Bo Svensmark, svensmark@plen.ku.dk
Analytical Chemistry, PLEN, KU, 2015 - 2021

Tutorial for the Excel spreadsheet ChromSimHPLC and ChromSim_Fortran (Lab Exercise 2 Gradient HPLC)

You must have a basic understanding of the Linear Solvent Strength model (LSS) in order to use this spreadsheet. It can be used to predict chromatograms for different gradients from isocratic data.

There are 2 models in *ChromSimHPLC*:

1) **Linear model (Excel):** Calculation of chromatograms in Excel using the Linear Solvent Strength model. The inputs are *log(kw)* and *S,* e.g. the results from the Lab exercise in isocratic HPLC, for all compounds. Two different chromatograms can be calculated for different gradients and/or isocratic conditions on the same column.

2) **Polynomial model (Excel + VBA-Macros)** Calculation of chromatograms in *Visual Basic for Applications* by digital simulation of a Nonlinear or a Linear Solvent Strength model using a polynomium of 1st, 2nd or 3rd degree. The input is a table with *log(k)* as function of eluent strength (ϕ). These values can be from the isocratic runs in the Lab exercise 1 in HPLC. The nonlinear simulation in VBA can be compared to the linear model in Excel.

There is another workbook *ChromSim_Fortran.* The only difference is that the simulation of the polynomial model is faster than with VBA, but this version do not work on Mac computers. See section 5) for Windows computers.

All user input are in white cells. The colour in the following text refers to table headings and borders, not the cells themselves.

1) Predict retention times for two different gradients (Linear Excel)

This is the simplest function, and is very fast. But – the predictions will only be accurate in the case where *log(k)* vs ϕ is strictly linear in the whole range of elution strengths during the gradient.

To use this function:

1) **Enter *names of the analytes*, *log(kw)* and *S* in the Table 1**. These data can be copied directly from the spreadsheet for the isocratic exercise cells B37:E44 in *sheet LSS Calculations* (B36:E44 for Thiourea is a t_M marker and should not be copied here)..
2) **Enter the relative concentrations in Table 1, cells B4:B14**. Usually use 1 for a compound and 0 for rows without data. You can also use this to remove or amplify a compound for peak identification.
3) **Enter the run conditions** in Table 2. You may use t_M = 1 until you know t_M. For the Lab exercise, use the retention time of Thiourea as t_M. Enter the intensity of the dead time marker for both gradients in cell B42 (0.1 is a good value).

4) **Enter the parameters for two gradients in Table 3.** The meaning of the parameters should explain them self. Note though, that only a single linear gradient is allowed, but you may add an isocratic part of ϕ-*start* by adding the isocratic time to the dwell time and input the sum in t_D. Also if *tmax* is > t_G, an isocratic part is added in this time at ϕ-*end*. You can see the gradients in the plot below the tables. If ϕ-*end* = ϕ-*start* an isocratic run will be calculated and t_D and t_G are not used. The intensities and offset are used to amplify and position the chromatograms in the graph. Try different values to see the effect.
5) **Finally enter the retention times for a reference chromatogram Table 1, t_R(ref)**. If you do not have a reference chromatogram (e.g. experimental values) you can set the green chromatogram as a reference by use of the bottom above the graph. To remove the reference, just delete the values in Table 1.
6) **The chromatograms can be seen in the graph**, and the retention times, peak widths, relative retention etc. in Table 4. LSS parameters are shown in table 8 further to the right in cells AG29:AN55.

Please note that the peak order in the tables may be different from the elution order. In this case it is advised to sort the input according to the elution order – so press the bottom for sorting of the input.. Sorting of the input is according to the retention times in the reference chromatogram.

Now you can try to change the input and see the results immediately. Try making a gradient for the blue chromatogram. Start with the same parameters as the green chromatogram and try different variations. You may also play with the other parameters for the column, intensity and offset.

If you want to optimize the gradient you may get help by looking in the plots of *log(k)* vs ϕ to the right of the table. Note that the gradient retention factor *k** (or the retention factor *k'* for isocratic runs) are shown in the plot, and this may be used to optimize the gradients – and/or to understand how the LSS model works.

2) Examples of linear LSS

In cell Y1:AM15 in sheet *Models* there are examples of input values for 4 different cases: the regular and the irregular standard samples from the textbook, a simple case and a dataset from a previous course. You can copy this input to the appropriate input cells C2:F12 in the sheet *Chromatograms* and try various gradients on these examples.

