Programmer's Guide to MATSTAB

Distribution: Lotfi Belblidia, Gerardo Grandi

# Introduction

*Purpose*

The purpose of this document is to support the development of a frequency domain module in S3K. The document describes the matstab-code from a programmer's point of view. Special focus is spent on describing the matrices needed for the linearized system.

*Overview of MATSTAB*

Matstab solves the linearized system of differential equations and is used to evaluate the decay ratio for a BWR. In S3K, the non-linear differential-algebraic system (1) is solved.

(1)

Where the matrix **B** is the identity matrix, except that the diagonal elements for algebraic equations are zero. In matstab, the function **f** is linearized, and the margin to instability is evaluated by solving (2):

(2)

where

Equation (2) is solved iteratively for the eigenvalue that corresponds to the global mode. The starting guess for **e** is based on the power distribution with an experience based typcial phase-shift profile in the axial direction. The eigenvalue can be divided into its real and imaginary part:

(3)

The frequency is and decay ratio is given by:

(4)

(5)

A typical starting guess for could be:

The starting guess can be adjusted when evaluating at a non-typical flow rate (recommended to ensure convergence).

In the code, the following main steps are performed in respective function described below.

|  |  |  |
| --- | --- | --- |
| Function name | Purpose | Main output variable(s) |
| init\_msopt | Set up options for matstab | msopt |
| get\_inp\_sim3 | Reads input and history from s3k input file and restartfile | fue\_new,geom,fuel neu termo |
| steady\_state | Solves the steady state | steady, Xsec |
| build\_A | Builds the **A** and **B** matrix in eq. (2) | At,An,Bt, etc, matr |
| newt | Solves eq. (2) for λ and **e** | lam, e |

It would seem that the steady state solution should not be required, since there is a steady state solution available from Simulate. However, the LHS of eq. (1) has to be zero for the set of equations that will be linearized and solved in eq. (2).

# Matrix composition

*Subdivision of the matrices*

The matrices **B** and **A** are subdivided into its components in order to facilitate the solution algorithm. Both matrices are subdivided into neutronics (An), fission power (Aq), fuel temperature (Af), in-core thermohydraulics (At) and ex-core thermohydraulics (Aj). The A-matrix is subdivided according to eq. (6):

(6)

(6)

**,** andare basically identity matrices with some zeros on the diagonal. is actually the identity matrix, and in there is an exception to the rule that we will come back to later. At first, it may seem surprising that is zero, but the dynamics of the delayed groups can be dealt with in the RHS of eq. (2), and it is obvious that neglecting the pure terms have no impact on the results, since this dynamics is very fast and be considered to be algebraic equations. I want to stress that the only approximation is that the pure terms are neglected, the delayed groups *are* considered in the equations.

*Submatrix* **At** *(function* **A\_th***)*

The submatrix **At** contains the thermo hydraulics for the core. There are 7 variables per node,

. The variable numbering is given by:

matr.ibas\_t = 1:7:ntot\*7, where ntot is the total number of nodes.

Thus, the indices for α is 1,8,15,22,…,ntot\*7-6 or ibas\_t

and for tl 2,9,16,23,…,ntot\*7-5 or ibas\_t+1

The corresponding equations are:

 (At.1)

 (At.2)

At.2 is the eq. that does not have just a diagonal element in **Bt**. Here, there is a piece of  and  on the LHS. The internal energy is substituted:



 (At.3)

 (At.4)

 (At.5)

where





 (At.6a)

 (At.6b)

where (At.6a) is used for forced convection (no boiling) and (at.6b) is used for nucleate boiling.

 (At.7)

In the code, At.5 occurs before At.4, since some results calculated for At.5 is needed in At.4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eq. Name | Variable | Number | Index | Method | Note |
| Phasic mass balance | *α* | At.1 | ibas\_t | Analytical |  |
| Mixture Energy | *tl* | At.2 | ibas\_t+1 | Analytical RHS, FD LHS | Non-diagonal LHS |
| Wl=f(alfa,tl,jm,P) | *Wl* | At.3 | ibas\_t+2 | FD | Slip is not explicit |
| Volume expansion | *Wg (jm)* | At.4 | ibas\_t+3 | Analytical | Wg is not explicit |
| Evaporation Rate | *ΓV* | At.5 | ibas\_t+4 | FD | Links to **Af** (tw) |
| Linear heat generation rate | *qw'* | At.6 | ibas\_t+5 | FD | Links to **Af** (tw) |
| Volumetric flux | *jm (Wg)* | At.7 | ibas\_t+6 | Finite Differences (FD) |  |

