

Initializing Distillation Column Models¹

Roger Fletcher* and William Morton†

**Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland, UK.*

†School of Chemical Engineering, University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh EH9 3JL, Scotland, UK.

Numerical Analysis Report NA/191, July 1999

Abstract

Difficulties associated with the optimisation of distillation column models by nonlinear programming are considered. The paper presents a systematic procedure to enable these difficulties to be overcome and proposes a particular formulation of the distillation column model.

A certain limiting case of the column model is examined, that of infinite reflux or zero feed. This limiting case considerably simplifies the model and provides a system of nonlinear equations that is readily solved. The solution of this problem gives useful information about the purity that can be achieved in the general case and the number of plates needed to attain a given level of purity. The limiting problem provides starting values for the solution of the general column and suggests a homotopy that can be followed if difficulties arise in obtaining convergence. To obtain a stable form of the limiting case requires the general column model to be formulated in a certain way, which to our knowledge has not previously been considered. The ideas have been successfully tested on various multi-column flowsheets involving distillation columns with heat integration.

Keywords distillation column, mathematical model, zero-feed column, nonlinear programming, AMPL.

1 Introduction

In this paper we consider problems associated with the modelling of distillation columns and the optimisation of such models by nonlinear programming (NLP). Obtaining convergence of an NLP solver on distillation models is an uncertain process. The solver may fail to converge, or converge to an infeasible solution (that is a solution that does not satisfy the constraints of the problem) or may converge to a physically unrealistic

¹A preliminary version of this paper was presented at the Dundee Biennial Conference on Numerical Analysis, June 1999

solution. This paper aims to present a systematic procedure to enable these difficulties to be overcome. It also proposes a particular formulation of the distillation column model.

The main idea is to examine a certain limiting case of the column model where the feed goes to zero. This ‘infinite reflux’ case is shown to considerably simplify the model and provides a system of nonlinear equations that is readily solved, and whose solution gives useful information about the purity that can be achieved in the general case and the number of plates needed to attain a given level of purity. This feature is potentially useful in cases (e.g. of non-ideal systems) where the analytic solution available from Fenske’s equation which assumes constant relative volatility (King [4], p.426) does not apply.

The limiting problem also provides starting values for the NLP solution of the general column and suggests a homotopy that can be followed if difficulties arise in obtaining convergence. The general column model is formulated in a certain way which provides a stable form of the limiting case, and promotes convergence from the starting guesses provided by the infinite reflux model.

In Section 2 we describe the formulation which we use for the mathematical model of a distillation column. This contains some new features devolving from the later part of the paper, although many of the ideas are common to other models that have been used. In Section 3 the limiting case is described and developed, based on the solution of a column with zero feed (that is, at infinite reflux). In Section 4 an extra condition is determined so that the zero-feed solution becomes the limiting solution of the general column. The advantages of pre-solving this limiting case problem are described and discussed in Sections 4 and 5. Some experience with these procedures for solving complex multi-column flowsheets with heat integration is described.

Within the structure imposed by the limiting zero-feed column, there are still various alternative model formulations that might be used. Section 6 considers some alternative possibilities and presents evidence supporting the particular choice made in Section 2.

The experiments with different modelling formulations have been carried out using the modelling language AMPL (Fourer et al. [2]). This expedites the modelling very considerably and provides a number of useful structural features for flowsheets. The main advantage however is that it provides automatic generation of first and second derivatives of all nonlinear problem functions and hence frees the user from this onerous task, whilst ensuring that it is done accurately and efficiently (and with due regard to sparsity). Another advantage is that AMPL is now linked to various state of the art NLP solvers, which can be accessed over the web. An AMPL program which describes our distillation column model is given in the Appendix.

The solver that we have used in our calculations is a sequential quadratic programming (SQP) solver, referred to as `filterSQP`, which uses the idea of an NLP filter (Fletcher and Leyffer [1]) to promote convergence. This method has proved very reliable overall, although some of the QP subproblems generated by large column models have proved to be very ill-conditioned, causing difficulties (usually not terminal) for the QP solver.

Before proceeding to describe the column model we describe the terminology used in

the rest of the paper for a distillation column. The purpose of a distillation column is to split a given multicomponent feed stream into two separate streams, referred to as the *tops* and *bottoms*. The aim is that the tops stream should be composed of only the more volatile *light components*, whilst the bottoms stream is composed of only the less volatile *heavy components*. The so-called *light key* is the least volatile light component and the *heavy key* is the most volatile heavy component. In practice a totally sharp split between the keys cannot be attained and some level of impurity in the tops and bottoms streams must be accepted (of heavy and light components respectively). The column consists of a number of stages or plates, on each of which contact between vapour and liquid takes place, promoting the transfer of light components to the vapour phase towards the top and heavies to the liquid towards the bottom.

The feed can be separated into its constituent components by linking a number of distillation columns, and choosing a different light and heavy key for each column, as illustrated for example by Fraga and McKinnon [3].

We only consider the steady state solution of the column, and not the dynamics associated with starting up the column or the effect of fluctuations in the feed composition. We assume that equilibrium is reached on each plate, an assumption which for real columns requires modification by using a plate efficiency or a mass transfer based model. The systems studied so far have assumed ideal vapour-liquid equilibrium (activity coefficients equal to one) but we expect that the ideas presented here will also simplify the optimisation of column models involving non-ideally mixed components, which we propose to study in future work.

2 A Column Model

Mathematical models for distillation columns can be classified according to whether they consider the details of composition, temperature and flow on each plate (the so-called ‘rigorous’ models) or whether they provide an overall description of the column using fewer variables, based on some kind of interpolation between behaviours in the limiting cases of total reflux and minimum reflux (King [4], p.428).

