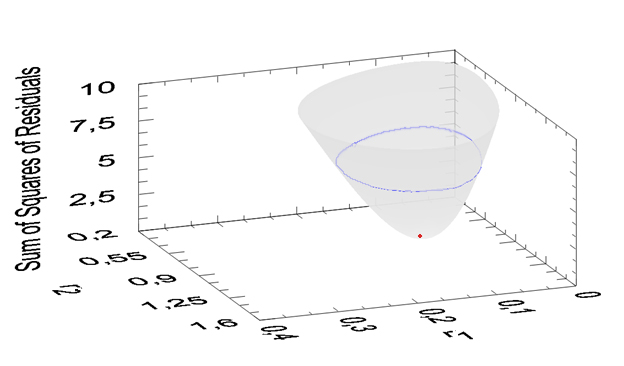
Description of Python code to determine reactivity ratios



*Sum of squares of residuals space (SSR in gray), optimum value of reactivity ratios (red) and 90% confidence limits (JCI) as a contour-line at a certain height in the sum of squares space (blue).*

Disclaimer: This code is open source software and no guarantee is given that it works correctly.

This Python code applies the IUPAC supported approach for the determination of copolymerization reactivity ratios using polymer compositional data and the terminal model. The files it reads are comma separated files contain *f0-X-F-*∆*F* data*.*

It is advised to prepare all data in Excel in the form of comma separated files (.CSV) where the last column is always the error in *F* (save as CSV MS-DOS option in Excel). Currently up to 200 datapoints can be loaded (easy to increase in the Python code).

You can run the program outside of a Python editor as long as you have a Python executable on your computer. For example you can prepare a .BAT file in Notepad (save with this extension under the selection of all files, not .txt file), something like this:

@echo off

cd "C:\Users\amvhe\Desktop\IUPAC\Pythoncode\"

"C:\Program Files\Spyder\Python\python.exe" "IUPAC f0XF method in PythonV15.py"

Pause

The first path is to the folder where your Python code is located, the second path is directly to the Python executable, followed by .py file. You might need to activate the python command in the command mode. If you run the program from a .BAT file, the program waits at the charts, you have the option there to change the axis and save the file in any format (eg TIFF). After clicking the X corner of the graph, the code runs further.

The data are sorted according to *f*0 and conversion for you to better spot systematic trends in the residuals. There also the 3D residuals plot can help.

The *f0-X-F* plane is analysed. The experiments are set-up by selecting an initial feed fraction of the monomer (*f*0) and performing a copolymerization experiment. The cumulative copolymer composition (*F*) is analysed and the total molar monomer conversion (*X*, for both monomers together and values between 0 and 1) at the moment of copolymer isolation is determined. An error in *F* needs to be estimated (∆*F*) and entered into the calculation, this value is used as a weighing factor in the fit. Alternatively one can use uniform weights of all data (choice of weighting scheme).

The IUPAC working group advises experimentalists to run several experiments at different *f*0 values and always determine the conversion. The data of *f*0-*F* (low conversion) experiments can still be used here, assuming that the conversion is also measured and present in the .CSV file. Even zero percent conversion can be dealt with (although practically not feasible). Also the high conversion experiments can be used, when *F* has been determined (or calculated from *f*) and several experiments at different *f*0 values can be included.

With the IUPAC supported approach, datapoints on the *f*0-*F*-*X* plane are used to determine the reactivity ratios and, as long as the conversion is known, low and high conversion experiments can be used and mixed.   
Regarding higher conversion data, one has to be aware that sometimes one of the two monomers is completely consumed. These data could be taken out of the set because now homopolymer is produced only and no new information is added to the dataset.

Which error scheme is to be used?  
If all the data have equal weight, uniformous weighing should be used, but in general it is preferred to use weighing by the error in each individual *F*.

If the error in *X* is substantial and we are not dealing with *F*-values calculated from *f* with proper error propagation (see below), the error in all variables method should be used (EVM).

