



SERAFIM FUTURES TECHNICAL REFERENCE

STRUCTURES, CALCULATIONS AND
CONVENTIONS USED IN THE FUTURES
PRODUCTION FORECASTING APPLICATION AND
DATABASE

26 July 2023

Written by:
Peter Cunningham, Jim McCann,
Ahmed Khamassi, Giel Krijger

SERAFIM

info@serafimltd.com

P. +44 (0)2890 421106

www.serafimltd.com

Contents

1. Introduction 3

 Software Installation and License Authorisation 4

 Adaptive and Fixed Versions..... 4

 ClickOnce 4

 Windows Installer (.MSIX) 6

 Pre-Requisites 6

 License Authorisation 7

 Update Policy..... 9

2. Structure and Data..... 12

 Models, Spectres and other Database content..... 13

 Database Schema and its Updating..... 14

 Database and Application Security..... 18

 Database Access 18

 Managing Access to a Model..... 18

 Approval Structures Within a Model 19

 Access to Functionality 20

 Structure within the Model – Scenarios, Projects and Forecasts 23

 Importing Historicals, Well Tests & Forecasts to FUTURES..... 26

 Providing Data to Other Databases and Applications 29

 Reading Compressed Data From the FUTURES Database..... 32

3. Calculations..... 34

 Standard, Roll-up, Variant and Difference Forecasts 35

 Input of Uptime 44

 Reporting of Uptime..... 46

Solving the Network	47
Network Splits	51
Pressure Calculations	52
Reservoir Inflow Performance Relationship.....	53
“Globally Approximate, Locally Exact” Algorithm for Incorporating Non-Linear (Pressure) Constraints	57
Pipeline Linear Coefficients (if no VFP Table).....	59
Rate and Reserves Multipliers.....	62
Free-gas: Liquid Ratio (FGLR) Method.....	65
C-curve Decline.....	67
Using UR as a Fundamental Decline Curve Parameter	70
Stochastic Calculations	72
Higher-Level (Reservoir etc) Decline Curve Analysis.....	75
Impact Calculations	77
Spreadsheet Calculations and Exports	78
Spreadsheet Calculations.....	78
4. References	82

1. INTRODUCTION

This document gives descriptions of the structures, calculations and conventions used in Serafim FUTURES. It should read in conjunction with the detailed “Help” and guidance notes in user dialogues in FUTURES.

Serafim FUTURES is a software application and database for forecasting and managing oil, gas and associated production profiles. The user specifies well production potentials through decline curve analysis or by importing externally generated profiles (e.g. from multiple simulation models) or by means of simple material balance calculations. FUTURES then calculates the effects of facility rate and pressure constraints, gas-lift allocation, new wells, shut-ins, debottlenecking etc. The resultant production and injection profiles can be used for long-term forecasting (for field development planning or reserves evaluations) or, if FUTURES is supplied with information about operational activities and shut-ins – for short-term forecasts (for operational scheduling and planning).

FUTURES formulates the required calculations as the mathematical optimisation of a large, but strictly linear function. Application of the classic Simplex algorithm and simultaneous, parallel calculation of multiple forecast enables FUTURES to deliver robust and rapid forecasts. Non-linear behaviour in pressure calculations is handled by an innovate version of successive linear programming, called GALE (“Globally Approximate, Locally Exact). In GALE, local solutions for individual wells (calculated with a fixed THP) are used to determine the conditions at which the well inflow performance and tubing performance are linearised for use in the global solution.

FUTURES stores its application data and output in one or more FUTURES databases, which can be on Oracle or SqlServer, or can be SqlServerCE files (for small databases for data transfer or individual work). The FUTURES application manages the database schema.

FUTURES is normally web-deployed i.e. the individual user installs it from the Serafim web-site, using Microsoft ClickOnce or Microsoft MSIX App Installer. Bug fixes are then downloaded semi-automatically, which allows over-night fixing of almost all bugs detected by users. In contrast, Serafim will deploy new functionality (as opposed to bug fixes in existing functionality) only after FUTURES’s main customers have had the opportunity for their own acceptance testing. If, instead, you wish to manage deployment and updating using Microsoft Software Centre or similar, it is also possible to download a Microsoft MSIX .appx file from the Serafim web-site.

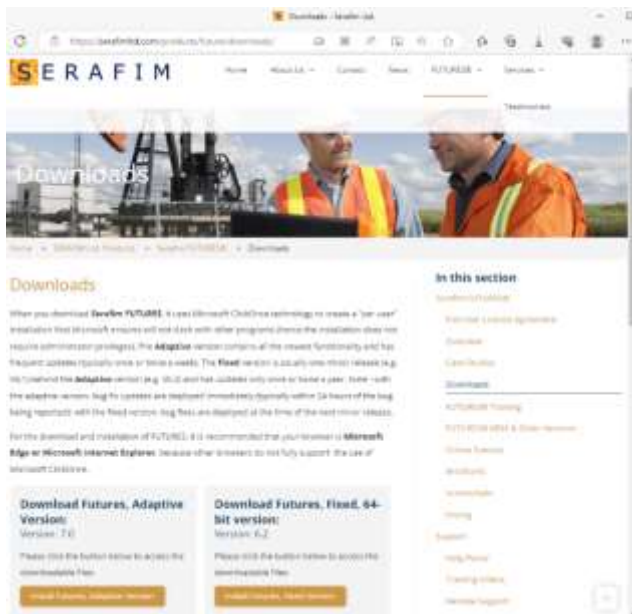
SOFTWARE INSTALLATION AND LICENSE AUTHORISATION

ADAPTIVE AND FIXED VERSIONS

FUTURES is available in “adaptive” and “fixed” versions. The Adaptive version contains all the newest functionality and has frequent updates (typically once or twice a week). The Fixed version is usually one minor release (e.g. V6.1) behind the Adaptive version (e.g. V6.2) and has updates only once or twice a year. Note - with the adaptive version, bug fix updates are deployed immediately (typically within 24 hours of the bug being reported); with the fixed version, bug fixes are deployed at the time of the next minor release.

CLICKONCE

It is recommended that FUTURES is installed directly from <https://serafimltd.com/products/future/downloads/> using Microsoft ClickOnce. With this method, users are prompted to install updates when these become available. FUTURES can also be installed from a Windows MSIX .appinstaller or .appx file (see below for details).

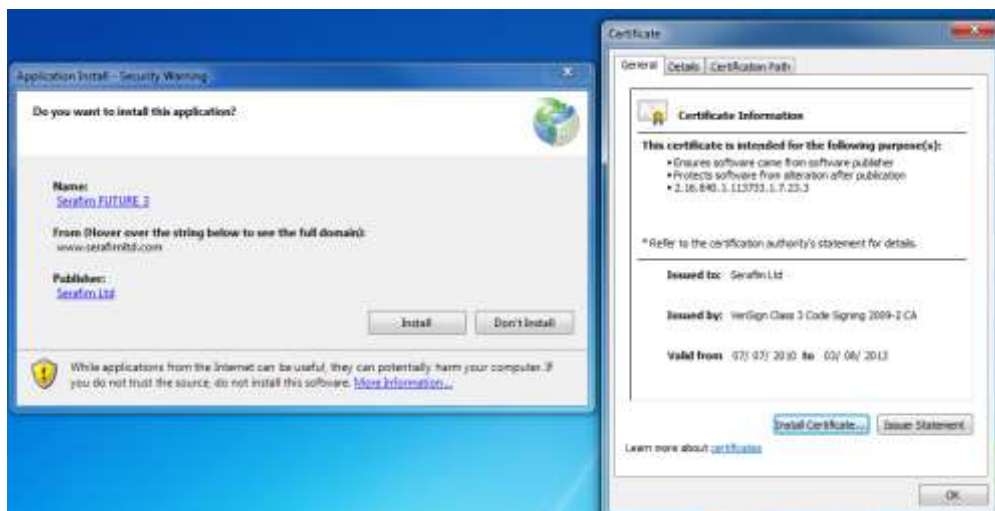


Microsoft ClickOnce is a standard Microsoft technology, designed to be a better alternative to Windows Installer for simple installation requirements. It has the following features

- ClickOnce-deployed applications are deployed “per user” (not “per machine”) and cannot cause the over-riding of .dlls used by other applications. In consequence, a ClickOnce-deployed application cannot cause other applications to break.

- As a result, Windows does not require Admin privileges for the installation of ClickOnce-deployed applications i.e. ordinary users can install them. (In contrast, .msi-deployment applications can, if they are set to use shared versions of .dlls, give rise to problems with existing applications. Hence, Windows requires Admin Privileges for their installation).
- ClickOnce can provide updates from the web. FUTURES makes use of this, so users will be prompted (but not forced) to update to the latest version of FUTURES.
- The ease of updates makes it possible to put out quick fixes of bugs (usually overnight). This, together with the fact that the ClickOnce-deployed version of FUTURES cannot cause other applications to crash (while an error in the .msi deployment might cause such a problem), means that ClickOnce delivers improved stability and reliability.

FUTURES is a full-trust application and so, in order to run, needs to be accepted as such by the Common Language Runtime (i.e. .NET Framework). ClickOnce default behaviour is to grant only restricted trust (“Internet Zone”) to software deployed from the web. The FUTURES assembly is signed by Serafim (using certification from Verisign). If Serafim has been set to be a “Trusted Publisher”, then FUTURES will be automatically given full-trust. Otherwise, the user will be prompted by ClickOnce, as in the screen-shot below, “Do you wish to install this application?” i.e. approve Privilege Elevation.



If Privilege Elevation is disabled (as it may have been done by corporate IT policy), then the installation of FUTURES from the web would fail until Serafim is added to the list of Trusted Publishers

More details are available at

<http://msdn.microsoft.com/en-us/library/t71a733d.aspx>

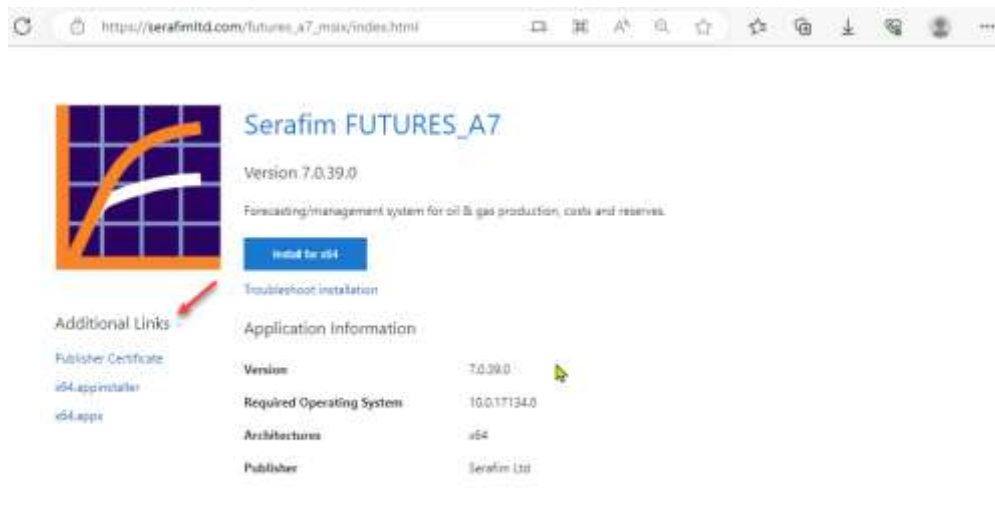
[http://msdn.microsoft.com/en-US/library/76e4d2xw\(v=vs.110\).aspx](http://msdn.microsoft.com/en-US/library/76e4d2xw(v=vs.110).aspx)

<http://msdn.microsoft.com/en-US/library/01daf08f.aspx>

WINDOWS INSTALLER (.MSIX)

FUTURES can also be installed using Microsoft .MSIX. The relevant links are at

https://serafimltd.com/futures_a7_msix/index.html



Clicking the “Install for x64” runs the MSIX .appinstaller file that installs FUTURES and sets up automatic updating similar to that of ClickOnce.

Alternatively, clicking on “Additional Links” (the red arrow in screen-shot above) enables you to download the MSIX .appx file which performs a traditional installation that can be managed by standard corporate IT tools such as Microsoft Software Center.

PRE-REQUISITES

If you are using an Oracle database and prefer not to connect FUTURES to Oracle in Devart “Direct Mode”, you will also need to install Oracle Client on the PC. NB – If you are using FUTURES(64-bit)_5, you will need to install the 64-bit version of Oracle Client; if you are using FUTURES_5, you will need to install the 32-bit version of Oracle Client. (Oracle Client can be downloaded from <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-win64soft-094461.html> and <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-win32soft-098987.html>)

Other software components used by FUTURES (Microsoft SqlServerCe 4.0, Devart dotConnect, Steema TeeChart etc) are installed as integral parts of the FUTURES installation (i.e. separate copies are made; these are installed locally with FUTURES and cannot be accessed by other applications). Given that the total size of the FUTURES installation files is approximately 200 MB, the extra disk space required for these local copies is sufficiently small as to be irrelevant.

LICENSE AUTHORISATION

To prevent unlicensed use of the program, FUTURES uses a system of license authorisation keys, which need to be supplied by Serafim. After installation (or after the end of the period of validity of an existing key), the user is prompted to provide a valid authorisation key:-



There are two types of license authorisation.

With “For this machine” authorisation –

- Inform Serafim (email future@serafimltd.com or telephone +44 28 9042 1106) of your six-figure “Installation ID” (red arrow above), which is specific to your machine.
- Serafim will generate a matching eight-figure authorisation key and inform you of it.
- Enter the Authorisation Key. FUTURES will display its main window and store the authorisation key on the machine, where it can be read automatically each time FUTURES is started.

With “For Database” authorisation (available if you have a global or production unit licence for FUTURES),

- Click on “For Database”
- Click the “Browse...” button, select the .ftrl file pointing to your Oracle or SqlServer FUTURES database (see next section for details on how this is first set up) and give the .ftrl password when prompted
- FUTURES will now display the nine-figure Database ID number (see red arrow on screen-shot below).
- If the database authorisation key has previously been entered, you will see displayed. Click “OK” and you are done. FUTURES will now display its main window.
- If you are the first person to use “For database” authorisation, you will need to obtain and enter the authorisation key
 - Inform Serafim (email future@serafimltd.com or telephone +44 28 9042 1106) of the nine-figure “Database ID” (red arrow below), which is specific to your database.
 - Serafim will generate a matching eight-figure authorisation key.
 - Enter the Authorisation Key. FUTURES will display its main window and store the authorisation key on the database.



The advantage of “For Database” authorisation is that you do not need to contact Serafim for a new authorisation key each time you wish to use FUTURES on a new machine.

It should be noted that it is possible to use one database for “For database” authorisation and, subsequently, another database for your work – FUTURES allows the use of multiple databases within a FUTURES session.

UPDATE POLICY

FUTURES is normally web-deployed i.e. the individual user installs it from the Serafim web-site, using Microsoft ClickOnce or MSIX. Bug fixes are then downloaded semi-automatically, which allows over-night fixing of almost all bugs detected by users. In contrast, Serafim will deploy significant new functionality (as opposed to bug fixes in existing functionality or minor new functionality) only after FUTURES's main customers have had the opportunity for their own acceptance testing.

Changes to software often introduce bugs and FUTURES is no exception. FUTURES users require a high degree of reliability, especially when FUTURES is being used for important business processes – reserves updates, operation planning etc – with tight deadlines. In order to prevent FUTURES reliability being harmed by the bugs introduced by software changes, it is necessary to ensure that there is suitable, extensive testing (both by Serafim and by its customers) prior to release to ordinary users of any software changes likely to introduce bugs. It is also necessary for customers to have some degree of control, so that, for example, major changes are not deployed at a critical time during the annual reserves update.

Hence, software changes that might give problems (most new functionality, all code refactoring etc, all database schema changes) are deployed first as Beta releases. Such a Beta release can be installed in parallel with the main release version, so customer management and IT can organise suitable internal testing on copies of their own FUTURES models. Typically, there is at least a month available for oil-company testing prior to the changes being deployed in the main release version. Furthermore, if any customer wishes for more time for testing, this is arranged. In other words, customers have control over the timing of the release of such software changes.

However, there are some software changes that have a very low probability of introducing further bugs – namely

- small bug fixes in areas of code that are used in a tightly controlled, restricted manner;
- new functionality that does not interact with existing functionality e.g. a new option for exporting data.

Given that Serafim can fix 95% of bugs over-night, it is logical to release the bug fixes as quickly as possible (without waiting for customer testing and IT approval etc), for the following reasons

- The resultant loss of user time from the bug is measured in hours – a couple of hours' when encountering the bug and informing Serafim; a few hours waiting prior to the bug fix being deployed.

- The bug is typically encountered by only one user before it is fixed. If known bugs were not immediately fixed, then a bug might be encountered by multiple users. This would result in the loss of considerably more user time.
- Each bug being encountered only once, this reduces the total number of encounters of bugs i.e. the software is more reliable for the average user.
- If bugs were not immediately fixed and were encountered and reported by multiple users, this would also lose Serafim time and reduce its ability to provide support to users.
- Users value very highly the prompt fixing of bugs – or, at least, the prompt fixing of the bugs that they themselves have encountered. It means that users do not have to remember work-arounds or have to explain oddities in results that come from unfixed bugs.

It should be noted that FUTURES has a modern object structure with a high degree of inheritance and some use of reflection. This means that most of code is used in a “tightly controlled, restricted manner” i.e. that most bug fixes are unlikely to introduce new bugs and are suitable for immediate deployment. Of course, there are always some bug fixes that have a significant risk of introducing new bugs. These bug fixes are *not* deployed immediately, but are incorporated in the next Beta version, that will receive proper testing prior to deployment.

FUTURES V4.3.10.13, released on 7 July 2016, gives an example of bug fix that was judged unlikely to introduce further bugs. Incremental profiles are shared out between one or more physical wells. At each time-step, FUTURES calculates what fraction of the incremental profile is given to each physical well and this fraction cannot be negative. Prior to V4.3.10.13, FUTURES had a quality-control check to ensure that the fractions were greater or equal to zero. The check was carried out without explicit allowance for round-off error (making use of the implicit treatment of round-off within .NET Framework’s double precision variables). However, one customer discovered this was insufficient when there were multiple negative incremental profiles. The problem was fixed with a modification of three lines of code (see below), tested by Serafim and then immediately deployed.

2. STRUCTURE AND DATA

In common with most modern applications, FUTURES stores its data in one or more relational databases. This means that there are two aspects to structure and data

1. Structure and data as seen by the user of the application.
2. Structure and data as seen from a database management or SQL point of view

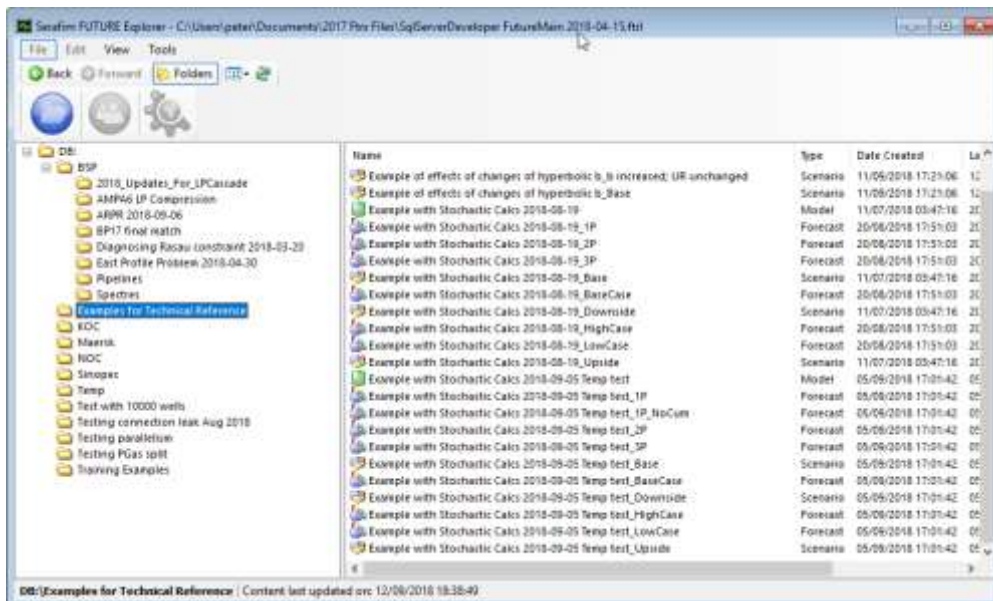
Most of the discussion below is about (1), but aspect (2) is also discussed, for the benefit of anyone who wishes to read data from the FUTURES database using database queries/views.