In this paper we focus on an idealised, rigorous column model with $N + 2$ equilibrium stages. These consist of N plates together with a condenser and reboiler. Our model assumes the use of a kettle-type reboiler (Sinnott [6], p.686). The plates are numbered as $i = 1, 2, \dots, N$ with plate 1 being at the bottom of the column and plate N at the top. Quantities associated with the reboiler are indexed by $i = 0$ and with the condenser by $i = N + 1$. The feed to the column is directed on to plate S and is composed of a mixture of m components. The components are numbered as $j = 1, 2, \dots, m$ in order of decreasing volatility, component 1 being the most volatile component, and so on. The model assumes that the number of stages (N) and the feed stage (S) are both fixed.

Heat input to the reboiler (at a rate Q^r) vapourises some of the material which passes up the column as vapour. At the top of the column, the vapour is condensed (with a rate of heat removal Q^c). Some of the liquid is returned as reflux to the top of the column.

The tops output stream is also taken from the condenser outflow, and the ratio of the reflux flow to the flow in the tops stream is referred to as the *reflux ratio*. The bottoms stream is drawn from the liquid in the kettle reboiler with a flow rate which preserves overall material balance.

The flow of liquid descending from stage i to $i - 1$ is denoted by L_i (liquid flow) and the flow of vapour upwards from stage i to $i + 1$ by V_i (vapour flow). The mole fraction of component j in the liquid L_i is denoted by x_{ij} and in the vapour V_i by y_{ij} . It is convenient to refer to these quantities by the m -component vectors \mathbf{x}_i and \mathbf{y}_i . The flow of each component of liquid may be expressed by $l_{ij} = L_i x_{ij}$ or by $\mathbf{l}_i = L_i \mathbf{x}_i$. Likewise $v_{ij} = V_i y_{ij}$ or $\mathbf{v}_i = V_i \mathbf{y}_i$ expresses the componentwise flow of vapour. The temperature of these quantities associated with stage i is denoted by T_i . The flow of heat in the liquid from stage i to $i - 1$ is denoted by H_i^L , and the flow of heat in the vapour from stage i to $i + 1$ by H_i^V .

Our model assumes that the column operates at a uniform pressure P . A more detailed model could allow for a pressure drop from stage to stage up the column but that is not done here. The liquid flows in the tops and bottoms stream are denoted respectively by D and B . The fractional composition of the tops stream is the same as that of the recycled liquid, and is denoted by \mathbf{x}_{N+1} . The fractional composition of the bottoms stream is the same as that of the reboiler liquid, that is \mathbf{x}_0 . The feed specification is given and is assumed to be liquid. The total flow in the feed is denoted by F and its fractional composition by \mathbf{x}^F . Componentwise flows for the feed, tops and bottoms streams are denoted by

$$\mathbf{f} = F\mathbf{x}^F, \quad \mathbf{d} = D\mathbf{x}_{N+1}, \quad \mathbf{b} = B\mathbf{x}_0 \quad (2.1)$$

respectively.

The variables are required to satisfy various constraints, for example all flows and compositions are non-negative and the fractional compositions must satisfy

$$\sum_{j=1}^m x_{ij} = 1 \quad (2.2)$$

and

$$\sum_{j=1}^m y_{ij} = 1. \quad (2.3)$$

There are equations for

- the material balance of each component at each stage of the column, and
- the overall heat balance on each stage.

We give a suitable definition of these constraints below. Since the vapour and liquid on each plate, and also at the reboiler, are assumed to be in equilibrium, the vapour and liquid compositions are related by

$$y_{ij} = K_{ij}x_{ij} \quad (2.4)$$

where, for ideal mixtures, the K_{ij} are nonlinear functions of T_i defined by

$$K_{ij} = \frac{1}{P} \exp \left(A_j + \frac{B_j}{C_j + T_i} \right). \quad (2.5)$$

The quantities A_j , B_j and C_j for $j = 1, \dots, m$ are known as Antoine constants and are given as data. A related condition is that the condenser outflow must be saturated liquid, which can be expressed by the bubble point equation:

$$\sum_{j=1}^m K_{N+1,j}x_{N+1,j} = 1. \quad (2.6)$$

We also note that equation (2.4) may be expressed as

$$\mathbf{y}_i = \mathbf{K}_i \mathbf{x}_i \quad (2.7)$$

where \mathbf{K}_i is a diagonal matrix with diagonal elements K_{ij} , $j = 1, \dots, m$. The heat flows H_i^L and H_i^V are also nonlinear functions of the T_i which can be derived by integrating expressions for the specific heat of the components, derived from experimental observation. We have used relatively simple quadratic expressions for enthalpies, of the form

$$H_i^L = \sum_{j=1}^m l_{ij}(a_j + a'_j T_i + a''_j T_i^2) \quad (2.8)$$

and

$$H_i^V = \sum_{j=1}^m v_{ij}(\beta_j + \beta'_j T_i + \beta''_j T_i^2), \quad (2.9)$$

where the coefficients a_j , a'_j , a''_j and β_j , β'_j , β''_j are given as data. The heat flow in the feed stream is likewise given by

$$H^F = \sum_{j=1}^m f_j(a_j + a'_j T^F + a''_j T^{F2}), \quad (2.10)$$

where T^F is the feed temperature (also given as data). Similar expressions hold for the heat flows H^D and H^B in the tops and bottoms streams, namely

$$H^D = \sum_{j=1}^m d_j(a_j + a'_j T_{N+1} + a''_j T_{N+1}^2) \quad (2.11)$$

and

$$H^B = \sum_{j=1}^m b_j (a_j + a'_j T_0 + a''_j T_0^2). \quad (2.12)$$

In stating the form of the mathematical model to be presented to the nonlinear programming solver, various considerations must be taken into account. It is usually advisable to avoid redundancy in the constraint formulation, as this can otherwise cause potential difficulties due to the loss of linear independence in any linearized constraints. There is often the option of using equations to define variables (essentially eliminating variables and constraints from the problem), or alternatively of leaving the variables and constraints explicitly in the formulation. The latter gives rise to larger nonlinear programming problems, but the equations are of a less complex nature and there is often more sparsity and structure in the resulting linear systems that can be taken into account. Other issues, such as suitable scaling of the variables and constraints, are often important.

