User Guide

Version 1.2.0

User Guide

Table of Contents

Section 1 Overview

The workflows for history matching, prediction uncertainty and optimisation are very similar. The main difference is the type of estimator point being defined.

An estimator point is a point for which there is a simulator response. It is defined normally by a well name (or a well group or the field), a variable name, and a time.

So for example an estimator point may be well “XYZ”, variable oil rate, and time 200 days.

Each simulation run generates a value for this, and an estimator is built up from the simulator responses and the values of the modifiers for each run.

Estimator points may also be extended to include RFT points and more esoteric capabilities.

Estimator points may be:

- History match points, for which there is a history value
- Prediction points, for which we wish to calculate a prediction S curve
- Optimisation points for which we wish to optimise some field performance objective.

1.1 History matching and prediction uncertainty

A pure history matching project defines history match points alone.

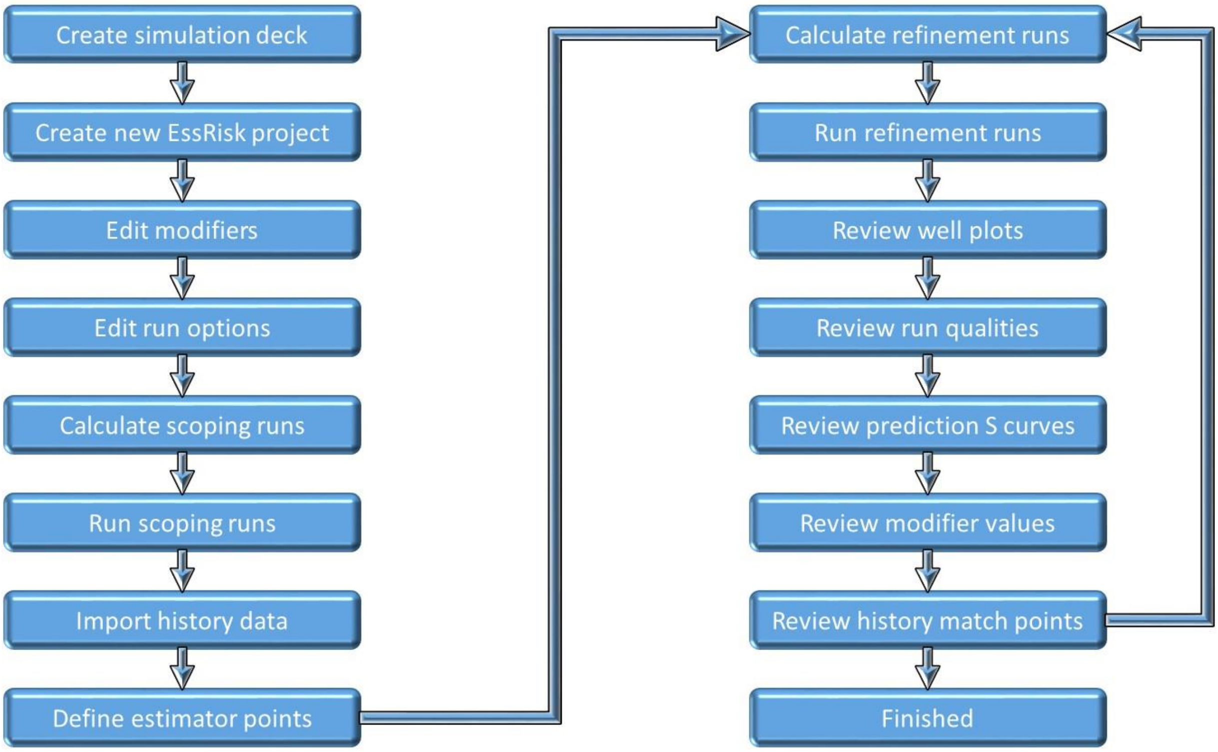
A prediction uncertainty project defines history match points and, in addition, prediction points.

1.2 Optimisation

An optimisation project defines optimisation points alone. The overall objective is a weighted sum of the individual optimisation points.

Section 2 Getting started

2.1 Workflow diagram



2.2 Description

2.2.1 Create deck

The first stage is to create a valid simulation deck, and check that it will run in a stable fashion to completion.

Next, the deck needs to be altered to define what values will be modified.

This is done by replacing any values with the name of the modifier.

E.g.

```
MULTIPLY
PORO  1.0 1 100 1 100  1  3    /
PORO  1.0 1 100 1 100  4  10    /
/
```

Is replaced by

```
MULTIPLY
PORO  %porosity-layers-1-3% 1 100 1 100  1  3    /
PORO  %porosity-layers-4-10% 1 100 1 100  4  10    /
/
```

This defines two modifiers, `porosity-layers-1-3` and `porosity-layers-4-10`, which modifiers porosity in the respective layers.

2.2.2 Create new EssRisk project

The created deck is then imported into EssRisk, using the menu File > New Project.

2.2.3 Edit modifiers

The modifier ranges and most likely value are then defined using the menu Tools > Edit Modifiers

2.2.4 Edit run options

The run options need to be defined. This is done in the menu Tools > Options.

The number of simultaneous runs is defined here.

The run commands for submitting the simulations have to be defined, the simulator has to be specified. There are additional options which are fully described in the user guide.

The run command is defined as text, and various tags are substituted when the individual run is submitted.

__DECK__ is a tag for the deck file name for an individual run (without directory or extension)

__RUN__ is a tag for the string "Run" followed by the run identifier.

__RUNID__ is a tag for the run identifier.

__PROJDIR__ is a tag for the deck directory

__REMOTEDIR__ is a tag for a remote access directory

__WORKDIR__ is a tag for the working directory

__EXP__ is a tag for the experiment name for an individual run.

An example for Eclipse is

eclrun.exe eclipse __PROJDIR____DECK__

2.2.5 Calculate scoping runs

The first step is to calculate the modifier values for the scoping runs. This is done with the menu Tools > Scoping runs. The number of scoping runs is chosen. The calculation is performed almost immediately.

2.2.6 Run scoping runs

The scoping runs are submitted to the simulator.

2.2.7 Import history

This is done using the menu File > Import History. A history file is chosen. This may be either:

- an Eclipse RSM file
- a csv file

2.2.8 Choose estimator points

The estimator points are chosen graphically. The well plots side tab is selected, and the plots chosen one by one. For each plot, the chosen history points are selected graphically by clicking on the history value.

Prediction points are chosen by clicking at some future time for the relevant plot.

For history match points, the tolerance can be adjusted graphically by pressing the mouse over the upper/lower value and dragging the mouse.

The estimator points may be further defined in the menu Tools > Estimator Points. The tolerances can be changed, and a log transform may be applied.

Also estimator points may be deleted.

2.2.9 Calculate refinement runs9

The calculation of refinement runs is performed in the Refinement Runs window (Tools > Refinement Runs). First, the total number of refinement runs is specified, and then the modifier values are calculated. This can take a little time, as it involves the building of the proxy model and the creation of S curves.

2.2.10 Run refinement runs

When the values have been calculated, the runs may then be submitted.

2.2.11 Review well plots

The well plots tab allows the user to review the well plots for each run. There is a run filter so the user can select runs.

2.2.12 Review run qualities

The qualities tab allows the user to review the quality for each run. There is a plot to show the evolution of run quality.

2.2.13 Review prediction S curves

The prediction tab allows the user to review the S curves for each prediction point, and the run values for that point. There is a run filter so the user can select runs and see how the run-based S curve matches the proxy S curve.

2.2.14 Review modifier values

The modifier value tab allows the user to review the modifier values for each run. There is a plot of the values and a run filter so the user can select runs.

2.2.15 Review history match points

Using a combination of these plots, the user can review the history match points and associated tolerances.

2.2.16 Finished

The user continues the process, submitting more refinement runs and reviewing the results, until the user gains an adequate understanding of the simulation model and possible inadequacies in matching the observed field, or is satisfied that there has been enough exploration of possible history matches and there is sufficient convergence in:

- The quality of the history match
- The evolution of prediction S curves
- The correspondence between proxy S curves and run- based S curves

Section 3 User Interface Style

3.1 Introduction

The user interface style is based on modern desktop applications. There is a conventional menu, but also some screens are available using a docking windows, or tab, style, which is becoming common.

The overall architecture is multi-threaded, which means that the algorithms and the user interface are kept separate and operate in different threads.

To the user this means that the user interface is responsive while simulation runs are taking place, where appropriate.

