05 0 0 0
## 64 2.97360801 0.00385 0.8 1.446287 9.198538e-05 0 0 0
## 65 1.63744474 0.00395 0.8 1.446287 9.198538e-05 0 0 0
## 66 2.68728486 0.00405 0.8 1.446287 9.198538e-05 0 0 0
## 67 2.04358321 0.00415 0.8 1.446287 9.198538e-05 0 0 0
## 68 3.23761586 0.00425 0.8 1.446287 9.198538e-05 0 0 0
## 69 2.55618957 0.00435 0.8 1.446287 9.198538e-05 0 0 0
## 70 0.84687597 0.00445 0.8 1.446287 9.198538e-05 0 0 0
## 71 0.68003614 0.00455 0.8 1.446287 9.198538e-05 0 0 0
## 72 1.77098300 0.00465 0.8 1.446287 9.198538e-05 0 0 0
## 73 -0.44801608 0.00475 0.8 1.446287 9.198538e-05 0 0 0

```

## 74	-0.74019664	0.00485	0.8	1.446287	9.198538e-05	0	0	0
## 75	-0.20820845	0.00495	0.8	1.446287	9.198538e-05	0	0	0
## 76	-0.84855083	0.00505	0.8	1.446287	9.198538e-05	0	0	0
## 77	1.30175475	0.00515	0.8	1.446287	9.198538e-05	0	0	0
## 78	0.88177169	0.00525	0.8	1.446287	9.198538e-05	0	0	0
## 79	0.76798229	0.00535	0.8	1.446287	9.198538e-05	0	0	0
## 80	-3.29641326	0.00545	0.8	1.446287	9.198538e-05	0	0	0
## 81	-1.41442202	0.00555	0.8	1.446287	9.198538e-05	0	0	0
## 82	-1.61533851	0.00565	0.8	1.446287	9.198538e-05	0	0	0
## 83	-0.76923618	0.00575	0.8	1.446287	9.198538e-05	0	0	0
## 84	0.28936531	0.00585	0.8	1.446287	9.198538e-05	0	0	0
## 85	0.02913935	0.00595	0.8	1.446287	9.198538e-05	0	0	0
## 86	1.31302951	0.00605	0.8	1.446287	9.198538e-05	0	0	0
## 87	-0.68492843	0.00615	0.8	1.446287	9.198538e-05	0	0	0
## 88	0.18431539	0.00625	0.8	1.446287	9.198538e-05	0	0	0
## 89	-2.22586623	0.00635	0.8	1.446287	9.198538e-05	0	0	0
## 90	0.88992907	0.00645	0.8	1.446287	9.198538e-05	0	0	0
## 91	-0.97916797	0.00655	0.8	1.446287	9.198538e-05	0	0	0
## 92	0.54293288	0.00665	0.8	1.446287	9.198538e-05	0	0	0
## 93	0.42419159	0.00675	0.8	1.446287	9.198538e-05	0	0	0
## 94	1.07216447	0.00685	0.8	1.446287	9.198538e-05	0	0	0
## 95	2.22449855	0.00695	0.8	1.446287	9.198538e-05	0	0	0
## 96	0.68213957	0.00705	0.8	1.446287	9.198538e-05	0	0	0
## 97	0.98976682	0.00715	0.8	1.446287	9.198538e-05	0	0	0
## 98	-1.55661928	0.00725	0.8	1.446287	9.198538e-05	0	0	0
## 99	-2.04850486	0.00735	0.8	1.446287	9.198538e-05	0	0	0
## 100	-0.40856510	0.00745	0.8	1.446287	9.198538e-05	0	0	0
## 101	-0.16777120	0.00755	0.8	1.446287	9.198538e-05	0	0	0
## 102	1.06186150	0.00765	0.8	1.446287	9.198538e-05	0	0	0
## 103	1.51612451	0.00775	0.8	1.446287	9.198538e-05	0	0	0
## 104	0.77186306	0.00785	0.8	1.446287	9.198538e-05	0	0	0
## 105	-1.41924805	0.00795	0.8	1.446287	9.198538e-05	0	0	0
## 106	-1.19090841	0.00805	0.8	1.446287	9.198538e-05	0	0	0
## 107	0.86978160	0.00815	0.8	1.446287	9.198538e-05	0	0	0
## 108	-0.10482927	0.00825	0.8	1.446287	9.198538e-05	0	0	0
## 109	1.33405187	0.00835	0.8	1.446287	9.198538e-05	0	0	0
## 110	1.21573244	0.00845	0.8	1.446287	9.198538e-05	0	0	0
## 111	0.58860990	0.00855	0.8	1.446287	9.198538e-05	0	0	0
## 112	-1.66255929	0.00865	0.8	1.446287	9.198538e-05	0	0	0
## 113	-0.30192756	0.00875	0.8	1.446287	9.198538e-05	0	0	0
## 114	1.26487946	0.00885	0.8	1.446287	9.198538e-05	0	0	0
## 115	0.13801946	