Proceedings of
11th European Conference on Turbomachinery Fluid dynamics & Thermodynamics
ETC11, March 23-27, 2015, Madrid, Spain

An Object Oriented Sequential Network Modelling Approach

Oscar Córdoba Muñoz - Inmaculada de Castro García

ITP, Madrid, Spain
oscar.cordoba@itp.es
inmaculada.castro@itp.es

ABSTRACT

This document is the result of the experience acquired creating a tool for Gas Turbines Performance and Fluid Systems modelling but most of the issues encountered are applicable to different networks problems (electrical lines, pipe routing or control). The text compiles mathematical, numerical and methodological issues encountered in the process.

In this work an object oriented modelling approach is followed to simulate a network. Once a topology is created, the next step is to establish the most appropriate set of algebraic equations to characterise the problem. This process comprises the proper selection of equations, unknowns and sets of initial conditions that will allow the correct resolution of the system using a Newton Raphson algorithm, trying to avoid Jacobian matrix singularities. When this is not possible, instead of changing the solution method, other alternatives, as a multilevel strategy, are used.

NOMENCLATURE

A	Geometric Area
C_d	Discharge coefficient
$\bar{cp}$	Mean specific heat capacity at constant pressure
ΔT	Temperature Difference $\Delta T = T_{outlet} - T_{inlet}$
FP	Flow parameter
M	Mach number
P	Static pressure
Pt	Total pressure
PR	Pressure Ratio $PR = \frac{P_{inlet}}{P_{outlet}}$
Q	Heat
Tt	Total temperature
TR	Temperature Ratio $TR = \frac{T_{outlet}}{T_{inlet}}$
Re	Reynolds number
SAS	Secondary Air System
W	Mass flow
γ	Ratio of specific heat capacities

INTRODUCTION

The main types of fluid systems found in a Gas Turbine are Oil, Fuel and Secondary Air Systems. These systems are connected to the main gas path. The main gas path discipline is known as Gas Turbine Performance. Secondary Air Systems will be used as example to illustrate the main content of this work. A Gas Turbine Secondary Air System model is usually understood as a model that represents the elements apart from the main gas path where the air passes usually for sealing and cooling tasks. The different elements encountered in this sort of systems are pipes, orifices, nozzles, etc. and they are represented as components. The different components are linked representing the different ways of the flow, resulting in a network, with a defined topology. These models are commonly referred as 0D since there is no continuous characterisation through the fluid lines. Air system description is only available in stations before and after the components. The biggest challenges in the Air Systems, compared to Gas Turbine Performance models, are choking and reverse flow phenomena. In the Gas Turbine Performance area, banks of initial guesses may be employed to converge

OPEN ACCESS
Downloaded from www.euroturbo.eu

Copyright © by the Authors

towards the solution, Reference 4. This strategy cannot be used in Secondary Air Systems where even the flow direction is unknown.

Once a network topology is created, the next step is to establish the physical equations, mass conservation, momentum and energy conservation, for the individual components, coupling and boundary conditions. Everything in the document is related to steady state although it is extensive to transient simulations. The mathematical model associated to this sort of physical problems in steady state is a system of nonlinear equations. The main targets are:

- To obtain which is the most appropriate set of equations and unknowns to be solved.
- To obtain which is the best set of initial conditions.
- To come up with the mathematical difficulties encountered applying the solving method.

The solution method for the algebraic nonlinear equations to be considered is Newton Raphson. If a set of equations is solved with Newton Raphson method, it will be considered robust enough. $\vec{F}\left(\vec{x}\right) = 0$ is the system to solve. $J = \nabla\vec{F}\Big|_{\vec{x}}$ is called Jacobian matrix at $\overrightarrow{x}$. Three main causes for method failure will be listed:

- The value $\vec{x}_n$ obtained during the iterative process is out of an expected range. It may lead to negative temperatures or pressures for instance.
- The inverse J^{-1} does not exist because the Jacobian matrix is singular. A local minimum or maximum is reached. A division by zero appears calculating the Jacobian matrix.
- The value J^{-1} does not exist because there is an infinite in the Jacobian matrix.

The three cases mentioned above may be overcome with an appropriate solver but other alternatives, as multilevel strategy or proper selection of tearing variables, will be discussed in the sequel.

Although there are commercial tools able to solve these problems, an in house development allows customisations and the control of the physical methodology used. The selected approach enables unique models with components from different disciplines, such as Performance, Oil or Air Systems, as well as tailored features not existing in the market at the beginning this work.

In the next sections the different alternatives to approach network models will be discussed. Based on the Secondary Air Systems problem, some issues found using the selected approach will be presented. Finally, according to the exposed examples, the main guidelines followed to address the resultant systems of nonlinear equations will be summarised.

EQUATION ORIENTED VERSUS SEQUENTIAL APPROACHES

An equation oriented approach creates an overall nonlinear algebraic system of equations to be computed globally. This approach is widely discussed in Reference 3. On the other hand, a sequential solver computes each network component getting outputs for given inputs, solving nested equations if necessary. Each nested system of equations must decrease the solving tolerance; otherwise a deeply nested process may conflict with the computer rounding error. If the nested systems are not accurately solved, the outer system may observe that for the same unknown inputs could obtain different residue values finding the solution.

The main idea is that there are better and worse approaches for specific problems rather than discussing the better approach. It is possible to find examples where one approach is better than the other. In Figure 1 the equation oriented approach fails using the Newton Raphson method for the initial values that make singular the Jacobian of the system of equations. In this example, the sequential approach never fails. In Figure 2 the equation oriented approach never fails and there are initial values with singular Jacobian in the sequential approach. Each specific problem requires a study of the best way to be performed.

The main advantages of an equation oriented approach are:

- Reduce problems with numerical tolerances. All the equations are solved at the same time with the same precision.
- The need of successive passes through the network and nested convergence at the unit operation level, make the sequential approach expensive in some cases.

On the other hand the advantages of sequential approaches are:

- Higher robustness may be implemented if inner iterations are properly done. There are some cases where some equations added to the overall system of equations may lead to convergence problems.
- Better debugging capabilities. The physical relations coded are not merged to be solved all together, as in the equation oriented approach. The execution is performed in the same sequence it is coded.