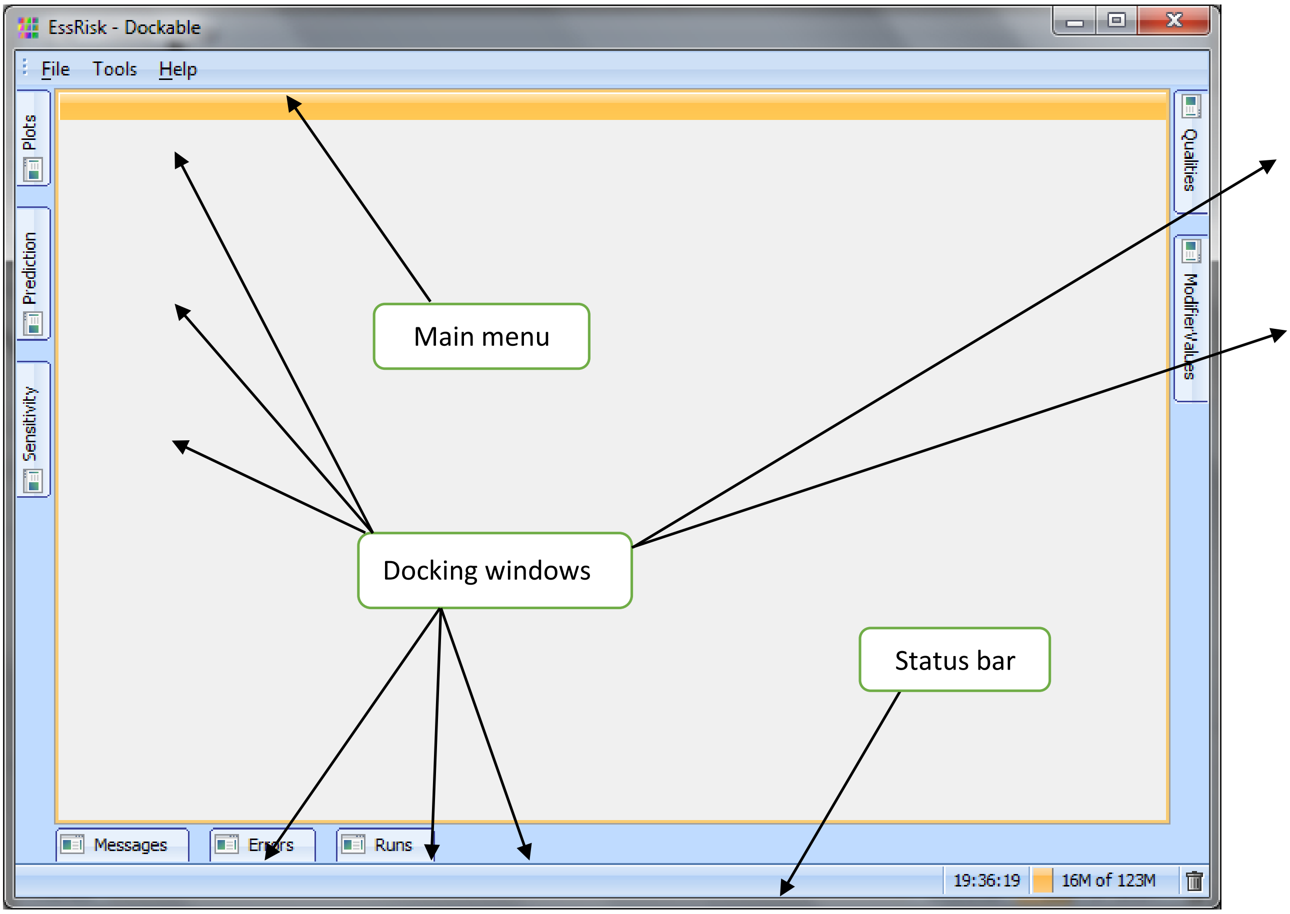
3.2 Databases and projects

The complete EssRisk information is stored in a database project file. This has the extension *.h2.db.

Whenever the user clicks ‘OK’, the data is immediately updated in the database. There is no need to perform an overall ‘Save’ operation when the program is exited.

When returning to an existing project, the project database is opened (File > Open Project). This makes the connection to the database and indexes the data for performance.

3.3 Overall window



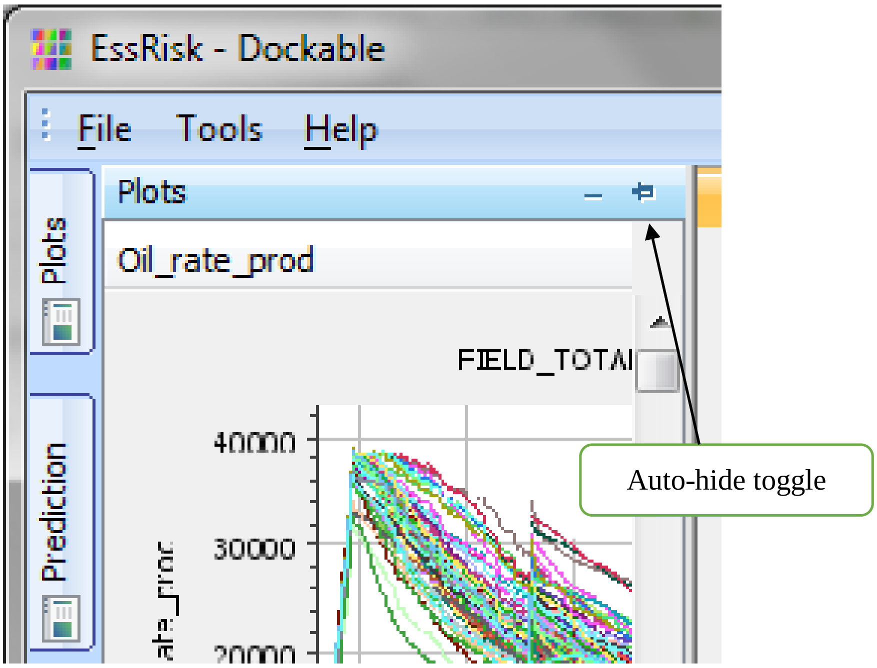
At the top of the window is the normal menu, which follows industry standards.

At the bottom of the window is the status bar. This consists of:

- A progress bar, indicating progress for long running activities
- A clock, showing time of day
- A memory status. This shows the current memory being used and the amount of memory allocated. This is controlled by the Java runtime engine, and goes up and down. This is only useful if the amount of memory starts to go high, and can be useful for diagnostic purposes.
- A dustbin. Clicking this makes the Java runtime engine consider a garbage collection, which can reduce memory. Users are not normally expected to require this, as it is handled automatically.

Around the edge of the main window are the docking windows with their tabs. If the mouse hovers over or clicks the tab, the window is displayed.

For example, if the mouse hovers over the plot tab, the following appears:



If the mouse moves away from the opened window, the window is hidden again.

If the user clicks on the auto-hide toggle, the window does not auto-hide, but remains visible and may be re-sized using the mouse. It can be made to hide by toggling the auto-hide button again.

At the bottom is the messages and errors docking tab. This opens a window showing messages, mainly about progress, and errors.

The runs docking tab shows status of runs.

3.4 Plots

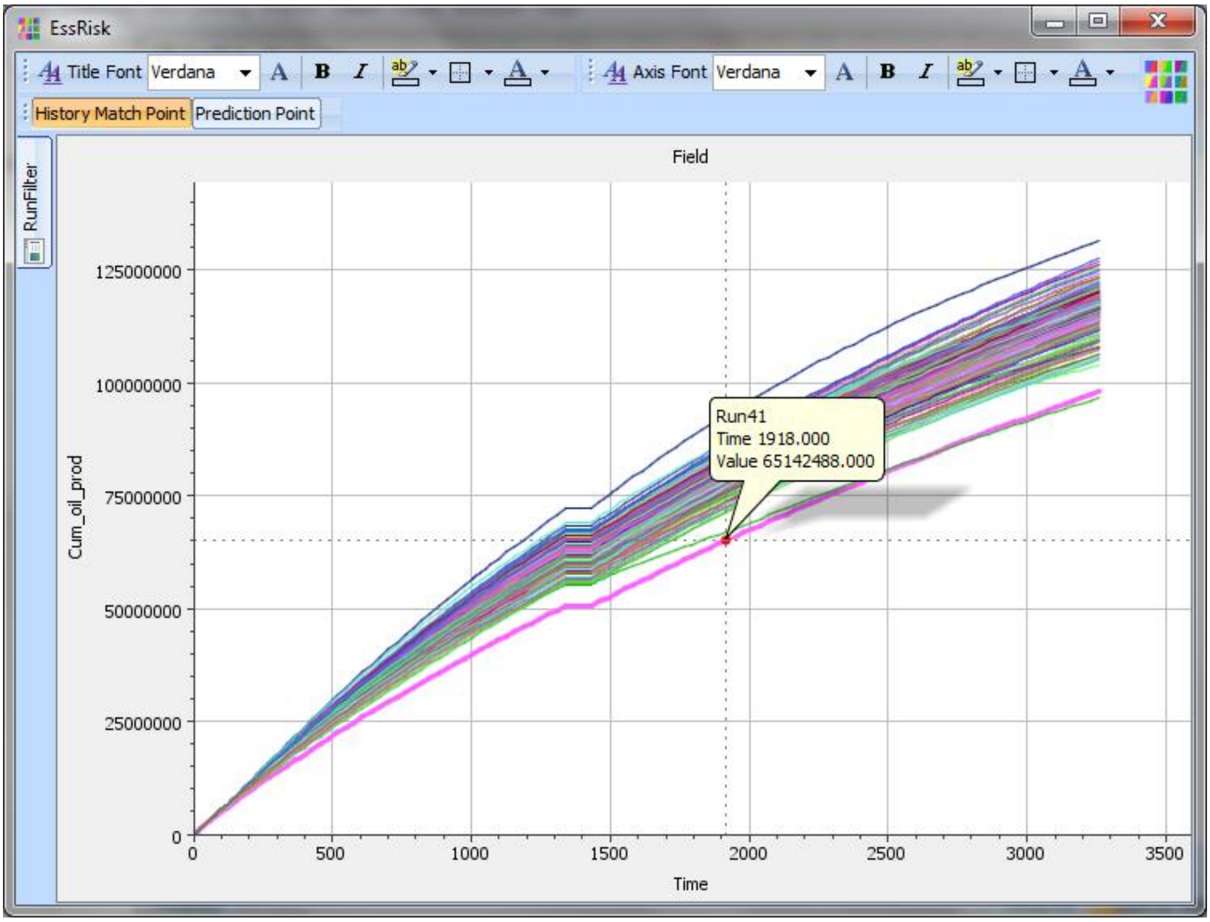
Plots follow a common overall style. They appear in a separate window.

