

# Probability-One Homotopy for Robust Initialization of Differential-Algebraic Equations

Michael Sielemann

Deutsches Zentrum für Luft- und Raumfahrt, Robotics and Mechatronics Center,  
System Dynamics and Control, Münchner Strasse 20, 82234 Wessling, Germany

## Abstract

An evolution of the recently introduced operator homotopy() is proposed, which further improves the solution of difficult initialization problems. The background and motivation for this approach are discussed and it is demonstrated how to apply it for electrical and fluid systems. The key difference to the earlier approach is the supporting theory, which guarantees that the method converges globally with probability one.

*Keywords: Initialization, DAE, homotopy, nonlinear equations*

## 1 Introduction

A dynamic model describes how the state variables and thus the entire system behave over time. The state variables define the current condition of the model and have to be initialized when simulation starts. For this purpose, Modelica provides language constructs to define initial conditions such as initial equation sections [12]. The resulting constraints and all equations and algorithms that are utilized during the simulation form the initialization problem. Based on its solution, all variables, derivatives and pre-variables are assigned consistent values before the simulation starts.

Mathematically, the resulting problem is an initial value problem for a differential algebraic equation system (DAE) with  $\dim(\mathbf{f}) = nx + nw$  equations:

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{w}, t) = \mathbf{0}, \mathbf{x}(t) \in \mathbb{R}^{nx}, \mathbf{w}(t) \in \mathbb{R}^{nw}, t \in \mathbb{R}$$

Here,  $\mathbf{x}$  is the vector of state variables and  $\mathbf{w}$  is the vector of algebraic unknowns. For simplicity of the discussion, we assume that the DAE has no hybrid part and is index-reduced, i.e. it has index 1, which means that the following expression is regular:

$$\begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}} & \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \end{bmatrix}$$

Note that all the following results still hold for hybrid, higher index DAEs with small adaptations. Initialization means to provide consistent initial values for  $\dot{\mathbf{x}}_0$ ,  $\mathbf{x}_0$ ,  $\mathbf{w}_0$  so that the DAE is fulfilled at the initial time  $t_0$ . Since these are  $2 \cdot nx + nw$  unknowns and the DAE has  $nx + nw$  unknowns, additional  $nx$  equations must be provided which are called "initial equations" in Modelica:

$$\mathbf{g}(\dot{\mathbf{x}}_0, \mathbf{x}_0, \mathbf{w}_0, t_0) = \mathbf{0}, \dim(\mathbf{g}) = nx$$

The most often used initial equations are:

$$\mathbf{g}(\dot{\mathbf{x}}_0, \mathbf{x}_0, \mathbf{w}_0, t_0) = \dot{\mathbf{x}}_0 = \mathbf{0}$$

that is, steady-state initialization.

The result is usually a nonlinear system of algebraic equations, which has to be solved numerically. This does not always work right away for industrial problems as the commonly employed gradient-based local algorithms [2, 10, 3], such as the damped Newton method, provide local convergence only (even when using globalizations such as trust regions).

Modelica allows users to describe any model mathematically, which makes it highly flexible and powerful for simulation of heterogeneous multi-domain physical systems. However, this also means that no knowledge of the mathematical character of the problem equations can be introduced into the solver. Instead, an algorithm has to work on a general numerical problem (in contrast to domain-specific algorithms for nonlinear problems).

As a result, the success to solve initialization problems of state-of-the-art implementations of Modelica tools depends on the choice of iteration variables and the guess values for these variables defined with the start attribute. As a result it may become difficult for a library developer to provide a robust initialization capability.

Since a model becomes useless whenever initialization fails, and the current state-of-the-art is not fully

satisfactory in this regard, we conclude that more reliable and robust methods are needed for a wider application of the Modelica modeling language by practitioners.

In a previous paper, we introduced a homotopy operator in [19]. It maps  $\text{homotopy}(\text{actual} = \dots, \text{simplified} = \dots)$  to  $\lambda \cdot \text{actual} + (1 - \lambda) \cdot \text{simplified}$ . Successful application examples were given for electronic circuits and multibody systems in [19] and for power plants in [1].

This homotopy operator is typically used to simplify governing equations of components, sweep boundary conditions and the like. The advantages of this approach are that the concept is simple and easy to understand. Also, it was successfully tested on relevant test cases. It has certain limitations however, in particular that the homotopy map is hard-wired into the language specification, that convergence is based on heuristics, and that a naive application can lead to singular problems (e.g., with a singular Jacobian at  $\lambda = 0$ ).

The objective of this contribution is thus to propose a more powerful homotopy operator, which can be used as the original one, enables a declarative definition of arbitrary homotopy maps, and allows global convergence via probability-one homotopy, an established method from topology.

