rSAS exercises with `rsas.py`

These exercises will help you understand and learn to think about rSAS functions. rSAS is a powerful tool, but is not usually very intuitive at first. By working through these and satisfying yourself that you understand them, you will build up this intuition.

Before you get started, be sure to join the google group at
https://groups.google.com/forum/#!forum/rsas_users

Uniform sampling, steady-state

You should have seen (and perhaps attempted yourself) the analytical solution for the uniform sampling steady-state case. The rSAS function in this case is a uniform distribution, and the transit time distribution is an exponential decay.

$$p_Q(T) = \frac{1}{T_0}\exp\left(-\frac{T}{T_0}\right)$$

Where $T_0 = S_0/Q_0$ and S0 and Q0 are the steady-state storage and flux. Set up a spreadsheet with the following columns and 1000 rows, populated with the following values:

- `timestep` : a list of integers indicating the timestep, starting from 0 and ending with 999
- `datetime` : a timestamp
- `J` = 3 mm/day for all days
- `Q` = 3 mm/day for all days
- `C_J` = 0 for all days, except on day 10, 400 and 800, when C_J = 100
- `ST_min` = 0 for all days
- `ST_max` = 210 mm for all days

Name this spreadsheet `uniformsteady.csv` and save it in the same directory as `ez_rsas.py`. Now in Anaconda, change to the directory with this file, and run the following code at the prompt:

```
run ez_rsas.py uniformsteady.csv uniform
```

Alternatively, at a shell command prompt you can run

```
python ez_rsas.py uniformsteady.csv uniform
```

For all the examples below, just replace `run` with `python` if you want to run from the command line / shell rather than from the IPython interpreter. This command asks python to run `ez_rsas.py` on the `uniformsteady`.csv file, assuming a uniform distribution for the rSAS function.

Assuming it runs without errors, you should now be able to reload `uniformsteady.csv` in excel and find that a new column has appeared. This column is the predicted concentration of the outflow.

Make a new spreadsheet (say, `all_exercises.xlsx`) to store the predicted concentration timeseries. You will be comparing this with other more complex cases.

Some things to try:

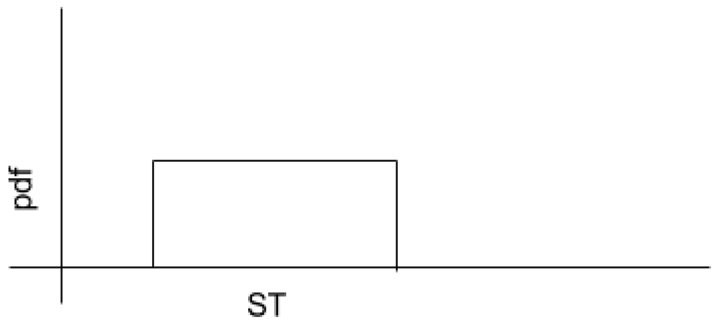
1. Compare the predicted concentration with the one predicted by the analytical result. The input is (approximately) a single ‘spike’ of input (why only approximately so?). The analytical result predicts that the outflow concentration in this case should be:
   $$C(t) = C_0 \exp\left(-\frac{t-t_0}{T_0}\right)\Delta t/T_0$$
   where t_0 is the time of the input (10, in this case), C_0 is the input concentration. The parameter T_0 is equal to $T_0 = S_0/Q_0$, where S_0 and Q_0 are `ST_max` and `Q` respectively.
2. Try changing the storage and the flow rates. Why does the flushing of the system depend only on the ratio of storage and flow rate, rather than on either one’s actual value?
3. Try making T_0 much shorter by changing either the storage or flow rates (perhaps just a few days) and re-run the code. How close to the analytical solution do you get? Try increasing the accuracy of the numerical solution by increasing the number of ‘substeps’ the solution takes from 1 (the default) to 3 or larger:
   `run ez_rsas.py uniformsteady.csv uniform --n_substeps=3`
4. What happens when you make the input concentration 100 for all days following day 10? Compare the result with the analytical solution:
   $$C(t) = C_0\left(1 - \exp\left(-\frac{t-t_0}{T_0}\right)\right)$$
   where C_0 = 100
5. The parameter `C_old` controls the assumed background concentration. Try setting `C_J` to zero for all timesteps, and set `C_old` to 100 by calling the code using the following command:
   `run ez_rsas.py uniformsteady.csv uniform --C_old=100`
   What analytical result would you compare this with?
6. Try making `C_J` a sine wave with a period of 1 year and an amplitude of 20. What does the output concentration timeseries look like? How is it different from the input?
7. Let’s say you don’t know what the storage volume is, but you have data showing that the precipitation oxygen-18 concentration timeseries follows the sinewave just described, but the discharge oxygen-18 concentration has an amplitude of only 2. If the flow rate it still 3 mm/day, can you work out what the storage volume must be? What is the mean transit time?