In the rest of this section we describe a formulation which is reasonably compact and well-scaled, and which has worked well in practice. It is based mainly on the standard MESH equations for rigorous column modelling (Sinnott [6], p. 452). The important feature of our approach is a particular selection of independent variables. We then use some of the MESH equations to solve for the dependent variables, so as to assist convergence of the solution of the column and to avoid unphysical variable values, e.g. negative component flows within the column.

We start by introducing a new independent variable θ , $0 \leq \theta \leq 1$, which measures the overall recovery in the tops stream, or the proportion of the feed liquid flow that is directed to the tops stream. This is somewhat different to other models that have been suggested, and is motivated by the ideas pursued in Sections 3 and 4. Thus we can express the total tops and bottoms flows as

$$D = \theta F \quad \text{and} \quad B = (1 - \theta)F. \quad (2.13)$$

The overall componentwise material balance for the column is $\mathbf{f} = \mathbf{d} + \mathbf{b}$, or alternatively, after using (2.1) and (2.13) and dividing through by F , as

$$\mathbf{x}^F = \theta \mathbf{x}_{N+1} + (1 - \theta) \mathbf{x}_0. \quad (2.14)$$

This equation is well-scaled and has a useful property which we exploit in Section 4.

We next consider the liquid and vapour flows L_i and V_i . These must satisfy the equations

$$V_i = L_{i+1} + D \quad i = S, \dots, N \quad (2.15)$$

and

$$L_i = V_{i-1} + B \quad i = 1, \dots, S. \quad (2.16)$$

This is a consequence of equating the net flow between stages with the output flow at the tops and bottoms. In our model we select V_0, \dots, V_{S-1} and L_{S+1}, \dots, L_{N+1} as independent non-negative variables, and use (2.15) and (2.16) to determine the remaining values of V_i and L_i . The reason for this is that the non-negativity of the remaining flows is automatically satisfied and does not have to be enforced as an inequality constraint in the model.

There are various ways in which the \mathbf{x}_i and \mathbf{y}_i variables and the equilibrium constraints might be handled, and this is discussed in more detail later in this section. We have chosen to have the \mathbf{x}_i , $i = 0, \dots, N$ as independent variables, and we use equations (2.4) to determine \mathbf{y}_i , $i = 0, \dots, N$, as dependent variables. The physical condition that the condenser does not affect the material composition may be stated as

$$\mathbf{x}_{N+1} = \mathbf{y}_N \quad (2.17)$$

and enables us to express \mathbf{x}_{N+1} as a dependent variable. As the total flows L_i and V_i are available, we can now determine the componentwise flows by using $\mathbf{l}_i = L_i \mathbf{x}_i$ and $\mathbf{v}_i = V_i \mathbf{y}_i$.

We are now in a position to state componentwise material balance equations for within the column. By equating the net flow between stages with the output flow, we deduce the equations

$$\mathbf{v}_i = \mathbf{l}_{i+1} + \mathbf{d} \quad i = S, \dots, N-1 \quad (2.18)$$

and

$$\mathbf{l}_i = \mathbf{v}_{i-1} + \mathbf{b} \quad i = 1, \dots, S. \quad (2.19)$$

In fact, equation (2.18) also holds for $i = N$ but this is already implied by (2.17) and (2.15).

In a similar way we can also state the heat balance equations. We choose Q^r to be an independent variable in the model. We can calculate the feed and output flows \mathbf{f} , \mathbf{d} and \mathbf{b} as in (2.1), and hence the corresponding heat flows can be calculated from (2.10), (2.11) and (2.12). An overall heat balance is achieved simply by defining the condenser heat duty Q^c by

$$Q^c = Q^r + H^F - H^D - H^B. \quad (2.20)$$

Heat balance within the column is obtained by equating the net heat flow between stages with the heat output at the top (or heat input at the bottom) of the column. This gives the equations

$$H_i^V - H_{i+1}^L = Q^c + H^D \quad i = S, \dots, N \quad (2.21)$$

and

$$H_i^V - H_{i+1}^L = Q^r - H^B \quad i = 0, \dots, S-1. \quad (2.22)$$

We have already remarked that the variables \mathbf{x}_i and \mathbf{y}_i need to satisfy the normalization conditions (2.2) and (2.3). In fact it is necessary to enforce only one set of these conditions. This can be seen by summing over the components in equations (2.18) and (2.19), and using (2.15) and (2.16). Imposing (2.2) as the normalization provides equations that are linear, which might be used to advantage by the nonlinear programming solver. In practice however, imposing (2.3) explicitly seems to provide somewhat better performance. Thus the conditions

$$\sum_{j=1}^m y_{ij} = 1 \quad i = 0, \dots, N. \quad (2.23)$$

are included in the model.

This completes the definition of the nonlinear equations used by the model. To these we add lower and possibly upper bounds on the independent variables in a suitable way. An AMPL model is given in the Appendix which precisely describes the definition of the independent and dependent (defined) variables, the constraints of the model, and the lower and upper bounds that are specified. There are no other inequality constraints that need be added to determine the solution. However we have found it useful to add the monotonic temperature condition

$$T_i \geq T_{i+1} \quad i = 0, \dots, N \quad (2.24)$$

which helps the nonlinear programming solver to avoid temperature estimates which are not physically acceptable. Much of the subsequent discussion in the paper proceeds on the assumption that none of the bounds or inequality constraints are active at a solution to the column model, which is usually the case in practice.