Start to copy the input from the isocratic Lab Exercise AH4:AL14 in *Models* to Table 1 (remember the reference tR(ref))and start with these gradients for the green and blue chromatogram:

Table 2

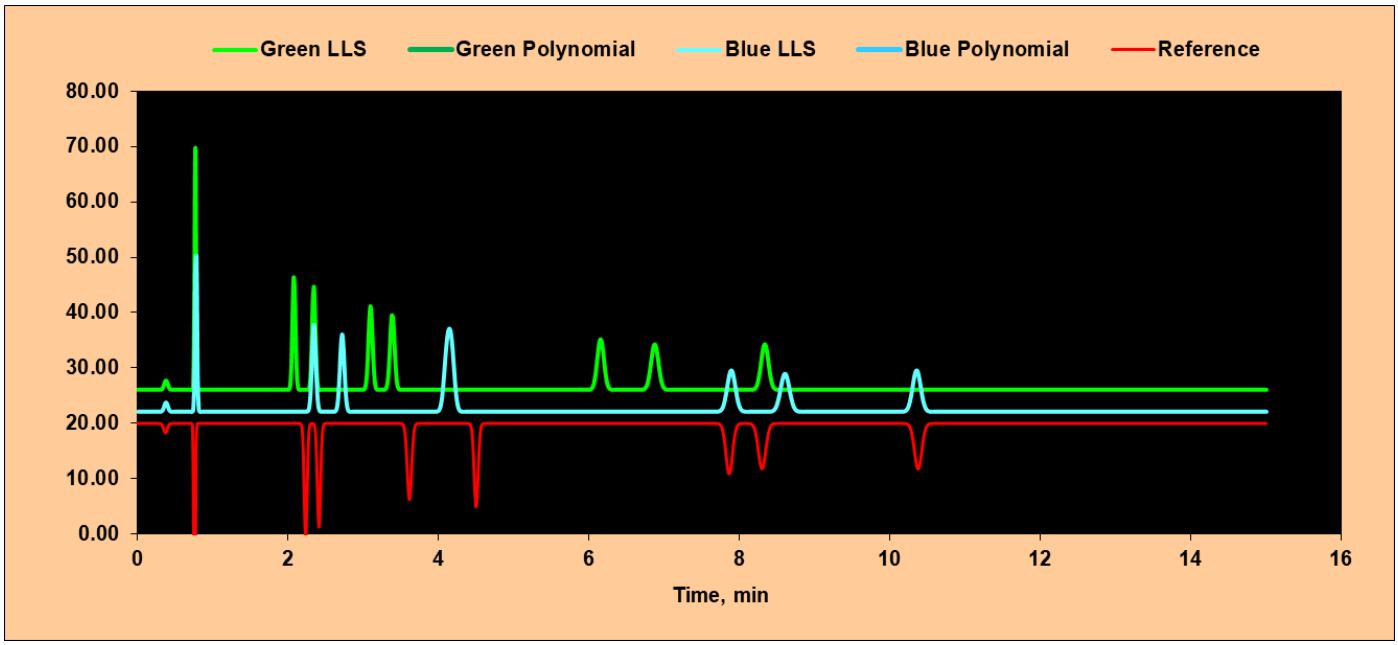
HPLC Settings	
tM, min	0.374
N	4000
tmax, min	15
tM intensity	0.1
4/sqrt(N)	0.063
dt	0.015

Table 3

Gradients	ϕ start	ϕ end	tD	tG, min	Δϕ	Degree	Offset	Intensity
Green LLS	0.2	0.6	0	12	0.4	1	26	1
Green Polynomial						1	25	0
Blue LLS	0.2	0.6	0	12	0.4	1	22	1
Blue Polynomial						1	30	0
Reference							20	-1

You will see that the peaks are not shown in elution order (see the Table 4). To correct this, press the bottom at T27: *Sort input…..* Peak 4&5 are not in the same order in the prediction and the reference – this may be due to a wrong peak assignment in the reference or to imperfections in the calculations.