If the well plots tab is selected, a table of all available well plots appears. The user can scroll across and up and down to find the plot of interest. If a plot is then clicked, a separate window appears for that plot. Plots can be added to the window.

The wells and variables can be filtered by clicking the buttons at the top.

Other plots appear by clicking on the ‘Plot’ button on the relevant docking window. For example, if the ‘Qualities’ docking window appears, there is a button at the bottom left which can be clicked.

Once a plot appears, they all follow similar guidelines. We illustrate with a well plot.



Plot of field Cumulative Oil v. Time.

If the mouse hovers over the curve for a particular run, a pop-up balloon top appears with some summary information about that point. The complete curve is also highlighted.

Plots may be zoomed using the middle mouse wheel, and panned left/right by holding down the left mouse key and dragging.

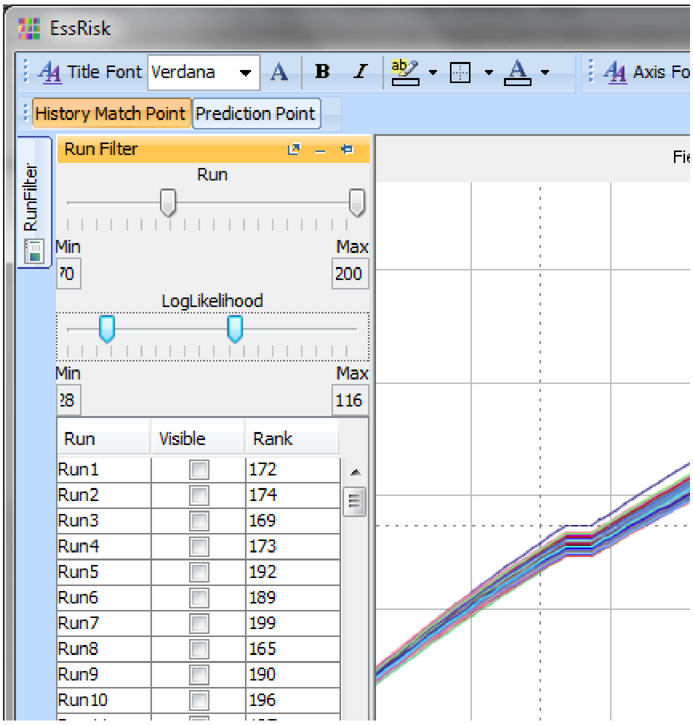
Holding down the CNTRL key and using the left mouse button allows you to draw a rectangle and zoom in. CNTRL key and right button zooms out again.

Right mouse button brings up a dialogue allowing you to change axis scales.

Sliders to the right and at the bottom allow adjustment of scales.

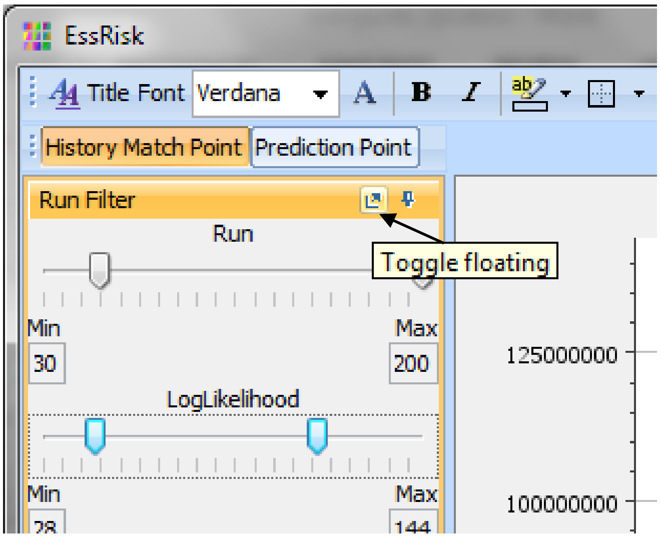
The menus at the top allow some editing of the plot, such as title and axis text styles.

The Run Filter tab on the left is a docking window tab, which brings up a run filter.

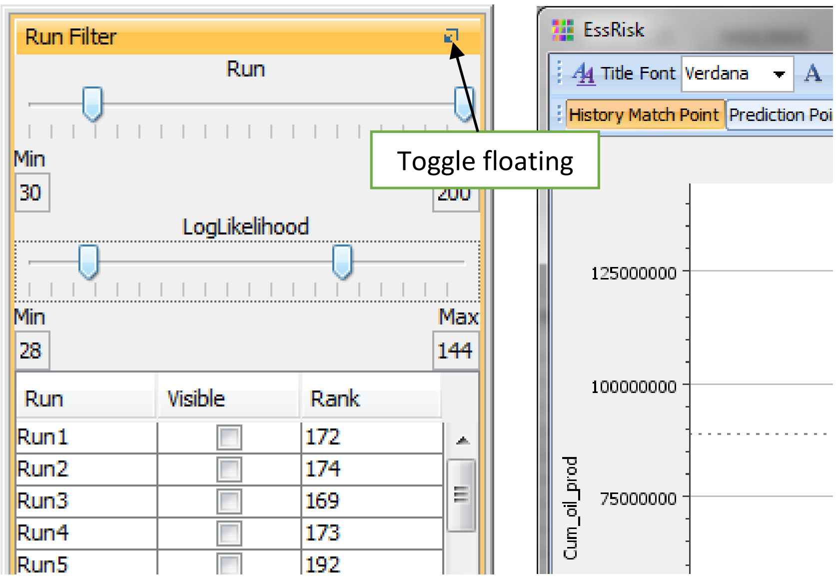


This allows the runs to be filtered according to run number and likelihood function value. Similar to other docking windows, the auto-hide capability can be toggled.

In addition, the floating capability may be toggled



So that the run filter window appears in a separate floating window. The floating may then be toggled again.


Run Filter
Run
Min
30
200
LogLikelihood
Min
Max
28
144
Run Visible Rank
Run1 172
Run2 174
Run3 169
Run4 173
Run5 192
Toggle floating
EssRisk
Title Font Verdana
History Match Point
125000000
100000000
75000000
Cum_oil_prod

Section 4 Creating the deck

The first stage is to create a valid simulation deck, and check that it will run in a stable fashion to completion.

Next, the deck needs to be altered to define what values will be modified.

This is done by replacing any values with the name of the modifier.

E.g.

```
MULTIPLY
PORO  1.0 1 100 1 100  1  3    /
PORO  1.0 1 100 1 100  4  10    /
/
```

Is replaced by

```
MULTIPLY
PORO  %porosity-layers-1-3% 1 100 1 100  1  3    /
PORO  %porosity-layers-4-10% 1 100 1 100  4  10    /
/
```

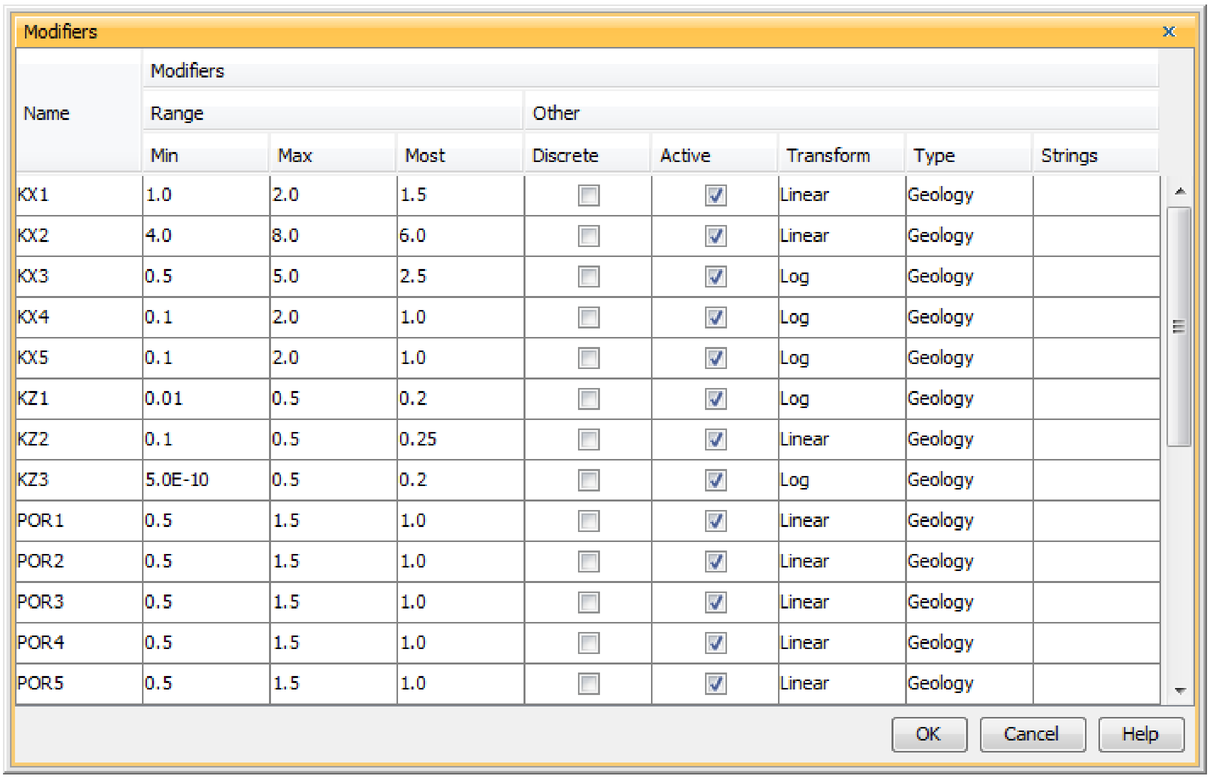
This defines two modifiers, `porosity-layers-1-3` and `porosity-layers-4-10`, which modifiers porosity in the respective layers.