It is a useful exercise to count the number of degrees of freedom inherent in the column model. We shall assume that the column pressure P and the variables N and S , describing the column configuration, are fixed. Of course the effect of varying these quantities is also of interest, but here we concentrate on the more simple case. Indeed, N and S are integer variables and cannot be varied unless we extend the formulation to one requiring solution by mixed integer nonlinear programming. The independent variables of the model are the split parameter θ , the reboiler heat duty Q^r , the temperatures T_0, \dots, T_{N+1} , the flows V_0, \dots, V_{S-1} and L_{S+1}, \dots, L_{N+1} , and the fractional compositions $\mathbf{x}_0, \dots, \mathbf{x}_N$. This gives a total of $mN + m + 2N + 5$ variables. Any other quantities that arise can be defined as functions of these independent variables. The nonlinear equations which remain, and which we impose on the model, are the $N + 1$ normalization conditions (2.23), the mN material balance equations (2.18) and (2.19), the $N + 1$ heat balance equations (2.21) and (2.22), the m equations (2.14) arising from overall material balance, and the single equation (2.6) relating to the condenser outflow, giving a total of $mN + m + 2N + 3$ independent equations. Thus **the distillation column model has 2 degrees of freedom.**

This freedom may be taken up in various ways: for example it is usually desirable to minimize the operating cost of the column, of which the reboiler heat duty Q^r is a major

item, whilst imposing constraints on the amount of impurity in the tops and bottoms streams.

An important property possessed by any solution to the constraints of the model is that it is *scalable* in the sense that if the feed flow F is changed by some factor, then an equivalent solution may be obtained by changing the flows L_i , V_i and Q^r by the same factor, and leaving the remaining independent variables θ , T_i and \mathbf{x}_i unchanged.

A major difficulty with the solution of distillation column models is that of inducing convergence in the nonlinear programming solver. Particularly it is valuable to be able to obtain estimates which satisfy the constraints of the model to good accuracy. If poor estimates are generated during the solution process then the linearized model used by a solver can be unsatisfactory, which can lead to failure to converge, or convergence to a point at which the constraint linearizations are inconsistent. It is not too difficult to provide reasonable estimates for some of the independent variables, for example the T_i and the \mathbf{x}_i , but the amount of flow within the column (L_i and V_i) is not easy to estimate, if a particular purity is required. For a fixed column configuration, if the flows increased without limit, the product purities asymptotically approach limits which depend on the number of plates above and below the feed and the relative volatilities of the key components. In the rest of the paper we look at means whereby good initial estimates for all independent variables might be obtained.

Before passing on to this topic, we note that our choice of independent variables is but one of a number of alternative possibilities for defining the model. The approach that we have used is to some extent motivated by the ideas described in Sections 3 and 4, but is also guided by practical experience with a variety of possible choices. We describe some of the alternative possibilities that we have investigated in Section 6.

3 The Zero-Feed Column

The new ideas for column initialization in this paper have come about by considering what happens to the column as $Q^r \rightarrow \infty$, the case of infinite reflux ratio. By scalability, taking this limit is essentially equivalent to fixing Q^r and decreasing the feed flow rate F to zero. Fixing Q^r removes one degree of freedom, and we shall remove the other degree of freedom by adding a single impurity constraint. The form we have chosen for the impurity constraint is

$$\frac{x_{0,lk}}{x_{0,hk}} = \frac{x_{N+1,hk}}{x_{N+1,lk}} \quad (3.1)$$

where lk and hk index the light key and heavy key components respectively. Since the heavy key component is the major source of impurity in the tops stream, and likewise for the light key component and the bottoms stream, this constraint essentially equates the fractional impurity in the light and heavy components. Of course one might equally argue the case for a different measure of impurity. However, the main point is that the exact level of impurity is not specified and this constraint achieves a symmetry between

the relative key compositions in the tops and bottoms products. Increasing the heat rate tends to increase the sharpness of the split, but as Fenske's equation makes clear there is a limiting product impurity at infinite reflux which prevents a sharp split. This limiting impurity can be reduced by increasing the number of plates, N .

Adding the impurity constraint and fixing Q^r , transforms the general column model of Section 2 into a well-determined system of $nM + m + 2N + 4$ variables and constraints, with zero degrees of freedom.

We now consider the special case of this system in which $F = 0$ (the *Zero-Feed Column*). This causes considerable simplifications in the nonlinear system. First of all the overall material balance for the column is trivially satisfied by $\mathbf{f} = \mathbf{d} = \mathbf{b} = \mathbf{0}$ and so we need not include the variable θ . Also the vapour and liquid material flow between the plates must balance, so it follows that

$$V_i = L_{i+1} \quad \text{and} \quad \mathbf{y}_i = \mathbf{x}_{i+1}, \quad i = 0, \dots, N. \quad (3.2)$$

We use these equations to eliminate the V_i and \mathbf{y}_i variables. In regard to heat flow, there is a constant net heat flow Q^r upwards between each stage. Thus the heat balance equations may be expressed as

$$V_i q_i = Q^r \quad i = 0, \dots, N \quad (3.3)$$

where

$$q_i = \sum_{j=1}^m x_{ij} ((\beta_j + \beta'_j T_i + \beta''_j T_i^2) - (a_j + a'_j T_{i+1} + a''_j T_{i+1}^2)). \quad (3.4)$$

Thus, once the T_i and \mathbf{x}_i are known, equations (3.4) and (3.3) enable the flows V_i and hence L_i to be determined. Thus we are left with a problem in which only the variables T_i and \mathbf{x}_i are present.