## 2 Theory

### 2.1 Definitions

We first define a generic nonlinear algebraic problem with a vector of unknowns  $\mathbf{z} = [\mathbf{x}_0; \mathbf{x}_0; \mathbf{w}_0]$  and residuals  $\mathbf{F} = [\mathbf{f}; \mathbf{g}]$ .

Then, a homotopy is a continuous deformation from one map to another via the homotopy map  $\rho(\mathbf{z}, \lambda)$ . The homotopy map is a map with one higher dimension as it additionally depends on  $\lambda$ , the homotopy or continuation parameter. The corresponding under-determined system of equations  $\rho(\mathbf{z}, \lambda) = \mathbf{0}$  can be followed using continuation algorithms.

Homotopy maps are carefully constructed such that for one value of the homotopy parameter, e.g.,  $\lambda = 0$ , the equation system is easy to solve and for another value, e.g.,  $\lambda = 1$ , the equation system is the one of interest, i.e.,  $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ .

Then, the root finding procedure works as follows. A curve  $(\mathbf{z}, 0)$  is followed from  $\rho(\mathbf{z}, 0) = \mathbf{0}$  along  $\rho(\mathbf{z}, \lambda) = \mathbf{0}$  until  $\lambda = 1$  as  $\rho(\mathbf{z}, 1) = \mathbf{F}(\mathbf{z})$ . This curve  $\{\mathbf{z} | \rho(\mathbf{z}, \lambda) = \mathbf{0}\}$  is called the root curve  $\rho^{-1}(\mathbf{0})$ .

Root curves have to be followed numerically and

therefore they must not contain singularities such as bifurcations or divergence to  $\pm\infty$ . Also, they must not be closed loops without crossing  $\lambda = 1$  (so called isolaes).

### 2.2 Probability-one homotopy

It is possible to prove that a problem satisfies these requirements using a particular type of homotopy method called probability-one homotopy. This method allows to avoid running into one of the ill-posed traces and thus delivers global convergence. It requires posing the problem equations  $\mathbf{F}$  and  $\rho$  in a particular fashion and was used with vast success in domain-specific simulation to resolve the convergence issues motivating this paper, in particular in analog electronic circuit simulation [14, 24, 16].

Informally, the key elements of probability-one homotopy are

- A well-defined random element to guarantee the full rank of the Jacobian matrix of  $\rho$ ,
- A boundedness argument, and
- An embedding, which essentially corresponds to the simplifications of component governing equations applied in [19, 1].

In order to summarize the supporting theory, transversality to zero [26] is defined.

**Definition 1.** Let  $U \subset \mathbb{R}^m$  and  $V \subset \mathbb{R}^n$  be open sets, and let  $\rho : U \times [0, 1] \times V \rightarrow \mathbb{R}^n$  be a  $C^2$  map.  $\rho$  is said to be transversal to zero if the Jacobian matrix  $\partial\rho$  has full rank on  $\rho^{-1}(\mathbf{0})$ .

Here,  $n = nx + nw$  holds. In the definition, an additional parameter dependency on a random vector  $\mathbf{a} \in \mathbb{R}^n$  is shown. This is the random element mentioned above. The Jacobian matrix of  $\rho$  is  $\partial\rho$ . It is a  $n \times (2n + 1)$  matrix and can be written as concatenation of three sub-matrices.

$$\partial\rho = \begin{bmatrix} \frac{\partial\rho}{\partial\mathbf{a}} & \frac{\partial\rho}{\partial\lambda} & \frac{\partial\rho}{\partial\mathbf{z}} \end{bmatrix} \quad (1)$$

Similarly to  $\rho^{-1}(\mathbf{0})$  introduced above, we can now consider  $\rho_{\mathbf{a}}^{-1}(\mathbf{0})$  as a set of root curves. Formally, we define it as follows.

$$\rho_{\mathbf{a}}^{-1}(\mathbf{0}) = \{(\mathbf{a}, \lambda, \mathbf{z}) | \mathbf{a} \in \mathbb{R}^n, \\ 0 \leq \lambda < 1, \\ \mathbf{z} \in \mathbb{R}^n, \\ \rho(\mathbf{a}, \lambda, \mathbf{z}) = \mathbf{0}\}$$

The following theorem, which is based on differential geometry and the Parametrized Sard's Theorem, is a generic formulation of probability-one homotopy methods.

**Theorem 1.** *Let  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  be a  $C^2$  map,  $\rho : \mathbb{R}^n \times [0, 1) \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  a  $C^2$  map, and  $\rho_{\mathbf{a}}(\lambda, \mathbf{z}) = \rho(\mathbf{a}, \lambda, \mathbf{z})$ . Suppose that*

1.  $\rho$  is transversal to zero, and, for each fixed  $\mathbf{a} \in \mathbb{R}^n$ ,
2.  $\rho_{\mathbf{a}}(0, \mathbf{z})$  has a unique nonsingular solution  $\mathbf{z}_0$ ,
3.  $\rho_{\mathbf{a}}(1, \mathbf{z}) = \mathbf{F}(\mathbf{z})$ .