Section 5 Creating a project

The created deck is then imported into EssRisk, using the menu File > New Project.

Section 6 Defining modifiers

The modifier ranges and most likely value are then defined using the menu Tools > Edit Modifiers



Modifiers

Name	Modifiers							
	Range			Other				
	Min	Max	Most	Discrete	Active	Transform	Type	Strings
KX1	1.0	2.0	1.5	☐	☑	Linear	Geology	
KX2	4.0	8.0	6.0	☐	☑	Linear	Geology	
KX3	0.5	5.0	2.5	☐	☑	Log	Geology	
KX4	0.1	2.0	1.0	☐	☑	Log	Geology	
KX5	0.1	2.0	1.0	☐	☑	Log	Geology	
KZ1	0.01	0.5	0.2	☐	☑	Log	Geology	
KZ2	0.1	0.5	0.25	☐	☑	Linear	Geology	
KZ3	5.0E-10	0.5	0.2	☐	☑	Log	Geology	
POR1	0.5	1.5	1.0	☐	☑	Linear	Geology	
POR2	0.5	1.5	1.0	☐	☑	Linear	Geology	
POR3	0.5	1.5	1.0	☐	☑	Linear	Geology	
POR4	0.5	1.5	1.0	☐	☑	Linear	Geology	
POR5	0.5	1.5	1.0	☐	☑	Linear	Geology	

OK Cancel Help

The columns are:

- **Modifier name.**
- **Min, Max, Most.** Minimum, Maximum and Most Likely values. Min < Most < Max. These are the values for which the engineer feels express the range for that modifier. It should represent the greatest reasonable range, the range of reasonable possibilities rather than probabilities.
- **Discrete.** Check this if the modifier is discrete and takes integer values.
- **Active**. Check if active (i.e. it is being used to modifier the deck).
- **Linear/Log**. Where the modifier has a large range (Max/Min > 10, say), then a log transform can greatly help the history matching process. This is only applicable if the range is >= zero. Zero is handled with a very small correction factor. If, for example, a fault transmissibility has a range 0 – 1, then a log transform is recommended.
- **Geology.** This is to distinguish between geology modifiers (which are unknown, but over which no engineer has control) and decision or optimisation modifiers, over which the engineer has control.
- **String**. For discrete modifiers, this column contains a string which represents the possible set of strings which are substituted in a deck. It may, for example, be different names of include files.

Section 7 Defining estimator points

7.1 Overview

For any given project, the user defines a set of estimator points. These estimator points may either be history match points or prediction points.

The estimator points are specified by

- well (or group of wells such as a field)
- variable (such as BHP, or oil rate)
- time

An estimator point may either be a history match point, or a prediction point. A history match point is a point which is being history matched. It is defined, in addition to the information above, by:

- History value
- Effective tolerance

For RFT data, a history match point has additional information about depth.

A prediction point is an estimator point for which we are trying to quantify uncertainty. An S curve of uncertainty is calculated for each prediction point. They do not have a history value.

Optimisation points are points which are being optimised. They have a weighting, defined by the tolerance, which gives the weighting when multiple prediction points are defined. Optimisation points do not have a history value. There is no RFT optimisation point.

Every individual estimator point has a separate independent proxy model.

The effective tolerance for a history match point is a range within which the engineer can reasonably expect to achieve a history match. The effective tolerance can be considered as:

Effective tolerance = measurement error + model error

where the measurement error is the error in measurement of the history value, and the model error is the error between the simulation model and the actual physical reservoir.

The model error is under the control of the reservoir modeller. In practice, no reservoir simulation model is perfect, but there is usually some idea, gained from experience or from previous similar projects, what the expectation might be as to how closely the simulation model fits the physical reservoir.

The history match points and their tolerances directly affect the uncertainty quantification of the prediction points.

For example, when analysing prediction uncertainty for field cumulative oil, it is normally sensible to choose a history match point for field cumulative oil at the end of history, and define a fairly small tolerance if it has a small measurement error. However, it might be that the reservoir simulation model has difficulty matching both field cumulative oil and field oil rate, so this might lead to a significant over or under estimation of prediction uncertainty. It may be better to acknowledge this with a larger cumulative oil tolerance, and also include a field oil rate history match point.

In summary, the engineer, when choosing history match points, needs to carefully consider the competing priorities of producing an apparent good history match v. producing an accurate uncertainty quantification of future production.

Given that each history match point has a tolerance which represents the engineer's views on what an acceptable match might be, the overall best history matches should more or less be within the tolerance for all history match points. There is often competition between history match points – either well A is matched within expectations, or well B is matched within expectations, but not both. In this case, the expectations need to be adjusted, and the tolerances increased for each well.

Clearly the history matching and reservoir modelling process is iterative; ideally the lessons gained from the history matching process will lead to a re-evaluation of the reservoir model, and the history match is repeated. It is therefore important that the history match process helps the engineer to learn as much as possible about the static and dynamic behaviour of the reservoir. The benefits of history matching tools such as Spark is that it allows, with the good diagnostic tools, the engineer more time to perform this analysis, using the engineer's experience and understanding, together with input from the rest of the asset team.

History match points serve two purposes. The first is obvious – to define a history match objective. The second is more subtle – to gain information. Every reservoir simulation run generates information, and this information is used to update the proxy model for every estimator point. Information about a bad history match is useful, because it tells the proxy model where not to search for a history match. We want to both search for good history matches and eliminate bad history matches. If there is a set of simulation runs which appear to give good history matches, according to the values of the objective function, it is still necessary to examine these matches and identify whether there are any wells or variables for which some of these runs give a poor history match. If this is the case, then a history match point needs to be added, so that those simulation runs can be eliminated from the set of good history matches.

7.2 Guidelines for history match points

- Iterate and visualise. Choosing history match points and tolerances is an iterative process, and ideally all well plots need to be visualised to come to a proper judgement. There are diagnostics which can be used to help identify problem history match points, but it is more difficult, except using the human eye, to identify points which have not been chosen but which should be.
- Start with relatively few points. The proxy model algorithms scale linearly with the number of history match points, so it is more efficient to start with fairly few. Moreover, if it is a new history match study, it is likely that the engineer wants to quickly discover whether a history match is possible at all or not. If it looks promising with a few points (such as just field rates, cumulatives and pressures), then more points can be added and the study can continue without restarting.
- Do not define too many points. From examination of well plot curves, the shape of each run, and the shape of history, can be evaluated. If all the shapes are similar, then only a single history match point needs to be defined – if that point is matched, then all other points will be matched. It is rare, for example, that more than two points are needed on any cumulative curve.

- Start with wide tolerances. This will help the algorithms to explore the full range of possible history matches, and at the early stages, the engineer does not know what the model error is – the history match criteria set by the asset team may be unrealistic for the given reservoir model.
- Choose points where the simulation is fairly stable. Do not choose points where the variable is changing rapidly. For example, do not choose history match points immediately after a well test. The reason for this is that (a) the measurement time may be inaccurate (b) the measurement value may 'look' close to history, but is in fact far from history. Of course, if the study is using mainly well test data in a brown field reservoir, then one may want to very carefully match the well test pressure curve data.
- What is important? Set the tolerances according to what is important. For example, often in history matching and prediction quantification, the field history match is the most important. In which case, set the field tolerances according to the engineer's beliefs, and set well tolerances wider.
- Cumulatives or rates? Often cumulatives are known more accurately than rates, but cumulatives can also average out certain reservoir behaviours. One approach is to have history match points at both early and late time for cumulatives. However, additional rate history match points can provide good information – if, by examining the well plots, it appears that simulation runs have a wide range of history matches for rates, then it is useful to include history match points for rates at those times.
- Choose where there is variation. Choose history match points where there is plenty of variation between runs. There is no point choose points where all runs are a very good match to history, or where all runs are identical. If there is variation between the runs, it indicates that the underlying physics is varying, so there is good opportunity for the proxy model to learn.
- Take care about null values. For some simulators, zero is taken as a null value. Be careful about choosing these points.
- Evaluate whether it makes sense. In some cases, all runs are far from the history value. In these cases, exercise extreme caution, as otherwise it will throw off all the other history match points. Is the history value correct? Is the simulation model correct? Normally one would expect runs both above and below any given history value.
- Does any run match a point? If there is an estimator point for which no runs go within the tolerance, then probably the tolerance is too low.
- Estimator points v. number of modifiers. In general, there should be at least as many estimator points as there are modifiers. Otherwise, there is not enough information to generate a good prediction uncertainty – there is too much 'slack' in the simulation model. Of course, there are brown field cases where this is unavoidable – in this case the study is intermediate between a full appraisal, and a history match study. If there are very few history points, the ranges on modifiers becomes more important. Conversely, where there is a lot of history, most modifiers should be within their range. It is always worth examining, at the end of the history match study, whether the modifier ranges need re-evaluating.
    - Failed runs. In some cases, certain runs can terminate early, because of convergence issues, errors, or reaching a defined economic limit. However, the information up to that point is still useful and can contribute to the proxy model, and in these cases some history match points should be chosen at early times. This learning will also help

avoid those modifier value combinations in future runs, particularly if the failed runs have a poor history match up until the time of failure.

7.3 Choosing history match points

The estimator points are chosen graphically. The well plots side tab is selected, and the plots chosen one by one. For each plot, the chosen history points are selected graphically by clicking on the history value.