The equilibrium conditions (2.7) become

$$\mathbf{x}_{i+1} = \mathbf{K}_i \mathbf{x}_i \quad i = 0, \dots, N. \quad (3.5)$$

Given the T_i and hence \mathbf{K}_i , these are simply recurrence relations from which the vectors \mathbf{x}_i may be calculated. Thus we can use (3.5) to eliminate all but m of the x_{ij} variables. However there is an aspect of numerical stability to consider here, and it is important to run the recurrence relations **up** the column for the heavy components (as $x_{i,j} = K_{i-1,j} x_{i-1,j}$ for increasing i), and **down** the column (as $x_{i,j} = x_{i+1,j} / K_{i+1,j}$ for decreasing i) for light components. This ensures that the components are exponentially decreasing and avoids any difficulty due to exponential growth. (Such ideas are familiar from the old Lewis-Matheson stage-by-stage design procedure for multicomponent columns (see King [4], p 450).)

The resulting problem has $N + 2$ variables T_i and m variables $x_{i,j}$ (these are the $x_{0,j}$ for the heavy components and $x_{N+1,j}$ for the light components with which the recurrences

are initiated). The constraints remaining are the $N + 2$ normalization equations (2.2) for $i = 0, \dots, N + 1$, the saturation condition (2.6) and the impurity condition (3.1).

Counting variables and equations we see that this system has $m - 2$ degrees of freedom. Yet we started out with a general system with zero degrees of freedom. When this issue first arose, the explanation for this mismatch was by no means obvious, and the means of resolving it even less so, especially with the modelling equations that were in use at the time. The resolution of this issue is described in the next section.

4 The Limiting Zero–Feed Column (LZFC)

The answer to the questions posed above is that the zero–feed column does indeed have $m - 2$ degrees of freedom. However it does not quite answer the question posed in Section 2 as to what is the **limiting** solution of the general column equations as $F \rightarrow 0$. In this case, although the overall material balance equation $\mathbf{f} = \mathbf{d} + \mathbf{b}$ is trivial in the limit, with zero on both sides, the equation

$$\mathbf{x}^F = \theta \mathbf{x}_{N+1} + (1 - \theta) \mathbf{x}_0 \quad (4.1)$$

obtained by dividing out F is still valid in the limit. This is obvious in retrospect, having seen this equation in Section 2, but it was certainly not obvious when this study began. In fact it was the resolution of this issue that led us to model the general column in the way described in Section 2. The equation has good numerical properties for use in the general column model. It is well scaled (with terms of order unity), and it also enables the general column model to be solved reliably for very small values of F , thus permitting the effective use of homotopies for the general case, starting from $F = 0$. Furthermore we see why the reflux ratio would be an unsatisfactory choice of independent variable, since it goes to infinity as $F \rightarrow 0$.

As hoped, equation (4.1) solves the problem of the extra degrees of freedom. It adds an extra variable θ to the problem and m extra constraints. However one of the product stream normalization constraints becomes redundant, as it is implied by summing over the components in (4.1). It probably does not matter which of these normalizations is dropped. To preserve symmetry, we have dropped both the product stream normalizations. Instead we retain (2.2) for $i = 1, \dots, N$, and impose the extra condition

$$\sum_{j=1}^m (x_{N+1,j} - x_{0,j}) = 0. \quad (4.2)$$

This gives a well-determined nonlinear system with $m + N + 3$ equations and variables.

There are many advantages of using the solution of an LZFC problem as a preliminary to solving a general column problem. We itemize these as follows and expand on the points in the subsequent text.

- The LZFC problem is much smaller than the general column problem.

- The LZFC problem is much easier to initialize and converge.
- The solution of the LZFC provides estimates of the minimum impurity that can be obtained for any given number of plates.
- The LZFC solution provides good estimates of T_i , \mathbf{x}_i and the flows L_i and V_i , with which to initialize the general case.
- The LZFC solution may be used to initialize a homotopy approach by which to solve the general case.
- Good results have been obtained on a variety of flowsheets involving linked distillation columns, with and without heat integration.

The most obvious advantage of solving the LZFC problem is that it is much smaller than the general column problem, having $m + N + 3$ variables as against $mN + m + 2N + 5$ variables. The LZFC problem does not involve the nonlinear heat balance constraints, and the flow variables L_i and V_i , whose values are not easily estimated. This avoids some of the difficulty inherent in the general problem. Moreover good estimates of T_i and the required \mathbf{x}_i variables are readily provided for the LZFC.

Very good estimates of θ and the required \mathbf{x}_i variables are readily obtained by assuming that the split is sharp. Thus we set

$$\theta = \sum_1^{lk} x_j^F, \quad (4.3)$$

that is the sum of light component mole fractions in the feed, (or the fractional tops stream recovery assuming a perfectly sharp split). Then for a light component we set

$$x_{N+1,j} = x_j^F / \theta \quad (4.4)$$

and likewise for a heavy component

$$x_{0,j} = x_j^F / (1 - \theta). \quad (4.5)$$

These settings are close to being normalized (equation (4.2)), once the remaining $x_{N+1,j}$ and $x_{0,j}$ variables are determined from the recurrence relations. The extreme temperatures T_0 and T_{N+1} can now be accurately estimated by solving the nonlinear equations

$$\sum_{j=hk}^m K_{0,j} x_{0,j} = 1 \quad \text{and} \quad \sum_{j=1}^{lk} K_{N+1,j} x_{N+1,j} = 1. \quad (4.6)$$

For T_0 , it is sufficient just to tabulate $\sum K_{0,j} x_{0,j}$ for a likely range of temperature and then use linear interpolation, and similarly for T_{N+1} . The effect on T_0 and T_{N+1} of varying the column pressure P can also be investigated at this juncture. Intermediate

values of T_i can be estimated by some interpolation procedure. We have found that linear interpolation is quite adequate. The resulting LZFC problems are typically solved from this initial approximation in about 6 or 7 iterations of our NLP solver (effectively implementing some version of the Newton–Raphson method).