It is important the capability to decide which variables must be considered as unknowns in an outer iteration. It must be

System of equations
$x^2+y-z^2-1=0$ $x+y-1=0$ $y+z-1=0$

Equation Oriented	**Sequential**
No nested equations	Nested equations $x+y-1=0$ $y+z-1=0$
Global unknowns x y z	Global unknown y
Global system $x^2+y-z^2-1=0$ $x+y-1=0$ $y+z-1=0$	Global equation $y-1=0$
Jacobian $J=\begin{bmatrix}2x & 1 & -2z\\ 1 & 1 & 0\\ 0 & 1 & 1\end{bmatrix}$	$Global\,Jacobian\,J_y=1$ $Nested\,Jacobian\,J_x=1$ $Nested\,Jacobian\,J_z=1$
Singular in the plane $2x-2z-1=0$	No singular

Figure 1: Equation Oriented failure example

System of equations
$x+y^2-z-1=0$ $x+y^2-1=0$ $y+z^2-1=0$

Equation Oriented	**Sequential**
No nested equations	Nested equations $x+y^2-1=0$ $y+z^2-1=0$
Global unknown x y z	Global unknown y
Global System $x+y^2-z-1=0$ $x+y^2-1=0$ $y+z^2-1=0$	Global equation $-\sqrt{1-y}=0$
Jacobian $J=\begin{bmatrix}1 & 2y & -1\\ 1 & 2y & 0\\ 0 & 1 & 2z\end{bmatrix}$	$Global\,Jacobian\,J_y=\frac{1}{2\sqrt{1-y}}$ $Nested\,Jacobian\,J_x=1$ $Nested\,Jacobian\,J_z=2z$
No singular	Singular in $y=1$ and $z=0$

Figure 2: Sequential failure example

also possible to decide which equations must be solved internally each time that the outer system iterates. In this work the sequential approach has been selected but if necessary, all the equations may be computed globally rather than in the inner loops. At the time of designing a simulation approach, the next is to be decided first:

- Which are the equations to be solved in the outer and nested iterations. In following sections nested iterations utility and applicability will be showed.
- Which are the unknowns to be solved for the proposed equations. Some modelling and mathematical issues must be considered before deciding the iteration scheme.

OBJECT ORIENTED AND NODE ORIENTED MODELLING

In the node oriented philosophy a fluid line is delimited by nodes. Nodes are the points where links may join or separate. The way to model with this approach is to decide on nodes, join them with links, and afterwards insert the components in the middle of these links. Links acquire importance in contrast to components which are merely obstacles in the flow path. This is only possible when components except nodes are single input - single output. On the other hand, in the object oriented approach, the components acquire a major importance. In the object oriented approach, nodes are another kind of components and the links join components; they do not join node to node. A node oriented approach could

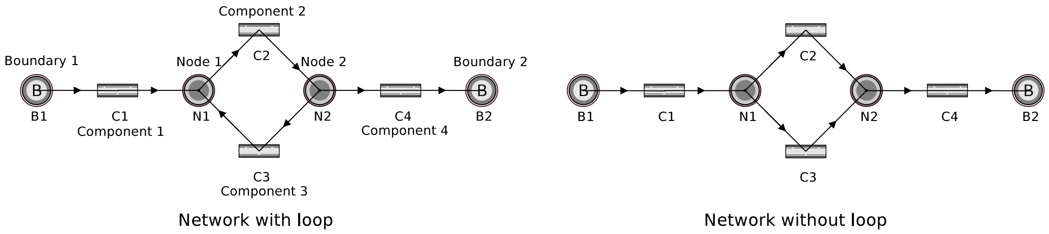


Figure 3: Network Loop

be considered as a particular case of the more general object oriented approach. In the object oriented approach, as there is no single input - single output components assumption, the topology defined by the user is kept during the resolution.

Main advantages for node oriented modelling:

- Reverse flow is easier to handle. The fluid lines are identified at the time to build the network. The mass flow sign in each line will determine the flow direction. In the node oriented approach at any moment a node to node link represents the beginning and the end of a flow line. In the object oriented approach the fluid lines have to be identified following the components connections and detecting which are components that perform as nodes.
- Easier to transform closed loop in open ones. A specific way to solve these networks could easily reverse, not only the flow changing the sign, but also the user topology. An example network is shown in Figure 3, with a topological loop. A node oriented approach could deal with this problem transforming the left scheme problem into the right one without loop.

In object oriented modelling, the closed loop must be detected and additional unknowns and closure equations have to be added to the left scheme in Figure 3. The link from component 3 to node 1 has to be internally eliminated. The system of equations increases the number of unknowns and equations. Each variable with the information proceeding from component 3 to feed the inputs in N1 will be added as iteration variable. An open loop network can then be solved. The additional closure equations equal the added unknowns with values from C3 with analogous value in N1. It must not be confused, the topology reversibility with the reverse flow effect. For an unchanged topology reverse flow may be overcome with the sign of the flow.

Main advantages of objects oriented modelling:

- Flexibility. The components may not be single input - single output, higher level components can be defined. It allows compatibility with between different disciplines like Gas Turbine Performance and Secondary Air Systems. In this sort of models components may exchange different kind of values with different neighbour components in different inputs and outputs. For example, a Gas Turbine Compressor may have different connections to other components exchanging different kind of magnitudes. It may be connected to the Secondary Air Systems, may also be connected with the mechanical shaft to the turbine and may also be connected with the neighbour gas path components, other compressors, an inlet or a combustion chamber.
- Encapsulation. Components are connected through the links and this is the only way of communication of a component with the exterior world.
- Re-usability. Due to inheritance feature, code and physical effects are reusable. Inheritance allows to easily develop more and new features in components.
- Feedback systems, where there is only physical description in one direction and no topology reversibility is possible, are easier to handle.