The last exercise mimics a typical way transit time distributions have been used in the past to understand catchment storage volumes and residence times.

Non-uniform sampling, steady-state

Now we're going to see what happens when the sampling isn't uniform.

A simple non-uniform case is simply a uniform distribution 'shifted' away from zero, like this:



Reset the spreadsheet you made before (save it as `Exercise_2.csv`), but now change the ST_min values to be 90 mm and the ST_max values to all be 300 mm. Once you re-run the code you should see the tracer doesn't break through until around timestep 40. Why not? (Hint: think about the meaning of ST_min)

To look at a more complex non-uniform rSAS function we will use a distribution called a [Kumaraswami distribution](), which has the benefit of having a simple equation:

CDF: $\Omega(S_T) = (1-(1-x^a)^b$ 喵喵喵？ lack a bracket

PDF: $\omega(S_T) = abx^{a-1}(1-x^a)^{b-1}$

where:

$x = \frac{S_T - S_{Tmin}}{S_{Tmax} - S_{Tmin}}$

Try the following:

1. First, skim over the Wikipedia page and find the equation for the CDF and PDF of this distribution. Make an excel spreadsheet that plots the PDF and allows you to vary the values of parameters `a` and `b`.
2. When `a=1` and `b=1`, the Kumaraswami distribution reduces to the uniform distribution. Keeping `b=1`, see how varying `a` affects the shape of the distribution. What effect do you think this will have on the age sampling?
3. Try to think of a physical system in which young water is preferentially removed relative to old water. Choose a value of '`a`' to represent this system. Try this value out

by adding an ‘a’ and a ‘b’ column to `Exercise_2.csv` and running ez_rsas.py using the following command:
`run ez_rsas.py Exercise_2.csv kumaraswami`

kumaraswami function doesn't have invcdf attribution?

4. Do the same for a system where old water is preferentially removed. Compare your results with the uniform case.

Non-uniform infinite-store sampling, steady-state

Sometimes it can be reasonable to approximate the shape of the rSAS function using a function that does not have an upper bound. A convenient probability distribution with this property is the gamma distribution.

Try the following:

1. First, skim over the gamma distribution Wikipedia page. Make an excel spreadsheet that plots the PDF (use can the function GAMMA.DIST) and allows you to vary the values of the `shape` parameter alpha and the `scale` parameter beta.
2. Why do the shape and scale parameters have these names?
3. When the shape parameter alpha=1, the distribution reduces to an exponential. Recall from the problem set you worked on earlier what the corresponding transit time distribution will be.
4. Make a new spreadsheet `Exercise_3.csv`, and add columns for the `shape` and `scale` parameter. Set `scale` to 210 mm, and `shape` to 1, and set `ST_max` to `inf`. Run using the command:
`run ez_rsas.py Exercise_3.csv gamma`
5. Compare the shape of this with the uniform rSAS shape. How are they different? You may want to plot them in log-log axes to look at the shape of the tail of the distribution.
6. Try varying the value of alpha to 0.5 and 2. What happens to the shape of the gamma distribution? What happens to the initial breakthrough from the ‘spike’? What about the tail?

External variability

There are two kinds of variability in transit time distributions: External and Internal. External variability refers to the case where transit time distributions vary in time, but only because the flow through the system is changing, not because the character of the transport is changing. This corresponds to a fixed rSAS function.

Choose one of the cases you explored before, and change it in the following ways:

1. Add several more ‘spikes’ of input later in the timeseries
2. Set the values of J and Q so that they increase gradually from 1 to 6 over the 1000 timesteps

What happens to the breakthrough curves? How is the transit time distribution changing over time?

Now try constructing some wacky flow rate (but always keeping `J` = `Q`), perhaps including some sine waves and random values (but no negative flow rates). You can get interesting results using “`= 3 + 1/RAND()`.” What do the breakthrough curves look like now?

Internal variability

Internal variability refers to variability in transit time distributions that cannot be accounted for by variations in the overall flow rate, and is instead the result of variations in the way older and younger water is sampled. It can be represented by time-variability in the rSAS function.