Prediction points are chosen by clicking at some future time for the relevant plot.

For history match points, the tolerance can be adjusted graphically by pressing the mouse over the upper/lower value and dragging the mouse.

The estimator points may be further defined in the menu Tools > Estimator Points. The tolerances can be changed, and a log transform may be applied.

Also estimator points may be deleted.

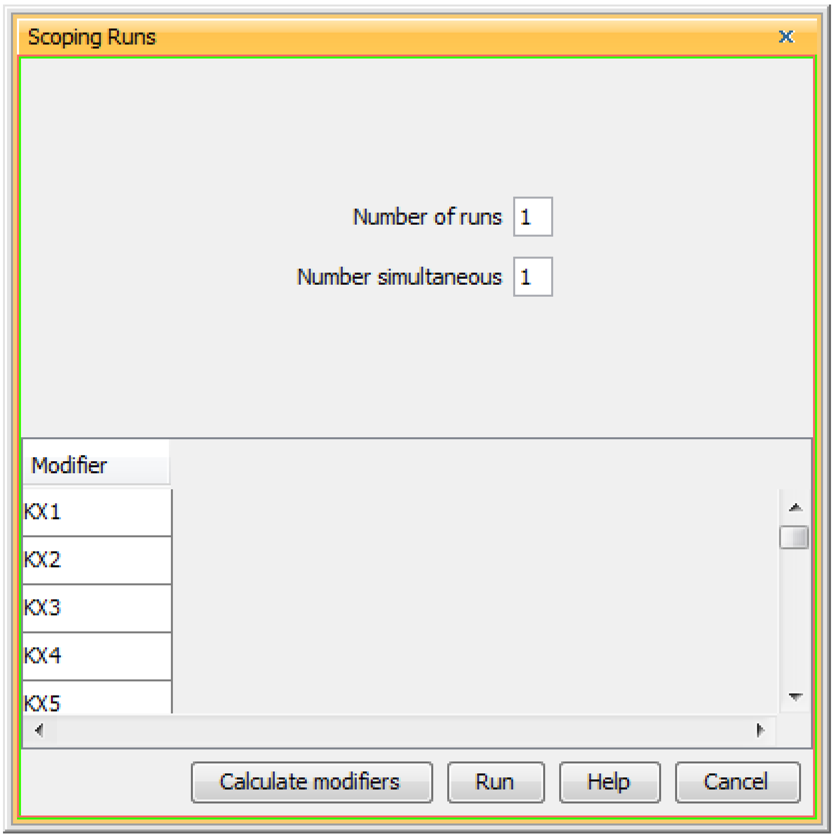
7.4 Choosing prediction points

The process for choosing prediction points is very similar to choosing history match points. On the well plot window, choose the tab ‘Prediction Points’, and then select a time of interest.

7.5 Choosing optimisation points

The process for choosing optimisation points is very similar to choosing history match points. On the well plot window, choose the tab ‘Optimisation Points’, and then select a time of interest.

Section 8 Scoping runs



Go to menu Tools > Scoping runs.

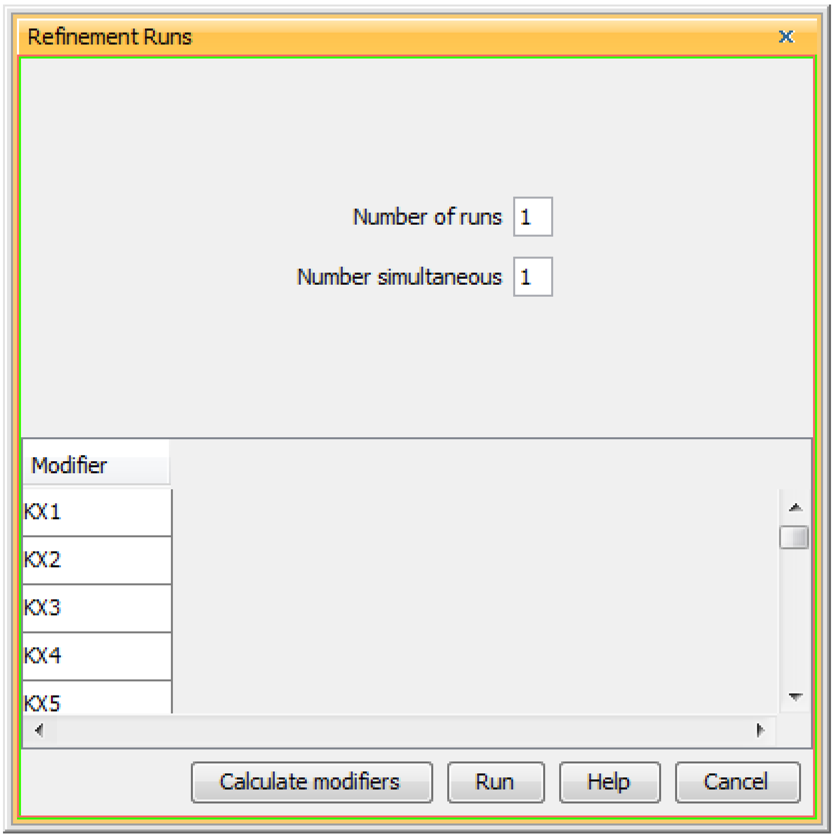
Select how many runs will be performed (25 is the usual guideline, but more may be needed if there is a large number of modifiers. The number should not normally exceed 100.)

Select how many simultaneous runs will be performed. This depends on the cluster and simulation licensing. If there are available resources, choose the same number as the total number. Scoping runs are not inter dependent, so it will not affect results if they are all done simultaneously.

Click on 'Calculate modifiers'. This operation is fast, and the results will be returned almost instantaneously.

Click on 'Run' to submit runs.

Section 9 Refinement runs



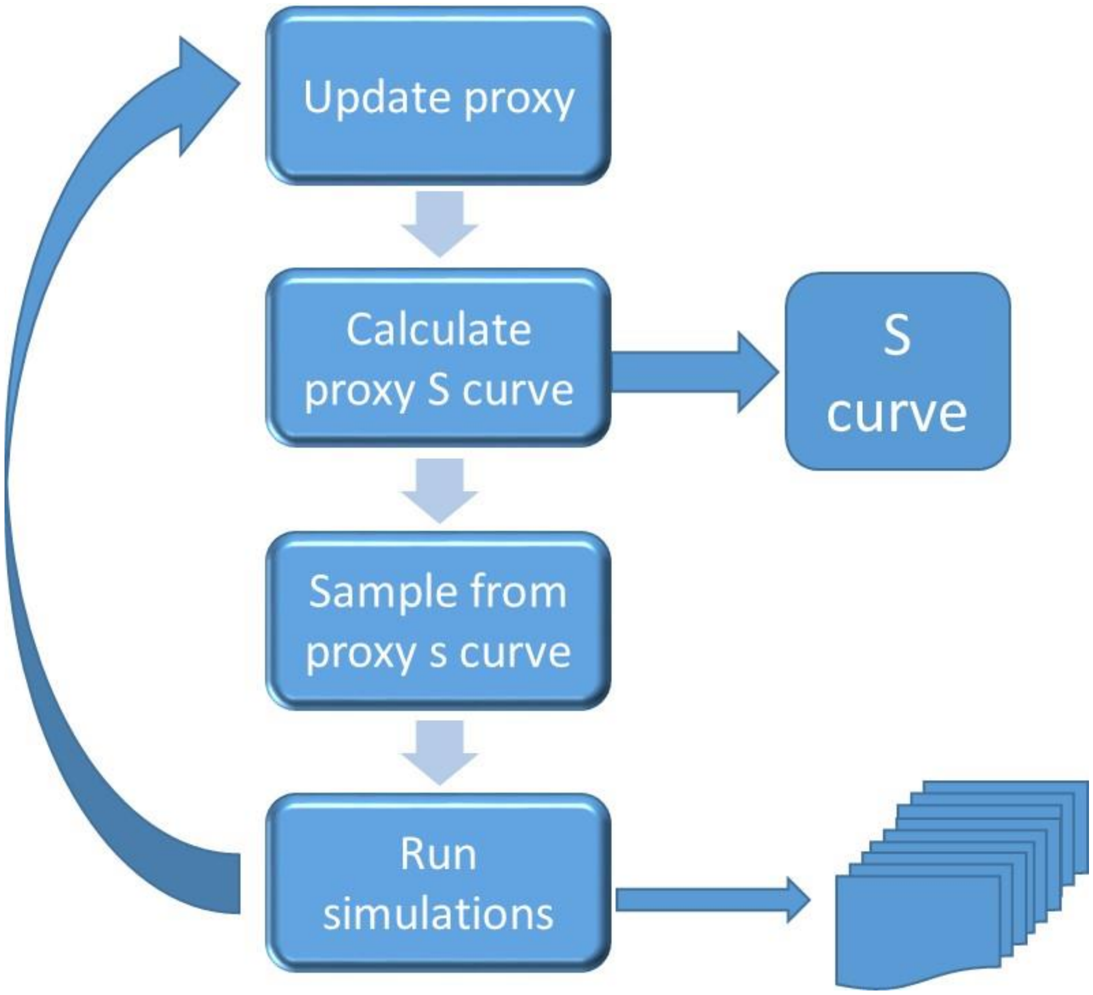
Go to menu Tools > Refinement runs.

This screen is almost identical to the scoping runs screen, and works in a similar manner.