THE OBJECT ORIENTED SEQUENTIAL APPROACH

In this work, an object oriented sequential approach has been followed. The components are connected with the rest of the model through ports that are in charge of transferring the information between components. A port is basically a set of variables. There are different sorts of ports. Ports of the same type transfer the same variables between components. In the object oriented approach each component contains input and output ports, inputs at the left and outputs at the right. The physics of the component therefore defines the output from known input, Figure 4. The nomenclature followed is the usual for Object Oriented approaches, where a point separates the different levels of hierarchy:

ComponentName.ComponentVariableName

ComponentName.PortName.PortVariableName

Three different types of ports, in terms of topology, are defined, INPUT, OUTPUT and NODE ports. A component

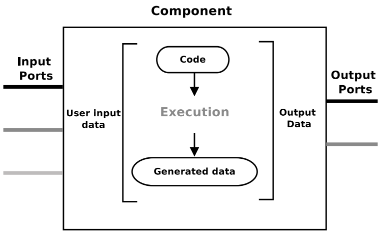


Figure 4: Component scheme

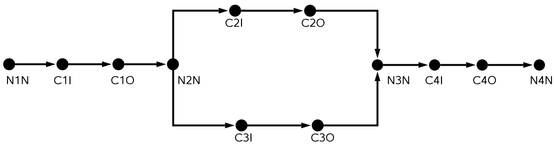


Figure 5: Network topology example

may contain several ports of different topological types. It must be noted that ports may have their own code with physical principles to fulfil, using the information of the connected ports. For example, a fluid INPUT port adds the incoming mass flows from the connected OUTPUT ports and averages the incoming total pressures and temperatures. This is also applicable to NODE ports. Figure 5 shows the network for the open loop scheme in Figure 3 in terms of ports. The represented ports belong to components with a node (name with N) and components with an input and an output port (name with C, suffix I for input and O for output)

For this network the incidence matrix may be built following a random order, perhaps the order that user followed inserting the components in the network, Figure 6. The incidence matrix accounts if a port is directly dependent on the others or not.

In the left part of Figure 6 the incidence matrix is not lower triangular. It means that this is not a good sequence of ports to run (executing the port if it is INPUT, the component if it is OUTPUT and both if it is NODE) because not all the necessary preceding components and ports have already been executed. The lower triangular matrix exists because there are not closed loops in the network topology. There are algorithms that are able to find a lower triangular incidence matrix changing columns and rows, as for example Reference 2. The right part of Figure 6 contains a sequence with a lower triangular incidence matrix. If the lower triangular matrix does not exist is because there are closed loops. Some algorithms that search for lower triangular matrices are able to detect the closed loops and help to convert the problem in an open loop one.

It will be assumed that the problem can be reduced to an open topology ordered with a proper incidence matrix. The remaining work is to find a set of unknowns or tearing variables with some initial values, run the network components and ports in a proper sequence and to evaluate some closure equations that the problem must match. This process is iterated with a solving algorithm until getting the unknown values that satisfy the closure equations.

	N1N	N2N	N3N	N4N	C1I	C1O	C2I	C2O	C3I	C3O	C4I	C4O
N1N	1	0	0	0	0	0	0	0	0	0	0	0
N2N	0	1	0	0	0	1	0	0	0	0	0	0
N3N	0	0	1	0	0	0	0	1	0	1	0	0
N4N	0	0	0	1	0	0	0	0	0	0	0	1
C1I	1	0	0	0	1	0	0	0	0	0	0	0
C1O	0	0	0	0	1	1	0	0	0	0	0	0
C2I	0	1	0	0	0	0	1	0	0	0	0	0
C2O	0	0	0	0	0	0	1	1	0	0	0	0
C3I	0	1	0	0	0	0	0	0	1	0	0	0
C3O	0	0	0	0	0	0	0	0	1	1	0	0
C4I	0	0	1	0	0	0	0	0	0	0	1	0
C4O	0	0	0	0	0	0	0	0	0	0	1	1

Not lower triangular

	N1N	C1I	C1O	N2N	C2I	C2O	C3I	C3O	N3N	C4I	C4O	N4N
N1N	1	0	0	0	0	0	0	0	0	0	0	0
C1I	1	1	0	0	0	0	0	0	0	0	0	0
C1O	0	1	1	0	0	0	0	0	0	0	0	0
N2N	0	0	1	1	0	0	0	0	0	0	0	0
C2I	0	0	0	1	1	0	0	0	0	0	0	0
C2O	0	0	0	0	1	1	0	0	0	0	0	0
C3I	0	0	0	1	0	1	1	0	0	0	0	0
C3O	0	0	0	0	0	0	1	1	0	0	0	0
N3N	0	0	0	0	0	1	0	1	1	0	0	0
C4I	0	0	0	0	0	0	0	0	1	1	0	0
C4O	0	0	0	0	0	0	0	0	0	1	1	0
N4N	0	0	0	0	0	0	0	0	0	0	1	1

Lower triangular

Figure 6: Incidence Matrices

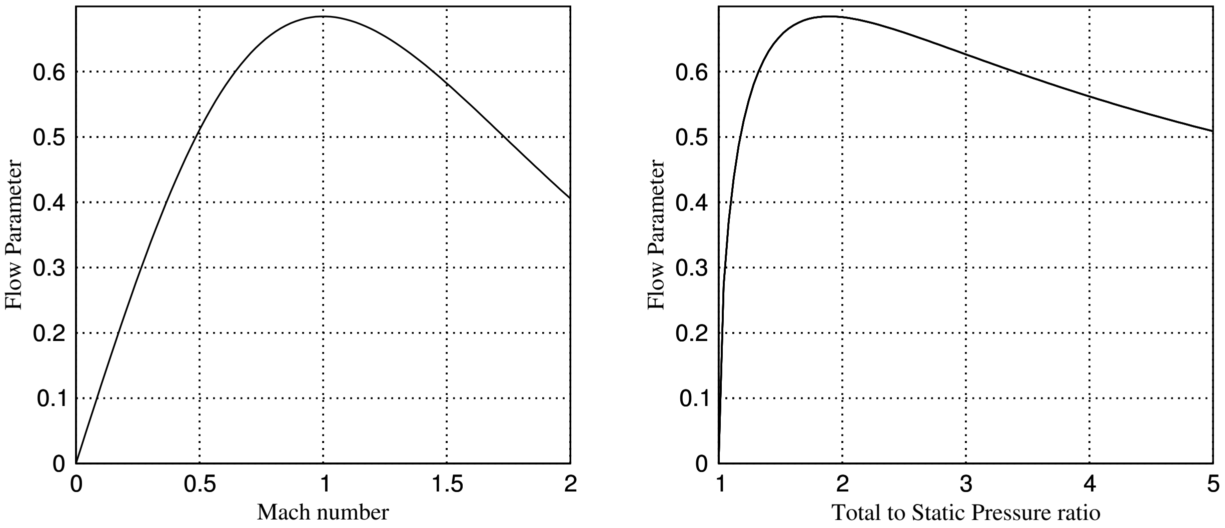


Figure 7: Flow parameter as function of Mach number and Total to Static pressure ratio