MODELS, SPECTRES AND OTHER DATABASE CONTENT

While the way the FUTURES application works is designed to be similar to Microsoft Excel, there is an immediate difference. In Excel, the user opens and edits a spreadsheet that is stored as a file on the computer file system, the user in FUTURES opens and edits a “Model” that is stored as a pseudo-file in a pseudo folder structure in the FUTURES database.

Besides Models, there are a number of other items held in the FUTURES pseudo-file system. The full list is

- Models – these are what are opened and edited. As described later, a Model can have the structure to define and calculate not only one, but a whole range of different possible forecasts.
- Scenarios – Each Model has one or more Scenarios. Each Scenario represents a separate set of possible properties. The most standard set-up has three Scenarios – “Low”, “Medium” and “High”.
- Forecasts – Each is a possible outcome; the equivalent of a simulation run.
- Spectres – A SPECTre is a “Set of Profiles Externally Created” that has been imported into FUTURES. The set of profiles might be of individual well historical profiles or of predicted well profiles from a simulation run.
- Pipelines – Tables of pipeline pressure changes.
- Spreadsheets – Loaded into the FUTURES database and used for custom calculations.
- Images – Bitmaps etc used as background for network diagrams.



DATABASE SCHEMA AND ITS UPDATING

FUTURES uses a very simple database schema, comprising of simple tables and their associated indexes. The schema contains no links between tables and no views/queries, because their logic is handled by the application, not the database schema.

The FUTURES application, when it starts up for a user session, manages the FUTURES database schema and checks that it is up-to-date. The checks usually take a fraction of a second and are necessary to ensure that FUTURES can read and write data to the database without errors.

As part of this process, the FUTURES application adds any missing database tables and columns, in line with modern practice with Agile software development (as described by terms such as “Database Release Automation”).

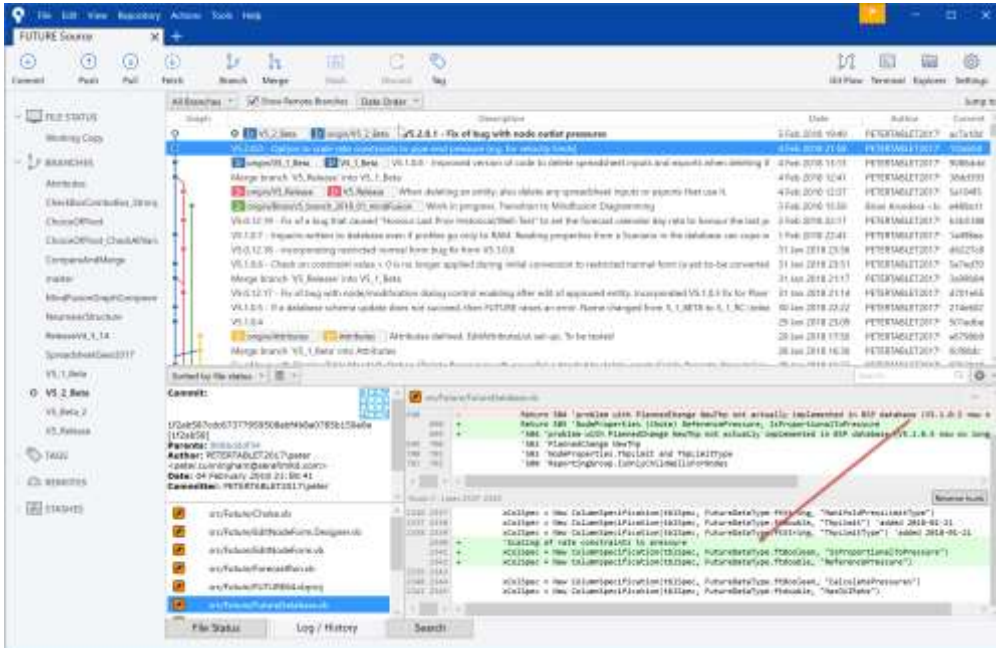
In order to ensure that such database schema changes are tested and verified in line with customer change management procedures, Serafim makes such changes only in Beta versions. In other words, routine updates of the release version of FUTURES (as used for quick bug fixes) *do not update the database schema*.

FUTURES’s use of Database Release Automation methods makes it possible for customers to carry out their change management procedures quickly and efficiently. Since the changes are purely additive, any failure to fully implement them (as can easily happen because of locks, time-outs etc) leaves the database in a situation where it can still be used by earlier versions of FUTURES.

FUTURES is designed so that it is possible to test a FUTURES Beta version (and its database schema changes) on a production database. The design features that allow this are

- The changes are purely additive, so the failure of any of them leaves the database still containing all the tables, columns and data required by earlier versions of FUTURES.
- If there is a partial or total failure of an update to the schema, FUTURES prevents the current session (running the Beta version that needs the new columns etc) from accessing the database, hence avoiding errors and consequent data corruption;
- If there is a partial or total failure of an update to the schema, the next Beta version session checks for the existence of each of the required new columns etc before submitting the “CREATE” SQL DDL command, so the update process can work successfully even if a previous attempt failed or partly failed.

- The logic for the updates (and the associated tests for existence etc) is contained in 500 lines of FUTURES object-oriented code that has been in use for many years and has been extensively tested. This code is normally never touched. Instead, typical changes to the database schema are implemented with high-level commands of the form illustrated below.



Indeed, it is recommended that, unless you are notified otherwise by Serafim, Beta versions are tested on your production database, for the following reasons

- As discussed above, this will not harm your production database.
- Even if you test schema updates on a test database, you still need to test them on the production database. It is possible for the updates to fail on the production database even if they succeeded on a very similar test database (for example, because of locks or time-outs). If you do not test the schema updates on the production database prior to deployment of the updated release version of FUTURES, you run the risk of the schema update failing and users being unable to access your production database (N.B. – in this situation, users could, in Windows, choose to revert to using their previous version of the FUTURES release and so continue working).
- Users often wish to combine testing of the new features with real work and so would not want to run of the risk of losing the results of work carried out with the Beta version. If such work has to be carried out in test database, then you need to ensure that the test database has adequate back-ups etc.

The database tables belonging to FUTURES are as follows (for V5.0 in August 2017):-

Table Name	Relevant for typical external queries	Remarks
ACTIVITYIMPACTS	Yes	Output of effects of individual Curbs, Planned Changes and new Wells
ADJUSTABLETABLES		
ASSETS		
ASSETTYPES		
CASEMEMBERS		
CASES		
CUMULATIVESANDPRESSURES	Yes	Profiles in uncompressed format (one record per date per entity per run)
CURBPROPERTIES		
CURBS		
DATABASEPROPERTIES		
DEPENDENCYLINKS		
DIFFERENCEFORECASTS		
ENTITYACCESSLIST		
EXPORTFORPHDWIN		
EXPORTFORVOLTS		
FLOWPRESSUREARRAYS		
FLOWPRESSUREDATA		
FLOWRATEPAYMENTTERMS		
FUTUREFOLDERS	Yes	Describes the folder structure
GUIHCFIELDS		
GUINETWORKS		
GUIPLANNEDCHANGES		
GUIPLATFORMS	Yes	Node properties
GUIRESERVOIRS	Yes	Reservoir properties
GUIWELLS	Yes	Well properties
IMAGES		
LOGENTRIES		
PAYMENTSMADE		
PIPELINETABLES		
PREVIOUSVERSIONSOFRUNS		
PROFILEARRAYS	Yes	Profiles in compressed format (one record per date per entity per run)
PROJECTGROUPS		
PROJECTS		

RECURRENTPAYMENTS		
REPORTINGGROUP	Yes	Defines membership of Reporting Group
RUNACCESSLISTS		
RUNS	Yes	To find Run GUID from long name etc
SPREADSHEETCALCULATIONS		
SPREADSHEETEXPORTS		
SPREADSHEETINPUTS		
SPREADSHEETOUTPUTS		
SPREADSHEETS		
STANDARDPROJECTS		
TEMPSTRINGLISTS		
TIMESTEPSSPECIFICATIONS		
USERGROUPMEMBERS		
USERGROUPS		
USERWINDOWSIDS		
WELLSFLOWINGTONODEORGROUP	Yes	List of wells belonging to Node, Field, Reservoir, Reporting Group, Project etc
WORKITEMS		
WORKSEQUENCES		

DATABASE AND APPLICATION SECURITY

There are four aspects to the management of access to FUTURES data:

1. Database access – Normally, direct access to the FUTURES database is restricted to the FUTURES application – the ordinary user is unable to access the database, except indirectly using the FUTURES application and a suitable .ftrl file (which contains the encrypted log on information needed for the FUTURES application to access the database).
2. Access to a Model – For each Model, you can assign each user either Read/Write privileges, Read privileges or no access. After all work is completed, Models can be “Frozen”, which means that nobody can alter them further.
3. Approval structures within a Model – To keep track of who has done what, users can sign-off individual Wells, Nodes, Planned Changes and Curbs within a Model.
4. Access to functionality

DATABASE ACCESS

Security always comes at a cost in terms of performance – because of the time and added complexity of security checks. In a software application, it is possible – for example, by ensuring that the relevant checks are carried out only once – to manage the impact on performance, to the point where it becomes negligible. In a database, it is possible to implement any desired security system (typically by giving users access to suitably defined views, rather than to the underlying tables), but this usually has a significant cost in terms of speed or complexity.

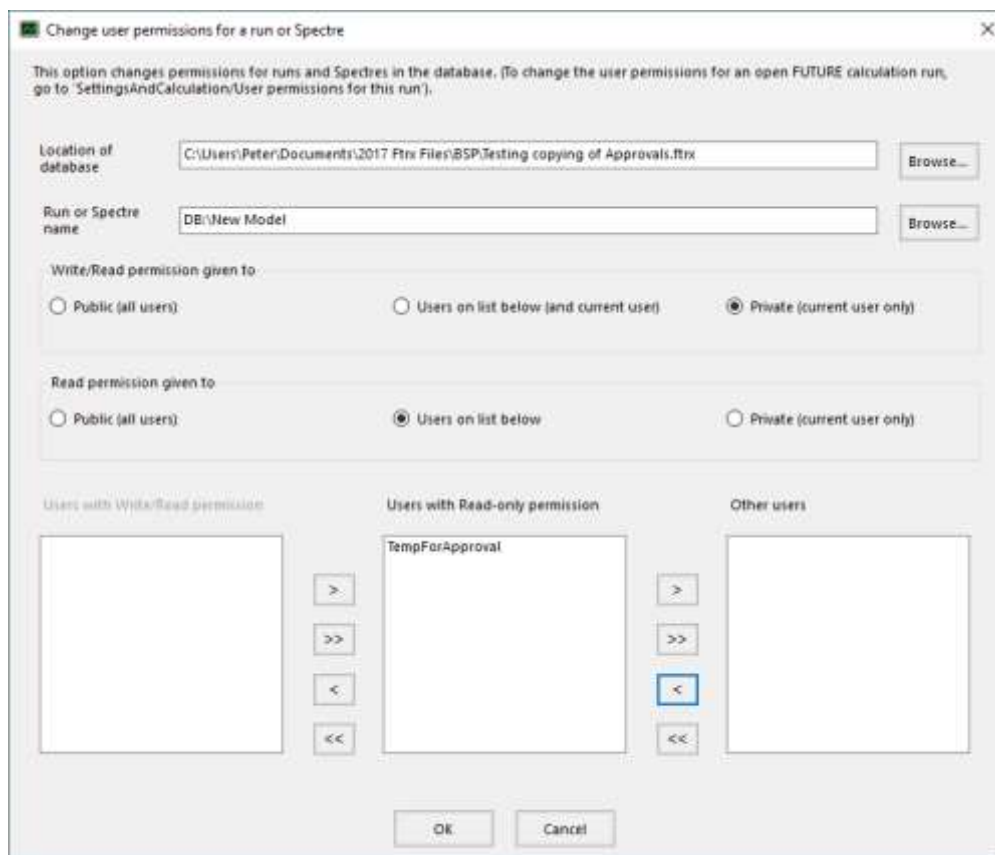
FUTURES (and many other applications that require fast reading and writing of large amounts of data) is designed to implement security within the application. FUTURES itself has high-privileged access to all the data, but the high-privilege access credentials are kept hidden from the user, preventing ordinary users from having high-privilege direct access to the database. Instead, they are limited to accessing it via the FUTURES application (i.e. if they try to use Oracle or Microsoft database management tools or Excel, Spotfire etc, they would not have knowledge of the high-privilege credentials used by FUTURES – they could of course be provided with separate, low-privilege credentials that would give them access to suitably defined views created for this purpose using database management tools).

MANAGING ACCESS TO A MODEL

Given that the ordinary user does not have direct high-privilege access to the database, FUTURES data can be stored in simple tables that contain data of various levels of confidentiality. Management of access to the data is carried out by the

FUTURES application, using permissions defined for each Model. Each user can be given one of the following three levels of permissions for a Model

- Write/Read – can open and subsequently save the Model; when the Model is opened, the database then puts a lock on it, so that other users are restricted to read-only access (to prevent the edits of one user over-writing those of another). Only users with Write/Read permissions for a Model can delete it.
- Read-Only – can open a read-only copy of the Model. If it necessary, the user can save the Model under a new name (or in a separate database).
- No access



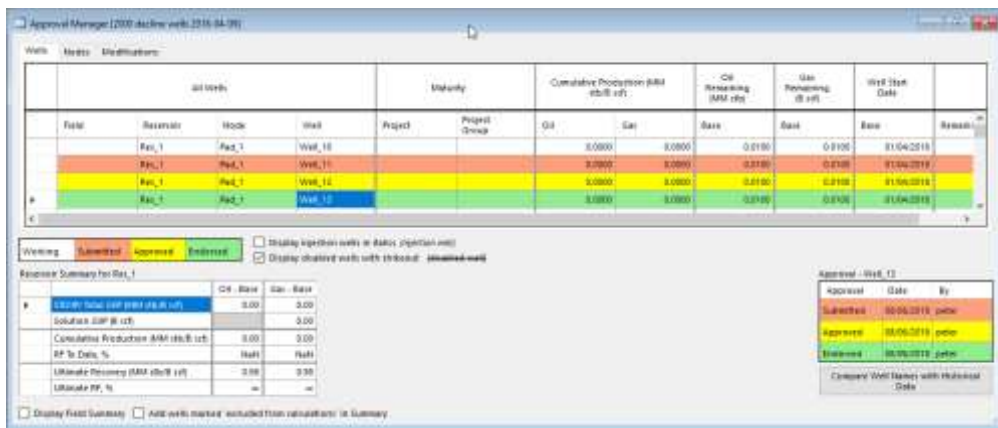
Final versions of Models can be “Approved and Frozen” – that is, made read-only for everyone (including users with Write/Read permission). It is recommended that whenever a Model’s Forecasts are exported from FUTURES and used in subsequent business processes, the Model is frozen, so that a proper audit trail can be maintained.

APPROVAL STRUCTURES WITHIN A MODEL

Freezing a Model allows you to sign-off the Model as a whole. However, a typical Model is constructed and updated by many people. In order to keep track of who

does what, it is useful to be able to sign-off individual elements within the Model. FUTURES provides a system to allow users to sign-off the individual Wells, Nodes and Modifications (Planned Changes and Curbs) that they have been editing. The system also allows Team Leaders (or similar) to approve the sign-offs and finally the relevant manager to endorse the sign-offs, as illustrated below with four wells – one in each of the following categories

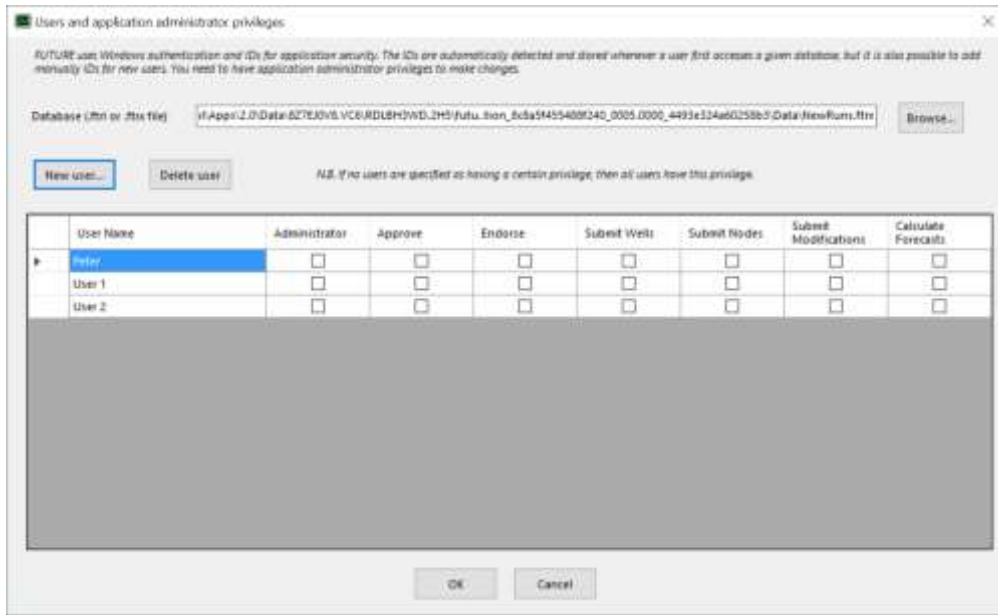
- Working – “Well_10”, with white background; not signed-off; can be edited as required.
- Submitted for Approval – “Well_11”, with red background; cannot be further edited (unless the submission is cancelled and it is set back to “Working” category).
- Approved – “Well_12”, with yellow background (as above, cannot be further edited unless reset to “Working” category)
- Endorsed – “Well_13”, with green background (as above, cannot be further edited unless reset to “Working” category).



Use of the approval structures within Models is subject to the user having appropriate privileges, as discussed below.

ACCESS TO FUNCTIONALITY

Within each database, FUTURES maintains a list of user privileges (FUTURES Main Menu > Database > Users and Administrators), as follows



The different privileges (which apply to all Models for which the user has Read/Write access in the given database) are as follows

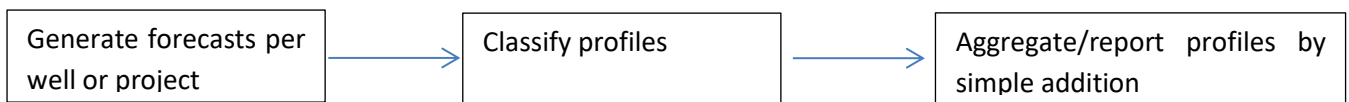
Privilege	Powers
Administrator	Assign user privileges. Cancel the freezing of a Model. Manually remove locks on Models. Force deletion of data (Spectres, Pipeline tables etc) used by Models.
Approve	Approve "Submitted" Wells, Nodes and Modifications. Approve and freeze a Model.
Endorse	Endorse "Approved" Wells, Nodes and Modifications. Endorse an approved/frozen Model.
Submit Wells	Edit and then sign-off a Well and submit it for approval. (Without this privilege, users will be able to edit any Well that is in the "Working" category, but will not be able to sign-off the Well).
Submit Nodes	Sign-off a Node and submit it for approval.
Submit Modifications	Sign-off a Modification (Curb or Planned Change) and submit it for approval.
Calculate Forecasts	Use "Save with Calculations" command and so generate or update Forecasts.

If you need to give the same user different detailed levels of privileges for different Models (so that, for example, the user needs to be allowed to edit and submit Wells in one Model, and to edit and submit Nodes in another), then it may be necessary to work with multiple databases.

STRUCTURE WITHIN THE MODEL – SCENARIOS, PROJECTS AND FORECASTS

One of the key features of FUTURES is its ability to take proper account of facility and pipeline constraints. With such constraints, output and aggregate profiles will not always be simple sums of input profiles. One consequence of this is that a simple scheme of classification of input profiles (as could be used in a system where output profiles are simple sums of input profiles) would not work. Instead, FUTURES provides a structure of “Scenarios”, “Projects” and “Forecasts” that is suitable for the full range of normal, forecast-related business processes (reserves reporting, operations planning, development planning etc.). However, the general-purpose nature of the FUTURES structure means that it is necessary for each business unit or company to choose the most appropriate implementation of the structure.