*Then, for almost all  $\mathbf{a} \in \mathbb{R}^n$ , there exists a zero curve  $\Gamma_a$  of  $\rho_{\mathbf{a}}$  emanating from  $(0, \mathbf{z}_0)$ , along which the Jacobian matrix  $\partial \rho_{\mathbf{a}}$  has full rank. If, in addition,*

4.  $\rho_{\mathbf{a}}^{-1}(\mathbf{0})$  is bounded, then  $\Gamma_a$  reaches a point  $(1, \mathbf{z}^*)$  such that  $\mathbf{F}(\mathbf{z}^*) = \mathbf{0}$ . Furthermore, if  $\partial \mathbf{F}(\mathbf{z}^*)$  has full rank, then  $\Gamma_a$  has finite arc length.

This theorem is due to Watson [26] and is therefore called Watson's Theorem in this work. In order to apply this theorem, homotopy maps are constructed to meet prerequisites (2) and (3) by design. Prerequisite (1) may be trivial to verify for some homotopy maps and harder for others, in which  $\lambda$  and  $\mathbf{a}$  are involved nonlinearly. According to [27], prerequisite 4 may be hard to verify and often is a "deep result" as (1)–(4) holding implies the *existence* of a solution to  $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ .

A remark is in order on the statement of probability one. This characteristic of the theorem is inherited from the Parametrized Sard's Theorem and is motivated by probability of failure being 0 in the sense of a Lebesgue measure. Figuratively speaking, this means that the set of points leading to failure forms at most an  $n - 1$  dimensional manifold inside  $n$ -dimensional space, that is, it does not occupy any "volume".

Informally, Watson's Theorem can be understood as a statement on the probability of singularities along a continuation path. A bifurcation for instance may occur on a problem fulfilling this theorem. But a random variation of the parameter vector  $\mathbf{a}$  will be sufficient to avoid the singularity on a following attempt (with probability one). On problems with more than one solution, this choice of  $\mathbf{a}$  determines what solution the homotopy map converges to.

A number of additional theorems on probability-one homotopies are reviewed in [18]; herein, the given one shall suffice.

## 3 Implementation in Modelica Tools

### 3.1 Convergence proofs

For applications, the key issue is to show how a problem satisfies the given theorem. While some applications successfully utilize general physical principles such as conservation of energy (see [23] for instance), research by the authors [18] shows that it is not possible to generalize such proofs to arbitrary physical domains. This was mentioned in reference [19] already. Instead, problem-specific arguments are used in this contribution. They are introduced together with homotopy maps below.

Conceptually, they work via a general no-gain property, as exposed by, e.g., electric resistors, diodes, and transistors, and via saturation (an amplifier for instance has a constant gain only until the amplified signal reaches the supply voltage).

### 3.2 Declarative definition of arbitrary homotopy maps

Modelica is meant to allow a declarative problem description. That is one in which no information has to be provided on *how* to solve the problem. Instead, the problem itself is described. The solution algorithms are encapsulated in the language compilers and simulators.

In order to be useful for practitioners, the notion of problem-specific homotopy maps has to be integrated into simulation tools. The goal was thus to extend the declarative description to homotopies.

Using Modelica, one structures a model in terms of classes and objects. Therefore, it is proposed to specify a homotopy map  $\rho_{\mathbf{a}}(\mathbf{z}, \lambda)$  on the level of the equation set of the model classes, too.

In order to implement the suggested approach, it is proposed to utilize a built-in operator, `lambda()`<sup>1</sup>. Any expression involving  $\lambda$ , which describes the problem-specific homotopy map, can be written using the operator `lambda()`. It is used for each occurrence of variable  $\lambda$ . This operator may return a value in  $[0, 1]$  during the numeric solution of algebraic equation systems and strictly 1 during the generation of simulation results. If the operator `lambda()` is used without an argument then a *single-phase homotopy map* is implemented. If integer arguments are used then a

<sup>1</sup>Note that the previously proposed operator can be expressed using this operator. Furthermore, except for the cases with multi-stage homotopies, the previously proposed homotopy-operator can be used since `lambda()=homotopy(simplified=0, actual=1)`.

homotopy map is implemented, which consists of  $n$  phases, where  $n$  is the maximum over all arguments of the operator `lambda()`. For example, when using `lambda(1)` and `lambda(2)`, then a homotopy map is implemented in which  $\lambda_1$  values are first swept from 0 to 1. After this is finished,  $\lambda_2$  values are swept from 0 to 1.  $\lambda$  has to be swept from 0 to 1 during these sequential continuation runs of  $\lambda_i$  in order to infuse the random element required by theory<sup>2</sup>.