0.00895	0.8	1.446287	9.198538e-05	0	0	0
## 116	1.75117179	0.00905	0.8	1.446287	9.198538e-05	0	0	0
## 117	0.06955914	0.00915	0.8	1.446287	9.198538e-05	0	0	0
## 118	-0.89293739	0.00925	0.8	1.446287	9.198538e-05	0	0	0
## 119	-2.36748426	0.00935	0.8	1.446287	9.198538e-05	0	0	0
## 120	-1.79141490	0.00945	0.8	1.446287	9.198538e-05	0	0	0
## 121	0.73518538	0.00955	0.8	1.446287	9.198538e-05	0	0	0
## 122	-0.71282464	0.00965	0.8	1.446287	9.198538e-05	0	0	0
## 123	0.74609846	0.00975	0.8	1.446287	9.198538e-05	0	0	0
## 124	0.37603973	0.00985	0.8	1.446287	9.198538e-05	0	0	0
## 125	-0.12243665	0.00995	0.8	1.446287	9.198538e-05	0	0	0
##								
##								

```

## $output
## $output$J.dif.up
## [1] 8.149375
##
## $output$J.adv.up
## [1] 0
##
## $output$J.dif.down
## [1] -0.004949227
##
## $output$J.adv.down
## [1] 0
##
## $output$R.int
## [1] -8.192326
##
## $output$overview
##      x          C      J.tot      J.dif J.adv      R.vol
## 1  0.00005 208.817505859  8.149375405  8.149375405    0 -2404.6358931
## 2  0.00015 197.906260361  7.909681200  7.909681200    0 -2389.2482033
## 3  0.00025 187.319692591  7.671525764  7.671525764    0 -2373.8605134
## 4  0.00035 177.055711498  7.434909097  7.434909097    0 -2358.4728236
## 5  0.00045 167.112226030  7.199831200  7.199831200    0 -2343.0851337
## 6  0.00055 157.487145136  6.966292071  6.966292071    0 -2327.6974439
## 7  0.00065 148.178377765  6.734291711  6.734291711    0 -2312.3097540
## 8  0.00075 139.183832867  6.503830120  6.503830120    0 -2296.9220642
## 9  0.00085 130.501419389  6.274907298  6.274907298    0 -2281.5343743
## 10 0.00095 122.129046281  6.047523245  6.047523245    0 -2266.1466845
## 11 0.00105 114.064622491  5.821677961  5.821677961    0 -2250.7589946
## 12 0.00115 106.306056969  5.597371446  5.597371446    0 -2235.3713048
## 13 0.00125  98.851258664  5.374603700  5.374603700    0 -2219.9836150
## 14 0.00135  91.698136523  5.153374723  5.153374723    0 -2204.5959251
## 15 0.00145  84.844599497  4.933684515  4.933684515    0 -2189.2082353
## 16 0.00155  78.288556533  4.715533076  4.715533076    0 -2173.8205454
## 17 0.00165  72.027916581  4.498920406  4.498920406    0 -2158.4328556
## 18 0.00175  66.060588590  4.283846505  4.283846505    0 -2143.0451657
## 19 0.00185  60.468051517  4.083374675  4.083374675    0 -2108.2173744
## 20 0.00195  55.029030562  3.860333685  3.860333685    0 -2105.5606600
## 21 0.00205  49.953505018  3.631313061  3.631313061    0 -2082.0870513
## 22 0.00215  45.174645006  3.415853254  3.415853254    0 -2053.4412065
## 23 0.00225  40.707188310  3.197655035  3.197655035    0 -2015.4321481
## 24 0.00235  36.465625942  2.997902131  2.997902131    0 -1988.1030605
## 25 0.00245  32.571801834  2.807648415  2.807648415    0 -1942.4334541
## 26 0.00255  28.782309049  2.603748058  2.603748058    0 -1935.1598645
## 27 0.00265  25.352237193  2.402377390  2.402377390    0 -1903.2678212
## 28 0.00275  22.253884634  2.187983996  2.187983996    0 -1851.3272978