However, there are some points to note:

- Refinement runs are inter dependent, so it is not advisable to define the number of simultaneous the same as the total number of runs. Each complete run updates the proxy model and this is used to generate modifiers for the subsequent run.
- The total number of runs to achieve history match rarely exceeds 200, and is often less than 100. If a match is not emerging within 100 runs, it normally indicates something wrong with the model or history data. It is inadvisable to submit, for example, 1000 runs – the use of a proxy model makes large numbers of runs unnecessary.
- When quantifying prediction uncertainty, it may be that a combined total (all scoping plus refinement runs) of over 200 runs is needed, but again it rarely exceeds 400 runs.
- The ‘Calculate modifiers’ can take some time, depending on hardware and the number of estimator points.

The overall process when running refinement runs includes most of the algorithmic aspects of EssRisk. It is shown in outline below:


Update proxy
Calculate proxy S curve
S curve
Sample from proxy s curve
Run simulations

Section 10 Results calculator

The results calculator calculates new results from existing results. The results are calculated for each well and for each run.

The calculated results are then treated in exactly the same way as other results, and may be used to define history match, prediction or optimisation points.

When the results calculator is defined, it will calculate using all existing runs, using the results stored in the database. It will then, for subsequent runs, calculate new runs on a run by run basis.

The results calculator can call external processes which might, for example, read from files. These files have to exist in order for results to be calculated.

The results calculator also calculates results for history curves.

The results calculator uses time based results. It cannot use RFT depth based results.

The results calculator is dynamically typed, and can manipulate arrays of floats, scalar floats, and strings. It will, for example, create an array of floats if an array of floats is multiplied by a scalar float.

Each results calculator variable has a user defined name, which consists of letters, numbers, and underscore. It must start with a letter. It is case sensitive. Spaces are not allowed.

10.1 Assignment operator

The assignment operator is ‘=’

10.2 Unary operators

The basic unary arithmetic operators are:

sin	sin
cos	cos
log	log to base e
log10	lots to base 10
exp	exponential
abs	absolute value

10.3 Binary operators

The basic binary arithmetic operators are:

'+'	add
'-'	subtract
'*'	multiply
'/'	divide

'%' modulus

'^' power

10.4 Logical operators

The logical operators are

== equals

!= not equals

< less than

<= less than or equals

> greater than

>= greater than or equals

&& and

|| or

! not

10.5 Time-based operators

Operators which calculate over the time array are:

integrate

cumsum

diff

gradient

10.6 Special operators

Special operators are:

modifier(name) returns the value of this modifier

well(name) logical operator, true or false depending whether current well is the named well

command('text') submits a subprocess defined by text, which returns a scalar value

10.7 Control operators

Control operators are:

while

for

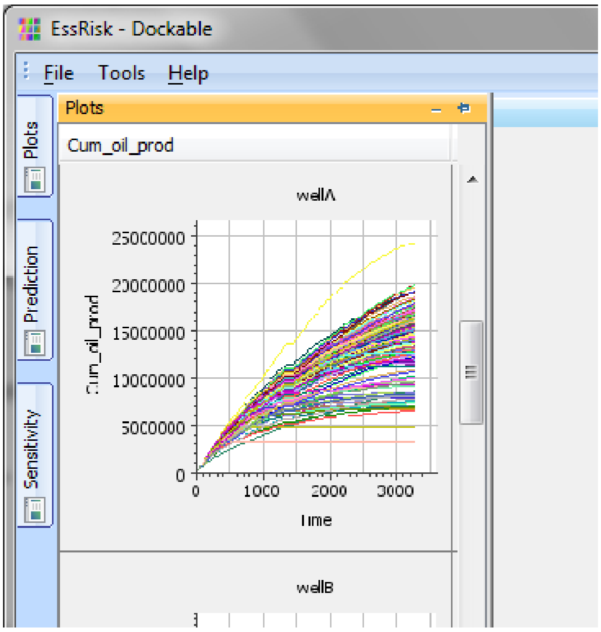
10.8 Saving results

To save a result, use

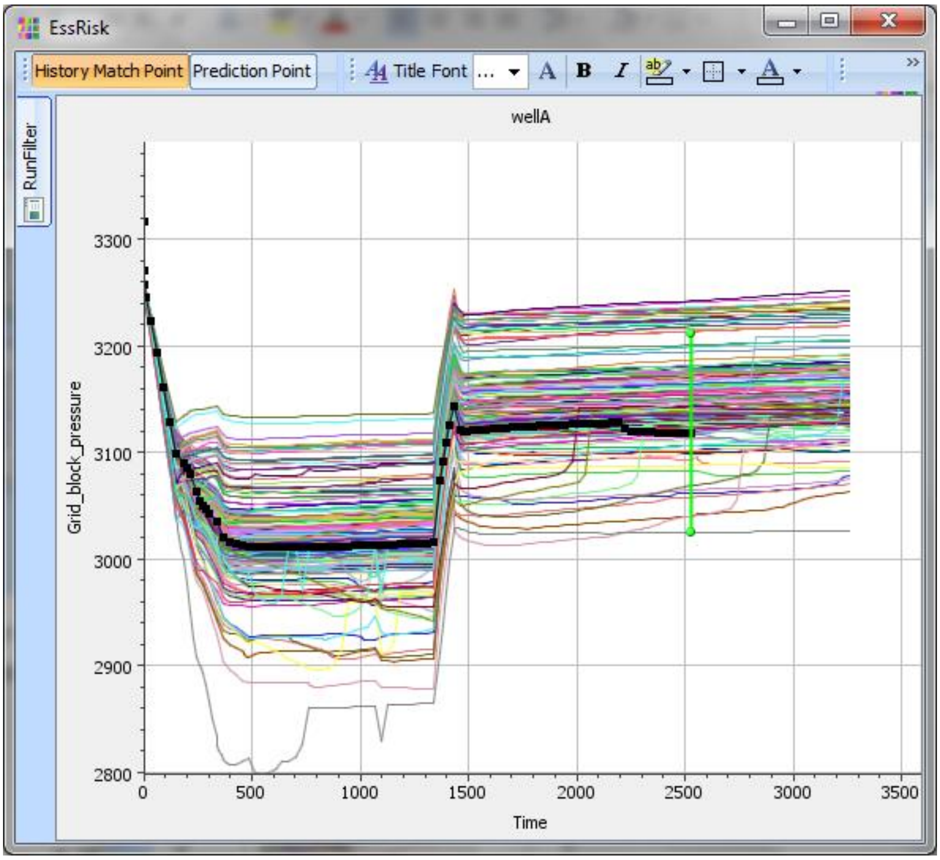
save(name)

'name' can then be used just the same as any other result, so can be used in an observation, prediction or optimisation file.

Section 11 Well plots

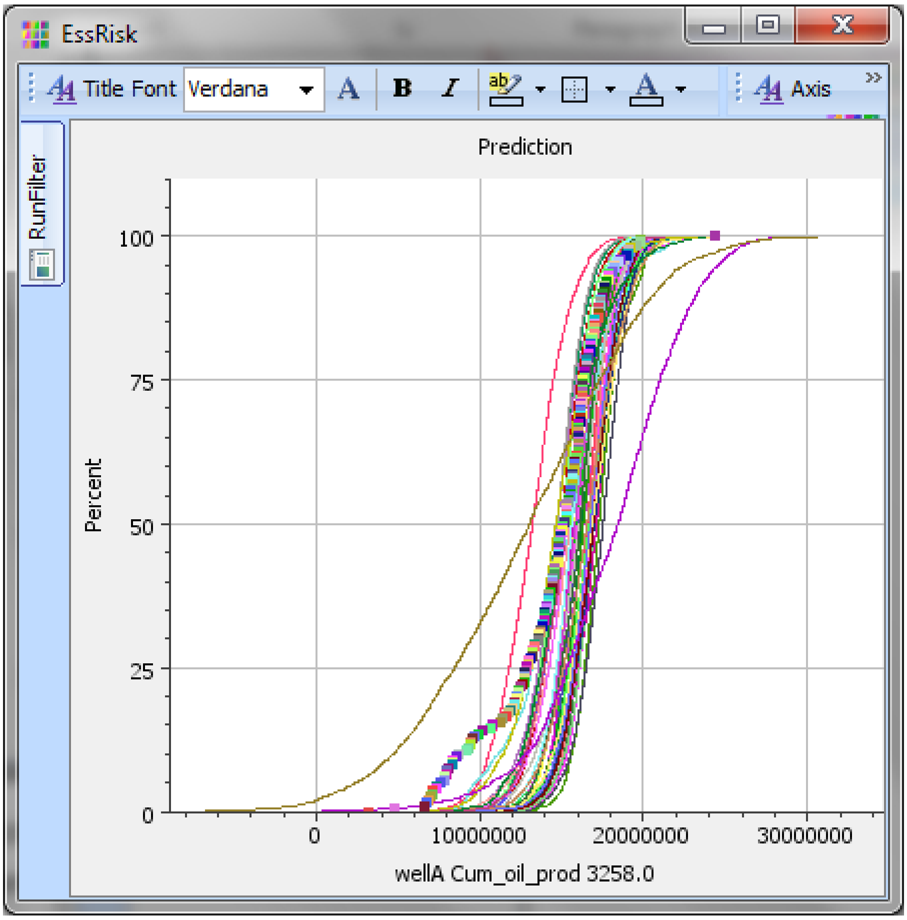


For well plots, move the mouse over the 'Plots' tab, which will bring up a table of all well plots. Scroll to the individual well plot desired, and click. The well plot will then appear in a new window.