As in equation (3.1), we have measured the impurity of the LZFC solution by looking at the ratio of the fractions of the light key and heavy key components in the \mathbf{x}_0 and \mathbf{x}_{N+1} variables. The function of the impurity constraint has been to equate the relative impurity of the tops and bottoms streams. The actual level of impurity given by the LZFC solution is a lower bound on the impurity that can be obtained in the general column with finite reflux. If we are interested in obtaining a particular level of impurity (say 1%) in the solution of the general column, then we might try solving the LZFC for increasing values of N until this level of impurity is attained. In fact it is important to look for a lower level of impurity in the LZFC, so that the value of Q^r in the general case is not too large. (Recall that the LZFC solution essentially corresponds to $Q^r = \infty$ when $F > 0$).

Of course the variation of N can also be investigated at a later stage of the process but this procedure provides a useful way of obtaining reasonable estimates. In particular it provides estimates for which the resulting general NLP problem has a feasible solution. This is very important in that many NLP solvers perform badly when faced with an infeasible problem.

5 Solution of Columns with Finite Reflux

One main purpose of the LZFC solution is to provide good estimates of the variables for the general column problem, with non-zero feed and finite reflux ratio. In this respect, we shall consider the form of the model having two degrees of freedom (or three degrees of freedom if P is included – see Section 2). Thus F is fixed and Q^r becomes an independent variable. Because of scalability, it suffices to take $F = 1$ and this also ensures that the material balance constraints are well scaled. Exactly what objective function is selected is a matter of choice. So far we have considered simply minimizing Q^r which is related to both column operating and capital costs (through steam requirements and column diameter needed to accommodate the vapour flow without flooding, respectively). This ignores the effect of column height (number of plates) on capital cost but then in our model N is fixed in any case. We have fixed a particular level of impurity, typically 1%, and imposed the inequality constraints

$$x_{0,lk} \leq 0.01 x_{0,hk} \quad \text{and} \quad x_{N+1,hk} \leq 0.01 x_{N+1,lk} \quad (5.1)$$

on the problem. These are usually both active at the solution as would be expected.

It has been observed that the T_i and \mathbf{x}_i from the LZFC problem are usually close to the values for the general problem. This is because even quite large relative changes in small impurity levels have little effect on the mole fractions of abundant components

(light above and heavy below the feed) and hence on bubble point temperatures. An initial value of Q^r has to be provided, and it is important to choose units of heat flow so that the numerical value of Q^r (and hence of the H_i^L and H_i^V) is around unity. This ensures that the heat balance equations are well-scaled. Given Q^r , we can use the LZFC solution, together with equations (3.4), (3.3) and (3.2), to initialize the flows L_i and V_i that are required in the general case. A feature of this procedure is that the initial point for the general column is usually reasonably close to feasibility. The main issue is then that of how to provide a good estimate of Q^r . It may be possible to do this on the basis of *a priori* knowledge about the column and where possible we aim to do this. If the effective relative volatilities of the components are known then the Underwood equations (King [4], p.420) can be solved to obtain the minimum reflux ratio, R_m . If we suppose the actual reflux ratio in our optimised column is somewhat greater than this (say by a factor 1.25 as in a well-known rule of thumb (Peters and Timmerhaus [5] p.371), and the feed is saturated liquid, then the vapour boilup can be estimated as

$$V_0 = (1.25R_m + 1)\theta F$$

This value should be compared with the solution of the LZFC problem, and Q^r and all internal column flows scaled accordingly. As an alternative to direct use of the thermodynamic package incorporated in the model, the effective relative volatility of the keys (α_K) can be calculated from the Fenske equation rearranged as

$$\alpha_K = \left(\frac{x_{0,lk}x_{N+1,hk}}{x_{0,hk}x_{N+1,lk}} \right)^{1/N_{ZF}}$$

where N_{ZF} is the number of plates in the LZFC problem.

Alternatively, if a good estimate of Q^r cannot be made (e.g. if the liquid mixture were non-ideal), then a homotopy approach to finding the solution can be very effective and is outlined in the next paragraph.

In the homotopy approach, we fix the value of $Q^r = 1$ and impose the impurity constraint (3.1) in place of (5.1). We have seen in Section 2 that this gives rise to a well-determined system of nonlinear equations. The LZFC solution is the solution of this problem for $F = 0$. We now solve the nonlinear system a number of times, for a sequence of values of F , starting from as close to $F = 0$ as is necessary to converge the system. At each stage we monitor the impurity in the output streams and adjust the value of F with the aim of converging to the required level of impurity (1% say). Each stage of the homotopy is solved using the best previous solution as an initial guess. Our experience is that each system solves very quickly, typically in as few as 2 or 3 iterations. Once the desired level of impurity is obtained, the flows (including Q^r) are rescaled to correspond to the actual feed flow F that is given. This approach can be automated using a path following technique. However we have found that the informal approach works very well and the AMPL language readily permits the manual adjustment and repeated solution of the nonlinear system in the manner described.