SELECTION OF TEARING VARIABLES

One option to sort out some mathematical issues solving a system of nonlinear equations is to change the tearing variables. These variables are the unknowns used to iterate in order to match the closing equations. To illustrate the problem, the following example is proposed here:

- Target: to get the Mach number and statics variables for a given area, total magnitudes and flow in an ideal gas fluid.
- Possible Tearing variables: Mach number M or Total to Static Pressure ratio $\frac{P_t}{P}$
- Closure equations

$$\frac{W\sqrt{RT_t}}{AP_t} = \sqrt{\gamma}M\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma+1}{2(1-\gamma)}} Flow\,parameter\,as\,function\,of\,M \tag{1}$$

$$\frac{W\sqrt{RT_t}}{AP_t} = \sqrt{\frac{2\gamma}{\gamma-1}\left(\left(\frac{P_t}{P}\right)^{\frac{\gamma-1}{\gamma}}-1\right)}\left(\frac{P_t}{P}\right)^{-\frac{\gamma+1}{2\gamma}} Flow\,parameter\,as\,function\,of\,\frac{P_t}{P} \tag{2}$$

Each one of the above equations is represented in Figure 7 for $\gamma = 1.4$. In the region next to zero flow these functions may be approximated by:

- Flow Parameter as function of Mach number

$$\left(\frac{W\sqrt{RT_t}}{AP_t}\right)_{M\to 0} \approx \sqrt{\gamma}M \tag{3}$$

- Flow Parameter as function of the Total to Static Pressure ratio

$$\left(\frac{W\sqrt{RT_t}}{AP_t}\right)_{\frac{P_t}{P}\to 1} \approx \sqrt{2\left(\frac{P_t}{P}-1\right)} \tag{4}$$

The behaviour of both functions is totally different in this region: while the first one is linear in the proximity of M=0, the second one has an infinite slope. This fact complicates the numeric solution and therefore the Mach number is preferred as variable to iterate.

This effect appears when the closure equation is of the type $y\sim\sqrt{x}$ in $x=0$. For these cases a variable change $z=\sqrt{x}$easies to solve an equation of the type $y\sim z$. Once a z value solution is obtained the operation $x=z^2$ returns the desired value. These types of variable changes that transform the closure equation removing singularities will be used, if possible.

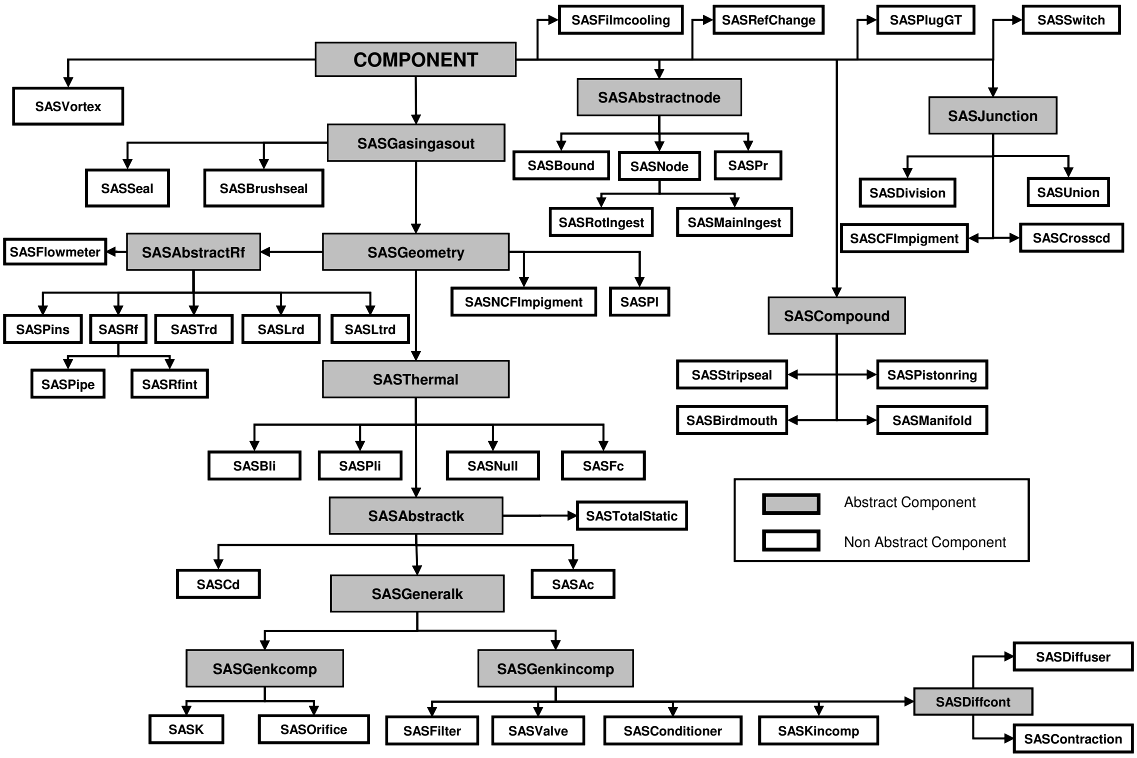


Figure 8: SAS inheritance tree

SAS, GAS TURBINE SECONDARY AIR SYSTEM LIBRARY

As mentioned above, the use of Object Oriented Simulation allows the inheritance tree described in Figure 8. It shows the amount of components and complexity of a library to model Secondary Air Systems. Many properties and methods are directly inherited from parents to children. This way to proceed helps to maintain the development code to characterise components. There are abstract components, which mean they cannot be used for modelling. They are used as objects, with attributes and methods common for the children who inherit. Except components inherited from nodes, unions and junctions, the rest are single input - single output, with one INPUT and one OUTPUT port. Following paragraphs describe some of the difficulties found during the SAS library implementation as well as the solution adopted. It is not the purpose to describe the physical methodology of each component, but only a description of some of the mathematical difficulties found. There is existing literature describing the physical behaviour of this sort of components like in Reference 1.