In some case one has several experiments at different *f0* where one is following the feed composition with conversion (*f-X*). Those data can be converted to *f0-X-F* data with a separate Excel sheet. The errors in *f* (and *f0*) and *X* each are propagated into an error in *F* which needs to be taken into account in the fit as a weighing factor (individual error per point). As the errors in *X* and *f* are already error propagated in an error in *F*, EVM should not be used.

The program uses numerical integration of the copolymerization equation and visualization of the sum of squares of residuals to find the optimal values of the reactivity ratios1,3. Each measurement is weighed in the fit with the use of the error in *F* (*∆F*). It can take minutes to display the solution, due to the numerical approach. One could start with a relatively large search range for *r*1 and *r*2 and then zoom in nearer to the minimum. In that way the joint confidence interval and the *r*-values get more accurate The errors are often not symmetric around the optimum because with our visualization approach we determine the true shape of the joint confidence interval and not the ellipsoid approximation.

The calculated table contains the calculated F and the residuals. If several points show an error larger than estimated by the user or the sum of squares of residuals value is bigger than the sum of squares of residuals of the fit, the fit is not adequate (or the errors are estimated too small and could be reconsidered). The 95% probability Joint Confidence Interval (JCI(95%)) is shown and can be saved in a TIFF file. Furthermore the residuals (*F*measured-*F*calculated) are shown in a 3D plot and can also be saved as a TIFF file. Systematic trends in this surface can be used to identify certain issues. The sum of squares of residuals surface can be saved also if desired (question at the end). The JCI is calculated with the χ2 Chi squared distribution, assuming that the errors in F are determined properly. If certain data related to a *f*0 value show strong deviations, you can revisit this *f0* value and see whether a mistake was made. If the data originate from *f0-f-X* data the individual experiments can be investigated for a small deviation in *f0* .   
It is very important to have accurate values of *f10*. Small errors can create a systematic error in the dataset. Even variation within experimental error of *f10* has an effect on the resulting reactivity ratios4.  
If a dataset based on only one *f10* value is selected, the option is given to optimize *f10* around the initial value (looking around this value ±0.005). The best *f10* value is then used to calculate the reactivity ratios. If the set is obtained after conversion from X-f data, this method should not be used, but instead the original X-f data should be optimized for *f10*.

The best procedures to determine reactivity ratios are shown below:

Start from *f0-f-X-ΔX* data (for example NMR experiments)

Optimize *f0* for each experiment.

Start from *f0-X-F -ΔF* data

Convert to *f0-X-F-ΔF* data, taking errors in *f, f0* and *X* into account (use Excel file).

Do a first fit and check the residuals. If needed vary *f0* for deviating experiments.

Trends in residuals?

Yes

No

Residuals mostly smaller than estimated errors?

No

Yes

Do a first fit and check the residuals (graph).

If not

Optimize f0 in single f0-f-X experiments again for deviating experiment(s)

Trends in residuals?

Yes

No

Adjust errors if reasonable

Residuals mostly smaller than estimated errors?

No

Yes

If still not

Adjust errors if reasonable

If not

**Starting from *f0-f-X-ΔX* data:**

After optimizing for *f10* for each individual experiment (see above), for a set of experiments of the same system one can convert to *f0-X-F-∆F* data (using equation 1 for *F* and equation 2 for *∆F*), to fit multiple experiments at the same time. If *f10* is changed, redo the conversion. An Excel file is supplied using these equations

When the error in *f*10 is also taken into account  should be added on the left under the square root and on the right hand side under the square root (provided in the Excel file if needed).

**Statistical background:**

The sum of squares (ss) of residuals as a function of, for example, the two parameters r1 and r2 is defined as:

ss(*r*1,*r*2)=wi (*F*i-f(*X*i,*r*1,*r*2))2  (3)

f(*X*i,r1,r2)is the function that relates the independent (errorless) variable Xi to the dependent variable Fi and where wi are the weighting factors for the n data pairs, which are

wi=1/σi2 (4)

where σi2 is the variance in *F*i. Using another measure of the error in *F*i, like for example the outer limits of a normal distribution (3σ) or the outer limits of a uniform distribution does not affect the relative weighting.