We have successfully solved a variety of columns using these ideas, not only as individual columns, but also within a flowsheet of linked columns for the purpose of separation. We have also tackled more complicated optimization problems involving heat integration between the columns. We have successfully solved all the problems described by Fraga and McKinnon [3] (see the flowsheets in Figure 3, (a) – (d) of [3], both for the fixed pressures given there, and also with the optimum variation of pressures. These are quite large problems having 4 columns and about 1000 – 2000 independent variables and constraints, depending on the numbers of plates used. Our experience is that an NLP solver can easily run into difficulty with such problems and either fail to converge, or converge to an infeasible solution. Our method of initialization provides estimates that are close to feasibility, and we have found it advantageous to maintain near-feasibility in the estimates generated by the NLP solver. Our filter-type SQP solver permits this to be done by imposing an upper bound on the amount of constraint infeasibility.

6 Alternative Column Models

In this section, we describe some alternative possibilities to that in Section 2 for defining the the general column model, and we compare their performance on some large flowsheets with linked columns. An obvious possibility is to include both the \mathbf{x}_i and \mathbf{y}_i as independent variables, and to impose the equilibrium constraints (2.4) explicitly. This increases the dominant term in the variable count from mN to $2mN$ and so approximately doubles the size of problem. Although there may be some advantage to be gained from having the extra variables present, it does put up some of the housekeeping costs associated with the nonlinear solver. Another interesting possibility is to designate $\mathbf{v}_0, \dots, \mathbf{v}_{S-1}$ and $\mathbf{l}_{S+1}, \dots, \mathbf{l}_{N+1}$ as the independent variables, in place of the L_i, V_i and \mathbf{x}_i . The information content of this representation is essentially the same. This cuts down the number of independent variables by about N , and can be thought of as a null-space parametrization of the material balance equations (2.18) and (2.19). In this case the equilibrium constraints (2.4) need to be explicitly included. In practice no significant gain for this version has been obtained, no doubt because the dominant term in the number of variables is still mN .

In Section 3 it is pointed out that for the infinite reflux column the dominant term mN can be removed by using recurrence relations to generate all but m of the variables x_{ij} . The same idea can be used for the general column, following the approach behind the McCabe-Thiele construction for binary column design, although the recurrence relations are considerably more complicated. The material balance equations (2.18), (2.19) and the equilibrium equations (2.4) are rearranged in a suitable way. As before, for a light component, we carry out the recurrence **down** the column, in which case the appropriate expressions are

$$x_{ij} = (L_{i+1}x_{i+1,j} + d_j)/(K_{ij}V_i) \quad i = N, \dots, S \quad (6.1)$$

and

$$x_{ij} = (L_{i+1}x_{i+1,j} - b_j)/(K_{ij}V_i) \quad i = S - 1, \dots, 1. \quad (6.2)$$

For a heavy component, the recurrence is carried out **up** the column, and the expressions are

$$x_{ij} = (V_{i-1}K_{i-1,j}x_{i-1,j} + b_j)/L_i \quad i = 1, \dots, S \quad (6.3)$$

and

$$x_{ij} = (V_{i-1}K_{i-1,j}x_{i-1,j} - d_j)/L_i \quad i = S + 1, \dots, N. \quad (6.4)$$

There is also a complication that is not present in Section 3. The recurrences involve \mathbf{d} and \mathbf{b} , and these are defined by $\mathbf{d} = D\mathbf{x}_{N+1}$ and $\mathbf{b} = B\mathbf{x}_0$. Thus all the components of \mathbf{x}_{N+1} and \mathbf{x}_0 are needed to initiate these recurrences. (In Section 3, only the light components of \mathbf{x}_{N+1} and heavy components of \mathbf{x}_0 are needed.) Thus both \mathbf{x}_{N+1} and \mathbf{x}_0 become independent variables in the model, and there are m extra constraints which equate the variables $x_{0,j}$ (for a light component) or $x_{N+1,j}$ (for a heavy component) to the values obtained on the last stage of the recurrences. Thus the independent variables appropriate to this formulation are the split parameter θ , the reboiler heat duty Q^r , the temperatures T_0, \dots, T_{N+1} , the flows V_0, \dots, V_{S-1} and L_{S+1}, \dots, L_{N+1} , and the fractional compositions \mathbf{x}_0 and \mathbf{x}_{N+1} . This gives a total of $2m + 2N + 5$ independent variables.

In Table 1 we show some numerical experience with these different formulations when applied to two instances of a linked column problem, starting from the same initial point. The heading $0mN$ relates to the formulation just described, in which the dominant mN term is absent, whilst $1mN$ relates to the formulation of Section 2, and $2mN$ relates to the extended formulation in which both x_{ij} and y_{ij} are independent variables. The first instance is that of column sequences (b), (c) and (d) in [3] but with no heat integration and the pressures fixed as in their solution. The second instance is similar, but allows the column pressures to vary subject to minimum temperature constraints on tops products (to allow water or air cooling) and a minimum pressure of one atmosphere.

formulation	# variables	# iterations	time (sec.)	sec/iteration
0mN	438	26	1209	46.5
1mN	893	29	215	7.4
2mN	1488	30	548	18.3
0mN	442	51	2397	47.0
1mN	897	61	483	7.9
2mN	1492	46	812	17.7

Table 1. Comparison of different column formulations

The different formulations all find the same solution and the number of iterations taken does not vary greatly within each problem instance. However the formulation of Section 2 is clearly superior on time per iteration, and this supports our preference for this approach. The reason for the poor performance of the $0mN$ formulation seems likely to be that the considerable increase in complexity of (6.1) through (6.4) causes a decrease in the sparsity of the Jacobian matrix, and hence an increase in the time taken to solve the QP problems in the SQP method. On the other hand, the extended formulation $2mN$ does not seem to provide any useful gains in sparsity to offset the increased size of the NLP problem.

7 Acknowledgements

We gratefully acknowledge the valuable contributions of Drs. David Gay and Sven Leyffer in regard to integrating the AMPL system and our NLP solver.