Choking effect

This effect will condition the selection of the system of equations to be solved. The choking effect may appear at any region in a network. In a fluid line, from one node to another node, a choked component may be found, and in some cases, more than one component may be choked. In a fluid line, the mass flow is common to every component inside. If a component is choked, the mass flow is the critical mass flow. The critical mass flow is defined as the maximum flow for a given inlet conditions and area. For flows beyond the critical one, there is no solution for the equations to satisfy. In the case of adiabatic flow, see Figure 7, the maximum flow is reached at Mach number one.

The critical Pressure Ratio is defined as the minimum Pressure Ratio where the component becomes choked. The main problem associated to a choked component is that nothing can be said about the Pressure Ratio. When a component is choked the mass flow is the critical one and the Pressure Ratio may be higher than the critical to satisfy conditions upstream and downstream. For a fluid line the simplest scheme is to iterate in mass flow matching the pressure at the end of the line. The mass flow guessed is passed from one component to the next and the component pressure drop is associated with the mass flow. This basic scheme is not valid to overcome the choking effect because a mechanism to unlink mass flow and pressure drop is needed if choking appears.

One solution is to add as tearing variable the Pressure Ratio of the components susceptible to be choked. The physics of the components relates mass flow with pressure ratio. The associated closure equation needed is that mass flow at the inlet of the component must be equal to the mass flow at the outlet, calculated with the pressure ratio guessed.

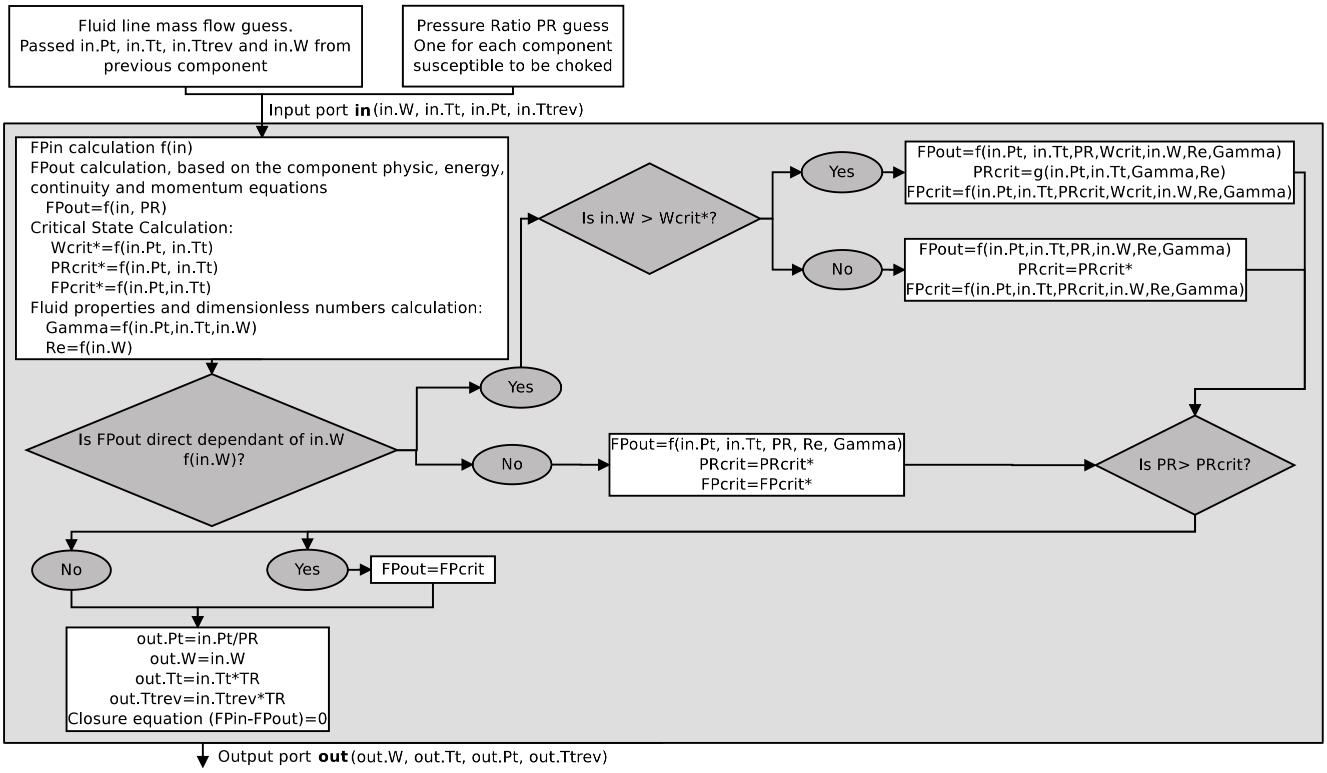

Fluid line mass flow guess.
Passed in.Pt, in.Tt, in.Ttrev and in.W from previous component
Pressure Ratio PR guess
One for each component susceptible to be choked
Input port **in** (in.W, in.Tt, in.Pt, in.Ttrev)
FPin calculation f(in)
FPout calculation, based on the component physic, energy, continuity and momentum equations
FPout=f(in, PR)
Critical State Calculation:
Wcrit*=f(in.Pt, in.Tt)
PRcrit*=f(in.Pt, in.Tt)
FPcrit*=f(in.Pt,in.Tt)
Fluid properties and dimensionless numbers calculation:
Gamma=f(in.Pt,in.Tt,in.W)
Re=f(in.W)
Is in.W > Wcrit*?
Yes
FPout=f(in.Pt, in.Tt,PR,Wcrit,in.W,Re,Gamma)
PRcrit=g(in.Pt,in.Tt,Gamma,Re)
FPcrit=f(in.Pt,in.Tt,PRcrit,Wcrit,in.W,Re,Gamma)
No
FPout=f(in.Pt,in.Tt,PR,in.W,Re,Gamma)
PRcrit=PRcrit*
FPcrit=f(in.Pt,in.Tt,PRcrit,in.W,Re,Gamma)
Is FPout direct dependant of in.W f(in.W)?
Yes
No
FPout=f(in.Pt, in.Tt, PR, Re, Gamma)
PRcrit=PRcrit*
FPcrit=FPcrit*
Is PR> PRcrit?
No
Yes
FPout=FPcrit
out.Pt=in.Pt/PR
out.W=in.W
out.Tt=in.Tt*TR
out.Ttrev=in.Ttrev*TR
Closure equation (FPin-FPout)=0
Output port **out** (out.W, out.Tt, out.Pt, out.Ttrev)