In order to simulate a given model efficiently, Mod-*elica* tools may apply symbolic preprocessing steps. A step that has to be considered in the context of homotopy is equation sorting. A typical example of a sorting algorithm used for equation-based, object-oriented modeling languages is the Block Lower Triangular (BLT) transformation [6], using a graph-theoretical algorithm by Tarjan [21]. Conceptually speaking, the continuation must be applied to the equation set as a whole. That is, all the equations that are either directly or indirectly influenced by the homotopy operator have to be solved simultaneously.

Note that if any of the probability-one homotopy theorems, such as the one introduced before, is fulfilled, then a large fraction of potential problems is avoided. For example, no singular Jacobian matrix at  $\lambda = 0$  can arise.

### 3.3 Test implementation

In order to validate the methodology, a test implementation was developed. It was based on the Modelica compiler Dymola<sup>®</sup> in versions 7.3 and 6.1. This test implementation utilized the LOCA continuation algorithms of Trilinos [8] and had the following properties.

- It provided three options for the treatment of the suggested homotopy operator. Normally, it was expanded according to a homotopy map. Alternatively, reduced equation sets were obtained by inlining the homotopy expression assuming  $\lambda_i = 1.0$  or  $\lambda_i = 0.0$ . In the latter case, maximum structural simplification of the equation system resulted.
- The user was able to manually prescribe whether to use homotopy initialization or not. This was

<sup>2</sup>When using the homotopy operator with integer arguments, several distinct continuation runs have to be started as the trajectories will in general not be smooth at the joining point of traces in any  $\lambda_i$  and  $\lambda_{i+1}$ . In general, the trajectories will be continuous but not differentiable. Even if a continuation algorithm manages to “hop over” such a joining point, starting continuation separately may be more efficient.

an important feature for library development and debugging, and may be useful for users, too.

- Verbose information on the homotopy was optionally provided, which was useful for library development and debugging. In particular, the homotopy traces were visualized. Like this, it was possible to reconstruct what happened during the solution of the simplified problems and the homotopy transformation.

Several implementation aspects such as automatic scaling and solver configuration via XML files have been described in [20, 18] and equally apply to this solver implementation.

## 4 Application Examples

As mentioned in the introduction, the use of probability-one homotopy is particularly well-developed in the area of analog electronic circuit simulation [23, 24, 25, 14, 13, 9, 7, 22, 11, 17, 29, 16]. First application examples are thus based on this work.

### 4.1 Operational amplifier $\mu A741$

The first example is an operational amplifier, which was discussed in [19] already. It uses bipolar junction transistors. Results are presented on probability-one homotopy using two different homotopy maps, the variable stimulus and the variable gain method. They are introduced next.

#### 4.1.1 Variable Stimulus

Melville et al. [14] proposed the Variable Stimulus Probability-One Homotopy. Its homotopy map is as follows.

$$\rho(\mathbf{z}, \lambda) = (1 - \lambda)G(\mathbf{z} - \mathbf{a}) + \mathbf{F}(\mathbf{z}, \lambda) \quad (2)$$

Here, the residual equations  $\mathbf{F}(\mathbf{z}, \lambda)$  are posed in the nodal analysis form [4] and the node voltages of the nonlinear elements are multiplied by  $\lambda$ . Therefore, the influence of the nonlinear elements is removed from the circuit at  $\lambda = 0.0$  and a linear circuit has to be solved. The matrix  $G$  defines the leakage from voltage sources of value  $\mathbf{a}$ . These voltage sources and the associated vector  $\mathbf{a}$  provide the random element needed in the probability-one approach. The leakage matrix  $G$  is a diagonal matrix with coefficients  $G_{\text{leak}}$ .

In order to substantiate that the Variable Stimulus Homotopy is globally convergent, Melville et al. [14] utilize Watson's Theorem as stated in section 2.2. Their arguments are as follows.

- The homotopy map (2) is twice continuously differentiable if and only if the device models used to assemble the residual equations in nodal form  $\mathbf{F}(\mathbf{z})$  are sufficiently smooth. It is assumed that this is fulfilled.
- The homotopy map  $\rho$  is transversal to zero as  $\partial\rho/\partial\mathbf{a}$  in (1) is a diagonal matrix with entries  $-(1-\lambda)\cdot G_{\text{leak}}$ . For  $\lambda < 1$ , this matrix has full rank.
- $\rho_{\mathbf{a}}(\mathbf{z}, 0)$  has a unique non-singular solution, because for  $\lambda = 0$  the circuit consists of resistors and voltage sources only. Such a linear problem has a unique non-singular solution.
- $\rho_{\mathbf{a}}(\mathbf{z}, 1) = \mathbf{F}(\mathbf{z})$  because the leakage circuitry is removed completely at  $\lambda = 1$  and each nonlinear device model is stimulated by the actual voltage.
- The zero set  $\rho_{\mathbf{a}}^{-1}(\mathbf{0})$  is bounded due to the no-gain property of the actual circuit and any partially stimulated circuit with leakage circuitry.