Go back to the case of a gamma rSAS function and set alpha=0.5, and construct a seasonally varying C_J using a sinewave. Remember that in this case the flow rate is constant at J=Q=3 mm/day.

Now, make the value of the shape parameter vary seasonally too. You can do this any way you like, but don’t allow the scale parameter to become negative. You might want to try. If it becomes too small your solution may also become inaccurate unless n_substeps is large (which will require lots of computation time!).

Uniform sampling, linear reservoir

In many cases of interest both internal and external variability will exist. A simple example is a well-mixed linear reservoir receiving intermittent inputs. Imagine a tank of water such that

- Inflows occur at random times and of random amounts (or more precisely in the example used here, flow into the tank is a marked poisson process of events with exponentially-distributed magnitudes)
- the contents are being continuously stirred (so uniform sampling of water of all ages occurs)
- Discharge is proportional to storage above an orifice, so $Q(t) = (S(t) - S_0)/t_c$, where tc is the ‘recession timescale’ parameter, and discharge goes to zero when storage approaches the minimum storage S0.

To model this, open the spreadsheet entitled “`LinearReservoir.xlsx`”. The “parameters” sheet lists some parameters that control the nature of the system:

- Minimum storage : this is S0, the volume of the tank when discharge = 0
- 1 / Rainfall frequency : the average time between rainfall events
- Mean annual rainfall : in mm

- Recession timescale tc : controls the time taken for the tank to drain back the minimum storage

Try changing these and look at the effect on the discharge in the “timeseries” sheet. Note that the minimum storage parameter only affects the value of ST_max.

Save the “timeseries” sheet as `linres.csv` and run it using:

```
run ez_rsas.py linres.csv uniform
```

Try comparing the result when the minimum storage is 10 mm, 100 mm, 1000 mm

Uniform sampling, linear reservoir - PQ, ST

We are going to take a deeper dive now and look at the structure of the transit time distributions and age-rank storage predicted by the model. To do this, reset the parameters of “`LinearReservoir.xlsx`” to something sensible, export `linres.csv` and run it using:

```
run ez_rsas.py linres.csv uniform --save_arrays
```

You should now see two more .csv files have been generated, linres_PQ_0.csv and linres_ST.csv. The first contains the time-varying (cumulative) transit time distribution, and the second the time-varying age-ranked storage. Open these and transfer the sheets into “`LinearReservoir.xlsx`”.

Go to PQ and set the zoom to 20%. Select all, and use conditional formatting by color scale to see the structure of the values in the sheet. Do the same for ST.

Rows in these arrays correspond with ages. Columns correspond with timesteps.

Make a new sheet in the workbook. Put the number 500 in cell G5. Create the following columns:

- Column A : copy of Column A from the timeseries sheet
- Column B : column header is “ST”, second row is “`=OFFSET(linres_ST.csv!$A$1,A2,$G$5)`”. Fill this down.
- Column C : column header is “PQ”, second row is “`=OFFSET(linres_PQ_1.csv!$A$1,A2,$G$5)`”. Fill this down.
- Column D : column header is “pQ”, second row is zero, third row is “`=C3-C2`”. Fill this down.

The value in cell G5 determines which timestep is copied from the two arrays and presented in the columns.

Make a plot with ST on the horizontal, and PQ on the vertical. What is this a plot of? Does this look like you expected?

Make a plot with timestep on the horizontal and pQ on the vertical. Explain what you are looking at.

Change timestep from 500 to 501, 502, etc, and see how the rSAS function and transit time distribution change in time.

Inverse storage effect

The inverse storage effect is a phenomena that appears to be common in hydrologic systems. It is a type of internal variability that is “inverse” in the sense that it is the opposite of the internal variability of the well-mixed linear reservoir.

In the case above, you’ll notice that the distribution becomes wider and flatter under wet conditions, and narrower under dry conditions. Consequently the most recent, say 10 mm that entered the system makes up a large part of the discharge at low storage, and a small part at large storage.

Hydrologic systems appear to do the opposite, in that they tend to preferentially release young water under wet conditions, and old water under dry conditions.

You can reproduce this by altering “`LinearReservoir.xlsx`” so that ST_max varies as S0 - Delta S, (without letting the value of ST_max become 0 or negative).

You might also try it with a gamma distribution by setting the scale parameter to be a linear function of storage with a negative slope.