Figure 9: Choking logic

For numerical convenience instead of the mass flow in the closure equations, mass flow parameter related to the inlet conditions will be used. In this scenario a new logic covering choking can be applied. If the Pressure Ratio found during the iteration is higher than the critical Pressure Ratio calculated with the inlet conditions, the flow parameter at the exit will be the critical one. With this scheme, Pressure Ratios over the critical one are allowed, not allowing the choked components to admit a mass flow over critical. Physical equations relate the Pressure Ratio with mass flow, and some considerations must be taken into account. If mass flow at the inlet is involved in the mass flow parameter calculation, some problems may appear while iterating since there are two independent tearing variables and the choking criteria has to be applied to both. Figure 9 accounts for a single iteration possible discontinuities in the equation residuals during the solution search.

Reverse flow effect

This problem arises as the direction of the flow is not known a priory, thus, it has to be possible to change the flow direction during simulation. Mathematical difficulties associated to zero flow must be handled. The user is not required to initialise solutions provided that the system solution and flow direction are completely unknown.

Due to the object oriented sequential approach, the component, rather than the global algorithm, must handle the reverse flow. The users define the positive direction of the flow when building the network topology and negative mass flows or pressure ratios below one must be possible. Reversible components are governed by the same equations in both directions while non-reversible components change completely their equations depending on the direction. A constant area duct is a reversible component while a non-constant area duct performs as a nozzle in one direction and as a diffuser in the other. It does not really matter if the component is reversible or not, the behaviour of the problem will be the appropriated one depending if the component pressure ratio guess is above or below one. In the object oriented sequential approach, to make a component working under reverse flow and to take into account that the user defined the direct flow from left to right, the component must detect this event and deal with the flow direction from right to left. To reuse the physical modelling code of reversible components, a simple variable change works. The variable change must swap inputs to outputs before executing, see Figure 10. This change has to be undone before going on with next components in the network.

The main difficulty associated to reverse flow is that temperature is only transferred downstream. This means that when the flow is reverse it is necessary to transfer the temperature from the outlet port (right) to the input port (left). A new variable, Ttrev, is introduced because the network topology keeps constant during the solution of the equations. This

Direct flow	Reverse flow	Reverse flow and reversible component $PR* = \frac{1}{PR}$ $TR* = \frac{1}{TR}$
Upstream Pressure in.Pt	Upstream Pressure $\frac{in.Pt}{PR}$	Upstream Pressure $in.Pt* = \frac{in.Pt}{PR}$
Upstream Temperature in.Tt	Upstream Temperature $in.Tt \cdot TR$	Upstream Temperature $in.Tt* = in.Tt \cdot TR$
Downstream Pressure $\frac{in.Pt}{PR}$	Downstream Pressure in.Pt	Downstream Pressure $out.Pt* = \frac{in.Pt*}{PR*}$
Downstream Temperature $in.Tt \cdot TR$	Downstream Temperature in.Tt	Downstream Temperature $out.Tt* = in.Tt^* \cdot TR^*$

Figure 10: Reversible components under reverse flow. Variable change to reuse physical modelling programming

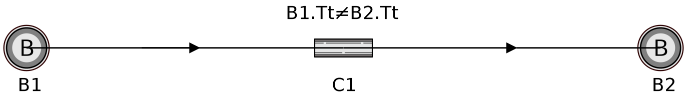


Figure 11: Temperatures discontinuities around PR=1

new variable is calculated through the line in the same way as the the total temperature, multiplying by the component temperature ratio ($out.Ttrev = in.Ttrev \cdot TR$). In the node where the flow line starts, the temperature for reverse flow is tearing variable and the correspondent closure equation is the temperature at the final node. The node at the beginning of the line decides to pass the direct or reverse temperature depending on the mass flow sign.

Initialisation and multilevel strategy

As reverse flow must be overcome, the philosophy is to initialise mass flow to zero and components pressure and temperature ratios to one. The system of equations itself, decides during convergence, the flow direction with the mass flow sign. Therefore, the components must deal with zero flow, and for a robust solution, the closure equations must have an appropriate behaviour around mass flow zero (continuity and residual slopes different from zero or infinite). This is not easy to achieve. For instance, a component between two boundaries may have discontinuities around zero mass flow, Figure 11. In the region next to PR=1, the expression used to calculate the mass flow parameter depends on the flow direction:

- Direct flow: $PR = 1 + \varepsilon \quad FPOut = f(in.W, P_{inlet}, PR, Tt_{inlet}) = f(in.W, B1.Pt, (1+\varepsilon), B1.Tt)$

$$\varepsilon \to 0 \qquad FPOut = f(in.W, B1.Pt, 1, B1.Tt) \tag{5}$$

- Reverse flow: $PR = 1 - \varepsilon \quad FPOut = f(in.W, P_{inlet}, \frac{1}{PR}, Tt_{inlet}) = f(in.W, \frac{B1.Pt}{(1-\varepsilon)}, \frac{1}{(1-\varepsilon)}, B2.Tt)$

$$\varepsilon \to 0 \qquad FPOut = f(in.W, B1.Pt, 1, B2.Tt) \tag{6}$$

B1.Tt is different from B2.Tt around PR=1. To deal with this discontinuity the system is solved in separate levels. Firstly, it is calculated an estimation of the mass flow using B1.Tt in the mass flow parameter expression. This estimation is used as initial value for the second level that uses the correct expression (either 1 or 2 depending on the flow direction).

It is somehow a continuity method, used to solve nonlinear systems of equations by successive approximations to the final problem. Although, there are two systems to solve, the procedure accelerates the computation time, provided good performance for the easier problem and the solution to initialise the whole problem. The decision of different solving levels is taken at the components and library creation. It is a task for component developers, not for final users.