If there is no interaction between wells or projects, then aggregate profiles are simple sums of individual profiles for individual wells or projects. Each individual profile can be classified (as e.g. “Developed Producing”, “Approved”, “Contingent” etc) and aggregation/reporting can be done by simple addition. A simplified workflow might be

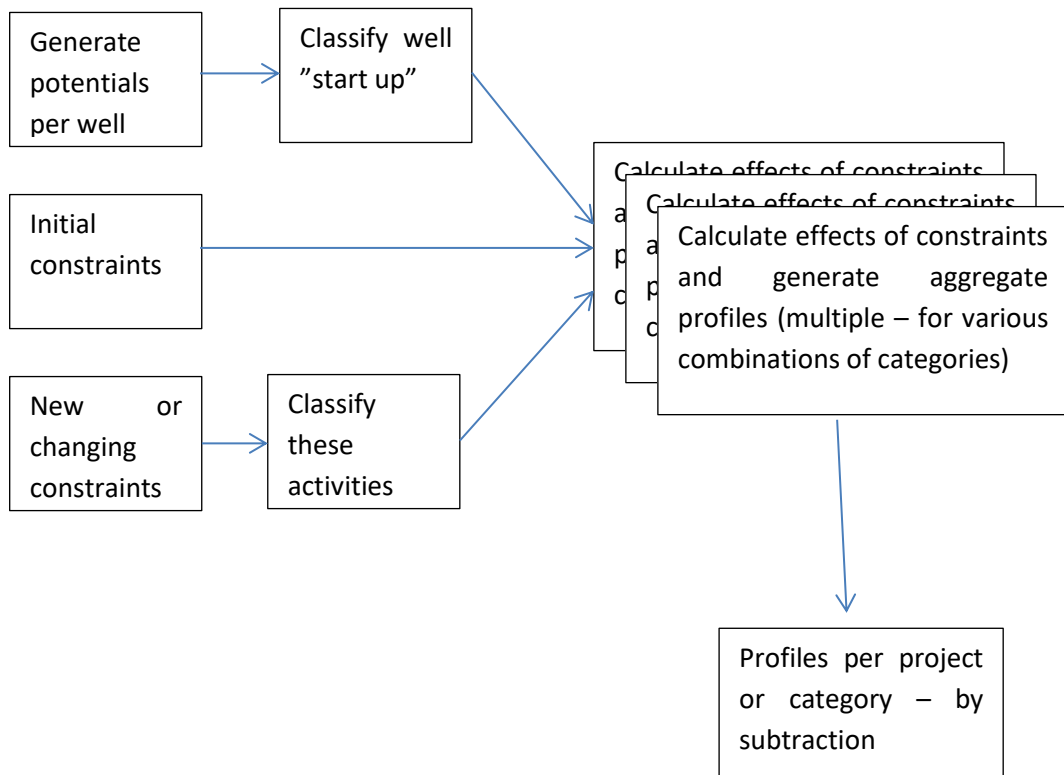


Where there are interactions (for example, when new wells would produce to existing facilities and cause production from older wells to be cut-back), it is usually important that project economics takes account of such interaction. So, the project profile for economic analysis is usually the difference

(Asset total profile if the project is carried out) – (Asset total profile if the project is not carried out)

This means that the calculation of the project profile of a given classification (e.g. “Contingent”) usually involves activities and wells of different classifications. Even production from an individual developed producing well may differ depending on whether or not a certain debottlenecking project is carried out. So, the one developed producing well may have some production that is “Developed Producing” and further production that should be in the “Approved” category. In this case, decline analysis of the well yields not a final forecast, but rather an estimate of the well potential (usually in terms of potential rate vs cumulative production).

A simplified work-flow for production forecasting with constraints might be



The structure within FUTURES for implementing such a work-flow uses the following elements

MODEL – The Model in FUTURES is what the spreadsheet is in Excel. It is the object that is opened, saved etc. A single Model should be set up to generate the whole family of required forecasts, using the other elements of structure as follows.

SCENARIO – Scenarios are used to describe the range of outcome uncertainties. Each Model can have as many Scenarios as required. Each Scenarios contains a separate set of properties (e.g. rate vs cumulative relationships) for each well, node or reservoir in the Model. Typically, you have three Scenarios, namely “High”, “Medium” and “Low”.

ACTIVITIES – There are three types of activities in FUTURES – PLANNED CHANGES (permanent changes to wells or facilities), CURBS (temporary changes to wells or facilities) and starting up wells (NB – even existing wells are counted as “starting up” at the beginning of prediction)

PROJECTS – In FUTURES, “Projects” are grouping of activities. A single project can contain a mixture of well start-ups, Curbs and Planned Changes.

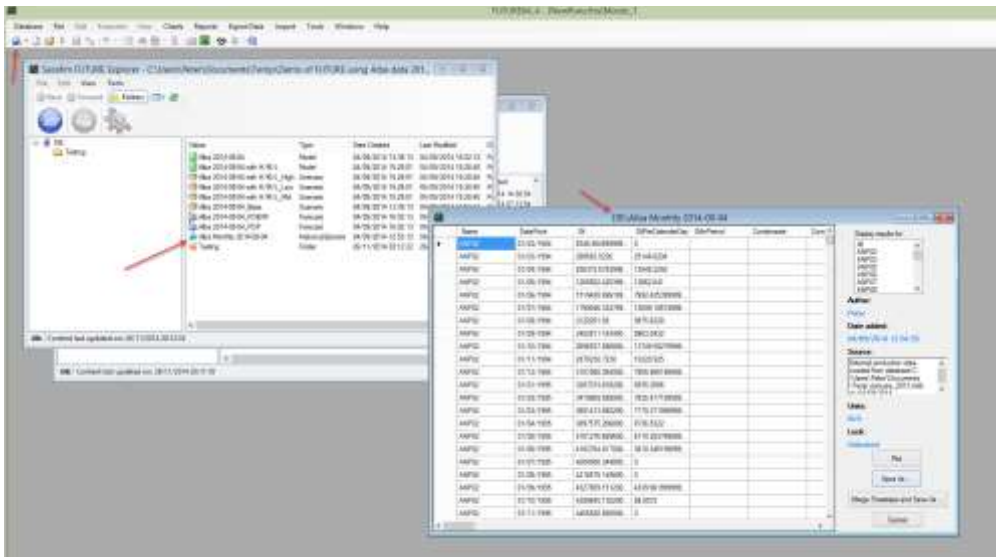
FORECAST – A Forecast is a prediction, based on a set of engineering and geological assumptions, of production rates and pressures for a defined set of wells, facilities and activities over a period of time. Specifying a Forecast in a FUTURES Model can be a simple matter of selecting a Scenario to use (a Scenario equates to the set of assumptions) and selecting which Projects to include, or it may be necessary to specify detailed stochastic settings and/or Forecast type settings (discussed in the “Calculations” chapter).

PROJECT GROUPS – An option to reduce the number of repetitive clicks to make. If you group some of your projects in “Project Groups”, then the selection of whether or not to include those Projects in a given Forecast is done for the relevant Project Group as a whole.

IMPORTING HISTORICALS, WELL TESTS & FORECASTS TO FUTURES

FUTURES makes use of a variety of production profiles that originate elsewhere – historical production allocation data, well-test data, simulator forecasts etc. FUTURES stores the relevant data (usually after some non-trivial processing – e.g. calculation of cumulative production at the date of a given well test) in its database. In other words, FUTURES does not maintain live links to other databases, but instead periodically, at user command, reads and stores the required data, either updating earlier production profiles or saving the data as a new, separate version.

Such external production profiles are stored in what FUTURES calls “Spectres” – Sets of Profiles Externally Created. Once loaded, a Spectre can be examined within FUTURES by opening the FUTURES Database Explorer window and double clicking on the Spectre.



The Spectre data is stored in the database in two forms

1. Summary information (meta-data) about the source of the Spectre, the date loaded etc is stored in the table “Runs”.
2. The production and injection data itself is stored either as database double precision numbers in the “CumulativesandPressures” table (this holds the production and injection data, with one line per date per entity (usually a well)), or it is stored as compressed arrays of numbers in the “ProfileArrays” table (with one line per entity)

Simulator profiles can be loaded directly from Eclipse RSM files, from suitably formatted Excel spreadsheets or from a suitable database query/view. Historical production allocation and well-test data can be loaded from Excel or from a suitable database query/view. Loading of well-test data needs to be done after the loading

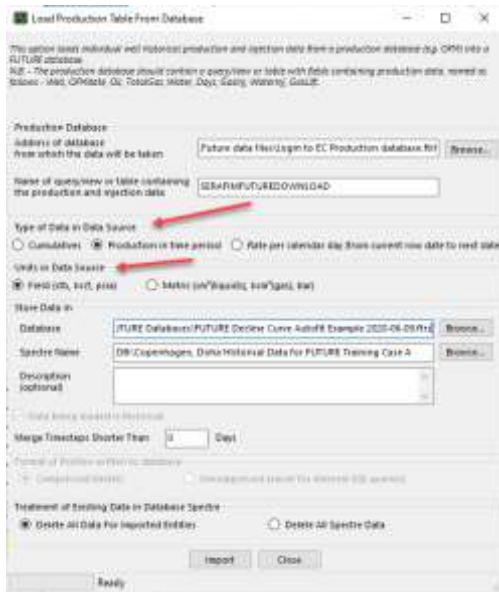
of the relevant historical production allocation data, since the latter is used to calculate, at the exact date of the well-test, the cumulative production, as required by decline-curve rate vs cumulative analysis.

A database query/view or table for the loading of historical production allocation data should have the following fields and should have one entry per time period (usually a month, but any dates, including irregular, are acceptable) per entity (usually a well).

- "Well" The name of the well
- "OFMdate" The date/time of the start of the time period in question (in DateTime data type)
- "Oil" Oil production
- "TotalGas" Total gas production (from the formation and from gas-lift)
- "Water" Water production
- "Days" Number of operating days(days)
- "GasInj" Gas injection (to the formation - not gas-lift) for the month
- "WaterInj" Water injection
- "GasLift" Lift-gas used
- "FBHP" Flowing bottom-hole pressure (optional i.e the field can be omitted)
- "FTHP" Flowing tubing-head pressure (optional)

As illustrated in the screen-shot below, the data can be in either field (stb, kscf, psia) or metric units (sm^3 for liquids, ksm^3 for gas, bara), while the type of data can be

- Cumulatives
- Production in the time period (e.g. in the month)
- Rate per calendar day (in which case "Days" should be operating days per calendar day i.e. uptime fraction)



A database query/view for the loading of well-test data should have the following fields and should have one entry per test. No two tests on the same well should have the same date/time – it is recommended that, if need be, the timings of one test is altered by a small time interval e.g. a minute.

- "Well" - The name of the well
- "DateOfTest" - The date/time of the test (in Date/Time data type)
- "OilRate" - Test oil production rate (in stb/day)
- "TotalGasRate" - Test total gas production rate (from the formation and from gas-lift) (in kscf/day)
- "WaterRate" - Test water production rate (in stb/day)
- "LiftGasRate" - Test lift-gas supply rate (in kscf/day)
- "FBHP" – Flowing bottom-hole pressure (in psia)
- "FTHP" – Flowing tubing-head pressure (in psia)
- "Choke" – Choke setting (in 1/64 inch)

PROVIDING DATA TO OTHER DATABASES AND APPLICATIONS

The main forms of data provided by FUTURES to other databases and applications are

- Production/injection profiles
- Cash-flow profiles
- Reserves reports
- Reports of the impact of activities

The production/injection profiles contain, for each report time-step

- Rates per calendar day
- Cumulatives
- "Capacity" - for Nodes, this is the capacity constraint, expressed in rate per operating day; for Wells and all other profiles (reservoirs, reporting groups etc), this is the sum of the well potential rates per operating day

Cumulatives are given for the exact date/time of the entry – so an entry whose date/time is 2018-01-01 (1 January 2018) gives the cumulative production/injection at 00.00 hrs on 1 January 2018 (so it does not include production/injection that would occur during January 2018). Rates per calendar day are averages over the time period beginning at the date/time of the entry and ending at the date/time of the next entry. So, if the next entry was 1 February 2018, then the rates per calendar day reported for 2018-01-01 are equal to the production/injection that occurs during January 2018 divided by 31 (the number of calendar days between 1 January 2018 and 1 February 2018).

The rates per calendar day, cumulatives and capacity are given for

- Oil (this is the total of hydrocarbon liquids, including condensate)
- Condensate – hydrocarbon liquids calculated, from the FUTURES material balance, to have been, at the time of production, vaporised oil within the reservoir gas-cap; PLUS hydrocarbon liquids that come from the treatment of gas, as described by Node fluid conversion factors
- Total gas – the physical gas flow in the production stream - I.e. produced gas from reservoir plus returning lift-gas (but not including injection gas)
- Produced gas – the part of "Total gas" that comes from the reservoir
- Solution gas – the part of "Produced gas" that was, at the time of production, dissolved in the reservoir oil-leg (as determined by the FUTURES material balance calculation)
- Water – the flow in the production stream
- Lift-gas – the amount of lift-gas supplied

- Gas injection
- Water injection

The production/injection profile also gives uptime and well count information:-

- UptimeFraction – for Wells and Nodes, this is the simple uptime (so a Node might have an uptime of 100% even if its Wells had uptimes of 80%); for all other entities (Reservoirs, Reporting Groups etc), this is the weighted average of the entity's Wells' uptime fractions. See discussion on uptime calculations for details.
- OperatingDays (called "ProductionDaysCumulative" in Export to Excel) - Cumulative operating days
- Producing Wells
- Injecting Wells
- Flowing Wells
- Shut-in Wells
- Active Wells
- Abandoned Wells

Forecast data is held in the database using the Forecast RunID, so this is needed by external queries/views. An easy way to find the Forecast RunID is by specifying the Forecast LongName, for example

```
SELECT Runs.Run, Runs.RunName, Runs.FolderID, Runs.LongName, Runs.RunType,
Runs.ParentRun, Runs.DateOfComputation, Runs.Author, Runs.BackupCopy
FROM Runs
WHERE Runs.LongName="DB:\MyFolder\MyModel_MyForecast";
```

Another way to find the Forecast Run ID is by specifying the LongName of the Model (this will give the Run IDs of all its Forecasts) as follows-

```
SELECT ModelRuns.LongName AS ModelLongName, ForecastRuns.LongName AS
ForecastLongName, ForecastRuns.Run AS ForecastRunID,
ForecastRuns.DateOfComputation, ForecastRuns.Author, ForecastRuns.RunType
FROM Runs AS ForecastRuns INNER JOIN Runs AS ModelRuns
ON ForecastRuns.ParentRun = ModelRuns.Run
WHERE ModelRuns.LongName='DB:\MyFolder\MyModel' AND
ForecastRuns.RunType="Forecast";
```

Once the Forecast RunID ("Run") is given in a query (called perhaps "ForecastIdsFromModelLongName"), it is straightforward to construct another query that either reads the data directly or joins the first query with the required table e.g. ActivityImpacts or CumulativesandPressures (if you have the option

“Uncompressed” for the Model setting “Format of Profiles written to database”) or ProfileArrays (this holds compressed profiles which require some work to decompress - see next section).

e.g. A simple query reading profile data directly from CumulativesandPressures using the RunID

```
SELECT Run, Name, DatePoint, OilPerCalendarDay, ProducedGasPerCalendarDay,
WaterPerCalendarDay, LiftGasPerCalendarDay, UptimeFraction FROM
CumulativesandPressures WHERE Run = '0458869f-94a7-4602-9cb7-0dbfe237a9a2'
```

e.g. A more complicated query reading Impact data from ActivityImpacts, looking up by ModelID

```
SELECT ActivityImpacts.Run, ForecastIdsFromModelLongName.ForecastLongName,
ActivityImpacts.Name, ActivityImpacts.ActivityType,
ActivityImpacts.NodeOrWellActedOn,
ActivityImpacts.NodeWhereImpactMeasured, ActivityImpacts.StartDate,
ActivityImpacts.EndDate, ActivityImpacts.IncludedInForecast,
ActivityImpacts.OilPerCalendarDay, ActivityImpacts.CondensatePerCalendarDay,
ActivityImpacts.TotalGasPerCalendarDay
FROM ActivityImpacts INNER JOIN ForecastIdsFromModelLongName ON
ActivityImpacts.Run = ForecastIdsFromModelLongName.ForcastRunID
WHERE ForecastIdsFromModelLongName.ModelLongName =
'DB:\MyFolder\MyModel' ;
```


READING COMPRESSED DATA FROM THE FUTURES DATABASE

FUTURES Forecasts and Spectres can be stored in “Compressed” format. When this format is used, the profiles are stored in the FUTURES database in the “ProfileArrays” table instead of the “CumulativeandPressures” table. This section describes how to read and decompress the data.

With uncompressed profile data stored in the “CumulativeandPressures” table, there is one record for each date point for each entity (well, reservoir etc) for each Forecast. With compressed profile data stored in the ProfileArrays table, there is one record for entity for each Forecast – i.e. each record holds an entire profile. Most of the fields in the ProfileArrays table – such as “DatePoint”, “Oil” (cumulative oil”, “OilPerCalendarDay” – are of Oracle type “Binary Large Object” or, equivalently, of SqlServer type “Image”. These fields are used to store bit-arrays or compressed bit-arrays of the array of dates or double-precision numbers of the original data.

FUTURES uses Microsoft .NET routines for both the array conversions (between the original date or double-precision arrays and the bit-arrays) and compression and decompression. The relevant FUTURES code, in VB.NET, for reading compressed data is given below. The code used the following .NET classes, which are documented in detail by Microsoft and whose .NET Core versions are available in open source.

System.BitConverter - <https://docs.microsoft.com/en-us/dotnet/api/system.bitconverter?view=netframework-4.6.1>

System.IO.Compression.DeflateStream - <https://docs.microsoft.com/en-us/dotnet/api/system.io.compression.deflatestream?view=netframework-4.6.1>

FUTURES code for reading compressed data

```
Public Function ConvertByteArrayToDoubleArray(ByVal ByteArray() As
Byte, Optional IsCompressedByteArray As Boolean = False) As Double()
    Dim uncompressedByteArray As Byte()

    If ByteArray Is Nothing Then Return Nothing

    If IsCompressedByteArray Then
        uncompressedByteArray = Decompress(ByteArray)
    Else
        uncompressedByteArray = ByteArray
    End If

    Dim nBytes As Integer = uncompressedByteArray.GetLength(0)
    Dim nDoubles As Integer = nBytes \ 8
    'Assert(nDoubles * 8 = nBytes, "nBytes must be divisible by 8")
    Dim retVal(nDoubles - 1) As Double
```

```

        Dim iByte As Integer = 0

        For iDb1 As Integer = 0 To nDoubles - 1
            retVal(iDb1) = BitConverter.ToDouble(uncompressedByteArray,
iByte)
            iByte += 8
        Next
        Return retVal

    End Function

    Public Function ConvertByteArrayToDateArray(ByVal ByteArray() As
Byte) As Date()
        If ByteArray Is Nothing Then Return Nothing
        Dim nBytes As Integer = ByteArray.GetLength(0)
        Dim nDates As Integer = nBytes \ 8
        'Assert(nDates * 8 = nBytes, "nBytes must be divisible by 8")
        Dim retVal(nDates - 1) As Date
        Dim iByte As Integer = 0

        For iDb1 As Integer = 0 To nDates - 1
            retVal(iDb1) =
DateTime.FromBinary(BitConverter.ToInt64(ByteArray, iByte))
            iByte += 8
        Next
        Return retVal

    End Function

    Public Function Decompress(ByteArray As Byte()) As Byte()
        Dim decompressedArray As Byte()
        Using decompressedStream As New System.IO.MemoryStream
            Using compressedStream As New
System.IO.MemoryStream(ByteArray)

                Using defStream As New
System.IO.Compression.DeflateStream(compressedStream,
IO.Compression.CompressionMode.Decompress)

                    defStream.CopyTo(decompressedStream)

                    decompressedArray = decompressedStream.ToArray

                End Using
            End Using
        End Using
        Return decompressedArray
    End Function

```

3. CALCULATIONS

After the user has described Reservoir, Network, Node and Well properties (including the crucially important well potential rate vs cumulative relationships), FUTURES needs to calculate the output profiles. Since this can take several minutes, these calculations are done when user selects the command **Main Menu > File > Save with calculations**. FUTURES then time-steps through each Forecast. At each time-step, FUTURES calculates flow-rates by

1. Formulating the well flow-rates and some additional parameters as unknown variables.
2. Defining an objective function using the oil, gas, water, injection water etc values specified (per reservoir) by the user. This objective function will be a linear function of the variables defined in Step (1).
3. Expressing rate and pressure constraints as linear equations in these variables.
4. Calculating with the Simplex algorithm the well flow-rates that maximise the objective function while honouring the rate and pressure constraints.