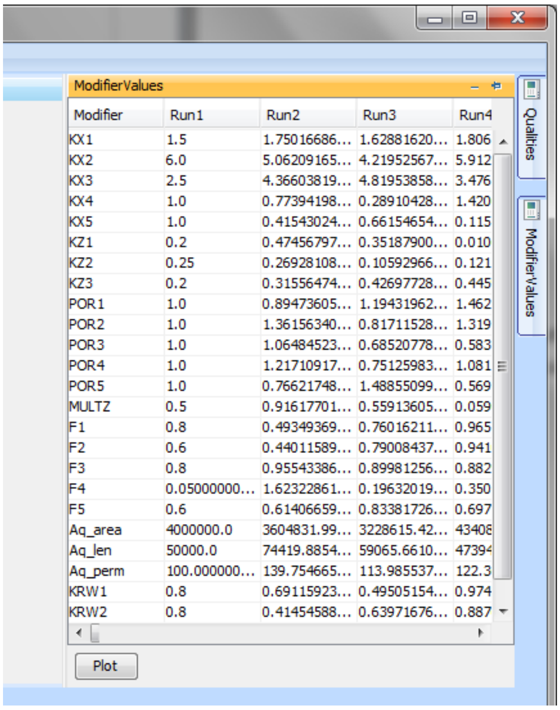
This plot can be used to select estimator points. If a history match point is to be added, select the 'History Match Point' tab, and then click on a history point (shown as a black point). The tolerance can be adjusted by moving the mouse over an end point (green circle), holding the mouse button down, moving the mouse up or down and then releasing the mouse button.

Section 12 Prediction plots



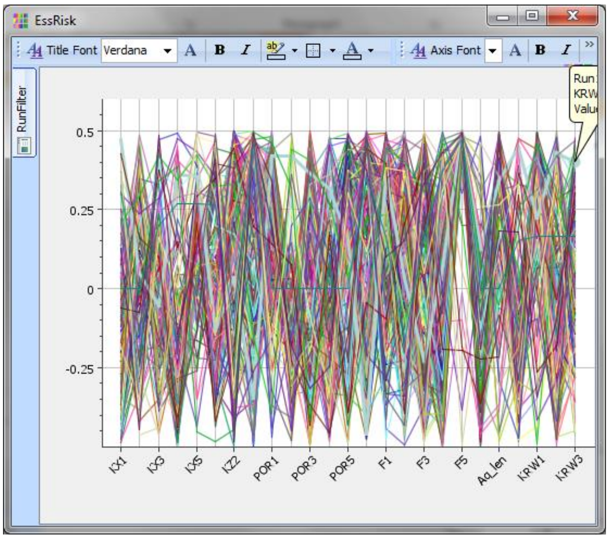
Prediction plots show the sequence of S curves generated from the proxy model (as continuous lines), and an S curve created from individual runs. By using the run filter to eliminate scoping runs and runs with poor history match quality, a set of runs can be generated for which the S curve is a good approximation to the proxy S curve.

Section 13 Modifier values



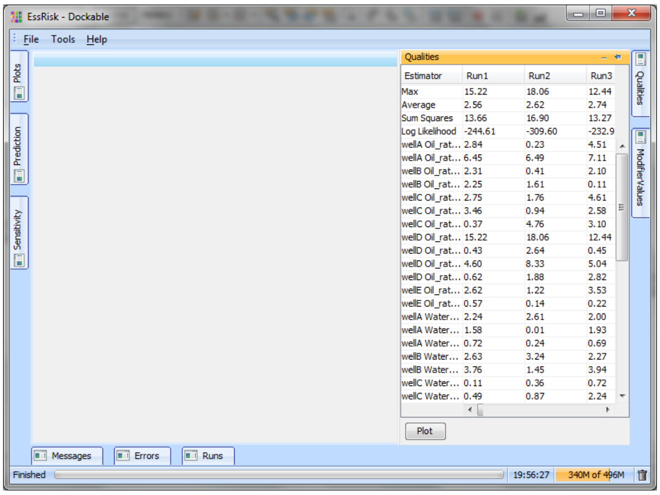
Modifier	Run1	Run2	Run3	Run4
KX1	1.5	1.75016686...	1.62881620...	1.806
KX2	6.0	5.06209165...	4.21952567...	5.912
KX3	2.5	4.36603819...	4.81953858...	3.476
KX4	1.0	0.77394198...	0.28910428...	1.420
KX5	1.0	0.41543024...	0.66154654...	0.115
KZ1	0.2	0.47456797...	0.35187900...	0.010
KZ2	0.25	0.26928108...	0.10592966...	0.121
KZ3	0.2	0.31556474...	0.42697728...	0.445
POR1	1.0	0.89473605...	1.19431962...	1.462
POR2	1.0	1.36156340...	0.81711528...	1.319
POR3	1.0	1.06484523...	0.68520778...	0.583
POR4	1.0	1.21710917...	0.75125983...	1.081
POR5	1.0	0.76621748...	1.48855099...	0.569
MULTZ	0.5	0.91617701...	0.55913605...	0.059
F1	0.8	0.49349369...	0.76016211...	0.965
F2	0.6	0.44011589...	0.79008437...	0.941
F3	0.8	0.95543386...	0.89981256...	0.882
F4	0.05000000...	1.62322861...	0.19632019...	0.350
F5	0.6	0.61406659...	0.83381726...	0.697
Aq_area	4000000.0	3604831.99...	3228615.42...	43408
Aq_len	50000.0	74419.8854...	59065.6610...	47394
Aq_perm	100.000000...	139.754665...	113.985537...	122.3
KRW1	0.8	0.69115923...	0.49505154...	0.974
KRW2	0.8	0.41454588...	0.63971676...	0.887

The modifier values screen shows the modifier values for each run. By clicking on the 'Plot' button, a plot of modifier values can be seen.



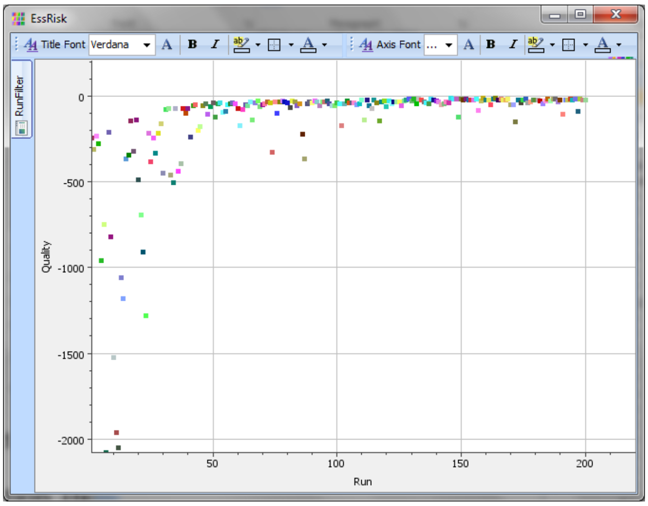
Each line represents the modifier values for a particular run.

Section 14 Qualities



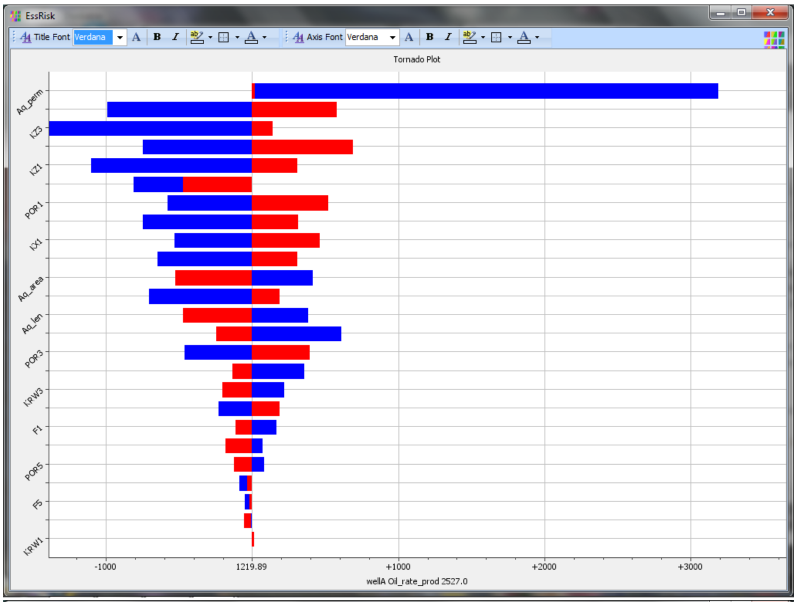
The qualities screen shows the history match quality for individual runs. Each history match point has a quality, which is the deviation from history divided by tolerance. The first row (max) shows the maximum quality over all modifiers, the second row (average) shows the average quality over all modifiers, and the third row (Sum squares) shows the sum of squares of qualities divided by the number of modifiers. The fourth row (Log Likelihood) shows the log likelihood value for each run.