Additionally, Melville et al. [14] make the engineering assumption that the Jacobian of  $\rho_{\mathbf{a}}$  has full rank at the solution  $\mathbf{z}^*$ .

This Variable Stimulus Homotopy can be implemented on analog circuits using the proposed homotopy operator. First, a model of a NPN bipolar junction transistor is provided (see listing 1).

Here, three functions `iCollectorNpn()`, `iEmitterNpn()`, and `iBaseNpn()` are used to establish the collector, emitter, and base currents respectively. In order to implement the leakage circuitry, a model instance of a class is attached to each connection set (see listing 2).

Note the negative sign in front of the summation of the currents of the pins. This is necessary as the nodal analysis form [4] summarizes the currents going into the components attached to a node.

According to the experiments of Melville et al. [14], the solution trajectories of this homotopy are "much smoother" than those of the generic homotopy maps mentioned in section 2.1 of [19]. Additionally, "the action is spread out evenly over all values of  $\lambda$ ".

#### 4.1.2 Variable Gain

Melville et al. [14] also proposed the Variable Gain homotopy, which is similar to the Variable Stimulus homotopy but addresses bipolar transistors differently. Instead of multiplying the terminal voltages of all nonlinear elements by  $\lambda$ , the forward current gain  $\alpha_F$  and the reverse current gain  $\alpha_R$  are multiplied by  $\lambda$ . The simplified problem with  $\alpha_F = 0$  and  $\alpha_R = 0$  therefore consists of resistors, voltage sources, and diodes only.

$$\rho(\mathbf{z}, \lambda) = (1 - \lambda)G(\mathbf{z} - \mathbf{a}) + \mathbf{F}(\mathbf{z}, \lambda\alpha) \quad (3)$$

Again, the residual equations  $\mathbf{F}(\mathbf{z}, \lambda\alpha)$  are posed in the nodal analysis form [4]. Due to the diodes, the leakage circuitry is not necessary to avoid floating nodes. However, it is still included in this homotopy to provide the random element to avoid bifurcations [14].

Originally, the Variable Gain homotopy was implemented as a two-stage procedure. First, the Variable Stimulus homotopy was used to solve the  $\lambda = 0$  problem of the Variable Gain homotopy. Then, continuation was started on the Variable Gain homotopy map (3) and the actual problem was solved. Today, Variable Gain Homotopy is commonly understood as what was originally labeled the "hybrid approach" in reference [14]. A local gradient-based algorithm is used to solve the  $\lambda = 0$  problem and the continuation is applied directly on the Variable Gain homotopy map. The robust convergence of a local gradient-based algorithm on the  $\lambda = 0$  problem is justified by Melville et al. [14] in case of norm-reducing algorithms (algorithms using so-called globalizations) by the work of Duffin [5]. The single-stage procedure is "two to three times faster than using homotopy alone" [14].

In order to show that the Variable Gain Homotopy is globally convergent, Melville et al. [14] again utilize Watson's Theorem. Their arguments are as follows.

- As before, the homotopy map (3) is twice continuously differentiable if and only if the device models used to assemble the residual equations in nodal form  $\mathbf{F}(\mathbf{z})$  are sufficiently smooth. Again, it is assumed that this is fulfilled.
- The homotopy map  $\rho$  is transversal to zero as  $\partial\rho/\partial\mathbf{a}$  in (1) is a diagonal matrix with entries  $-(1-\lambda)\cdot G_{\text{leak}}$ . For  $\lambda < 1$ , this matrix has full rank.
- $\rho_{\mathbf{a}}(\mathbf{z}, 0)$  has a unique non-singular solution, because for  $\lambda = 0$  the circuit consists of resistors, voltage sources, and diodes only. Duffin [5] proved that such a problem has a unique solution.

```

1 model NPN
2 // Connectors
3 Modelica.Electrical.Analog.Interfaces.Pin C "Collector";
4 Modelica.Electrical.Analog.Interfaces.Pin B "Base";
5 Modelica.Electrical.Analog.Interfaces.Pin E "Emitter";
6
7 // Parameters
8 parameter Real af = 0.995 "Forward current gain";
9 parameter Real ar = 0.5 "Reverse current gain";
10
11 equation
12 C.i = iCollectorNpn(
13     lambda()*B.v, lambda()*C.v, lambda()*E.v, af, ar);
14 E.i = iEmitterNpn(
15     lambda()*B.v, lambda()*C.v, lambda()*E.v, af, ar);
16 B.i = iBaseNpn(
17     lambda()*B.v, lambda()*C.v, lambda()*E.v, af, ar);
18 end NPN;

```

Listing 1: NPN transistor model using variable stimulus

- $\rho_a(\mathbf{z}, 1) = \mathbf{F}(\mathbf{z})$  because the leakage circuitry is removed completely at  $\lambda = 1$  and each nonlinear device model uses the nominal forward and reverse current gains.
- The zero set  $\rho_a^{-1}(\mathbf{0})$  is bounded as Melville et al. [14] showed. This is due to the results of [28], which showed that bipolar transistors exhibit the no-gain property as long as the absolute values of the current gains remain less than or equal to one.