If all of the Model's constraints are rate constraints (which are linear functions of the well flow-rate variables), then we are done. If some of the constraints are pressure constraints (which can be non-linear), then -

5. The global solution from Step 4 is followed by a local exact solution, for each node or well with pressure constraints, taking boundary pressures from the global solution.
6. The pressure constraints are linearised again at the solution points from the local exact solutions in Step 5.
7. Iterate from Step 4 onwards until convergence.

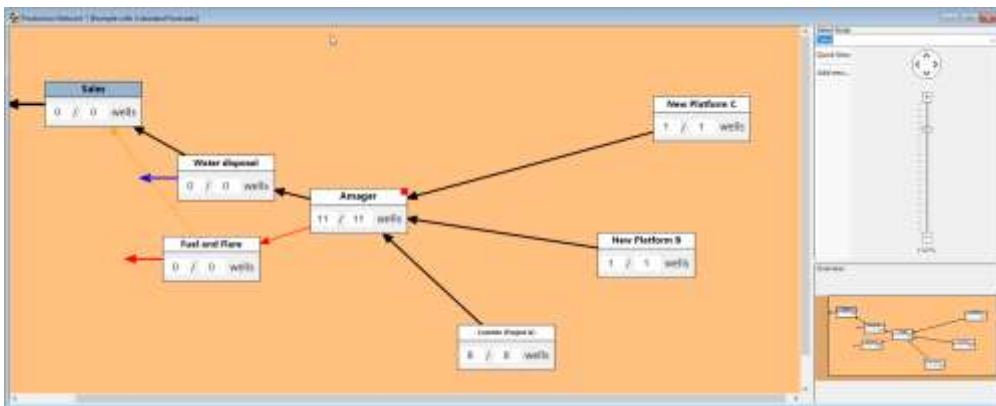
The following sections describe this process in more detail.

STANDARD, ROLL-UP, VARIANT AND DIFFERENCE FORECASTS

There are four types of Forecast generated by FUTURES

1. Standard – the most common type, representing a possible set of physical flows in time.
2. Roll-up – this sits on top of a group of Standard Forecasts; it calculates rates in the same manner as a Standard Forecast and then, for each designated “Sales Point” Node, calculates the incremental sales volumes and shares these out to the relevant Projects and Project Groups.
3. Variant – used to calculate properties (rates) that are variants of the rates of a Standard or Roll-up Forecast, such as “Well Potential” rates without the effects of facility constraints.
4. Difference – the result of subtracting one of the above Forecasts from another.

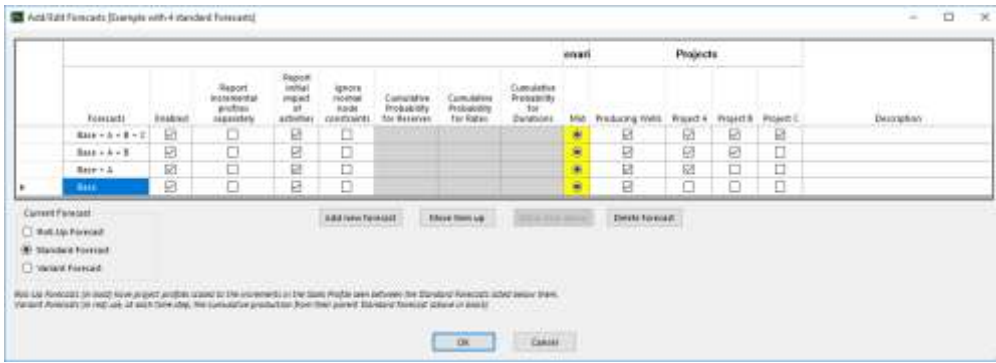
The various types of Forecasts are demonstrated below in a simple imaginary field example, with an existing platform “Amager” and three projects, A, B and C, which will tie-in new production in 2014, 2018 and 2022.



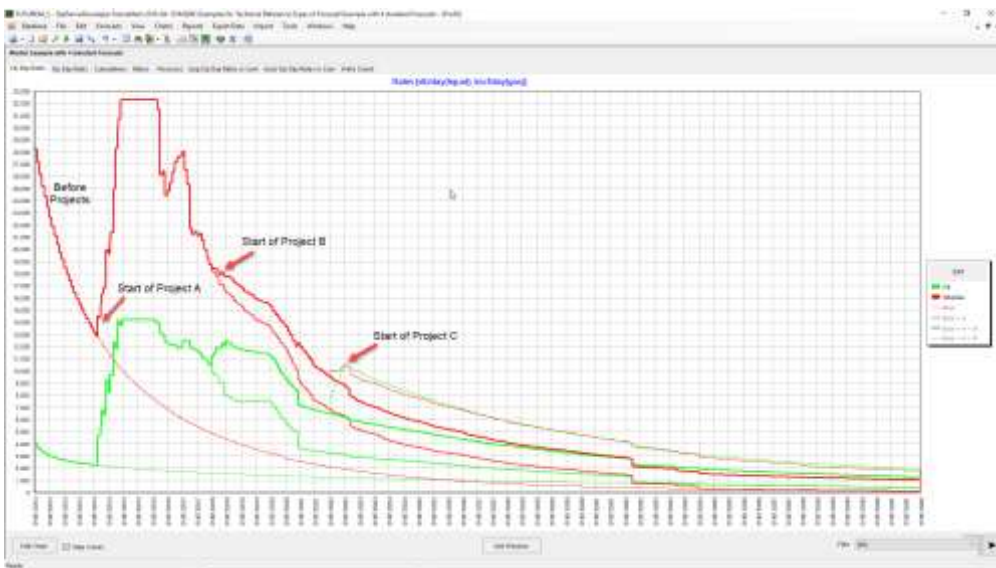
In our example, the Amager platform is facilities-constrained, so production from new wells causes some loss of production from existing wells. The most straightforward to start looking at this is to set up four separate Standard Forecasts –

1. A base forecast, with only the existing wells
2. A forecast “Base + A” with the existing wells and Project A
3. “Base + A + B” – existing wells, Project A and Project B
4. “Base + A + B + C” – existing wells and all three Projects.

The set-up of the forecasts in FUTURES would be as follows (note that here, without Roll-up or Variant forecasts, the vertical ordering of the forecasts has no effect; with Roll-up or Variant forecasts, the vertical ordering is significant).



The resultant Amager platform production profiles for the four forecasts are illustrated below.



In Standard forecasts, FUTURES reports Project profiles that are simply the sum of the profiles of the wells belonging to the Project in question. Unfortunately, such simple Project profiles are often not suitable for evaluating the economics of a project. When evaluating the economics of a project, it is usually necessary to use a profile representing the net effect on sales volumes (oil and gas) of doing the project – i.e.

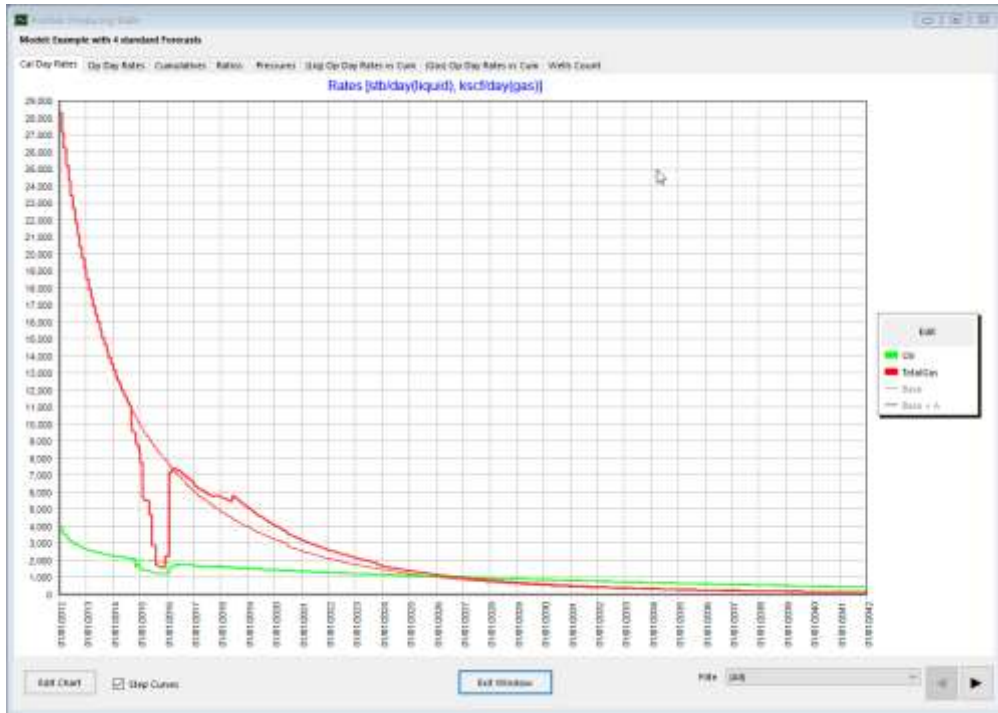
(Sales oil and gas in a Forecast with the Project) – (Sales oil and gas in an otherwise identical Forecast without the Project)

Such an “incremental sales” project profile may differ from the simple sum-of-the-wells project profile for the following two reasons:-

- Production from this project may cause a reduction in production from existing wells (because of facility constraints or interference in the reservoir).

- Some of the new production may be lost because of fuel and flare.

In our example, Project A causes a considerable reduction in production from existing wells, as can be seen in the profiles for the existing wells in the “Base” forecast (without Project A) and in the “Base + A” forecast.



One possible way of obtaining the necessary incremental sales profiles would be to export the Standard Forecast profiles for the “Sales” node and calculate the difference externally e.g. in a spreadsheet. Another alternative is to instruct FUTURES to calculate the differences, using “Difference Forecasts”. In our example, three Difference Forecasts have been set up, as follows. For each Difference Forecast, FUTURES calculates the difference profiles for every entity within FUTURES – i.e. for all Wells, Reservoirs, Nodes, Fields, Reporting Groups, Projects, Project Groups and Categories.

Add/Edit Difference Forecasts [Example with 4 standard Forecasts]

Note on terminology: Difference Forecast = (Minuend Forecast) - (Subtrahend Forecast)

Difference Forecasts	Enabled	Minuend Forecast	Subtrahend Forecast	Description
Increment from A	<input checked="" type="checkbox"/>	Base + A	Base	
Increment from B	<input checked="" type="checkbox"/>	Base + A + B	Base + A	
Increment from C	<input checked="" type="checkbox"/>	Base + A + B + C	Base + A + B	

Buttons: Add new forecast, Move item up, Move item down, Delete forecast, OK, Cancel

The resultant sum-of-wells profile – from the Standard Forecast “Base + A” and the incremental sales profiles for Project A (without Projects B and C) – from the Difference Forecast “Incremental from A” - are illustrated below.

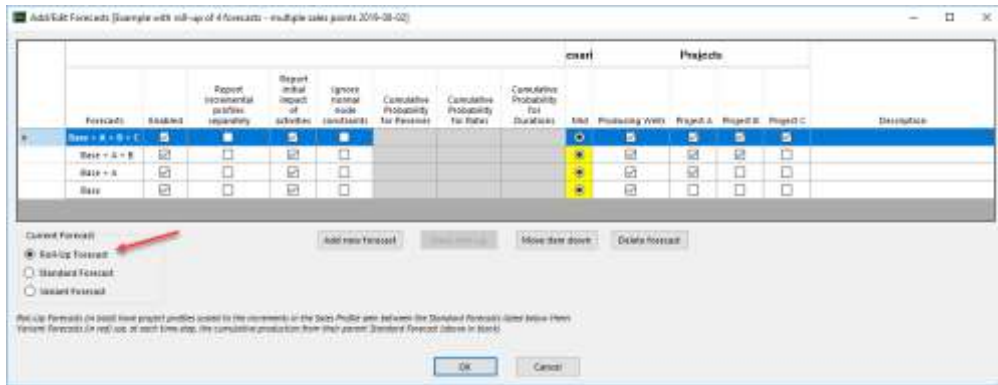


It would be more convenient if

1. the incremental sales profiles for a given Project could be reported under the name of the Project
2. when we have many (twenty or more) projects, we could generate approximate incremental sales profiles without needing 20+ Forecasts and 20+ Difference Forecasts

Roll-up Forecasts provide the mechanism to do this. A Roll-up Forecast calculates incremental sales profiles for all the Projects (and Project Groups) in the group of Standard Forecasts that are linked to it. The Roll-up Forecast is the Forecast with all the relevant Projects being executed. The group of Standard Forecasts are all the Standard Forecasts directly below the Roll-up Forecast (and above any subsequent Roll-up Forecasts) in the FUTURES listing of Forecasts.

In our example, “Base + A + B + C” is the Forecast that executes all the relevant Projects. So, in a new Model called “Example with roll-up of 4 forecasts”, it is designated as the Roll-up Forecast, while the other three Forecasts need to be listed below it (the precise order of these three Forecasts does not matter – FUTURES orders them in sequence of increasing number of Projects).



It is also necessary to specify what constitutes the “sales” profiles – in our example, we want to see the profile of the Node after the removal of fuel and flare etc, which we have called “Sales”, and we also want to see how the fuel and flare totals (in the Node “Fuel and Flare”) are affected. So, each of these two Nodes needs to be set to be a “Sales Point”, as illustrated below.

Edit Node Properties - Sales [Example with roll-up of 4 forecasts - multiple sales points 2019-08-02]

Scenario: Mid

Basic Data | Constraints | Split, Pressures & Abandonment | Conversion | Remarks

Basic Node Data

Name: Sales

Network: Production Network 1

Enabled *If disabled, the whole branch will be ignored in calculations*

Parent: <HEAD NODE>

Show Nodes from other networks of the same type

Step-Parent:

Supplies lift-gas from production *Step-Parent Node receives flow not carried through to Parent Node; can be left blank if there is no need to track split flow*

Uptime Factor and Uptime Calculation

Factor: 1 Uptime = Parent Uptime x Factor Uptime = Factor

Key Node *Constraints to be honored even in Forecasts with "Ignore normal node constraints" option chosen*

Sales Point (for impact reporting and Project roll-up) *In "Edit Forecasts", at least one Forecast must be "Roll-up" and/or have the "Report Initial Impact" option selected.*

Map Coordinates

X-Coordinate:

Y-Coordinate:

In the Main Menu, click on "Window/Move node icons to honour their X and Y coordinates" to display the relative positions of the Nodes in the Network Maps

Options in Network Diagram

Visibility of this Node

From Parent Visible Always Hidden Always Display lower Nodes

Line to Parent

Line color: Black

Line width: 3

Line to Step-Parent

Line color: Black

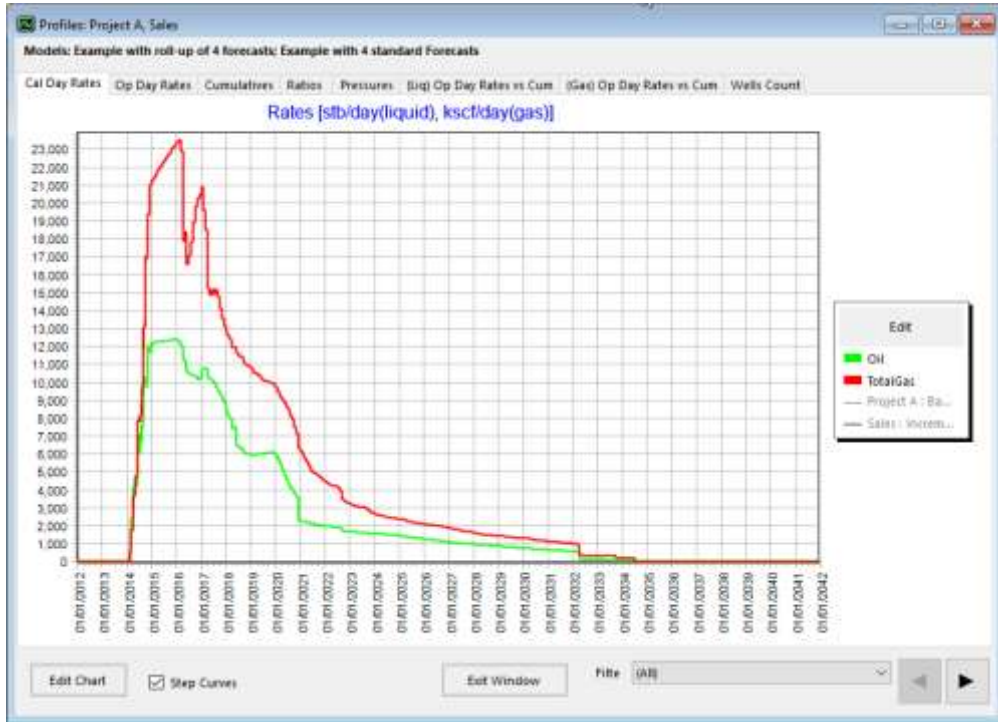
Line width: 1

Copy to all scenarios | Restore from back-up | OK | Delete | Cancel

The roll-up process takes the forecast with the lowest number of executed Projects (in our example, the forecast called, unsurprisingly, “Base”) to be the base. The next forecast, (in our example, “Base + A”) is taken to represent the base plus a first tranche of projects. So FUTURES calculates the sales increment between these two forecasts and the profiles for the projects in this tranche are then scaled so that the project profiles add up to the sales increment (NB – there are usually separate scaling factors for each fluid). Then the next forecast (in our example, “Base + A + B”) is taken to represent the consequences of the second tranche of projects, so the profiles for

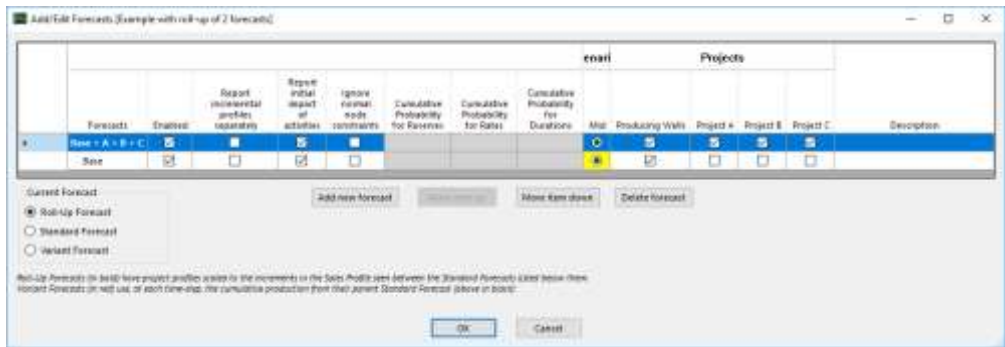
these projects are now scaled to the increment between “Base + A” and “Base + A + B”.

In our example, each tranche of project has only one project. Hence, the scaled project profile is exactly equal to the sales increment, as illustrated below (where the “Project A” profile from the “Example with roll-up of 4 forecasts” is plotted together with the “Sales” Node profile from the “Increment from A” difference forecast from the original “Example with 4 standard forecasts” Model.

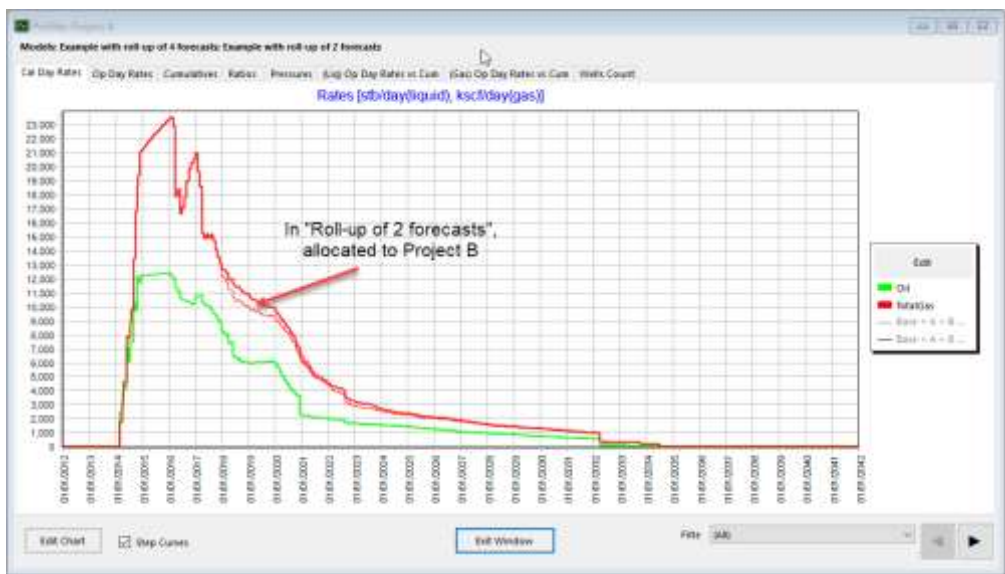


In our example, with only three projects, it is feasible to have one forecast per project. If there were 100 projects (as may be the case in corporate planning), this would be inconvenient. Fortunately, the roll-up functionality provides a simple approximate solution to this problem of evaluating 100 project profiles. Instead of having 100 forecasts – i.e. 100 tranches of projects in the roll-up, with tranche having a single project – it is possible to have, for example, five forecasts/tranches, with 20 projects in each tranche.