The multilevel strategy is also adequate for more advanced components, where the complexity of the modelling methodology requires a proper initialisation. More simple levels could solve components and get a choking estimation or the direction of flow and for higher levels would be computed most sophisticated calculations, as some zooming capabilities with other program interactions.

Scaling the problem

In the mathematical equations solution, the use of non-dimensional parameters is recommended; if it is not possible, it is recommended to scale both, the tearing variables and the closure equations. It helps to achieve a well-conditioned system of equations. This scaling process is sometimes useful to avoid numerical problems derived from the magnitude units.

To scale or reduce the mass flow, a value for each fluid line must be used that will depend on the nature of the different components in the fluid line. For pressures in the nodes which are used as unknowns a typical pressure value is enough. ISA pressure at sea level is a good value. With this scale, values of pressures over 100 or under 0.01 are not expected in typical SAS models. This value can also be used to scale the closure equations for pressure equalities at nodes. For temperatures and closure equations equalling temperatures, the ISA temperature at sea level is adequate. As expected

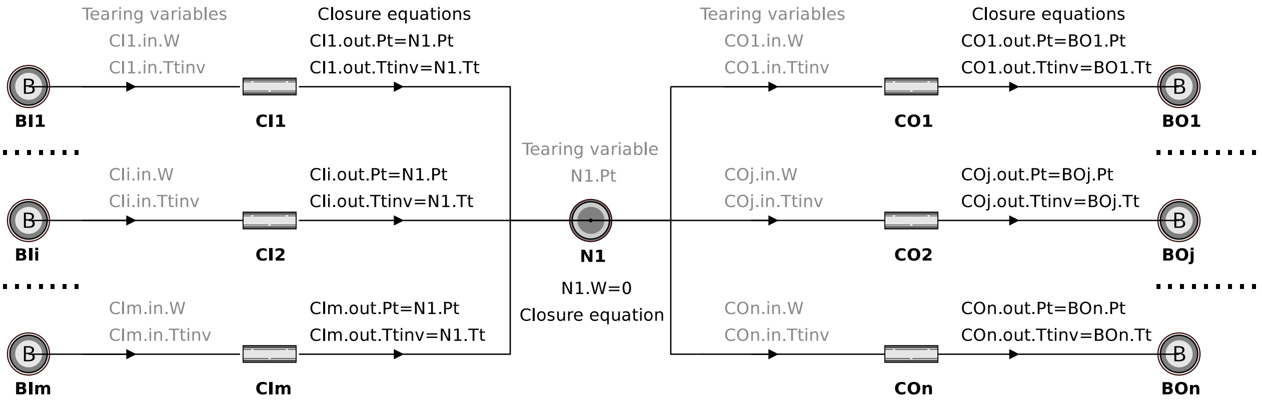


Figure 12: Node example

temperatures in a Gas Turbine are in the range 150-2000K, the scaled values may vary around 0.1-10. Pressure ratios and temperature ratios are not needed to be scaled.

For additional unknowns and closure equations the scaling issue must be studied in detail. The objective is to control the residual value and slope of the closure equations to improve the system convergence. For example, to solve the choking effect, additional equations equalling the mass flow parameter at the component input and output have been used. The flow parameter is expected to be of order unity and this closure equation is expected to be of unity order. For components whose geometry is defined by the area it works but, for some other with additional geometrical data, as pipes with the associated length, an additional value to scale this closure equation is needed.

SAS node

This is one of the most important components. It is detailed because it is in charge of making the network to work. In Figure 8 node is organised as an abstract component and three components inherit from it, boundary (SASBound), Pressure Node (SASPr) and internal node (SASNode). Nodes are multiple inputs-multiple outputs. This section affects to all types of nodes although it is focused on the internal node. The missions of the nodes are:

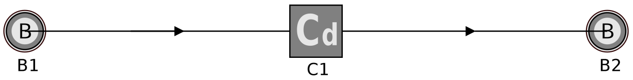
- To balance pressures.
- To satisfy mass flow continuity.
- To satisfy the energy equation.

Figure 12 shows an example of a very basic network with e internal node. There are m boundaries with lines that input to the node and n lines that output from the node to their respective boundaries. Each line has a component that relates pressure drop with mass flow. From the possible solutions to satisfy the governing equations, the selected one is to keep all the lines independent from the rest at the iteration level, maintaining the same behaviour in all of them . The node pressure is introduced as tearing variable and, for every input line there is a closure equation that equals the total pressure at the end of the line with the node pressure. The calculated node pressure is passed to every output line. Mass flow continuity equation is not used inside the node calculations but it is added as closure equation.

Some other options could be not to have the pressure node as unknown and equal the incoming lines pressure to a preferred one. It is one less unknown and one less equation than the proposed system of equations. On the other hand, one of the output mass flows could have been function of the other mass flows and the continuity equation would have been automatically satisfied in all iterations. The solutions with a preferred line may fail during convergence, due to numerical issues. For example, a node where many lines with big flows and a leakage line that is the preferred line. All the iterations iteration, the leakage flow is calculated as the sum of the flow in the remaining lines (with the respective sign). A small quantity as the sum of big quantities may lead to unexpected numerical behaviours.

Inputs and outputs for a node are defined by the user when the network topology is constructed. New concepts are “effective input” and “effective output”, that represent if really flow is entering or not to the node which has to be taken into account in the thermal balance. “Effective inputs” are inputs with positive mass flow or outputs with negative mass flow. “Effective outputs” are outputs with positive mass flow or inputs with negative mass flow. Total enthalpy entering the node is calculated with “effective inputs” and the node temperature is passed only to the effective output lines.

$$Tt_{node} = \frac{\sum_{effective\,inputs} H}{\sum_{effective\,inputs} W} \tag{7}$$



Unknowns	Closure equations
$C_d.in.W$	$EQ1 = \frac{B_2.P_t - C_d.out.P_t}{P_{tref}} = 0$
$C_d.in.Tt_{rev}$	$EQ2 = \frac{B_2.T_t - C_d.out.T_{trev}}{T_{tref}} = 0$
$C_d.\beta$	$EQ3 = FP_{out} - FP_{in} = 0$