This Variable Gain Homotopy can be implemented on analog circuits using the proposed homotopy operator. Again, a model of a NPN bipolar junction transistors is given (see listing 3).

As before, three functions `iCollectorNpn()`, `iEmitterNpn()`, and `iBaseNpn()` are used to establish the collector, emitter, and base currents respectively. Instead of the terminal voltages, the current gains are multiplied with  $\lambda$ . The leakage circuitry can be implemented using model instances of the class listed in section 4.1.1 and is not repeated here.

According to Melville et al. [14], this is their fastest converging homotopy map. In particular, “the time required to solve a system of operating point equations with this homotopy [map] is not more than two to three times slower than the time required to solve the same equations by less widely convergent methods”.

### 4.1.3 Results

Figure 1 shows robustness profiles [20] of both homotopy-based solvers and a local gradient-based algorithm [15] for comparison. A robustness profile, in essence, shows the probability of convergence over the quality of a start iterate. The probability of convergence  $P_{conv}$  is estimated by sampling. The quality of the start iterate in turn is measured by a scaled distance of the start iterate  $\tilde{\mathbf{z}}$  to the next solution  $\tilde{\mathbf{s}}_j$ . Algorithms, which deliver constantly full probability of convergence, i.e.,  $P_{conv} = 1$ , independently of the quality of the start iterate, are called globally convergent herein.

## 4.2 Inverter Chain

This example involves Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs). The circuit it is based on is not found in practical devices, however, it can be scaled via the number of inverters in the chain. For the results discussed here,  $n = 50$  inverters were used. For MOSFETs, the ATANSH homotopy is currently state of the art.

### 4.2.1 Arc-Tangent Shichman-Hodges

The Arc-Tangent Shichman-Hodges or ATANSH model was proposed by Roychowdhury and Melville [16, 17] for probability-one homotopy and large-scale integrated circuits of metal-oxide

```

1 model ElectricalNode
2 // Connectors
3 parameter Integer n=0 "Number of pins"
4 annotation(Evaluate=true, Dialog(connectorSizing=true));
5 Modelica.Electrical.Analog.Interfaces.Pin pin[n] "Pin array";
6
7 // Parameters
8 parameter Real Gleak "Leakage";
9 parameter Real a "Random source voltage";
10
11 equation
12 0 = -sum(pin[:].i) + (1.0 - lambda())*Gleak*(pin[1].v-a);
13 for i in 1:n-1 loop
14   pin[i].v = pin[i+1].v;
15 end for;
16 end ElectricalNode;

```

Listing 2: Electrical node class

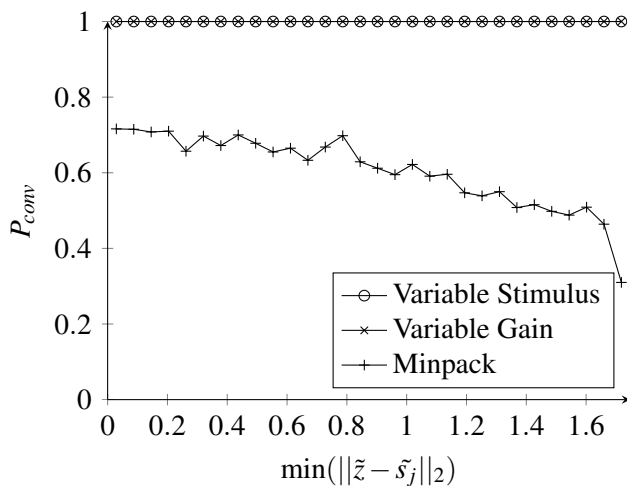


Figure 1: Robustness profiles [20] for Operational Amplifier 741 (60/60/1000 samples per bin)

semiconductor field-effect transistors. Conceptually, it is similar to the Variable Gain homotopy in that it varies key nonlinearity in component models. The ATANSH model uses two homotopy parameters  $\lambda_1$  and  $\lambda_2$ . Parameter  $\lambda_1$  influences the drain-source driving point characteristic without affecting the gain. Parameter  $\lambda_2$  in turn controls the transfer characteristic, i.e., the gain, without affecting the driving point characteristic.

$$\rho(\mathbf{z}, \lambda, \lambda_1, \lambda_2) = (1 - \lambda)G(\mathbf{z} - \mathbf{a}) + F(\mathbf{z}, \lambda_1, \lambda_2) \quad (4)$$

The ATANSH MOS homotopy model is a single-piece model. The drain-source current  $I_{ds}$  is given via

the following equation [16].

$$I_{ds} = \frac{\beta}{2} [V'_{gs}(V_{gb}, V_{db}, V_{sb}, \lambda_2, \lambda_1)]^2 \cdot h(V_{db} - V_{sb}, \lambda_1) \quad (5)$$

Roychowdhury and Melville [16, 17] remark that their probability-one homotopy map is a heuristic. In an attempt to justify its success, Watson's Theorem as stated in section 2.2 is considered.