8 References

- [1] Fletcher R. and S. Leyffer, Nonlinear Programming Without a Penalty Function, Dundee Numerical Analysis Report NA/171 (1997), submitted to Mathematical Programming.
- [2] Fourer R., D.M. Gay and B.W. Kernighan, AMPL, A Modelling Language for Mathematical Programming, Boyd and Fraser Publ. Co., Danvers, MA (1993).
- [3] Fraga E.S. and K.I.M. McKinnon CHiPS: A Process Synthesis Package *Chemical Engineering Research and Design*, **72**(A3), 389-394 (1994).
- [4] King, C.J., *Separation Processes (2nd edition)*, McGraw-Hill, New York (1980).
- [5] Peters M.S and K.D. Timmerhaus, *Plant Design and Economics for Chemical Engineers (4th edition)*, McGraw-Hill, New York (1991).
- [6] Sinnott, R.K., *Chemical Engineering, Volume 6: Design (2nd edition revised)*, Pergamon Press, Oxford (1996).

9 Appendix

```

# column.mod
# Original AMPL coding by Roger Fletcher

# Program to find a solution of the distillation column equations.

# Components are given in order of decreasing volatility.

set COMPONENTS ordered;    param H_KEY symbolic in COMPONENTS; # heavy key

# thermopack coefficients

param Aant {COMPONENTS}; param Bant {COMPONENTS}; param Cant {COMPONENTS};
param alp0 {COMPONENTS}; param alp1 {COMPONENTS}; param alp2 {COMPONENTS};
param bet0 {COMPONENTS}; param bet1 {COMPONENTS}; param bet2 {COMPONENTS};

# fixed column parameters

param N integer > 2;           # number of trays
param S integer > 1, < N;     # feed tray
param P > 0;                   # column pressure
param xF {COMPONENTS} >= 0;   # feed mole fractions
param F >= 0;                 # total feed flow
param TF;                     # feed temperature
param T_lb;    param T_ub > T_lb; # stage temperature bounds

param f {j in COMPONENTS} := F*xF[j]; # feed flows

# independent variables

var Qr >= 0;                   # heat duty at reboiler
var theta >= 0, <= 1;         # tops/bottoms split
var T {0..N+1} >= T_lb, <= T_ub; # stage temperatures
var z {0..N} >= 0;           # total flow towards the feed
var x {0..N, COMPONENTS} >= 0; # liquid compositions

# NB: for i < S, z[i] is a vapour flow, and for i >= S, z[i] is a liquid flow

# defined (dependent) variables

var K {i in 0..N+1, j in COMPONENTS} =
    exp(Aant[j]+Bant[j]/(Cant[j]+T[i]))/P; # equilibrium constants
var y {i in 0..N, j in COMPONENTS} = K[i,j]*x[i,j];

```

```

# vapour compositions

var D = theta*F;
var d {j in COMPONENTS} = D*y[N,j]; # tops flows
var B = (1-theta)*F;
var b {j in COMPONENTS} = B*x[0,j]; # bottoms flows

var L {i in 1..N+1} = if i<=S then z[i-1]+B else z[i-1]; # total liquid flows
var V {i in 0..N} = if i<S then z[i] else z[i]+D; # total vapour flows

var l {i in 1..N+1, j in COMPONENTS} = # componentwise liquid flows
    if i>N then L[N+1]*y[N,j] else L[i]*x[i,j];
var v {i in 0..N, j in COMPONENTS} = V[i]*y[i,j]; # componentwise vapour flows

var HL {i in 1..N+1} = 1e-6*sum {j in COMPONENTS}
    l[i,j]*(alp0[j]+T[i]*(alp1[j]+alp2[j]*T[i])); # liquid heat flows
var HV {i in 0..N} = 1e-6*sum {j in COMPONENTS}
    v[i,j]*(bet0[j]+T[i]*(bet1[j]+bet2[j]*T[i])); # vapour heat flows

param HF := 1e-6*sum {j in COMPONENTS}
    f[j]*(alp0[j]+TF*(alp1[j]+alp2[j]*TF)); # feed heat flow
var HD = 1e-6*sum {j in COMPONENTS}
    d[j]*(alp0[j]+T[N+1]*(alp1[j]+alp2[j]*T[N+1])); # tops heat flow
var HB = 1e-6*sum {j in COMPONENTS}
    b[j]*(alp0[j]+T[0]*(alp1[j]+alp2[j]*T[0])); # bottoms heat flow
var Qc = Qr + HF - HD - HB; # cooler duty

# normalization equations

subject to nm1 {i in 0..N}: sum {j in COMPONENTS} y[i,j] = 1;

# material balance equations

subject to mb1 {i in 0..S-1, j in COMPONENTS}: l[i+1,j] = v[i,j] + b[j];
subject to mb2 {i in S..N-1, j in COMPONENTS}: v[i,j] = l[i+1,j] + d[j];

# heat balance constraints

subject to hb1 {i in 0..S-1}: HV[i] + HB = HL[i+1] + Qr;
subject to hb2 {i in S..N}: HV[i] = HL[i+1] + Qc + HD;

# condenser recycle is saturated liquid

```

```
subject to crs : sum {j in COMPONENTS} y[N,j]*K[N+1,j]=1;

# consistency of feed and tops/bottoms outflow

subject to ctb {j in COMPONENTS}: xF[j] = theta*y[N,j] + (1-theta)*x[0,j];

# monotonic temperatures

subject to mt1 {i in 0..N}: T[i+1] <= T[i];

# purity condition

subject to pr1: x[0,prev(H_KEY,COMPONENTS)] <= 0.01*x[0,H_KEY];
subject to pr2: y[N,H_KEY] <= 0.01*y[N,prev(H_KEY,COMPONENTS)];

# objective function

minimize cost: Qr;
```