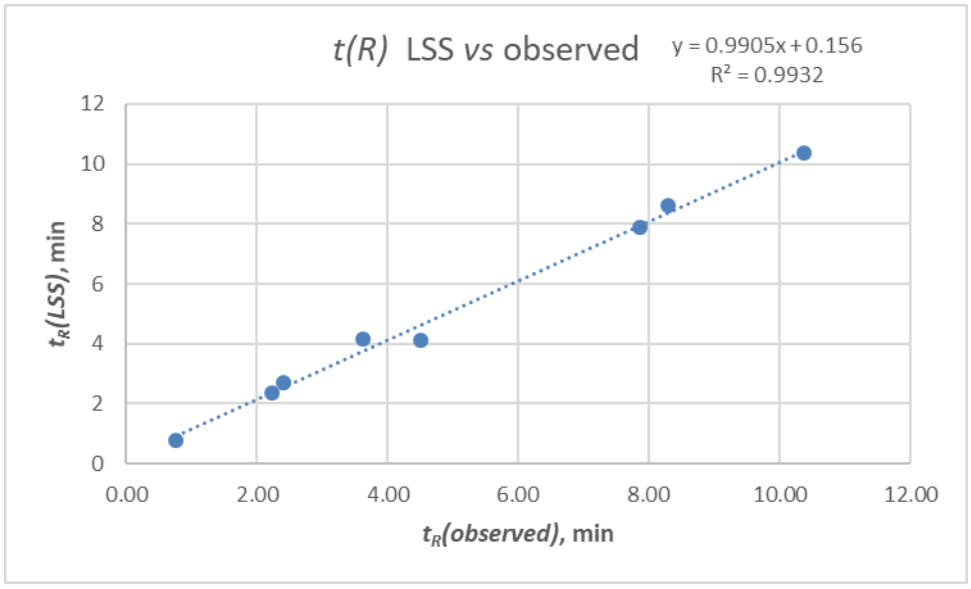
As you can see, the late eluting peaks do not fit too well. This could be because the calculation uses t_D =0, but in reality there is a dwell time >0. Try different t_D-values and see the result (blue chromatogram). In the end you should get results like this:



Why are the late eluting peaks affected more by the dwell-time, than the early eluting peaks?

When you compare the predicted and experimental chromatogram in the graph, it may not look too good (the blue chromatogram is the best, but still peak # 4-6 are quite different):

But if we plot the predicted vs the observed retention times, it looks much better:



So in average the predictions are quite good: the trendline (solid black) is very close to theory – but for closely spaced peaks it may be bad (see the blue chromatogram), even for R^2 = 0.9932. This could be much better using the polynomial model, see next section.

Advanced users: In cell D70:G80 there is a scheme for a *What-If-Analysis*, i.e. you can get many results for variation of a single input parameter. As an example, you could see how the retention of 3 peaks changes with dwell time. It is a little complicated, but just follow the instruction in the spreadsheet and/or use the help function in Excel for *Data Table.*

3) Predict retention times by a polynomial model (Excel + VBA)

This function is slightly more complicated. One simulation takes from 5 seconds to a few minutes depending on the computer power and the accuracy you need. But – the predictions can be much more accurate in the case **when *log(k)* vs ϕ is not strictly linear** in the whole range of elution strengths during the gradient.

To use this function:

1) In the sheet *Models* build the polynomial models from experimental data for *log(k)* as function of elution strength ϕ in isocratic HPLC.
    a. Enter *Analyte names, Log(k) and* ϕ in the table at the top (*Analyte names* in A8:A18, ϕ in C7:V7, and *Log(k)* in C8:V18).
    b. Click on the button *Update Polynomial Models.*
    c. Look at the polynomial fit below and choose the degree of the polynomial to use in the sheet *Chromatograms*.
2) Go to the sheet *Chromatograms* and calculate the chromatograms.
    a. First copy the *Analyte names*, *log(kw)* and *S* to the table on top of the page by clicking the button *Get Anayte names, log(kw)….* .
    b. Enter or correct the *Run conditions* and *Programs* for the Green and Blue chromatogram (and look at the results of the linear model in Excel, the first graph).
    c. If the elution order are different from the order of the analytes in the input table, first sort the analytes in the sheet *Chromatograms*, and then – **important** – go to sheet *Models* and rearrange the input in the table A8:V18 in the same order – and make the model again (press *Update Polynomial Models).*
    d. Go to Table 3 and enter the degree of the polynomial for the polynomial fit.
    e. Click on the *Update buttons* below Table 3 to start the simulation. The calculations may take from 5 s to several minutes.
    f. The progress can be seen in the lower left corner of the window. If it takes way too long, press ESCAPE and then click on Ok in the message box.
    g. Look at the results in the graph and in Table 5 and 7 to the right.