Figure 13: A simple Cd model

SAS tearing variables and closure equations

Due to chocking and reverse effects it is necessary to introduce tearing variables to relate the inlet with the outlet in terms of total pressure and temperature:

- Pressure ratio: defined as $PR = \frac{in.Pt}{out.Pt}$ (over one in a direct flow condition)
- Temperature ratio: defined as $TR = \frac{out.Tt}{in.Tt}$ (over one if heat is added to the component and the flow is direct)

Pressure ratio

Pressure ratio appears in the mass flow parameter calculation but also in the closure equation to match the pressure at the end of the line. Figure 13 shows a simple model with its tearing variables and closure equations. If the starting point to iterate is $Cd.PR_0 = 1$ Newton Raphson estimates for the next iteration $Cd.PR_1$.

$$EQ1 = B2.Pt - \frac{B1.Pt}{Cd.PR} = 0 \qquad \left.\frac{dEQ1}{dPR}\right|_{PR=1} = B1.Pt \qquad Cd.PR_1 = 2 - \frac{B2.Pt}{B1.Pt} \tag{8}$$

For values of $B2.Pt > 2B1.Pt$ the first Newton Raphson step tries negative pressure ratios. This effect is caused by the pressure ratio definition. If the tearing variable is the inverse pressure ratio this problem disappears:

$$Cd.\beta = \frac{1}{Cd.PR} = \frac{Cd1.out.Pt}{Cd2.in.Pt} \quad EQ1 = B2.Pt - B1.Pt * Cd.\beta = 0 \quad \left.\frac{dEQ1}{d\beta}\right|_{\beta=1} = -B1.Pt \quad Cd.\beta_1 = \frac{B2.Pt}{B1.Pt} \tag{9}$$

According to this, although the PR parameter is maintained in all the component equations, the external tearing variable to iterate will be $\beta = \frac{1}{PR}$. The variable change is done before executing each component and undone before going on with next components.

Temperature ratio

If a fixed quantity of heat is added to the example of Figure 13 a new closure equation has to be added to calculate the temperature ratio TR:

$$EQ2 = Cd.in.W * \bar{cp} * Cd.in.Tt * (Cd.TR - 1) - Q_{added} = 0 \qquad \left.\frac{dEQ2}{dTR}\right|_{TR=1} = Cd.in.W * \bar{cp} * Cd.in.Tt \tag{10}$$

$$Cd.TR_1 = 1 + \frac{Q_{added}}{Cd.in.W * \bar{cp} * Cd.in.Tt} \tag{11}$$

For high values of Q_{added} combined with reverse flows, the first Newton Raphson step in the second level tries negative temperature ratios which lead to negative values of out.Tt and the consequent convergence failure.

$$Cd.out.Tt_1 = Cd.in.Tt_1 \cdot Cd.TR_1 \rightarrow reverse\,flow \rightarrow Cd.out.Tt_1 = Cd.in.Ttrev_1 \cdot Cd.TR_1 \tag{12}$$

This effect disappears if instead of the temperature ratio as tearing, $Cd.\Delta T$ is used instead:

$$Cd.\Delta T = \frac{Cd.out.Tt - Cd.in.Tt}{T_{ref}} = (Cd.TR - 1) \cdot \frac{Cd.in.Tt}{T_{ref}}$$

A negative value of TR implies also a negative value of ΔT but now it does not imply a negative value of out.Tt.

$$Cd.out.Tt_1 = T_{ref} \cdot Cd.\Delta T_1 + Cd.in.Ttrev_1$$

Although ΔT is negative the reverse temperature tries to make out.Tt positive.

This variable change is introduced in the same way as the one in the pressure ratio parameter, before entering the component and undone after the execution.

SAS Network Summary

- For a given network topology, representing a model, the incidence matrix can be constructed, closed loops can be detected and opened. The network can be sorted, to be executed in an appropriate order, bet keeping the topology defined.
- The nodes, including boundaries will contribute to build the system equations to be solved. For each output line the mass flow and reverse temperature will be added as tearing variables, mass flow to be solved in a lower level than reverse temperature. For the input lines two closure equations will be added, pressure balance in the lower level and temperature balance in the higher. Each node, not boundary, will add the total pressure as tearing variable and mass flow as closure equation in the lower level.
- For each component susceptible to be choked and reversed, the inverse pressure ratio will be added as tearing variable in the lower level and temperature difference (if necessary) in the higher level. Continuity of flows will be added as closure equation in the lower level $\left(\frac{FP_{out}-FP_{in}}{FP_{scale}}=0\right)$. Energy balance must be imposed at higher level using the scaled ΔT as tearing.
- Lower level solution starts iterating the tearing variables to match the closure equations. In all iterations, for given values of the unknowns, the network is executed in the right order and closure equations residues are then calculated.
- The lower level is solved. Keeping the results from the lower level, the higher levels are executed. Zooming and more accurate calculations will be added to higher levels of execution. It is possible to include design options by adding proper tearing variables and closure equations after the higher level of the network.

CONCLUSIONS

- An approach to network models solution is described. Advantages and disadvantages compared to different approaches have been taken into account. Some of the issues found in Gas Turbine SAS models have been used as example.
- Some of the most important mathematical difficulties have been arisen and a solution is proposed for them. A clear understanding of the problem details helps to find an adequate solution. Although further details as ingestion or some other complex components could be explained, the essential is showed in this paper.
- There are better and worse mathematical approaches to overcome the problem. This issue has been discussed and the approach presented in this work provides a robust solution. The resultant systems of nonlinear equations may be solved with the Newton Raphson method.

REFERENCES

1. ESDU Fluid Mechanics, Internal Flow, www.esdu.com
2. I. S. Duff , J. K. Reid, An Implementation of Tarjan's Algorithm for the Block Triangularization of a Matrix, ACM Transactions on Mathematical Software (TOMS), v.4 n.2, p.137-147, June 1978.
3. M. Shacham, S. Macchietto, L. F. Stutzman and P. Babcock, Review, Equation Oriented Approach To Process Flowsheeting, Computers & Chemical Engineering, Vol 6, No.2, pp. 79-95, 1982.
4. P. P. Walsh, P. Fletcher, Gas Turbine Performance, second edition, Blackwell Publishing, chapters 5, 6 and 7.