For a node (in the particular case, the most oscillating node in void), the numbers of the diagonal 7 by 7 block may look like:

*α* *tl  Wg Wl ΓV qw' jm* 

The *k-1* terms denotes dependence on neighbors and are thus in matrix **At** itself, whereas the *tw* terms can be found in **Atf**, the *qfiss* terms in **Atq**, and the *P* terms are stored in **Atj**.

The rest of the submatrix **At** consists of two sets of equation per channel.

1. Channel flow distribution, integrated momentum, I = sum(Gm) over each closed loop, 'main output' variable: Wl\_inn, 'input variable' In, where n runs over all channels
2. Impulse momentum balance dI/dt = ploss

The matrices **Atj Atq Atf** describe the dependence of the equations above to respective submatrix

*Submatrix* **Bt** *(function* **A\_th***)*

The submatrix **Bt** is given by the left hand side of equations At.1-At.7. **Bt** is block-diagonal, and each block has the following contents the supscript *i* denotes the *i*:th node:





*Submatrix* **Aj** *(function* **A\_th***)*

The submatrix **Aj** (sometimes denoted **AI**) describes the ex-core thermohydraulics.

The variables are: 2 (tl Wl) per section before the core (1-phase portion), and 4 (α tl Wg Wl) after the core. In order to be able to compare with earlier experience in MATSTAB, I have hardcoded the number of sections to be:

|  |  |
| --- | --- |
| Ex-core portion | nsec |
| Downcomer 1 dc1 | 3 |
| Downcomer 2 dc2 | 5 |
| Lower plenum 1 lp1 | 2 |
| Lower plenum 2 lp2 | 2 |
| Upper plenum upl | 4 |

Thus, there are 24 variables for the one-phase portion, and 16 variables for the two-phase portion.

The equations for the one-phase portion are quick and dirty at this point:

In the connection to the core, the sum of all channel flows equals the sum of the flow out of downcomer 2.

Of course, in hindsight it is sort of silly to have 24 variables when one (Wltot) probably would have been enough, but let's just leave it for now.

For the two-phase portion, the following four equations are used:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eq. Name | Variable | Number | Index | Method | Note |
| Phasic mass balance | α | Aj.1 | bas\_u+1 | Analytical |  |
| Mixture Energy | tl | Aj.2 | bas\_u+2 | Analytical RHS, FD LHS | Non-diagonal LHS |
| Wl=f(alfa,tl,jm,P) | Wl | Aj.3 | bas\_u+3 | FD |  |
| Volume expansion | Wg (jm) | Aj.4 | bas\_u+4 | Analytical | jm is not explicit |

Since there is no steam generation in the riser, the equations At.1-At.4 collapses to Aj.1-Aj.4

 (Aj.1)

 (Aj.2)

 (Aj.3)

 (Aj.4)

In equation Aj.4, *jm* is not explicit. Instead, the linearized coefficients refer to *Wl* , *Wg* , *tl* and the system pressure, *P*.

In addition to this, there are two variables associated with the pump.

These variables are: pump speed and pressure increase over the pump. If the pump is a jet pump, the variables are: drive flow (Wdr) and pressure increase.

The last variable in **Aj** is the system pressure.

*Submatrix* **Af** *(function* **A\_tfuel***)*

There are *Mf* (*Mf* =4 in Matstab) differential equations describing the *Mf* fuel-zones:

 (Af.1)

 (Af.2-3)

i=2,..., *Mf*

 (Af.4)

*i=Mf(=4)*

With the fuel pellet surface temperature:



There are two equations describing the fuel gap dynamics:

 (Af.5)

 (Af.6)

Equations Af.1-Af.6 are linearized with finite differences by using the function eq\_DtfDt.

The wall temperature is given by:

 (At.7)

Equation At.7 is linearized with finite differences on the function eq\_tw.

The matrix is block diagonal and a block could look like this for a low power node:

*tf1 tf2 tf3 tf4 tc1 tc2 tw*



The *qfiss* terms are stored in **Afq**, the *tl* term is stored in **Aft** and the pressure term (*P*) is stored in **Afj**.