In this ss(*r*1,*r*2) space the optimum set of the parameters is found, e.g. ȓ 1 and ȓ 2, at a lowest value of the sum of squaresss(*ȓ*1, *ȓ* 2).

The advantage of this method is that, when the user defines a proper region where the solutions for the parameters are located as well as the complete joint confidence interval (JCI), the JCI has the correct shape. The program is designed to perform four different tasks:

a Calculate the ss(*r*1,*r*2) in a particular region for the two parameters and looking for the best values of *r*1 and *r*2. Very important is the choice of weighting the data; absolute error constant (wi=1) or individual weighting for each data point (wi=1/σi2). The sum of squares values are calculated in a grid of 100 × 100, this means that a particular minimum can shift a little bit, depending on how the grid is chosen.

b Visualize the ss(*r*1,*r*2) as a grid surface in three-dimensional space where the *r*-values form the x and y axis and the sum of squares of the residuals the z-axis. This array can be saved.

c Show the experimental *F* values relative to the fitted *f*10-*X*-*F* surface with colors on whether the residuals are positive or negative. This graph can be saved as a TIFF file (overwriting previous files with the same name).

c The JCI determined from the surface is also shown and can saved as a TIFF file (overwriting previous files with the same name).

The χ2 distribution is used to construct the JCI2:

ss(*r*1,*r*2)z ≤ ss(*ȓ* 1, *ȓ* 2)+σ2χ2z(p) (5)

where σ2 corresponds to the average variance of the dependent variable (e.g. *F*) and is calculated from the known errors as entered by the user.

When entering the data, also delta has to be entered. This error is used to calculate the theoretical maximum value of the sum of squares of the residuals:

ss(*r*1,*r*2)theor=wi (deltai)2  (6)

where wi can have its usual meaning as in eq. 4. In the case that individual weighting per data-point is chosen, these delta-values are also used for the weighting. In the case of the absolute error constant (uniform weighting), these delta-values are not used for weighting but are used only for calculating the theoretical sum of squares.

In principle this theoretical sum of squares should be larger or in the same order of magnitude as the sum of squares in the minimum. If this is not the case, the errors in the measurements are badly estimated or the model is not appropriate for the data. This can also be checked when the residuals per point are investigated. In principle these residuals should be generally smaller than the estimated errors and also the residuals as a function of the independent variable (*f*0 and *X*) should not show systematic deviations (if this is the case, the chosen model might be wrong).

The estimated standard deviation in the parameters is conservatively estimated from the joint confidence intervals. Therefore the complete JCI should be encapsulated in the search region. It is however preferred to use the JCI itself to report the uncertainty in the parameters.

The numerical integration is conducted by taking *f* = (*f*10-*F*cum \* conv)/(1-conv)). What is actually calculated is the sum of all *F*cums (F1totest) which is then divided by the number of iterations to give the current average *F*cum.

In the Python code one can make changes. For example the stepsize for the numerical integration, stepsizeconv, is now set to 0.001 which gives accurate numerical integration but the calculation time is in the order of up to a minute. Increases this number (eg to 0.003) will reduce calculation time but reduces accuracy of the calculation. In theory one could also increase the gridsize for *r*1 and *r*2 (now 100x100) but that will dramatically increase the calculation time and one has to change the array sizes accordingly. Another option would be to automatically change the search region if one of the r-values is on the border of the search region, etc.

The IUPAC recommended method is extensively described by the IUPAC working group on “*Experimental methods and data evaluation procedures for the determination of radical copolymerization reactivity ratios”* 4.

21-04-2024

**References**

1 A.M. van Herk, *J. Chem. Educ.,* **72**, 138 (1995)

2 P.J. Rossignoli, T.A. Duever, *Polym. React. Eng*., **3(4)**, 361 (1995)

3 A.M. van Herk, T. Dröge, *Macromol. Theory Simul*., **6**, 1263 (1997)

4 A. A. A. Autzen, S. Beuermann, M. Drache, C. Fellows, S. Harrisson, R. A. Hutchinson, A. Kajiwara, D. Keddie, B. Klumperman, G. T. Russell,A. M. van HerkPolymer Chemistry, 2024, DOI: 10.1039/D4PY00270A