## 29 0.00285  19.440933510  1.985803467  1.985803467    0 -1788.1393736
## 30 0.00295  16.830593382  1.799470304  1.799470304    0 -1732.7353586
## 31 0.00305  14.508964612  1.591511475  1.591511475    0 -1661.2721783
## 32 0.00315  12.512779566  1.415543132  1.415543132    0 -1562.2132838
## 33 0.00325  10.741199133  1.268397583  1.268397583    0 -1459.4649182
## 34 0.00335   9.093181449  1.108544569  1.108544569    0 -1373.8846563
## 35 0.00345   7.690990618  1.010572336  1.010572336    0 -1275.0549505
## 36 0.00355   6.314963523  0.877929965  0.877929965    0 -1213.5491114

```


## 37	0.00365	5.208033871	0.770918076	0.770918076	0	-1126.9406474
## 38	0.00375	4.357610222	0.632214085	0.632214085	0	-1013.8975591
## 39	0.00385	3.560390869	0.534918956	0.534918956	0	-919.2184658
## 40	0.00395	2.832243639	0.439409718	0.439409718	0	-837.5140832
## 41	0.00405	2.191164130	0.367233881	0.367233881	0	-762.2925669
## 42	0.00415	1.583986884	0.306351090	0.306351090	0	-705.2658346
## 43	0.00425	1.066929468	0.243394876	0.243394876	0	-650.7557188
## 44	0.00435	0.750866871	0.231418498	0.231418498	0	-572.5255702
## 45	0.00445	0.512147197	0.190779536	0.190779536	0	-498.4606258
## 46	0.00455	0.244151621	0.154104417	0.154104417	0	-449.4218575
## 47	0.00465	0.110875306	0.101465166	0.101465166	0	-382.5345949
## 48	0.00475	0.019696267	0.114325428	0.114325428	0	-320.4539526
## 49	0.00485	-0.030983589	0.099534041	0.099534041	0	-263.1850622
## 50	0.00495	-0.210110517	0.082024716	0.082024716	0	-247.1286019
## 51	0.00505	-0.245546698	0.085886216	0.085886216	0	-206.6748527
## 52	0.00515	-0.334113161	0.083793478	0.083793478	0	-188.7684708
## 53	0.00525	-0.470039247	0.079898439	0.079898439	0	-190.6743892
## 54	0.00535	-0.486411267	0.063179557	0.063179557	0	-170.9858455
## 55	0.00545	-0.481551740	0.052530263	0.052530263	0	-152.2312867
## 56	0.00555	-0.421592197	0.010436936	0.010436936	0	-123.6587096
## 57	0.00565	-0.347881855	-0.025769270	-0.025769270	0	-92.3188549
## 58	0.00575	-0.396705256	-0.051620631	-0.051620631	0	-90.9287542
## 59	0.00585	-0.393283154	-0.062068411	-0.062068411	0	-78.6647399
## 60	0.00595	-0.439985168	-0.064511039	-0.064511039	0	-81.3561702
## 61	0.00605	-0.409491006	-0.049723667	-0.049723667	0	-69.8591341
## 62	0.00615	-0.328473891	-0.041607480	-0.041607480	0	-50.8130576
## 63	0.00625	-0.210294319	-0.059888622	-0.059888622	0	-24.4185254
## 64	0.00635	-0.114834634	-0.068111519	-0.068111519	0	-2.9314282
## 65	0.00645	-0.045870551	-0.040788946	-0.040788946	0	10.7347408
## 66	0.00655	-0.014894150	-0.039400868	-0.039400868	0	13.7895303
## 67	0.00665	-0.055747757	-0.030224339	-0.030224339	0	-0.6565529
## 68	0.00675	-0.057408373	-0.021813936	-0.021813936	0	-6.7664944
## 69	0.00685	-0.054789843	-0.027715488	-0.027715488	0	-12.3134946
## 70	0.00695	0.133444736	-0.028183675	-0.028183675	0	27.0676832
## 71	0.00705	0.247418089	-0.008344967	-0.008344967	0	48.3928989
## 72	0.00715	0.306985891	0.029780509	0.029780509	0	54.9394786
## 73	0.00725	