- The homotopy map (4) is twice continuously differentiable if and only if the device models used to assemble the residual equations in nodal form  $\mathbf{F}(\mathbf{z})$  are sufficiently smooth. For the given MOS model this is fulfilled.
- The homotopy map  $\rho$  is transversal to zero as  $\partial\rho/\partial\mathbf{a}$  in (1) is a diagonal matrix with entries  $-(1 - \lambda) \cdot \text{Gleak}$ . For  $\lambda < 1$ , this matrix has full rank.
- $\rho_{\mathbf{a}}(\mathbf{z}, 0)$  has a unique non-singular solution, because for  $\lambda = 0$  the circuit consists of resistors, voltage sources, and simplified MOS transistors only. At  $\lambda_1 = 0$  and  $\lambda_2 = 0$  the simplified MOS devices become two-terminal almost-linear resistors. It is a reasonable engineering assumption to assume that such a problem has a unique non-singular solution.
- $\rho_{\mathbf{a}}(\mathbf{z}, 1) = \mathbf{F}(\mathbf{z})$  because the leakage circuitry is removed completely and each MOS device model is restored to its original form.

```

1 model NPN
2 // Connectors
3 Modelica.Electrical.Analog.Interfaces.Pin C "Collector";
4 Modelica.Electrical.Analog.Interfaces.Pin B "Base";
5 Modelica.Electrical.Analog.Interfaces.Pin E "Emitter";
6
7 // Parameters
8 parameter Real af = 0.995 "Forward current gain";
9 parameter Real ar = 0.5 "Reverse current gain";
10
11 equation
12 C.i = iCollectorNpn(
13     B.v, C.v, E.v, lambda()*af, lambda()*ar);
14 E.i = iEmitterNpn(
15     B.v, C.v, E.v, lambda()*af, lambda()*ar);
16 B.i = iBaseNpn(
17     B.v, C.v, E.v, lambda()*af, lambda()*ar);
18 end NPN;

```

Listing 3: NPN transistor model using variable gain

- The zero set  $\rho_a^{-1}(\mathbf{0})$  is bounded due to the no-gain property of the actual circuit and the simplified one with leakage circuitry and simplified MOS device models.

Additionally, one can make the engineering assumption that the Jacobian of  $\rho_a$  has full rank at the solution  $\mathbf{z}^*$ .

This MOS model for probability-one homotopy can be implemented using the proposed homotopy operator. Listing 4 illustrates this on an n-channel MOS transistor.

Function `idsNchannel()` implements equation (5) for this type of transistor. Note how the `lambda()` operator is used as described in section 3.2 with an integer argument. As Roychowdhury and Melville [16] first ramp  $\lambda_2$  and then  $\lambda_1$ , their homotopy is implemented using  $\lambda_2 = \text{lambda}(1)$  and  $\lambda_1 = \text{lambda}(2)$ . The leakage circuitry can be implemented using model instances of the class listed in section 4.1 and is not repeated here.

Roychowdhury and Melville [16, 17] report that local gradient-based algorithms are two to three times faster than the ATANSH homotopy on average *if they converge*. They additionally provide data to show however that the ATANSH homotopy took “considerably less time to obtain the DC operating point of the circuit than conventional methods took to give up” on their test cases. This illustrates that the extra wall time is an acceptable price to pay for robust convergence on large-scale problems.

## 4.2.2 Results

See figure 2 for results on using probability-one homotopy methods<sup>3</sup> and on using local gradient-based algorithms in comparison.

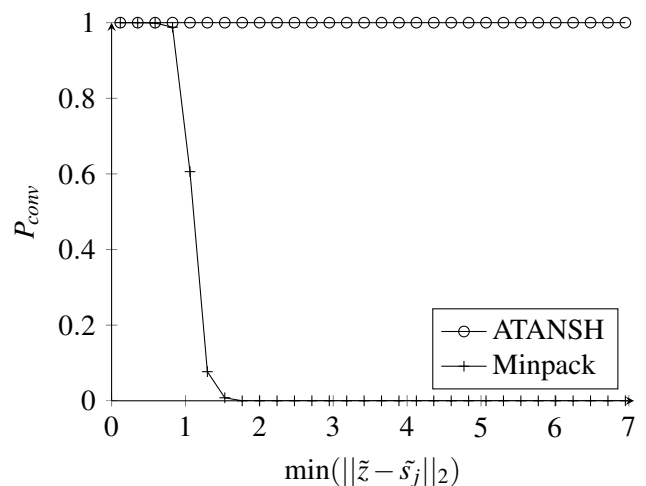


Figure 2: Robustness profiles [20] for Inverter Chain (60/1000 samples per bin)