For a high power node (same node as for **At**above), the numbers could be:

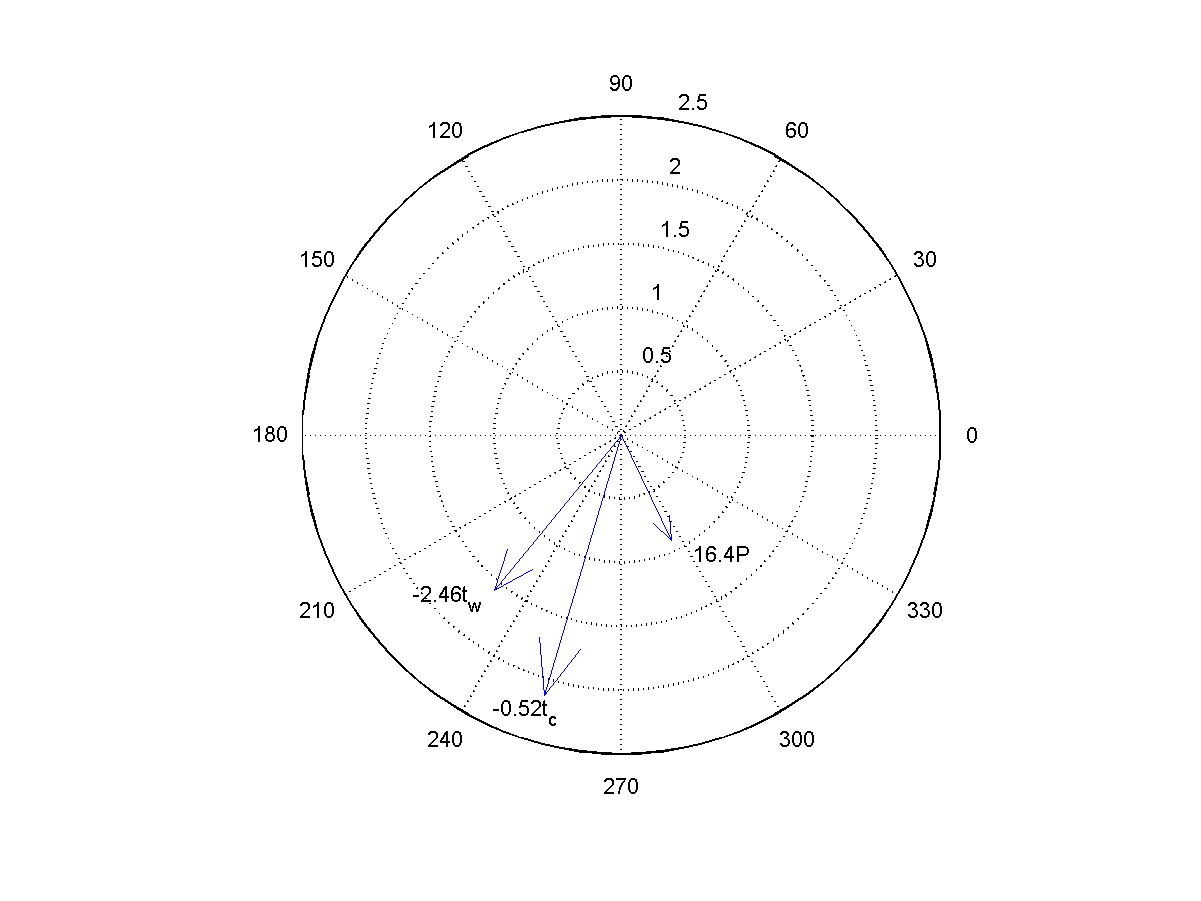
*tf1 tf2 tf3 tf4 tc1 tc2 tw*



For the high power node, there is no dependence on *tl*, since *tfl=tsat*.

For the case under consideration, the phasors corresponding to the last equation is depicted in the figure.

Note that the *tc* component has been depicted with negative sign. Thus the sum of the components corresponding to *tw* and *P* should add up to the vector representing -0.52*tc*.



The numbers vary from node to node, but they should be in the general range indicated in the example matrices above. The matrix **Bf** is almost the identity matrix, except that At.7 is algebraic, thus a block of **Bf** is given by:

