When models fail

Modelling and simulation programs are no substitute for years of practical experience in plant design, argues **Sean Moran**

THERE is no such thing as process design. Chemical engineers may be known as process engineers in professional life, but we do not design processes – we design process plants. Engineers design physical artefacts, and a process is not an object. Process plants, however, are – they are made of concrete and steel, wires and pipes, tanks and pumps. Processes happen in them.

While modelling and simulation programs can construct a very approximate virtual model of the process happening inside a process plant, we, as designers, should not forget that the model they produce is not the plant itself.

to the drawing board

The process plant designer specifies the physical sub-components of the plant and how these are to be connected and controlled in order to carry out the process safely, reliably and economically. The process is an emergent property of the specified collection and interconnection of parts. The job of selecting and specifying the parts and their interconnections involves a great deal of professional judgement, as well as the judicious application of engineering, science and mathematics.

Documenting these choices is done largely by means of drawings. Drawings allow communication with other engineering disciplines, which is necessary to optimise the plant design. The people who will build the physical plant need drawings to do their jobs. The plant itself is the ultimate deliverable, but the immediate deliverables are mostly drawings.

The people who will build the physical plant need drawings to do their jobs.

This is process plant design, a rather messy, intuitive, collaborative, multi-disciplinary, multifactorial business. It involves knowing the needs of electrical, software and civil engineers, equipment suppliers and of those who will procure, commission and operate the plant. It also involves communicating and negotiating with these other disciplines.

The precise process conditions to be used are actually not that important a part of the whole activity. If we are honest with ourselves, we cannot as designers predict the conditions within the plant as constructed to a high degree of precision.

A good process plant designer makes sure that the plant design envelope encompasses the range of conditions that the plant is likely to see, that it is robust enough to maintain adequate performance across that envelope, and is understood well enough to be reliably controlled under all foreseeable circumstances.

The plant design process as commonly practised relies upon a chemical engineer to produce an approximate but sufficiently realistic model of the physical and chemical processes happening in the plant. This model is a useful abstract approximation which informs the professional judgement of the plant designer. It usually considers the most important aspects of unit operation design, mass and energy balance, a bit of chemistry, and so on. Over the course of my career it has gone from being written by hand on paper to being written in a spreadsheet.

The simpler such a model is the more useful (but unfortunately less precise) it is. If you doubt the truth of this statement, consider the extreme condition: it would be quicker to build the plant than to build a completely true model of the plant, and this model would tell us nothing we could not measure on the plant itself.

theory vs practice

In 1999 IChemE's CAPE Special Interest Group produced a set of guidelines[1] which, in summary, caution against the uncritical use of any computer model, even the humble spreadsheet. Process plant design is the same process as it ever was, and while computers may be faster these days, they aren't any closer to being people than they were in 1999.

The guidelines say that if a computer is used by a chemical engineer in the course of their professional work, it is the sole responsibility of the practitioner to verify the validity of the inputs, to validate the applicability of the program used for the application to which it is put, and to understand all defaults and assumptions built into that program. It is the legal and professional responsibility of the practitioner to distrust the output of the program, and to check the outputs thoroughly for sense.

Those expert in the use of such programs know all of this, but they also know that less skilled or rigorous users are commonplace, and that these programs are seductively easy to misuse. It is a great deal easier to use them badly than to use them properly. As a consequence of this, there are many who think that successful modelling (often defined by the worst users as simply getting the recycle streams to converge) proves that a design is viable. These are not just less able students, there are professors advocating proof by modelling, and the use of models to generate rules of thumb for design.

I am no expert in the use of these programs, but such programs are generally written by software engineers, miscellaneous physical science graduates, and graduate chemical engineers. The bright young engineers involved in software development have had no opportunity to develop professional judgement through engineering practice.

Exercising such judgement is the most crucial component in process plant design. Allowing new graduate engineers and non-engineers to produce a product which looks like a process model – but is not – seems very risky indeed.

plugging the gaps

If you are writing a program which models or simulates a process plant, you have to build in many assumptions and take many shortcuts in order to get it to work. Writing such a program is itself a kind of engineering, so you can no more write it mathematically than you can design a plant from first principles. You have to use heuristics to plug the gaps and uncertainties in your knowledge.

Such plant design shortcuts are known to experienced process plant designers, but these rules of thumb are frequently not in the public domain, and are unlikely to be known to those writing these programs.

For example, even something as apparently simple as sizing pumps for acid and alkali addition to control pH in aqueous systems is not something which can be done from first principles. The overwhelming majority of the duty can be dictated by the buffering capacity of the system. There may be a number of buffering systems, each of which might have a number of ions in equilibrium with each other in proportions which vary with respect to pH and temperature. When you add in consideration of varying flow, pH and key buffering ion concentrations, the system defies rational analysis. There is a technique which allows pump sizes to be practically determined from a small number of analyses of feed water pH, and alkalinity. I can't tell you what it is though – it's know-how.

know your limits

An expert process plant designer has acquired the professional judgement, and knowledge of design practice to understand a model's uses and limitations, whereas a graduate engineer has not and is at risk of being seduced by the potential benefits because they don't know what they don't know.

Programmers have to set certain variables to default values to make the program easy to get going. They have to simplify mathematical models of physical sciences to work reasonably quickly on readily-available computers by making assumptions which become invisible, implicit features of the program.

Judgement is the most crucial component in process plant design.

Such programs usually offer users standard databases of physical properties, whose original sources of data would have attached margins of error and ranges of validity to the measured properties. Programmers may well arrange for their software to helpfully generate intermediate values by interpolation, and (worse yet) out-of-range values by extrapolation.

Most unhelpfully of all, many programmers get around the difficulty of only including a limited range of unit operations by allowing users to set up imaginary unit operations called something like 'separators' to stand in their stead. These allow the user to set up a non-existent process step which, by unexplained means, divides an incoming stream according to the user's wishes.

There are in fact normally three ways to address the common problem of a missing unit operation. One can substitute it on the program's flowsheet with something the user thinks is pretty similar, (thus I have seen students substitute absorption columns for cooling towers in hysys); use a separator; or (much more difficult) write your own reasonably accurate module of the missing unit op. Human nature being what it is, I don't see too much of this last option.

Then there is the fact that very few users will set up a dynamic model instead of the (far-easier-to-set-up) static steady state scenario. In the real world, there is no steady state. Steady state is an imaginary scenario which we set up in academia to reduce the complexity of process engineering to the point where beginners can make a start on it. When we start designing plants which are actually going to be built, we have to leave the training wheels of steady state design behind. We need to design a plant which can cope with all of the things it might see during its life.

The most common way to do this is to set up a number of steady state designs which address the outer limits of the design envelope. We might for example have a high-flow/low feedstock quality scenario, a low flow/ high feedstock quality scenario, and so on.

all dressed up, no place to go?

Process plant designers tend to set up their mass and energy balance models in MS Excel, in such a way that it is easy to vary the parameters which define the design envelope, and quickly generate multiple scenarios, allowing them to carry out the sensitivity

It would be quicker to build the plant than to build a completely true model of the plant, and this model would tell us nothing we could not measure on the plant itself.

analysis which the CAPE SIG guidelines require.

Unfortunately, the reasonably slick-looking outputs of modelling programs seem to encourage unreasonable trust in their default outputs. Generating multiple scenarios and analysing sensitivity are often thought unnecessary. The ease with which these programs are misused, and their ability to shortcut understanding of the process to produce an apparent solution, can be a problem if they are not used as intended.

These programs allow a 'process' to be 'designed' and supposedly optimised in a model space which has no limitations in physical layout, no distinctions between the commercially-available types or sizes of equipment, no consideration of the requirements of other disciplines, no real consideration of cost impacts, and little consideration of many safety, environmental and QA issues. Physical plants may be optimised with tools such as pinch analysis, but anyone 'optimising' a virtual plant with these tools needs to understand that optimising the map does not optimise the territory.

For example, I can apply pinch analysis to a computer model of a plant which has been defined only in the broadest terms, and has not been fed with lots of data from a real plant. I can get the recycles to converge, but the resolution of the model is broadly similar to the resolution of the cost estimate associated with it, maybe +/- 50%.

There are limits on the ability to control and measure a real plant, but they probably don't define real-world optimisation. For example, on a real plant, we might save 5% of a heat exchange duty by switching to a spare heat exchanger. Any optimisation technique which gives me lower savings than this is probably not worthwhile acting on. I can however carry out a pinch analysis on the real plant, and I might find some savings which I can be reasonably sure will materialise in practice.