If there is more than one project in each tranche of projects, then the sales increment (resulting from doing all the projects in the tranche) is shared out amongst all these projects. For a given project, the resultant profile may be slightly different to the incremental profile that would have been the case if the increment of that project alone had been calculated. To continue our example, we created a Model “Example with only a single tranche of projects, i.e. with two forecasts – “Base” and “Base + A + B + C” – defined as follows.



When there are several projects in a given tranche, then the benefits of the use of facility capacity are shared out amongst all the projects. The effects of this can be seen in the profiles of Project A. The higher curve is the increment in sales that occurs when Project A is executed without Project B and C. The lower curve is the share-out (in proportion to sum-of-wells production rates) of the increment in sales when Project A is executed in conjunction with Project B and C. The two differ from 1 Jan 2018 onwards (when Project B comes on stream).



Each Standard or Roll-up Forecast can have one or more subsidiary Variant Forecasts. A Variant Forecast uses, to calculate well potentials at each time step, the well cumulatives from its parent Standard Forecast. A typical use is the calculation of various forms of network potential production rates – e.g. the production rates that would be achieved if all uptimes were 100%. If such network potential rates were calculated in a stand-alone Standard Forecast, then the rates in a year’s time would incorporate two effects – firstly, the effect of 100% uptime at this point in time (increasing rates); secondly, the effects of higher cumulative production over the

year past, causing the wells to be further along the well potential rate vs cumulative relationship (generally reducing rates). Variant Forecasts provide a method to generate profiles that incorporate only the first effect.

INPUT OF UPTIME

At the start of each calculation time-step, FUTURES calculates individual uptimes for each Node and Well and uses them to calculate the effective (calendar day) capacities and potential rates. The network is then solved in terms of these effective capacities and potential rates.

The calculation scheme for the individual Node/Well uptimes allows uptimes to be specified for a whole branch or the entire network. The calculation first evaluates an "**Uptime Factor**" for the Node/Well and then its resultant "**Uptime**", as follows.

- The calculations are done in order of ancestry, starting with Head Nodes; so when a Node/Well's uptime comes to be calculated, its Parent and Step-Parent Node uptimes are already known.
- Node/Well Uptime Factor = (As specified in initial Node/Well properties or in most recent Planned Change) - (Loss of uptime in applicable Curbs)
- Node/Well **Uptime** = EITHER (Node/Well **Uptime Factor**) x (**Uptime** of Parent Node) OR (Node/Well **Uptime Factor**)

In production forecasting, it is important to include the effects of all the operational factors that restrict production. Some of these – such as planned major shut-downs for annual maintenance – are best described as explicit, individual events in the FUTURES Model, using Curbs or Planned Changes (setting either rates or uptimes to zero). Other operational factors – particularly unplanned shut-downs and equipment failures – are best described with appropriate uptime values.

FUTURES allows you to assign uptime values to each node and well in the system. These uptime values are used to express constraints in terms of calendar days. Then the solution for network flow is calculated in terms of calendar day rates – so that the fundamental node mass balance (flow in = flow out) is automatically honoured for the production in the time period.

Rather than requiring input of uptime values for each node and well, FUTURES implements a simple scheme to allow you to specify uptime values for whole groups of nodes and wells. In terms of user input, each node and well has its own "Uptime Factor", but these can be mostly left at a value of 1, because for each node and well, the uptime value (as used in the calculation) is usually calculated as the product of the individual node or well's uptime factor with the uptime factors along its path of ancestors. So, if you assign an uptime factor of 0.90 to the Head Node and leave all other uptime factors as 1, then all nodes and wells have resultant uptime values of 0.9.

Where a Node has both a Parent and a Step-Parent, then the “Uptime of Parent Node” is evaluated as follows

- If the Node split is “Fixed Fractions” (so flow generally requires both Parent and Step-Parent to be operating), then “Uptime of Parent Node” is the **lower** of the Parent uptime and the Step-Parent uptime.
- Otherwise (i.e. Node split is “Variable Fractions” or “Single Variable Fraction”, and flow is generally still possible when only one of the Parent or Step-Parent is operating), “Uptime of Parent Node” is the **higher** of the Parent uptime and the Step-Parent uptime.

There is also an option, in the Node, to override this inheritance of uptime (where the Node uptime = (Uptime of Parent Node) x (Uptime Factor for this Node)

Planned Changes can modify Uptime Factors. Curbs can reduce Uptime Factors by a specified "Loss of Uptime" amount (which can be negative for the unusual situation of there being a temporary increase in uptime).

It should be noted that rate constraints can be specified as applying either to operating day rates or to calendar day rates. Pressure constraints (lift curve calculations) are interpreted as applying always to physical (i.e. “operating day”) flow rates and are converted into equivalent constraints in terms of calendar day rates.

REPORTING OF UPTIME

As discussed in the previous section, major planned shut-downs are often described explicitly in the FUTURES Model, while the effects of unplanned or minor shut-downs are approximated with uptime factors. When it comes to reporting, FUTURES reports the overall uptime, including the effects of downtime from Curbs and Planned Changes that set rates to zero.

In a given calculation step, for Nodes and Wells with non-zero rates, FUTURES calculates the uptime value described in the previous section (i.e. either equal to the uptime factor specified for the Node/Well itself or, more usually, calculated from the uptime factors of the Node/Well itself and its ancestor node uptime factors). For Nodes and Wells with all fluid rates equal to zero, FUTURES stores an uptime value of zero.

In a given reporting step (covering one or several calculation steps), FUTURES reports the uptime averaged over the calculation steps with weighting in proportion to the durations of the individual calculation steps. An mathematically equivalent and simpler description of the calculation is that the operating days in the reporting step is equal to the sum of the operating days in the calculation steps.

All other Entities – Reservoirs, Fields, Reporting Groups, Categories, Projects – report the weighted average of the uptimes of the Wells that make up the Entity. The weighting factor is the “Reference Fluid rate per operating day”, a simplified measure of the significance of production, defined somewhat arbitrarily as

Reference Fluid Rate = Oil rate (in stb/day) + 0.1 x Produced gas rate (in kscf/day) + 0.05 x Injection gas rate (in kscf/day) + 0.05 x Injection water rate

Again, there is a mathematically equivalent and simpler description, derived as follows

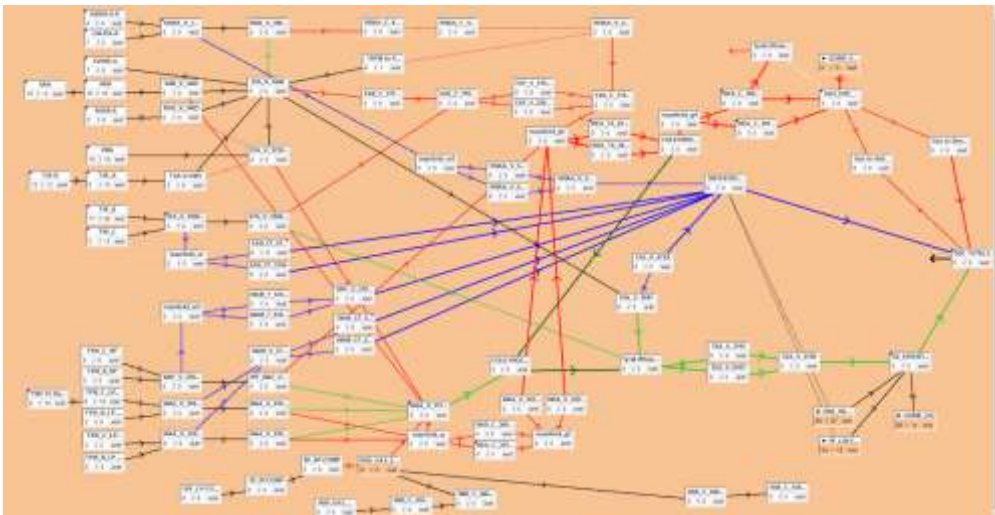
$$\begin{aligned}
 \textit{Entity uptime} &= \frac{\sum(\textit{Well uptime} \times \textit{Well reference fluid per operating day})}{\sum(\textit{Well reference fluid per operating day})} \\
 &= \frac{\sum(\textit{Well reference fluid per calendar day})}{\sum(\textit{Well reference fluid per operating day})} \\
 &= \frac{\textit{Entity reference fluid per calendar day}}{\sum(\textit{Well reference fluid per operating day})}
 \end{aligned}$$

It should be noted that roll-up and scaling of Project profiles does not affect their reported uptimes.

SOLVING THE NETWORK

At each time-step, FUTURES determines flow-rates, pressures etc by calculating the optimal network solution as follows

1. Well fluid flow-rates, lift-gas allocations and some pressures are represented by variables (“unknowns”, x_1, x_2 etc , whose values will be optimised).
2. Well oil, gas, water etc flow-rate, Node flow-rates and the remaining pressures are represented by functions of the variables ($f(x_1, x_2, \dots)$)
3. Constraints are represented in the form $f(x_1, x_2, \dots) \leq a$
4. The value of production is represented by a linear function of the variables
5. A successive linear programming algorithm is used to calculate the set of values for the variables that maximises the value of production while honouring all the constraints.



Finding the optimal operating conditions requires exploration of all the elements of the network. The oil production process from a series of well-heads is inherently a nonlinear complex process. The nonlinearity can be reduced to a sequence of linear segments and the optimal production configuration operates efficiently over these segments. The optimal production solution in FUTURES is provided using the cornerstone of *linear programming* technology. The *simplex method* is reliable, stable, efficient and robust in operating conditions for network flows. Our implementation provides an efficient, tried and tested solution that *guarantees* an optimal solution for the flow rates. The performance of this optimization method has been validated by our proprietary *interior-point* method, which is also available as an option (Cunningham, Moutari, & McCann, 2015). For details on the *interior point* package, customers are invited to contact Serafim Ltd.

FUTURES Simplex Method

The fundamental challenge is finding the optimal production throughput given the multiphase flows through the network. Typically the constraints arise from the fluid flows and pressure limitations of the well-heads and pipeline networks. For example, the input flows and output flows at a manifold must balance. The network optimization allows for all local network changes, simultaneously and automatically, and calculates their effect on the global performance of the production network. The engineering variables, and the constraints within the production system, are transformed (scaled and adjusted) to equivalent mathematical variables. The problem then is presented in mathematical terms, given n variables: x_1, x_2, \dots, x_n (which are connected or constrained) the aim is to find the values of these variables that maximise the *objective* function:

$$f = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \quad .$$

The set of constants: c_1, c_2, \dots, c_n are given (determined) by the production facility and operational constraints.

The variables, x_1, x_2, \dots, x_n are converted to standard form, namely that they are non-negative (*primary constraints*):

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \quad .$$

with an additional m linear constraints in the form:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad , \quad i = 1, \dots, k$$

$$a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n \geq b_j \quad , \quad j = k + 1, \dots, m$$

where m may be greater than, equal to, or less than n , and the coefficients a_{ik}, b_i, b_j are features of the particular production operation and derived from the linearization process. FUTURES converts these constraints to the mathematically-standard form where the b -values are non-negative.

The k inequalities are transformed to equalities by the addition of *slack* or *artificial variables*. The introduction of slack variables creates a total of $n + k$ variables. For

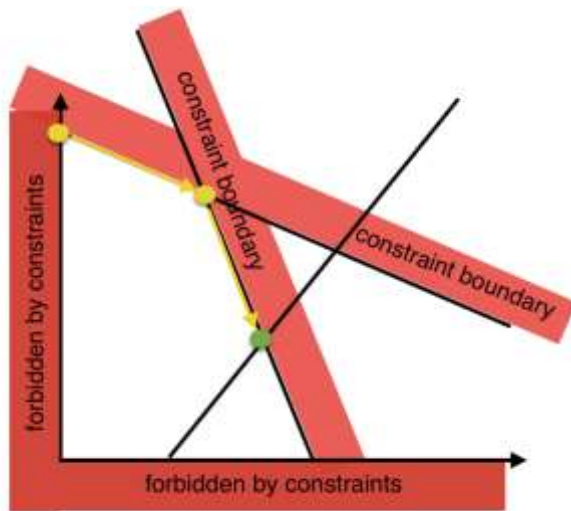
example, the first inequality:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1$$

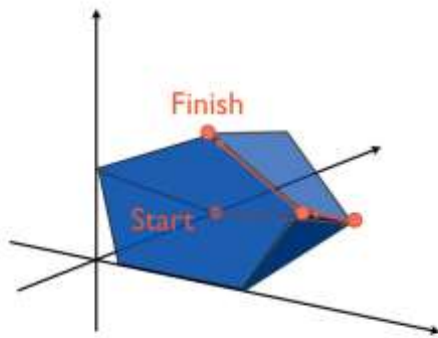
becomes, (Press, Teukolsky, Vetterling, & Flannery, 2007)

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - x_{n+1} = b_1$$

with the additional inequality constraint $x_{n+1} \geq 0$.



The *simplex method* uses the fact that the equations are linear and therefore operating 'planes' which intersect along lines or corners. These lines/planes/walls constrain the operating conditions defining the *simplex of feasible solutions* within which the facility is able to operate. The aim of the method is to find the best operating conditions, from all possible configurations, taking into account the overall operational limits. The *objective function* has an optimal value at one of the corners (or sometimes lines) of the feasible region.

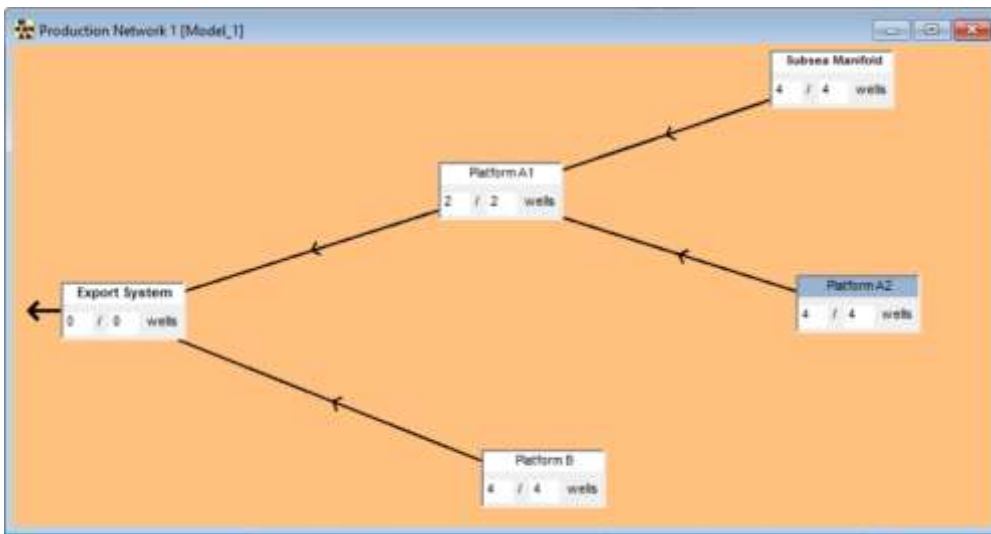


The simplex method systematically explores the *corners* of the simplex proceeding (step-by-step) from one corner to the next in such a way as to continuously improve the value of the objective function. The method progressively reduces the large set of variables in the equations by (Gauss-Jordan) elimination, with pivoting, creating a numerically stable and robust algorithm, until the *canonical/optimal* form is discovered.

NETWORK SPLITS

Networks in FUTURES are organised using a tree-and-branch structure, with a single “Head” (or “Root”) node and with every other node having a “Parent” node. In production networks, flow goes from child nodes to parent nodes, while in injection networks, flow goes from a parent node to its child nodes. For the occasions where the actual flow pattern in production networks is more complicated, FUTURES also allows a node to have a flow split, with part of the flow going to a “Step-Parent” node (or being dumped), while the remainder goes to the Parent node.

A simple tree structure, without network splits, is illustrated below. The node “Export System” is the Head node.



Let us consider the situation where Platform A2 sends part of its production to Platform B and the remainder to Platform A1. This can be described by specifying a flow split in Platform A2 and specifying that Platform A1 is its “Parent Node”, while Platform B is its “Step-Parent Node.” There are three options for describing the split:

1. Fixed fractions – constant proportions (possibly different for each component) of the oil, water and gas go to the Step-Parent Node, whilst the remainder goes to the Parent Node.
2. Variable fractions – FUTURES calculates what fractions of oil, water and gas should go to the Step-Parent and Parent Nodes. This calculation is part of the main network solving/optimising calculation.
3. Single Variable Fraction – As in (2), FUTURES calculates the split, but the fraction sent to the Step-Parent has to be the same for water and for gas as for oil.

PRESSURE CALCULATIONS

Under operating conditions, the well-bore and pipeline flow performance require the calculation of pressure and flow rates at different locations in the network within the reservoir, through the wellbore and between the well-head and the surface extraction point.

Pressure and flow rate are very closely connected both within the reservoir and production network in terms of the energy balance of the flow. Within the reservoir Darcy (and non-Darcy) flow provides the link between flow and pressure gradients. In the production pipelines the energy balance between pressure gradients, frictional forces, inertial forces and dissipation govern the flow. At the well-head interface between reservoir and pipeline, both the pressures and flow rates must match to create deliverability. That is the inflow performance relationship (IPR) which correlates the (flowing) bottomhole pressure (BHP) with the production flow rate, must connect with the vertical lift performance (VLP) data for the tubing head pressure (THP), while simultaneously maintaining the flow rates between reservoir output and production inflow. FUTURES determines the well-deliverability by solving the nonlinear equations in the pressure/flow variables between IPR and VLP.

RESERVOIR INFLOW PERFORMANCE RELATIONSHIP

The reservoir inflow performance relationship (IPR) for a given well defines the correlations between bottomhole pressure (BHP) and flow rate from the reservoir into the well-bore. FUTURES uses either linear IPRs or non-linear gas pseudo-pressure relationships.