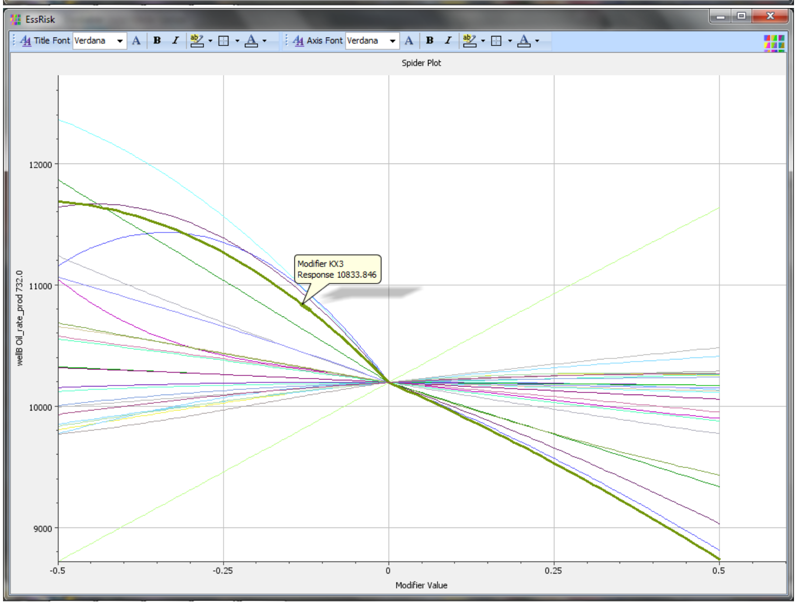
By clicking on the ‘Plot’ button, a plot of the evolution of the log likelihood can be seen.



Section 15 Sensitivity

The sensitivity screen shows a table of tornado plots and spider plots for all prediction points. By clicking on an individual plot, the plot can be seen.





Section 16 Qualities and log-likelihood

The history match objective is defined for each simulation run as a likelihood function. It is the likelihood that the run is within the tolerance for all history match points. The overall likelihood is the product of each individual match point – so the log of the overall likelihood is the sum of the individual log likelihoods.

Defining the density function and cumulative distribution function as:

$$Gaussian\ (normal)\ density\ function$$

$$\varphi(\omega,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\omega-\mu)^2}{\sigma^2}}$$

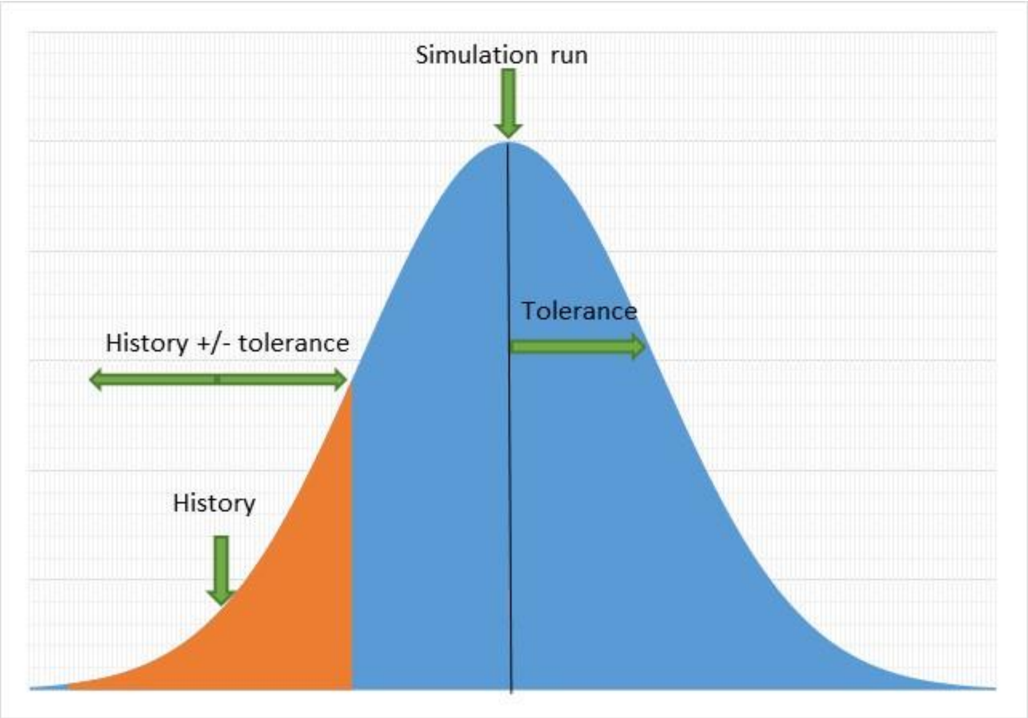
$$Cumulative\ Gaussian\ (normal)\ distribution\ function$$

$$\vartheta(\alpha,\mu,\sigma) = \int_{-\infty}^{\alpha} \varphi(\omega,\mu,\sigma)\, d\omega$$

$$Likelihood = \prod_{i=1}^{n_h} L_i$$

$$where\ L_i(h_i, tol_i,\ \mu_i, tol_i) \ = \ \vartheta(h_i + tol_i, \mu_i, tol_i) - \ \vartheta(h_i - tol_i, \mu_i, tol_i)$$

$$n_h\ is\ the\ number\ of\ history\ match\ points$$
$$h_i\ is\ the\ history\ value$$
$$\mu_i\ is\ the\ simulation\ run\ results$$
$$tol_i\ is\ the\ effective\ tolerance$$



Likelihood function L_i for single history match point i

In practice, the logarithm of this likelihood function is not too far from a much simpler function

$$\sum_{i=1}^{n_h} \frac{(h_i - \mu_i)^2}{tol_i^2}$$

but is flatter in the centre.

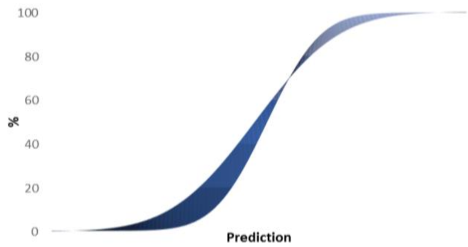
Section 17 Sensitivity

The value of the tolerance at each match point affects the individual match point likelihood, and the overall likelihood. Through the likelihood function, it also directly affect the shape of prediction S curves. By altering the match point tolerances, the engineer can adjust the prediction S curve.

The proxy model approach allows the calculation of the sensitivity of the overall likelihood and the prediction S curve to changes in the tolerance. It then ranks these sensitivities.

For the likelihood sensitivity, it is calculated by taking the derivative of the overall likelihood with respect to the relative change in tolerance.

For the S curve, it is calculated by examining changes in the shape of the S curve, and in particular the area marked blue in the diagram below. It is calculated by taking the derivative of the blue area with respect to the relative change in tolerance.



These sensitivities are used in two ways:

- To identify match points where tolerances need to be examined more closely. If a match point has very high sensitivity, it may be that the tolerance needs to be increased, as it is either too small for the given uncertainty in data and the model, or else the history data is far from the actual simulation run results. It may also be that a match point is very insensitive, and the tolerance is too high. It is up to the engineer’s judgement and knowledge of the data and the model whether or not to adjust (or even remove) these points.
- The sensitivities can also be used to understand how the engineer’s judgement on errors affects prediction uncertainty, and what influences prediction. Of course, it is expected that tolerance on field cumulative oil at end of history will have a high influence on field cumulative oil prediction. It may also be the case that other measurements, such as water cut or BHP at some individual well will also have a significant effect on field cumulative oil prediction, and the sensitivity calculation can help the engineer gain a better understanding of the underlying flow dynamics.

Section 18 Appendices

18.1 Well controls

Although not directly related to the subject of estimator points, the issue of how to control wells is very important to prediction uncertainty quantification.

Often wells are controlled by historic rates, and pressures are history matched. There is then an issue about how to proceed into the prediction phase.

In addition, sometimes the reported well rates are uncertain, whereas the group rates are more accurate.

A recommended alternative approach is to control wells on THP or BHP, and allow rates to vary and match on rates. This makes moving into prediction a much more natural process.

As for tolerances, this procedure may bring to light some well allocation issues, and the tolerances need to be adjusted to reflect this. The THP pressure tolerances need to reflect uncertainty in lift curves.

Full background and discussion of these issues may be found in SPE 100206 History Matching with Production Uncertainty Eases Transition into Prediction.

18.2 Number of runs

The engineer needs to specify how many scoping and how many refinement runs are to be submitted. There are several considerations:

- How long does each run take?
- How long before the engineer wants to review results (i.e. doing runs over a weekend or overnight or while on vacation)?
- How many simultaneous runs (depends on cluster and available hardware)?
- How many modifiers?
- How complex or difficult is the history matching problem?

From experience, most history match problems can see a significant improvement after less than 200 simulation runs, and, where a prediction study is being performed, a further 200 runs should result in a stable prediction uncertainty quantification.

Because the overall likelihood is built up from individual estimator point likelihoods, the total number of modifiers is not necessarily so important. If there is a large number of wells, and the modifiers are regional, then the modifiers will mainly affect the wells in that region. So the modifiers are effectively portioned into subsets, and the history match can proceed with fewer runs than the number of modifiers.

As an example, a reservoir may be partitioned into 8 regions which have little intercommunication. Each region may have 32 modifiers, so the total is 256. However, effectively we are doing 8 ‘almost independent’ simulations in parallel, so the necessary number of runs is still low.

This is also a good reason to choose individual well estimator points as well as field estimator points – the field estimator points depend on all modifiers, whereas well estimator points depend on a much smaller subset.

An additional consideration is the use of discrete modifiers. This makes the history match problem harder, and clearly if there are 200 possible discrete values for a particular modifier, then at least 200 (and probably a lot more) runs will be needed. It is recommended that discrete modifiers are kept to a minimum where possible.

Scoping runs are defined using a Latin Hypercube experimental design. This design ensures that each modifier is explored across its complete range, and the number of design points is independent of the number of modifiers.

Having said all the above, a rough starting point for the number of runs would be 25 – 50 scoping runs, followed by sets of 50 refinement runs. After each set of refinement runs, the results should be evaluated. A set of runs would normally itself consist of simultaneous runs, so 50 refinement runs may be performed with 10 simultaneously. Sets of runs are performed asynchronously – new runs can be started without waiting for completion of all simultaneous runs.