But in my rough computer model, I am working to only +/- 50% costing. What am I to make of pinch analysis telling me that an additional heat exchanger will save me 5% a year in energy costs, and the computer model telling me that this will add +10% to my overall capital cost? The correct answer is nothing whatever. These numbers are far below the resolution of my model. They don't mean a thing.

Only the most rigorous users will ascertain whether the thermodynamic and physical data is valid over the ranges of physical conditions used, and how large the product of all the uncertainties associated with the use of that data is. Few indeed will talk directly to program vendors and writers to understand all of the assumptions built into the program, as the CAPE SIG guidelines suggest.

Program default output is usually presented in a spuriously precise way. The program has not been told how large a margin of error is associated with its input data. It can be the case that when we add together all of the errors and uncertainties in the data, the assumptions and approximations and errors built into the program, and those of the program operator, our answer might be only accurate to the first two or three figures.

There is nothing wrong with data with two or three significant figures, unless you start to believe that the model is the thing itself, and that professional judgment is no longer needed. As the guidelines say, outputs need to look sensible, and if you don't know what sensible looks like, you should speak to someone who does. Modelling programs can produce nonsensical outputs even if properly operated, especially in systems with nested recycles.

In one example, a company used a modelling programme to generate rules of thumb used in a design manual which were obviously insufficient to those more experienced in designing the type of plant. By the time I spotted this and objected, several full-scale plants had been built which were far smaller than they needed to be, due to the effective removal of margins of safety. The investigation into the mistake consisted of using a different modelling program to test the usual rules of thumb, but the data which was input was chosen to make them look excessive. The effective failure of the real plant was not thought to be as strong a piece of evidence as computer modelling.

This is no trivial matter, as I hear that many now believe that this fallacious approach can produce plant designs which require no additional safety factors. This looks to me exactly like the process of erosion of safety margins due to complacent application of methods outside their safe operating envelope leading to the engineering disasters such as that of the Tacoma Narrows Bridge, which Henry Petroski discusses in many of his books. As he says in the context of structural engineering: "The more successful a design, the more likely it is to be a model for future designs. But because engineering and construction are influenced by aesthetics, economics, and, yes, ethics or their absence, designs tend to get pared down in time. This paring down can take the form of enlargement in size without a proportional increase in strength, in defiance of the size effect; streamlining in the sense of doing away with what is believed to be superfluous; lightening by the use of stronger materials or materials stressed higher than before; and cheating, which can take the form of leaving out some indicated reinforcement in concrete or deliberately substituting inferior materials for specified ones. The cumulative effect of such paring down of strength is a product that can more readily fail. If the trend continues indefinitely, failure is sure to occur."

A simple pressure vessel is usually designed with a safety factor of 4, meaning it is four times as strong as it theoretically needs to be in a world without error, incompetence or cheating. Designers of far more complex process plants need the humility to acknowledge our persistent inability to control and predict both nature and art. We do not yet understand or control the physical world to the point where first-principles designs (which is the best that modelling programs can produce) are safe, robust or economical. I would invite anyone who disagrees to fly in an aircraft with the safety factor of 1 – exactly as strong as it needs to be in a perfect world – which some of our plants are now approaching.

guilty until proven innocent

These programs are potentially useful tools, but the CAPE working group's advice to treat their outputs as guilty until proven innocent is often nowadays reversed by those who do not understand their limitations. Their outputs are now considered by many as almost equivalent proof of a design to actually building the plant, and consequently able to be used to optimise a process plant in virtual reality. **tce**

Sean Moran (sean.moran@nottingham.ac.uk) is director of Expertise Limited, associate professor at the University of Nottingham, and author of the forthcoming IChemE/Elsevier book *An Applied Guide to Process and Plant Design*.

1. http://bit.ly/1gGqpyL

Chemical Engineering Matters

The topics discussed in this article refer to the following lines on the vistas of IChemE's technical strategy document *Chemical Engineering Matters*:

Health and wellbeing *Lines 8–9, 21*

Visit **www.icheme.org/vistas1** to discover where this article and your own activities fit into the myriad of grand challenges facing chemical engineers