In the gas pseudo-pressure calculation, the flow rate in the reservoir at the well bore is simulated by a non-Darcy flow (Geertsma, 1974). A radial-flow model is adopted at the well-bore. This flow equation is highly nonlinear in pressure, however it can be simplified using the gas pseudo-pressure (Kirchoff integral transformation). The pseudo-pressure transformation greatly simplifies the solution of the (radial) gas flow equation: the connection between flow-rates and pressure. The pseudo-pressure, $m(p)$, is defined as (Al-Hussainy, Ramey, & Crawford, 1966):

$$m(p) - m(p_{wf}) = 2 \int_{p_{wf}}^p \frac{p'}{\mu(p')Z(p')} dp'$$

where μ is the (pressure-dependent) gas viscosity and the Z-factor (compressibility factor) has its usual meaning (Dake, 1978). Pseudo-pressure is measured in psia²/cp. The reference pressure, p_{wf} , is the bottomhole flowing pressure (BHP). There is a one-to-one correspondence between p and $m(p)$:

$$m(p) \leftrightarrow p$$

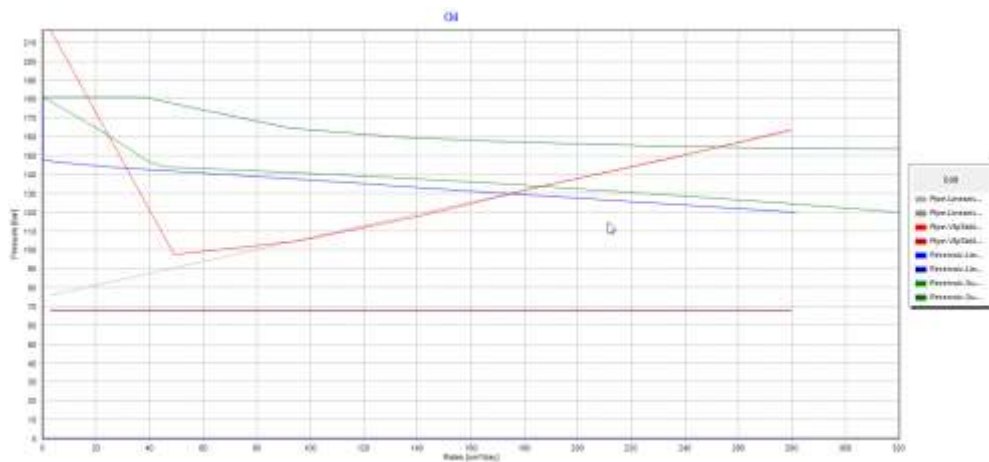
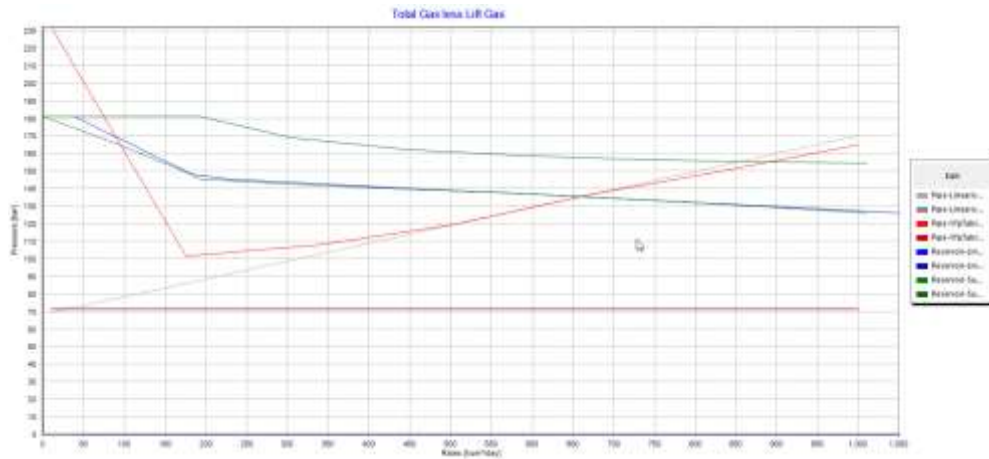
FUTURES references tables of the viscosity (Carr, Kobayashi, & Burrows, 1954) and Z-factor (Dranchuk & Abou-Kassem, 1973) over a range of pressures, and temperatures and gas composition, means that the integral can be calculated to evaluate $m(p)$.

In order to create a conversion from pressure to pseudo-pressure, $p^* \rightarrow m(p^*)$, a table is first generated. This is done by numerical quadrature (the trapezium rule is used) over a pressure grid and interpolation methods are employed to look up viscosity and compressibility. Given a set of N pressure values $\{p_1, p_2, p_3, \dots, p_N\}$, with corresponding viscosity and Z-values $\{\mu_i, Z_i\}$ for a given temperature and gas composition, FUTURES creates a corresponding table of pseudo-pressures:

$$m(p_i) = \sum_{j=1}^{i-1} \left(\frac{p_{j+1}}{\mu_{j+1}Z_{j+1}} + \frac{p_j}{\mu_j Z_j} \right) (p_{j+1} - p_j)$$

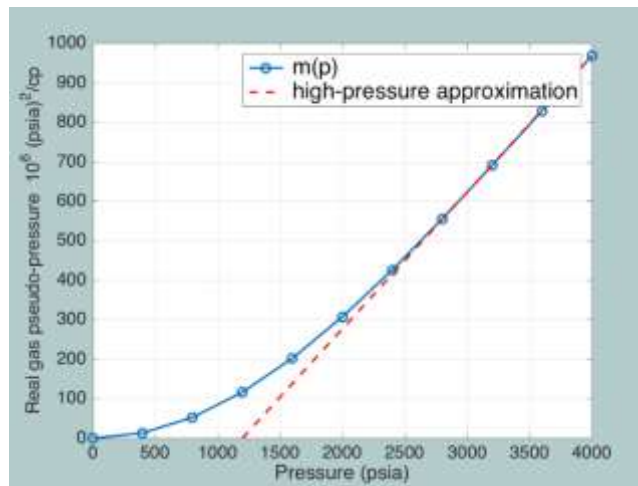
If you specify for a Well or a Node representing a multi-completion well, “Report Flow-Pressure Relationship”, then FUTURES will generate, at each time-step,

- the reservoir inflow performance and tubing performance curves for conditions (reservoir pressure, tubing-head pressure and fractional flow) at that time
- the corresponding curves for the linear approximations used by the global solver.



Multi-completion well with one zone shut-in by the solver

FIGURE 1: PSEUDO-PRESSURE VS PRESSURE (FOR GAS GRAVITY 0.85, TEMP. 200 °F)



The relationship is illustrated in Figure 1 for a typical gas composition and temperature. Note that at high gas pressures, there is an approximately linear relation between the true pressure and pseudo-pressure as shown by the red-dotted line.

Given the bottom-hole well flowing pressure is denoted by, p_{wf} , then the excess pressure leads to a (gas) flow rate, Q , given by:

$$m(p) - m(p_{wf}) = AQ + FQ^2$$

where the first term on the right is the usual Darcy flow, and the second term represents a (non-Darcy) correction. This correction is analogous to a skin-factor, but in fact owes its origins to an inertial resistance arising from the unsteady flow of gas through the pores. In principle the Darcy-flow coefficient A can be estimated from a reservoir simulation for the oil flow (along with the gas/oil ratios), however, the non-

Darcy term cannot. In practice, both coefficients A and F can be extracted from well testing data for the oil flow rate. FUTURES allows these values to be inserted manually by the user. The inverse process, calculating p given the pseudo-pressure, $m(p): m(p) \rightarrow p$ is implemented by interpolation from the $m(p)$ tables. The correlations, $Q \leftrightarrow m(p) \leftrightarrow p$, create the flow-pressure relationship (FPR) at each well-head. This provides the corresponding IPR curve: the relation between the oil flow rate and the bottom-hole pressure. An example of this is shown in Figure 2.

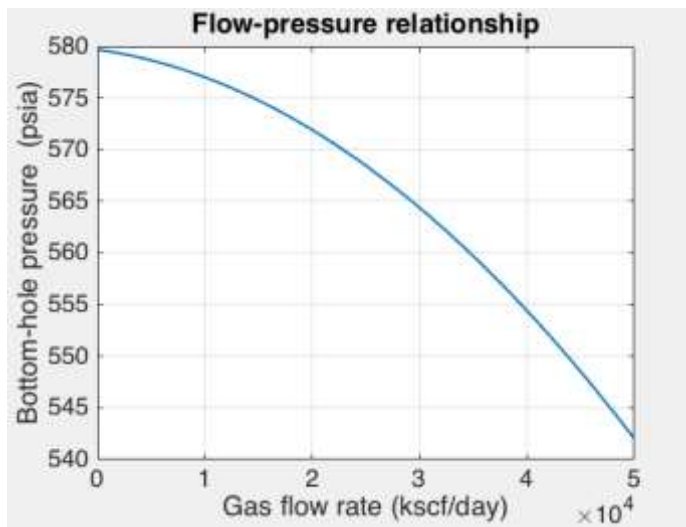
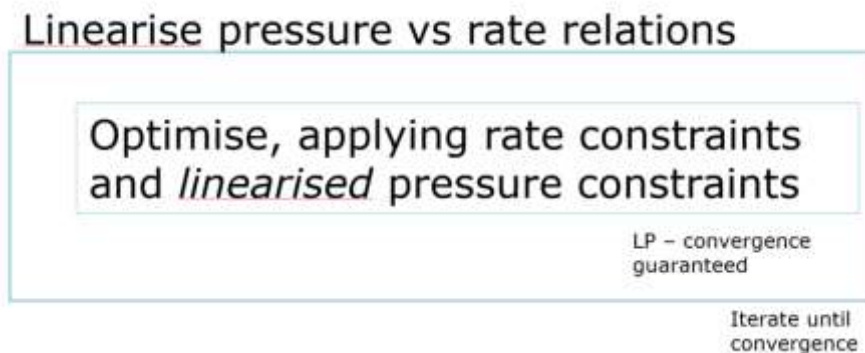


FIGURE 2: INFLOW PERFORMANCE RELATIONSHIP (IPR) CORRELATING BOTTOMHOLE PRESSURE (BHP) AND GAS FLOW RATE AT THE WELL-BORE

“GLOBALLY APPROXIMATE, LOCALLY EXACT” ALGORITHM FOR INCORPORATING NON-LINEAR (PRESSURE) CONSTRAINTS

The GALE algorithm was developed from an earlier algorithm that calculated the optimal network flow rates given a set of rate constraints, by applying the Simplex method of optimising a linear objective function subject to linear constraint functions. Linear approximations of pressure constraints were then included in the original inner algorithm, as illustrated below.



The basic idea is that, since all the rate constraints are linear and many pressure constraints are close to being linear, a purely linear description of the network will have the same solution as a full non-linear description providing we chose suitable linear approximations of those pressure constraints that are significantly non-linear. If, for the sake of argument, we already knew the solution of the full non-linear problem, then fitting linear approximations to the non-linear constraint expressions at the solution rates and pressures would indeed give a linear problem with the same solution as the non-linear problem.

We can then make the observation that, if the constraint expressions are continuous with continuous first derivatives, there is a region surrounding the solution point in which the linear approximations will be close to the non-linear expression. Within this region, a solution of the linear problem will be close to the solution of the non-linear expression.

In other words, an iterative scheme as described below

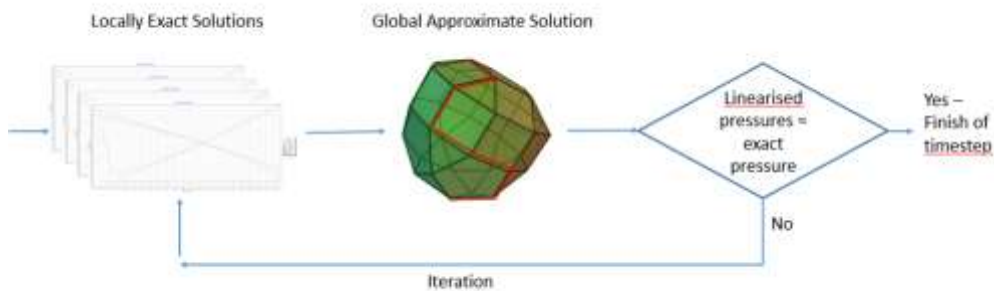
1. Linearise the non-linear expressions to create a linear problem that approximates the non-linear problem.
2. Solve the linear problem.
3. Linearise the non-linear expressions at the solution point found in (2)
4. Iterate (returning to (2)) until convergence.

would give the correct solution of the non-linear problem, providing the starting point (for the first linearization) was close enough to the solution point.

In practice, with real-life choices of starting points, such a simple iteration often fails to converge for typical oil and gas production networks, e.g. because of oscillation. However, two modifications to the scheme have been found to converge in all the test cases used: -

- For pipelines, at step (3), instead of linearising at the new solution point, we can linearise at a point approximately half-way between the previous point of linearization and the new solution point.
- We can apply our knowledge of how, approximately, wells behave. The global solution of the linear problem in (2) probably gives us a reasonable approximation of each individual well's THP. Taking this THP as fixed (for the moment), we can calculate exactly the intersection of reservoir inflow and tubing performance curves, as illustrated in **Fig. 1**. This intersection point is then used as the point at which to linearise the reservoir inflow and tubing performance relationships.

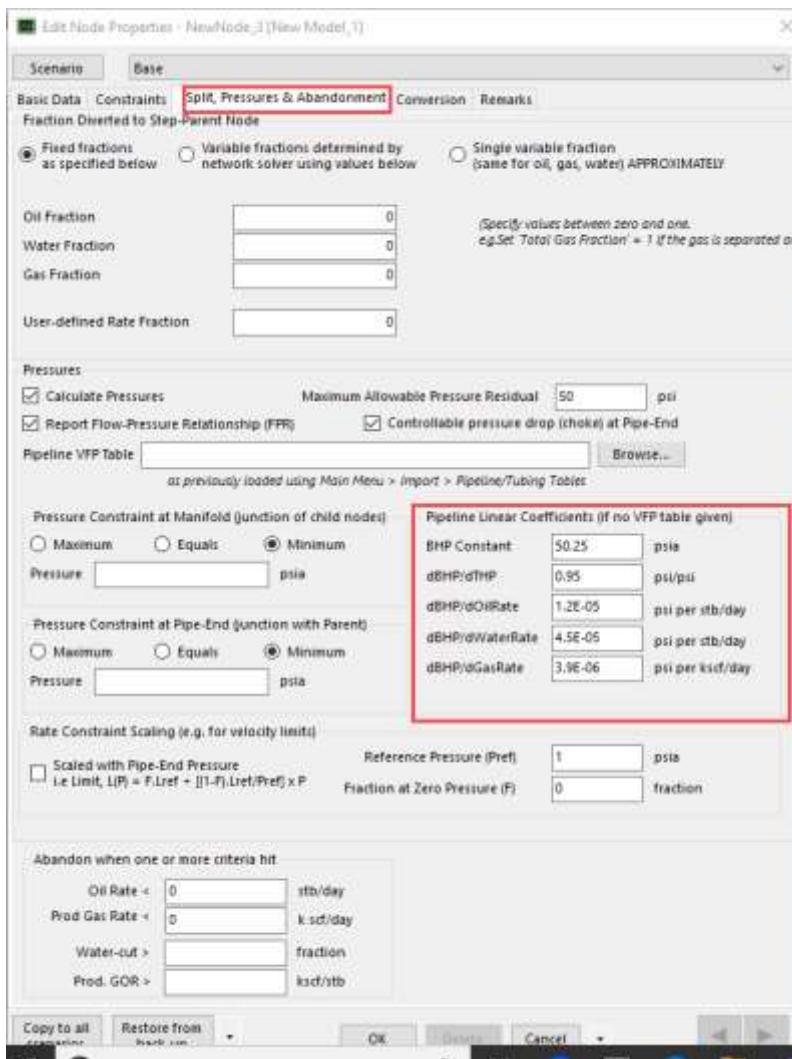
A schematic outline of the GALE algorithm is shown below.



PIPELINE LINEAR COEFFICIENTS (IF NO VFP TABLE)

There is an option in FUTURES to calculate pipeline pressure changes from user-defined linear coefficients, rather than from a VFP table. This can be useful if no VFP table is currently available or if it is decided that a simple description is adequate to model system behaviour.

The option is selected by setting the Node “Pipeline VFP Table” to be blank and then supplying appropriate values to the “Pipeline Linear Coefficients” (see screen shot below).



FUTURES uses the linear coefficients to express the manifold pressure (“BHP”) as a function of the oil rate, water rate, total gas rate and pipe-end pressure (“THP”), using the equation

$$BHP = a + b.TH P + c.(Oil\ rate) + d.(Water\ rate) + e.(Total\ gas\ rate)$$

where the constants are the user-supplied values

a - “BHP constant”

b – “dBHP/dTHP”

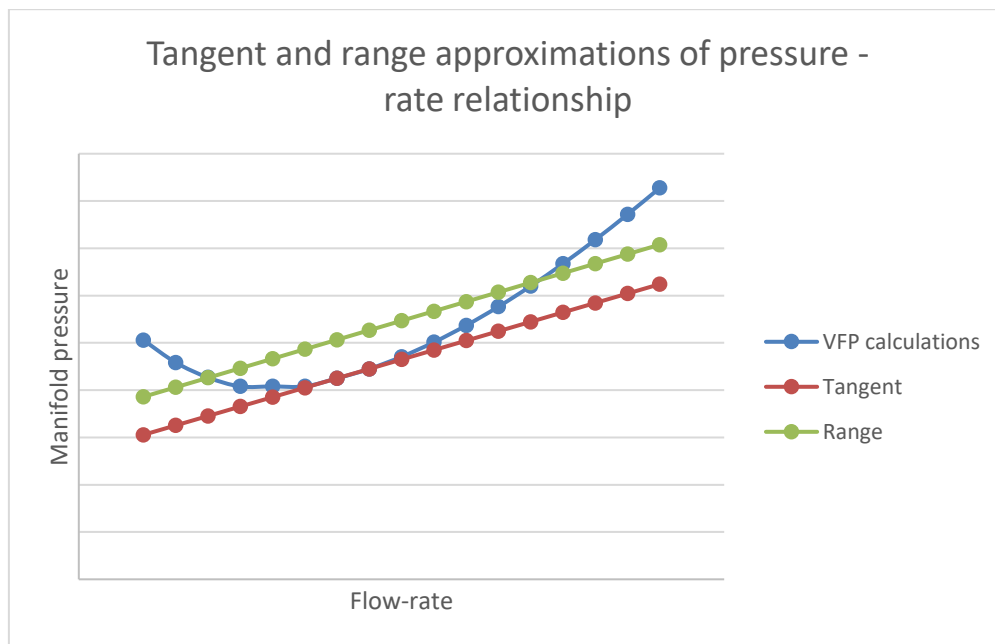
c – “dBHP/dOilRate”

d – “dBHP/dWaterRate”

e – “dBHP/dGasRate”

If you have a “VFP calculation” model of the pipeline flow and pressure changes (in PROSPER© or similar software), it is possible to set the linear coefficients to the partial derivatives at a point, as described in

https://en.wikipedia.org/wiki/Linear_approximation – i.e. to construct a multi-dimensional tangent approximation. However, this is not always the best option. It gives a very close approximation near the point of matching, but it may be preferable to try to give a reasonable linear approximation across the full range of rates and pipe-end pressures that are expected to be seen in forecasting. The difference between a tangent approximation and a range approximation is illustrated in one dimension below.



In this example, the “Range” coefficients are calculated (using Solver in Excel) to minimise the maximum distance between the “VFP calculations” pressure and the “Range” pressure.

If you do not have a suitable model of pipeline flow and pressure losses (e.g. with an old pipeline whose condition and degree of corrosion is uncertain), but do have measurements of actual flow-rates and manifold and pipe-end pressures, then it is possible to follow a similar procedure to the “range” matching of VFP calculation data, as follows

1. Set up a list (e.g. in Excel) of the flow-rate and pressure measurements
2. Make an initial estimate of the pipeline linear coefficients
3. For each measurement entry (i.e. set of oil-rate, water-rate, total-gas-rate, manifold pressure and pipe-end pressure), calculate a modelled manifold (“BHP”) pressure by applying the pipeline linear coefficients equation to the measured oil-rate, water-rate, total-gas-rate and pipe-end (“THP”) pressure.
4. For each entry, calculate an “Error term” – e.g. the absolute value of the difference between the measured and modelled manifold pressure.
5. Define an overall error term – usually either the average or maximum of the individual entry error terms.
6. Adjust the pipeline linear coefficients so as to minimise the overall error term (e.g. by using Solver in Excel).
7. Verify the reasonableness of the linear coefficients when it is necessary to extrapolate beyond the range of the measured data.

RATE AND RESERVES MULTIPLIERS

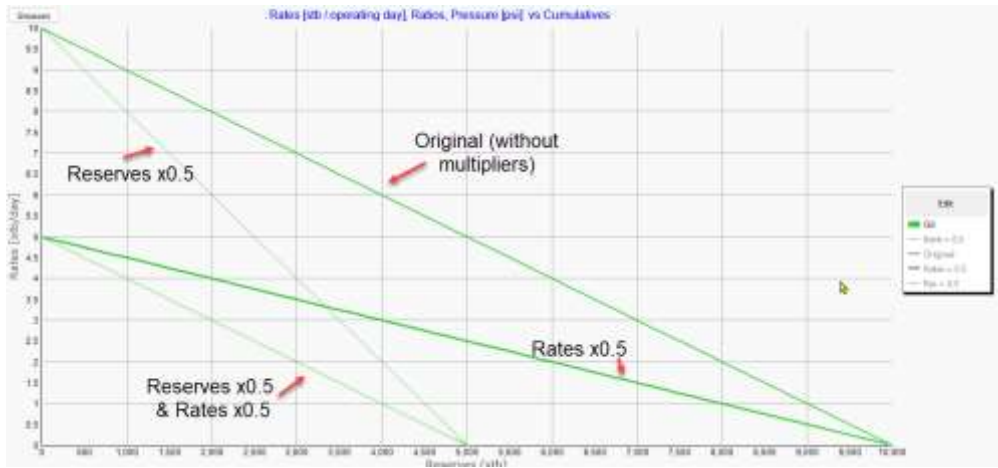
For quick adjustments and sensitivity calculations, FUTURES uses a system of rate and reserves multipliers. These stretch or compress the axes of the rates - cumulative relationships (and the water-cut, GOR, OGR, WGR – cumulative relationships).