0.234527164	0.016531164	0.016531164	0	30.3255808
## 74	0.00735	0.155754043	0.024702535	0.024702535	0	5.9088879
## 75	0.00745	0.130711875	0.008352854	0.008352854	0	-4.6506148
## 76	0.00755	0.160107529	-0.010481870	-0.010481870	0	0.1008769
## 77	0.00765	0.272064381	-0.026768629	-0.026768629	0	27.2083306
## 78	0.00775	0.322245260	-0.031658166	-0.031658166	0	41.9543747
## 79	0.00785	0.319713799	0.001652046	0.001652046	0	43.9713248
## 80	0.00795	0.297047693	0.003604329	0.003604329	0	40.8158194
## 81	0.00805	0.245683710	0.024308100	0.024308100	0	31.0884666
## 82	0.00815	0.099779637	0.011621844	0.011621844	0	-0.6907481
## 83	0.00825	0.100824078	0.017476530	0.017476530	0	3.6335279
## 84	0.00835	0.076872967	0.017740911	0.017740911	0	1.6540985
## 85	0.00845	0.052697078	0.017807350	0.017807350	0	-0.3253309
## 86	0.00855	0.028565399	0.017675845	0.017675845	0	-2.3047603
## 87	0.00865	0.004746916	0.017346398	0.017346398	0	-4.2841897
## 88	0.00875	-0.018489383	0.016819007	0.016819007	0	-6.2636191
## 89	0.00885	-0.040874512	0.016093674	0.016093674	0	-8.2430485
## 90	0.00895	-0.062139483	0.015170397	0.015170397	0	-10.2224779

```

## 91 0.00905 -0.082015310 0.014049178 0.014049178 0 -12.2019073
## 92 0.00915 -0.100233005 0.012730016 0.012730016 0 -14.1813367
## 93 0.00925 -0.116523582 0.011212911 0.011212911 0 -16.1607661
## 94 0.00935 -0.130618053 0.009497863 0.009497863 0 -18.1401955
## 95 0.00945 -0.142247433 0.007584872 0.007584872 0 -20.1196249
## 96 0.00955 -0.151142733 0.005473938 0.005473938 0 -22.0990543
## 97 0.00965 -0.157034966 0.003165061 0.003165061 0 -24.0784837
## 98 0.00975 -0.159655146 0.000658241 0.000658241 0 -26.0579131
## 99 0.00985 -0.158734287 -0.002046522 -0.002046522 0 -28.0373425
## 100 0.00995 -0.154003399 -0.004949227 -0.004949227 0 -30.0167719
##
##
## $parms
## $parms$env.parms
## $parms$env.parms$TC
## [1] 10
##
## $parms$env.parms$S
## [1] 35
##
## $parms$env.parms$P
## [1] 1.013
##
## $parms$env.parms$z
## [1] 0
##
## $parms$env.parms$Dmol
## [1] 0.0001330373
##
##
## $parms$continuous.parms
## $parms$continuous.parms$p
## [1] 3
##
## $parms$continuous.parms$bnd.upper
## [1] 1
##
## $parms$continuous.parms$bnd.lower
## [1] 1
##
## $parms$continuous.parms$n.C
## NULL
##
## $parms$continuous.parms$n.J
## NULL
##
## $parms$continuous.parms$n.R
## NULL
##
## $parms$continuous.parms$n.uniform
## [1] TRUE
##
## $parms$continuous.parms$optimal.window.size
## [1] "interactive"

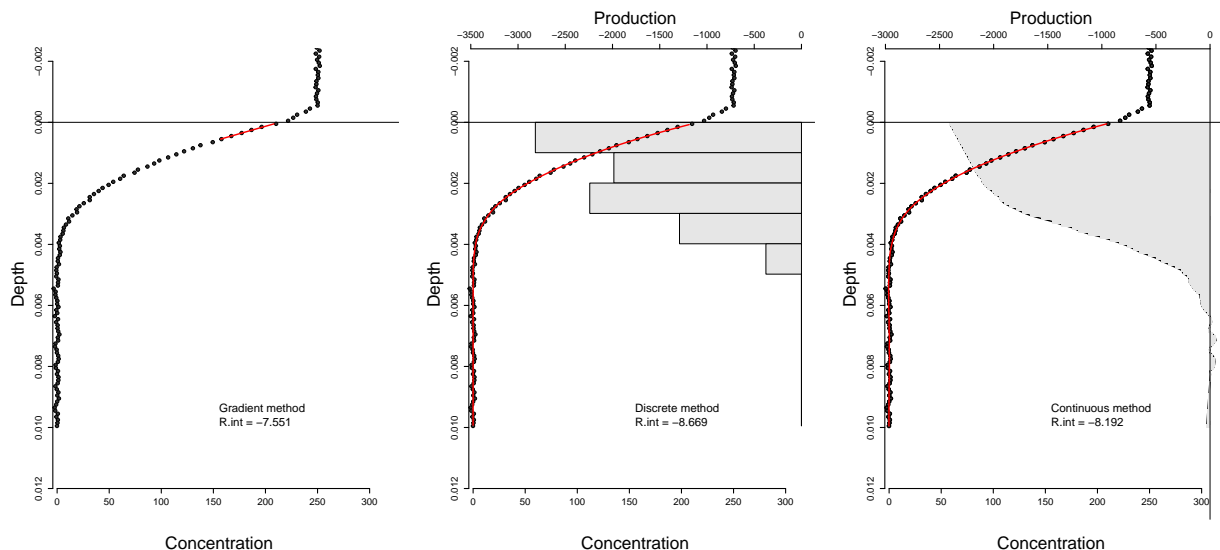
```

```
##
## $parms$continuous.parms$min.n
## [1] 2
##
## $parms$continuous.parms$max.n
## [1] 49
##
## $parms$continuous.parms$keep.graphics
## [1] FALSE
##
## $parms$continuous.parms$interpolation
## [1] "interpolate"
```