The computer runtime in VBA is proportional to the number of theoretic plates squared, N^2. Unfortunately, the accuracy of the simulation increases with the number of theoretic plates, so a longer runtime gives a better result. For a realistic value of *N*, say 10,000, the simulation will take a very long time (days). Now the retention times are almost independent of the number of theoretic plates, so $N = 200$ is used in the simulations, since this gives retention times with an accuracy of around 1 % at a relative short computer time. The peak width from the simulation is corrected to reflect the number of theoretic plates given in the sheet *Chromatograms*.

If you want to make a more accurate simulation, you can go to the sheet *Macro input*, and change *N* = 200 in cell A2 for the Green chromatogram and/or in cell L2 for the Blue chromatogram to 500 or 1000.

Advanced users: The VBA-Macros can simulate any gradient you like, also nonlinear and step functions. Just enter the time and the eluent strength (Phi) for the program in column E:F or P:Q.

4) Example of a polynomial model

Please try the example in the sheet before you enter your own input for the real Lab Exercise 2 in gradient HPLC. The example is real experimental data made by Simon Stevns Jakobsen, for UPLC of a set of analytes with a good set of *log(k)* vs ϕ for the whole range of ϕ. The input, Run conditions and Gradient Program can be seen in the sheet *Models* (on top of the sheet to the very right). There is also a column with the experimentally observed retention times $t_R(obs)$. Copy-Paste these experimental values from the sheet *Models* to the reference chromatogram in Table 1 in sheet *Chromatograms*.

After you have made the polynomial model and calculated the chromatogram with the Linear model in Excel (using the new experimental run conditions and gradient), try the following:

1) Set the degree in K40 = 1 and press the *Update Green* button. Now the VBA macros calculates the chromatogram with a 1st degree polynomial, i.e. a linear fit, which may be used to check the accuracy of the simulation. If you look in the tables, you will see small differences at the last digit of some retention times between the Table 4 from Excel and the Table 5 from VBA. You may try to increase the number of theoretic plates in the simulation (cell A2 in sheet *Fortran input*), and see the effect on the accuracy of the retention times – and on the computer runtime (VBA CPU-time is shown below the Table 7 together with the number of loops in the simulation).
2) Compare the calculated retention times to the observed retention times in the reference chromatogram. You will see that they do not fit very well. The average difference, Δt_R(estimated-reference), are shown below the Table 7.
3) Change the polynomial degree in K40 to 2 and calculate the chromatogram with a quadratic fit. You should see that this fits the experimental values much better.
4) Try finally a polynomial fit of 3rd degree. Is it better or worse that the quadratic and the linear fit?

Now you should be able to use both the Linear and the Nonlinear Solvent Strength model for predicting retention times, and to compare ChromSimHPLC's estimates with real chromatograms. This workbook can also be used for the theoretic chromatograms in the chromatographic part of the course. Have fun!

5) Predict retention times by a polynomial model with (Excel + Fortran)

Simulation of non-linear LSS in Visual Basic is somewhat slow, but should work on both Windows and Mac computers. A faster version using Fortran is available for windows: *ChromSim_4.0_Fortran.xlsm.*

The two versions (VBA and Fortran) are identical from the users point of view and gives exactly identical results[1], but the Fortran subroutine *ChromSim.dll* must be installed on the computer in order to use the Fortran version.

It is very simple to install the Fortran subroutine *ChromSim.dll* on your computer:

1) Make a folder ***F*** in the root of disc C: **C:\F**
2) Copy the file *ChromSim.dll* to this folder.

The subroutine is only ~134 kB. You may place this subroutine in another folder if you wish, but then you must correct the declaration before Macro 1 in VBA in Excel accordingly.

The difference in speed can be seen here, for an Intel i-7 2.6 GHz Windows 10 PC):

CPU-time	VBA	Fortran
n	200	500
Loops	1.8E+07	1.1E+08
s	17.9	8.7
µs/loop	1	0.08

In addition there is a ~5s overhead for call of the Fortran dll.

[1] The default number of theoretic plates *N* in the simulation is 200 for the VBA-version and 500 for the Fortran-version. In order to get identical results, correct 500 to 200 in cell A2 in sheet *Fortran input* otherwise the results from Fortran will be slightly more accurate and slightly different from the results from the VBA-version.