The rates and reserves multipliers are specified in the Well Element properties (see below) and can be modified in Planned Changes.

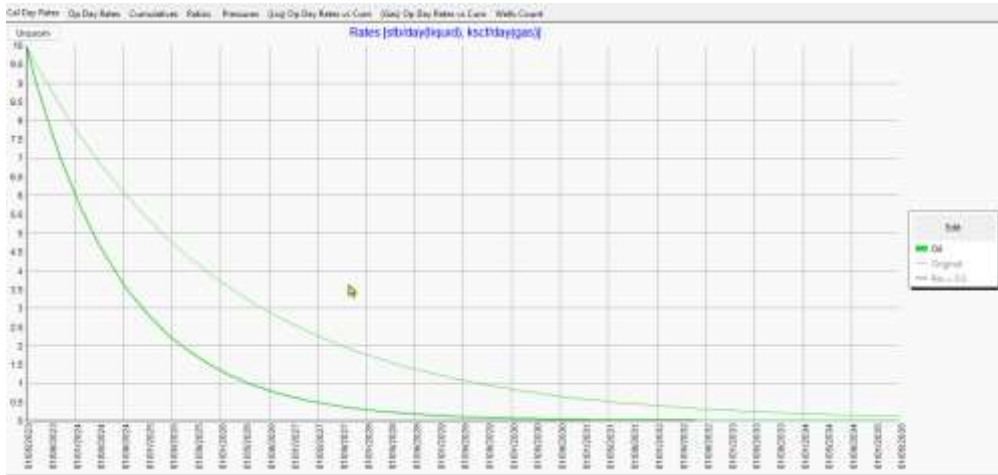
The screenshot shows the 'Edit Well Properties - Rates = 0.5 [Multipliers]' dialog box. The 'Scenario' is set to 'Base'. The 'Basic Settings' tab is active. The 'Name' is 'Rates = 0.5' and it is 'Enabled'. The 'Reservoir' is 'Reservoir 1' and the 'Parent node' is 'Head Node'. The 'Well uptime (fraction)' is '1'. The 'Drilling and start-up' section shows 'Date on stream' as '01/05/2023', 'Cost (MM)' as '0', 'Currency' as 'USD', and 'CashFlow' as 'CAPEX'. The 'Type of profile' is 'Standard'. The 'Take Acceleration From; Add Incremental Profile To:' section has 'Reservoir' selected. The 'Type of production model' is 'Decline (oil, gas, water)'. The 'Primary control on well' is 'Rates from Production Model'. The 'Multipliers to well potentials (not shown in Decline Charts)' section shows 'Rates multiplier' as '0.5', 'Reserves multiplier' as '1', and 'Value multiplier' as '1'. The 'Well Count Contribution' is '1' and the 'Project' is '(none)'. There are red arrows pointing to the 'Rates multiplier' and 'Reserves multiplier' input fields. At the bottom, there are buttons for 'Show Planned Changes', 'Show Curbs', 'Copy to all scenarios', 'Restore from back-up', 'OK', 'Delete', and 'Cancel'.

The chart below shows the effects on the oil-rate vs oil cumulative relationship of

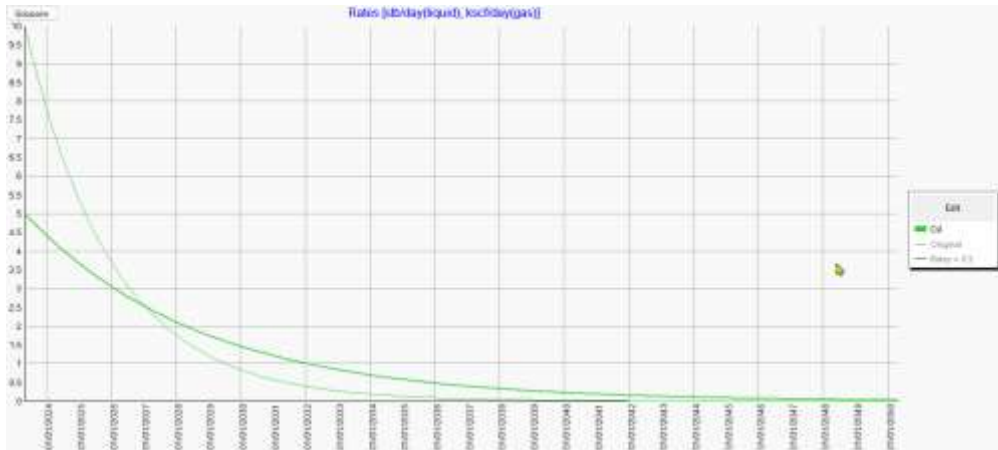
- A reserves multiplier of 0.5
- A rates multiplier of 0.5
- The combination of a reserves multiplier of 0.5 and a rates multiplier of 0.5



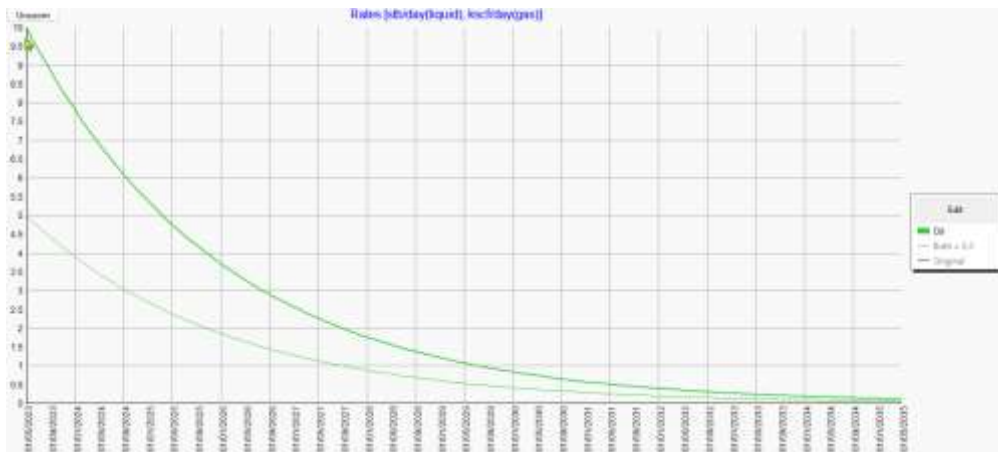
With a reserves multiplier of 0.5 and no rates multiplier (strictly speaking, a rates multiplier of 1), the initial rate of the well is unchanged, but, with lower reserves, the well declines faster. The resultant rate vs time chart is as follows.



With a rates multiplier of 0.5 and no reserves multiplier (strictly speaking, a reserves multiplier of 1), the initial rate of the well is half of the original, but, with the same technical reserves, the well declines slower. After a few years, the well with the rates multiplier of 0.5 goes on to produce at a higher rate than the original well, enabling it to produce the same reserves over the life of the well. The resultant rate vs time chart is as follows. Note that the areas under the two curves (which equate to the cumulative production) are equal.



With both a reserves multiplier of 0.5 and a rates multiplier of 0.5 (strictly speaking, a rates multiplier of 1), the well with the multiplier declines at the same rate and so tracks the original well, as illustrated below. So in the simple case where you want to apply a constant multiplier, m , to every point of the rate-time profile, you need to set **both the reserves and the rates multiplier to m** .



FREE-GAS: LIQUID RATIO (FGLR) METHOD

The equilibrium free-gas : liquid ratio (FGLR) method is a method that can be used in material balance calculations or to extrapolate trends in gas production in wells using decline curves for oil and water production. It is based on the idea of calculating the FGLR at which there would be no overall movement of the gas-liquid contact. It is then assumed that the field is tending towards such an equilibrium position i.e. the field FGLR converges towards the equilibrium FGLR. These assumptions are sufficient (together with a separate estimate of the rate of convergence) to give an approximate gas production profile.

Consider a saturated reservoir being produced from a number of wells in the oil-rim. If the wells can initially produce without coning-in free-gas, then there are two possible situations

- ② Reservoir pressure is dropping; the gas-cap expands until the gas cone just starts to reach the production perforations in one or more of the production wells
- ② Reservoir pressure is being fully maintained, through aquifer influx or water injection. There is no tendency for the gas-cap to expand, so the wells can continue to produce indefinitely without producing free-gas. The field can be considered to have an equilibrium FGLR of zero.

At anything other than extremely high pressures, gas is characterized by a very high mobility. Hence, if a gas-liquid contact moves and the gas cone reaches production perforations, the gas can potentially produce at very high rates – sufficiently high that the gas-liquid contact would not, under normal circumstance, move any further down. These factors suggest that, after a period, possibly of several years, of movement, the gas-liquid contact will tend to reach an equilibrium position.

If the contacts are not moving, it is straightforward to calculate the field FGLR for a simple one-tank model of the reservoir. If there is no injection, then

Equilibrium FGLR = (Gas-cap expansion per unit pressure drop) / (Oil-rim and water-leg expansion per unit pressure drop)

If there is injection, then

Equilibrium FGLR = ((Gas-cap expansion per unit pressure drop) x Pressure drop + gas injection) / ((Oil-rim and water-leg expansion per unit pressure drop) x Pressure drop + water injection)

Clearly, what happens during the process of moving towards this equilibrium is very important to oil recovery. However, for the purposes of a first-order estimate of expected gas production profiles (with oil reserves and water-cut development estimated using decline analysis) it is sufficient to assume that the field FGLR converges exponentially to the equilibrium free-gas liquid ratio, in the form

$$FGLR = FGLR_{initial} + (FGLR_{equilibrium} - FGLR_{initial}) \left(1 - e^{-Q_L/a}\right)$$

where

Q_L = cumulative liquid production

a = “gas breakthrough parameter”, a measure of the speed of convergence to equilibrium

Hence, the GOR equation used in the Muskat method can be replaced by

$$GOR = R_s + \frac{FGLR \cdot q_l}{q_o} = R_s + \frac{FGLR}{1 - BSW}$$

where

q_l = liquid production rate

q_o = oil production rate

C-CURVE DECLINE

The justification for exponential decline is usually quoted as the empirical observation that

$$\frac{dq_{oil}}{dt} / q_{oil} = -a \quad (A-1)$$

This is then extended to hyperbolic decline in the form

$$\frac{dq_{oil}}{dt} / q_{oil} = -a \cdot q_{oil}^b \quad (A-2)$$

However, examination of the physics of simple systems that exhibit exponential decline (such as the decay of radioactive particles, or production of a gas field under pure depletion) suggests that the fundamental driver is that the decay/production rate is proportional to the remaining population/reserves. In our case, this would be

$$\frac{dQ_{oil}}{dt} / (R - Q_{oil}) = a \quad (A-3)$$

where

Q_{oil} = cumulative oil production

R = ultimate recovery

The “C-curve” method is to extend this relationship to the more general form

$$\frac{dQ_{oil}}{dt} / (R - Q_{oil}) = (a + \beta(R - Q_{oil})^b) \quad (A-4)$$

or, in a dimensionless form

$$\frac{dr}{dx} / r = -(a + \beta r^b) \quad (A-5)$$

where $r = 1 - Q_{oil} / R$

and x is a measure of field aging, such as Q_{liquid} / R

The equivalent relationship in hyperbolic decline, with Arps b is

$$\frac{dr}{dx} / r = -\beta r^{b/(1-b)} \quad (A-6)$$

The key idea behind this approach is that it is the $dr/dx - r$ relationship that matters for creating life-of-well or life-of-field production profiles. The exact form of the relationship chosen does not matter much, providing it is sufficiently general to fit the shape of decline as observed in reality or as predicted in Eclipse. The C-curve relationship was chosen so as to be easily solvable to yield formulae that can be easily used and manipulated.

Solving for cumulative oil

Starting from the initial equation

$$\frac{dr}{dx} / r = -(a + \beta r^b) \quad (A-7)$$

the variables can be split as follows

$$\frac{1}{r(a + \beta r^b)} dr = -dx \quad (A-8)$$

Integrating both sides gives

$$\frac{1}{ab} \ln\left(\frac{r^b}{a + \beta r^b}\right) = -(x + \alpha) \quad (A-9)$$

where α is a constant

[Proof of integration of left hand side –

$$\begin{aligned} \frac{d}{dr} \left(\frac{1}{ab} \ln\left(\frac{r^b}{a + \beta r^b}\right) \right) &= \frac{d}{dr} \left(\frac{1}{ab} \ln\left(\frac{1}{ar^{-b} + \beta}\right) \right) \\ &= \frac{1}{ab} (ar^{-b} + \beta) \cdot \frac{-1}{(ar^{-b} + \beta)^2} \cdot (-abr^{-b-1}) = \frac{1}{r(a + \beta r^b)} \end{aligned} \quad (A-10)$$

]

Solving equation (A-9) for r gives

$$r = \sqrt[b]{\frac{a}{e^{ab(x+\alpha)} - \beta}} \quad (A-11)$$

The usual boundary conditions include

- a) production starts with dry oil i.e. $dr/dx = 1$ when $r = 1$
- b) At the start of production (i.e. when $r = 1$) $x = 0$

These boundary conditions allow us to express the α and β in terms of other variables, as follows

Condition (a) implies (from equation (A-7))

$$\beta = 1 - a \quad (\text{A-12})$$

Applying condition (b) to equation (A-9) gives

$$\frac{1}{ab} \ln \left(\frac{1}{a + (1-a)e^b} \right) = -(0 + \alpha) \quad (\text{A-13})$$

i.e.

$$\alpha = 0 \quad (\text{A-14})$$

Applying these values of α and β to equation (C) gives

$$r = b \sqrt[b]{\frac{a}{e^{abx} + a - 1}} \quad (\text{A-15})$$

Changing from r and x to R , Q_{liquid} and Q_{oil} gives

$$Q_{\text{oil}} = R \cdot \left[1 - \sqrt[b]{\frac{a}{a - 1 + e^{\left(\frac{a \cdot b \cdot Q_{\text{liquid}}}{R}\right)}}} \right] \quad (\text{A-16})$$

USING UR AS A FUNDAMENTAL DECLINE CURVE PARAMETER

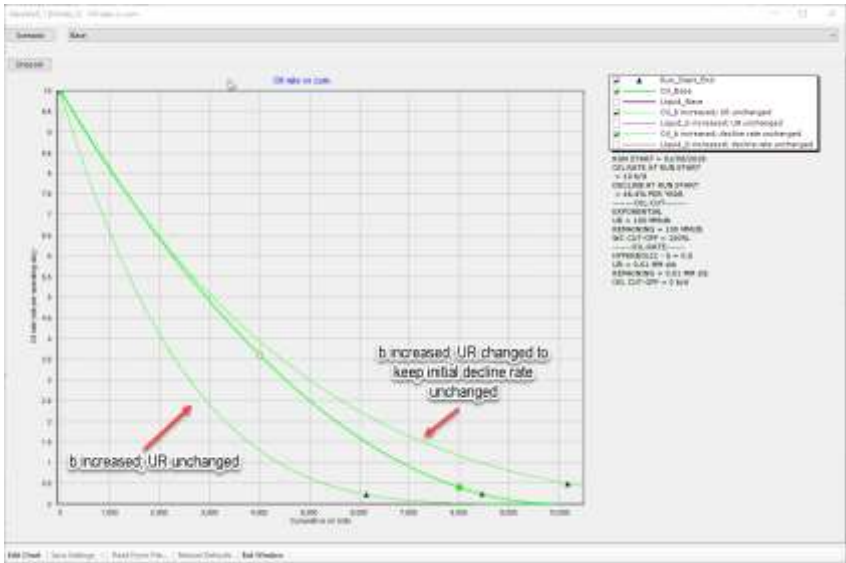
In the FUTURES's decline curve calculations, technical ultimate recovery (UR) is treated as a fundamental parameter, together with initial rate/cut etc, while the decline rate is calculated from the fundamental parameters. Other software treats the decline rate as a fundamental parameter and calculates technical UR. Although the two approaches allow for the same range of decline curves (and the same range of output oil, gas and water rates), they show some differences in behaviour during changes of a single parameter. It may be useful to explain the reasons behind this.

When decline analysis was carried out manually (without using computer applications), it was natural to fit a decline curve to the observed initial rate and initial decline rate (i.e. initial slope) and then calculate or derive graphically the UR. Many decline curve software applications provide a directly analogous work-flow – the UR is calculated from the initial rate and initial decline rate – but this is not a necessity (since, in an application, it is as easy to calculate initial decline rate from initial rate and UR).

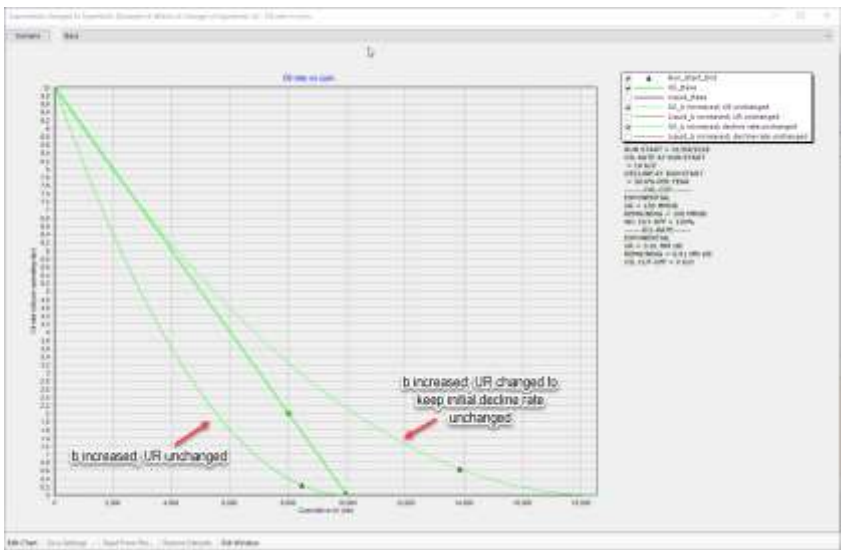
FUTURES is designed to be allow the user to update easily work from previous months or years. During such an update, the user updates production history and well-test data and moves the forecast start date. In the ideal case where the recent production history matches the previous forecast over this period, the aim is that the new forecast should require no changes to any parameters (other than the forecast start date). In the more normal situation where there are some fluctuations in the recent production history, FUTURES is designed to make the assumption that fluctuations from the decline trend reflect the variability in well production rates, rather than implying sudden fluctuations in the assessment of the well technical ultimate recovery. In other words, during the update process, FUTURES does not modify well ultimate recoveries – these are modified only if the user judges it fit. In contrast, FUTURES can be set to update automatically forecast well initial production rates in line with the most recent production history or well-test data. In order to implement such an update scheme, FUTURES needs to calculate well initial decline rates from initial production rates and UR values (and other decline parameters such as cumulative oil production at start, type of curve, hyperbolic b factor etc).

The well dialog in FUTURES asks the user to specify either the UR values or the initial decline rate. If the user modifies the b-factor by typing a new value with UR values specified or by moving the middle, light-shaded button in the interactive decline chart), FUTURES keeps the technical UR constant. The consequence, then, of an increase of the b-factor (illustrated below for a change from $b = 0.5$ to 0.75) is to give a decline curve with a higher initial decline rate i.e. a more pessimistic curve. This contrasts with the curve achieved if the b-factor is changed with the initial decline rate specified. The upper curve in the chart below is what is generated when b is

changed while keeping decline rate fixed. With such a curve, higher b values give more optimistic decline curves.



The same phenomenon – that it is pessimistic to increase b while keeping UR constant, but it is optimistic to increase b while keeping the initial decline rate constant – can be seen when changing from exponential decline to hyperbolic.



STOCHASTIC CALCULATIONS

FUTURES can carry out stochastic (probabilistic) calculations, so that, for example, your low reserves total is statistically reasonable. For the stochastic calculations, FUTURES does not use Monte Carlo methods, but calculates reserves, rate and/or duration multipliers for each well.