We still have the method and input values (the method now obviously shows “continuous”). The input section now also contains a **continuous.input** dataframe, which contains the input as it is processed for the continuous approach. The smoothing procedure needs equally spaced data, so FLIPPER interpolates the supplied data. There are also columns for the first derivatives of the porosity, diffusion coefficient and advection velocity. These are zero by default, but can also be supplied in the user-input dataframe (columns **dpor_dx**, **dDs_dx** and **dv_dx**). The output dataframe looks very similar as when you have run the discrete method, but it now contains continuous profiles for the diffusive flux J.dif, the advective flux J.adv and the volumetric production rate. And then finally there are the parameter lists.

Should you want to run all three methods at the same time, you can set the **method** to “all” or NULL. FLIPPER will then run through all three methods, and supply you with a combination of the output of the individual methods. You can then easily plot all three methods next to each other using the *plot.FLIPPER* function, which allows for a comparison of the different methods.

```
gradient.parms <- c()
discrete.parms <- list()
continuous.parms <- list(optimal.window.size="interactive",n.uniform=TRUE)
test <- FLIPPER.func(input=input.profile,species=c("O2"),por.cte=por.cte,tort.dep=1,
  env.parms=env.parms,
  gradient.parms=gradient.parms,
  discrete.parms=discrete.parms,
  continuous.parms=continuous.parms,
  method="all")
plot.FLIPPER(test)
```



Specific cases

Electro-active sediments

Under construction

Irrigated sediments

Under construction

references

- Berg, Peter, Nils Risgaard-Petersen, and Søren Rysgaard. 1998. "Interpretation of Measured Concentration Profiles in Sediment Pore Water." *Limnology and Oceanography* 43 (7): 1500–1510. doi:<https://doi.org/10.4319/lo.1998.43.7.1500>.
- Boudreau, Bernard P. 1996. "The diffusive tortuosity of fine-grained unlithified sediments." *Geochimica et Cosmochimica Acta* 60 (16): 3139–42. doi:[10.1016/0016-7037\(96\)00158-5](https://doi.org/10.1016/0016-7037(96)00158-5).
- . 1997. *Diagenetic Models and their Implementation*. Springer-Verlag Berlin Heidelberg New York.
- Fick, A. 1855. "Über Diffusion." *Annals of Physics* 94: 59–86.
- Jørgensen, Bo Barker, and David J. Des Marais. 1990. "The Diffusive Boundary Layer of Sediments: Oxygen Microgradients over a Microbial Mat." *Limnology and Oceanography* 35 (6): 1343–55. doi:<https://doi.org/10.4319/lo.1990.35.6.1343>.
- Meysman, Filip J. R., Bernard P. Boudreau, and Jack J. Middelburg. 2005. "Modeling reactive transport in sediments subject to bioturbation and compaction." *Geochimica et Cosmochimica Acta* 69 (14): 3601–17. doi:[10.1016/j.gca.2005.01.004](https://doi.org/10.1016/j.gca.2005.01.004).
- Savitzky, Abraham, and Marcel J. Golay. 1964. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry* 36 (8): 1627–39.
- Thibault de Chanvalon, A., E. Metzger, A. Mouret, J. Knoery, E. Geslin, and F.J.R. Meysman. 2017. "Two Dimensional Mapping of Iron Release in Marine Sediments at Submillimetre Scale." *Marine Chemistry* 191: 34–49. doi:<https://doi.org/10.1016/j.marchem.2016.04.003>.