As with other calculations, FUTURES aims, when characterising the uncertainty ranges with production forecasts, to provide calculations that are as simple as possible. In many situations, it may be sufficient to calculate deterministically “Low”, “Medium” and “High” Forecasts for a given development option. However, this can prove be over-simplification. Such a deterministic “Low” Forecast, where every well is taken to have low reserves may end up being not the desired P15 in cumulative probability of reserves, but rather something like P01, since, in reality, it would be very unlikely that every well proves to have low reserves. This situation is often addressed by using Monte Carlo analysis, but FUTURES, instead, gives the option of it calculating the probability distributions of the sum of reserves (and/or initial well rates and/or drilling durations) and then scaling each well reserves (and/or rates and durations) so that their sum equals the desired P15 (or P20 or whatever) of the sum of reserves distribution.

Strictly speaking, FUTURES does not calculate the probability distribution of the sum of reserves etc, but calculates the variance of sum of reserves etc and makes the assumption that the sum is normally distributed. The reasonableness of this assumption is discussed below. The fundamental mathematical result used is the expression for the variance of a sum of random variables, X_i ,

$$\text{Var}\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n \sum_{j=1}^n \text{Cov}(X_i, X_j) = \sum_{i=1}^n \text{Var}(X_i) + 2 \sum_{1 \leq i < j \leq n} \text{Cov}(X_i, X_j).$$

This approach, besides being faster than Monte Carlo analysis, has the benefit of focusing attention on the important question of what are the covariances, or, equivalently, what are the Pearson correlation coefficients (which are simply normalised covariances). This approach (whenever it is reasonable to assume that the sum is normally distributed) also highlights the relative unimportance of the shapes of the distributions of the individual well properties.

Control of stochastic calculations is accessed via

FUTURES Main Menu > Forecasts > Edit Stochastic Calculations

which displays the dialog below.

Edit Stochastic Calculations

In the Stochastic Calculations, FUTURE applies scaling factors to well rates and reserves and to drilling sequence durations so that the sums of rates, reserves and durations are those for the cumulative probabilities specified in the Forecast. FUTURE assumes that each individual well rate, reserves or duration follows a trapezoidal distribution with specific points in the distribution defined across the various Scenarios. The sums of the rates etc are approximated as normal distributions with variances calculated from the relevant Pearson correlation co-efficients.

Correlation Coefficients

	Reserves	Well Rates	Work Sequence Durations
between wells in the same reservoir (or between work items in the same work sequence)	0.6	0.6	0.6
between different reservoirs in the same field	0.5	0.5	
between different fields (or between different work sequences)	0.2	0.2	0.2

Pearson correlation coefficients close to one indicate strong correlations while coefficients closer to zero indicate lower correlations

Scenarios - Cumulative probabilities for individual wells

The cumulative probability values for work sequence durations, well rates, or reserves indicate the probability that the actual value will be less than or equal to the given values. Note that for work durations, optimistic scenarios should have lower values indicating reduced likelihood of extending the work duration.

	Scenario	Reserves	Well Rates	Durations
	Downside	0.15	0.15	0.85
	Base	0.5	0.5	0.5
▶	Upside	0.85	0.85	0.15

Forecasts - Cumulative probabilities for the sums of all wells

Using the correlation coefficients, the Forecast will calculate the probability distribution functions for the sums of rates, reserves and durations and hence the values for the sums for the specified cumulative probabilities. The individual well values are then scaled to match.

	Forecast	Reserves	Well Rates	Durations
	LowCase			
	1P	0.1	0.1	0.9
	BaseCase			
	2P	0.5	0.5	0.5

OK
Cancel

The Scenario cumulative probabilities specify the meaning to be given to the input data – they tell us, in our example, that the “Downside” Scenario should contain, for

each well, an estimate of the initial well rate such that there is a 15% probability that the actual initial well rate will be lower.

The Forecast cumulative probabilities specify the meaning of the output data. In our example, the “1P” forecast should give a value for the sum of the well initial rates such that there is a 10% chance that the actual sum of well initial rates will be lower.

Points to note:-

- If a Forecast cumulative probability is left blank, then the corresponding parameters are treated in a simple deterministic manner i.e. they take the values specified in the Scenario used in the Forecast, without any scaling.
- If a Scenario cumulative probability is left blank, then the Scenario is not used in the construction of the individual well parameter probability distributions.
- It is possible to specify that certain individual wells are omitted from a stochastic scaling calculation, (so their parameters are unscaled and their probability distributions are not used in the construction of the probability distribution of the sum of the parameters). This is done by unticking the relevant cells in the “Stochastic Reserves”, “Stochastic Rates” or “Stochastic Durations” columns in “Edit Table of Wells”.



If the correlation coefficients are set to zero, then the individual well parameters will be completely uncorrelated i.e. their covariances will be zero. In consequence, the portfolio effect (diversification of risk) will reduce considerably the range of forecasts. If, on the other hand, all the correlation coefficients are set to one, then the individual well parameters will be completely correlated and the range of forecasts will be similar to that of deterministic forecasts.

HIGHER-LEVEL (RESERVOIR ETC) DECLINE CURVE ANALYSIS

With FUTURES, it is usually best to carry out decline curve analysis (DCA) at a well or completion level, since the stochastic calculations and automatic DCA address the two most common problems encountered by well-level DCA (namely that it can be laborious and the sum of well-level low case declines can be excessively pessimistic).

Even when carrying out well-level DCA, it is important to verify that higher-level profiles (for reservoirs, fields, assets etc) show reasonable decline curve trends. This can be done by looking at the "Liquid Op Day Rates vs Cum" tab in "View Profiles with Historicals" etc.

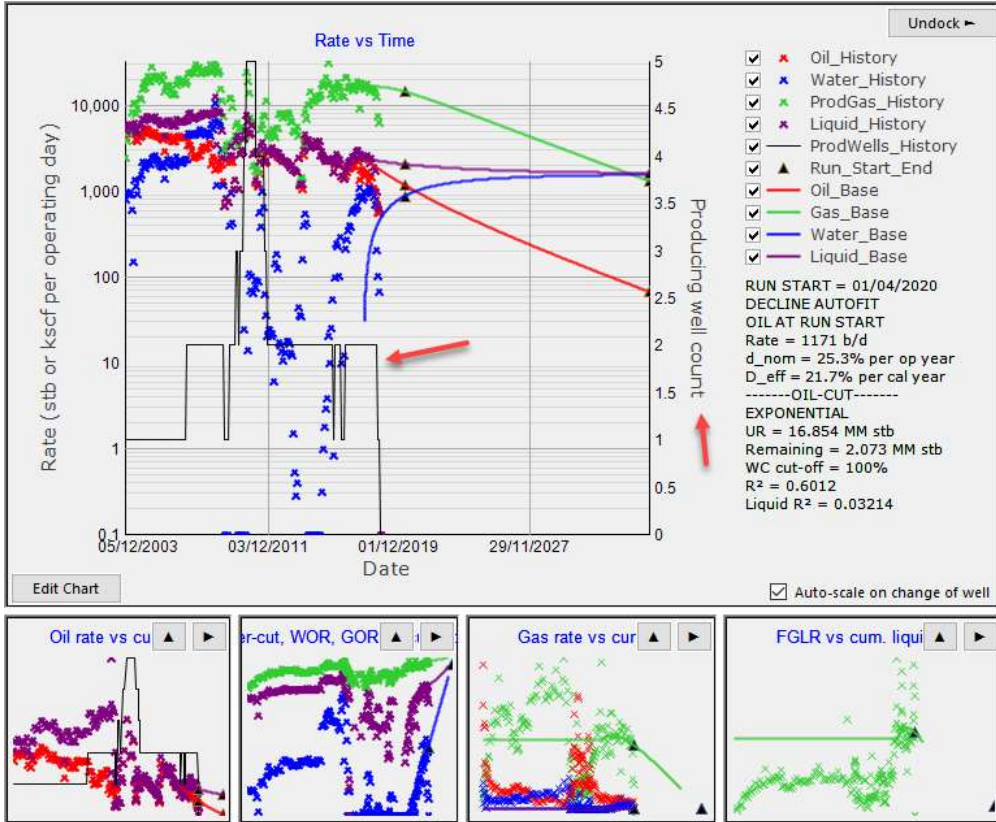


However, there can still be circumstances, such as problems with allocation, when it is best to carry out higher-level DCA. The most straight-forward way to do this is to create a separate FUTURES model where the FUTURES Well Elements represent the higher-level entities. Such a model will need to have a historical spectre containing the historical higher-level profiles. You can create such a historical spectre using the option (when you have the original well-based Model open)

FUTURES Main Menu > File > Save Aggregate Historical Profiles as....

FUTURES will then create a new historical spectre containing summed-up historical profiles for all the higher-level entities (i.e. Reservoirs, Fields, Nodes, Projects, Reporting Groups, Categories etc) that are contained in your original well-based model. The new historical spectre will include information about how many wells

were producing at any given time and this is displayed in the decline charts of the new FUTURES model.



IMPACT CALCULATIONS

When working with short-term activity plans and the like, it is often useful to consider the “Losses” or gains that result from each activity. FUTURES provides an option – Impact Calculations – to calculate such losses and gains.

Given that some activities – e.g. the bringing onstream of new wells – result in production gains, FUTURES does not use the customary term “Losses”. Instead, FUTURES uses the term “Impact”. If a FUTURES Impact has a positive value, this is a gain in production, while if an Impact has a negative value, this is a loss. (The reason for avoiding the term “Losses”, is that it would be confusing to represent a gain as a negative “Loss”, and even more confusing to represent a gain as a positive “Loss”).

Most activities have effects on production that stretch over many time-steps. Typically, a platform shut-down will cause a loss of production during the period of the shut-down, but the oil and gas is not permanently lost, only delayed. The shut-in wells will usually not have declined during the shut-in period, and so, on resumption of production, will produce at slightly higher rates than they would have if there had been no production. this loss of production will be gradually recovered over subsequent months and years.

SPREADSHEET CALCULATIONS AND EXPORTS

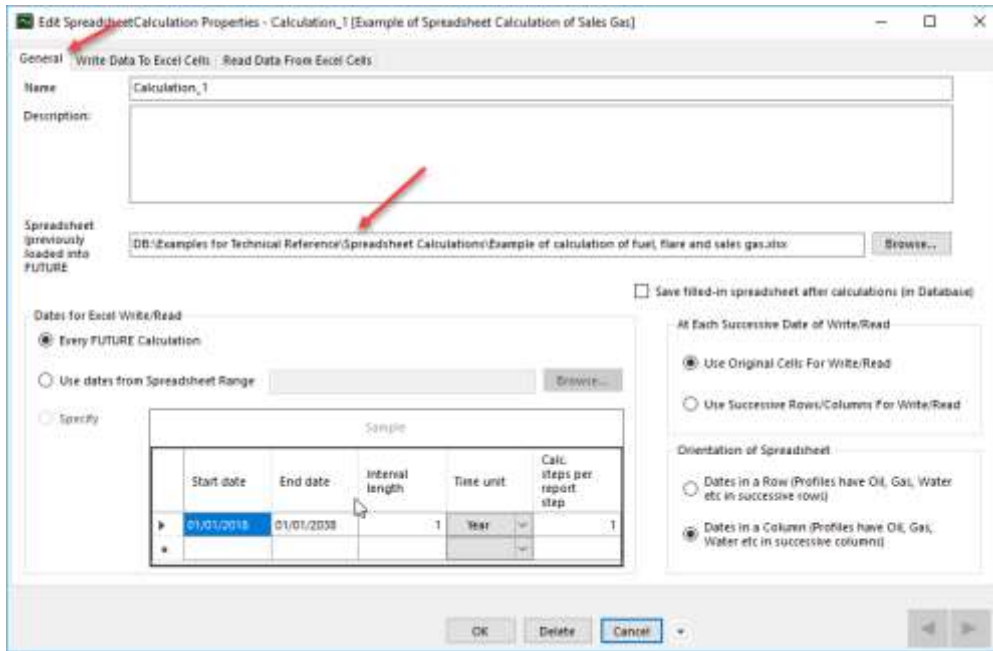
There are often times when it is useful to combine the power of FUTURES with the flexibility of Excel. There are three separate methods for combining FUTURES and Excel calculations and results –

1. “Spreadsheet Calculations” – Suitable for calculating and storing within FUTURES non-standard profiles e.g. sales gas entitlements. At each time-step, FUTURES feeds data into an internal spreadsheet (that has been previously loaded into the FUTURES database) and then reads one or more profiles etc from the spreadsheet. These profiles are then saved in the database together with all the normal well, reservoir, node etc profiles.
2. “Spreadsheet Exports” – For the automatic export of selected profiles to external spreadsheets.
3. “Export Production Profiles to Excel” – For the manual export of selected profiles to external spreadsheets (with options for choice of units and time-steps).

SPREADSHEET CALCULATIONS

A FUTURES “Spreadsheet Calculation” is an entity held within a FUTURES Model. These Spreadsheet Calculations are created and edited in much the manner of creating and editing Wells, Reservoirs, Nodes etc, using the functionality under **FUTURES Main Menu > Edit**. It should be noted that, unlike the situation with Wells etc, Spreadsheet Calculations properties are the same for all Scenarios.

Each Spreadsheet Calculation makes use of a specified Excel spreadsheet that has earlier loaded into the FUTURES database, using the functionality **FUTURES Main Menu > Import > Spreadsheet for user-defined calculations**.

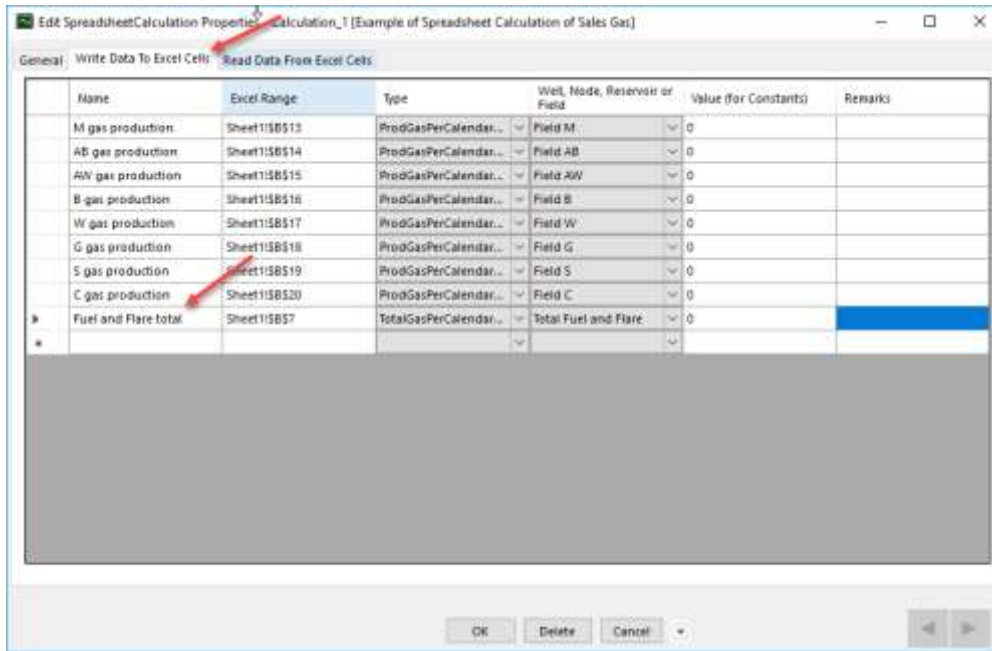


The “General” tab in the “Edit Spreadsheet Calculation Properties” dialog also allows you to specify the manner that FUTURES writes to and reads from the spreadsheet – at what time points; whether successive rows or columns are used etc.

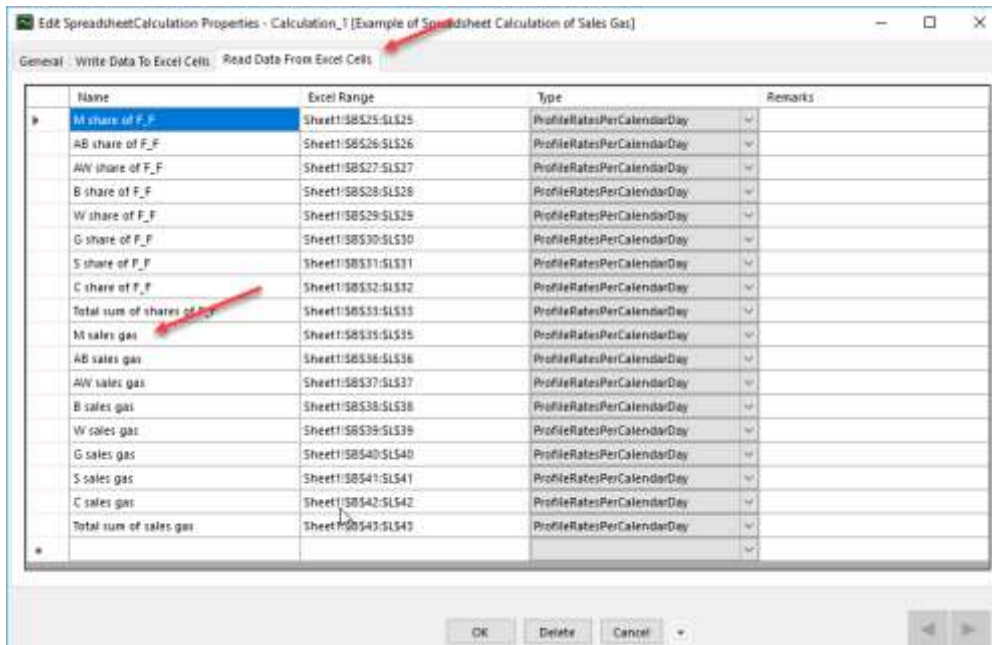
The “Write Data to Excel Cells” tab allows you to specify what data is to be written into the spreadsheet. A typical use of Spreadsheet Calculations would be to calculate, in a production system where multiple fields are processed together and fuel and flare gas is removed, what is the sales gas entitlement of each Field. The spreadsheet might then need to be told by the FUTURES Forecast for each time-step

- The fuel and flare total
- The gas production from each field (assuming that the allocation calculations are based on gas rates; if the allocation was more complicated and depended also on e.g. liquid rates, then it would be necessary to tell the spreadsheet the liquid production from each field).

Typical settings are illustrated below. It should be noted that the “Excel Range” cells allow you to browse the spreadsheet to select the range (enter the cell then click on the button that appears on the right of the cell).



Once the spreadsheet has calculated the sales gas entitlements etc, then the Spreadsheet Calculation needs to read these results, so that they can be stored as profiles in the Forecast (and then viewed, exported etc in the same manner as the profiles of Wells, Reservoirs, Nodes etc). The screen-shot below illustrates how the sales gas profile for Field M is specified as being read from the Excel range "Sheet1!\$B\$35:\$L\$35".



4. REFERENCES

- Al-Hussainy, R., Ramey, H. J., & Crawford, P. B. (1966). The flow of real gases through porous media. *J. Pet. Tech.* , 624-636.
- Beggs, D. H., & Brill, J. P. (1973). A study of two-phase flow in inclined pipes. *Journal of Petroleum Technology*, pp. 607-617.
- Carr, N. L., Kobayashi, R., & Burrows, D. B. (1954). Viscosity of Hydrocarbon Gases Under Pressure. *J. Pet. Technol.*, 47-55.
- Craft, B. C., & Hawkins, M. (1990). *Applied Petroleum Engineering*. NJ: Prentice hall.
- Cunningham, P., Moutari, S., & McCann, J. F. (2015). Optimization using the interior point method. *Serafim Ltd Report*.
- Dake, L. P. (1978). *Fundamentals of reservoir engineering*. Amsterdam: Elsevier.
- Dranchuk, P. M., & Abou-Kassem, J. H. (1973). Calculation of Z factors for natural gases using equations of state. *J. Can. Pet. Technol*, 34-36.
- Economides, M. J., Hill, A. D., & Ehlig-Economides, C. (1994). *Petroleum Production Systems*. NJ: Prentice Hall.
- Geertsma, J. (1974). Estimating the coefficient of inertial resistance in fluid flow through porous media. *Soc. Pet. Eng. J.*, pp. 445-450.
- Keunzi, H. P., Tzschach, H. G., & Zender, C. A. (1971). *Numerical Methods of Mathematical Optimization*. New York: Academic Press.
- Mehotra, S. (1992). On the implementation of a primal-dual interior point method. *SIAM Journal of Optimization*, pp. 575-601.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing (3rd ed.)*. . New York: Cambridge University Press.