<sup>3</sup>The ATANSH homotopy map cannot be compared to the variable gain or variable stimulus homotopy maps. The reason is that they are specific to a type of transistor, either the MOSFET or the BJT.



```

1 model NMOS
2 // Connectors
3 Modelica.Electrical.Analog.Interfaces.Pin G "Gate";
4 Modelica.Electrical.Analog.Interfaces.Pin D "Drain";
5 Modelica.Electrical.Analog.Interfaces.Pin S "Source";
6 Modelica.Electrical.Analog.Interfaces.Pin B "Bulk";
7
8 equation
9 // Drain-source current according to ATANSH
10 D.i = idsNchannel(G.v-B.v, D.v-B.v, S.v-B.v,
11     lambda(1), lambda(2));
12 S.i = -D.i;
13 // Gate, source
14 G.i = 0;
15 B.i = 0;
16 end NMOS;

```

Listing 4: MOS-FET model using ATANSH

### 4.3 Air distribution network

In this section, a basic but robust probability-one homotopy for thermo-fluid dynamic applications with unidirectional flow is introduced and applied to an Air Distribution test case. This is a thermo-hydraulic example with pipes transporting gases under wall friction and heat transfer, heat loads in cabin volumes, fans and so on. More details are given in [18].

#### 4.3.1 Unidirectional Thermofluid Probability-One Homotopy

The notion of a nodal approach for probability-one homotopy is adopted. Therefore, the mass and energy balances are addressed in this context. Pressure is a potential variable and thus the established approach of leakage circuitry used in sections 4.1.1, 4.1.2, and 4.2.1 can be applied trivially. Therefore, the components implementing the mass balance in the homotopy map are written in nodal form as follows.

$$\rho_{\text{hyd}}(\mathbf{z}_{\text{hyd}}, \mathbf{z}_{\text{th}}, \lambda) = (1 - \lambda) G_{\text{hyd}}(\mathbf{a}_{\text{hyd}} - \mathbf{z}_{\text{hyd}}) + \mathbf{F}_{\text{hyd}}(\mathbf{z}_{\text{hyd}}, \mathbf{z}_{\text{th}}, \lambda) \quad (6)$$

The subscript in  $\rho_{\text{hyd}}(\mathbf{z}_{\text{hyd}}, \mathbf{z}_{\text{th}}, \lambda)$  refers to the mass balance as hydraulic part. Consequently,  $\mathbf{z}_{\text{hyd}} = \mathbf{p}$ , i.e., the vector of unknowns of this part of the homotopy map is the vector of unknown pressures.  $G_{\text{hyd}}$  is the hydraulic leakage,  $\mathbf{a}_{\text{hyd}}$  is the vector of pressure values introducing the random element required by probability-one homotopy. The vector of residual equations  $\mathbf{F}_{\text{hyd}}(\mathbf{z}_{\text{hyd}}, \mathbf{z}_{\text{th}}, \lambda)$  for the hydraulic part are

the mass balances, that is the sums of the connection set mass flow rates. Of course these residual equations also depend on  $\mathbf{z}_{\text{th}}$ , the vector of thermal unknowns. These can be either temperatures or specific enthalpies. As it does only matter to the model of thermodynamic properties which one is used and all equations can be transformed accordingly, it is assumed without loss of generality that they correspond to temperature, i.e.,  $\mathbf{z}_{\text{th}} = \mathbf{T}$ .

For the thermal part the situation is more involved. As the temperatures or specific enthalpies  $z_{th}$  are not potentials (note that their values are not equal over all connectors in a connection set in the general case), a mechanistic application of the concept to the energy balance will fail. A modified nodal homotopy map component for the energy balance is written in the dimension of a specific enthalpy. It can equally be used with and without conduction (analogous to the leakage in the hydraulic part).

$$\rho_{\text{th}}(\mathbf{z}_{\text{hyd}}, \mathbf{z}_{\text{th}}, \lambda) = (1 - \lambda) G_{th}(\mathbf{a}_{th} - \mathbf{z}_{th}) + \mathbf{F}_{th}(\mathbf{z}_{\text{hyd}}, \mathbf{z}_{th}, \mathbf{a}_{th}, \lambda) \quad (7)$$

Here, the thermal node value  $\mathbf{z}_{th}$  is used in the specific enthalpy computation. The mass flow rate leaving the connection set is the sum of the mass flow rates over the outlet connectors plus the mass flow rate due to leakage in equation (6). The superscripts  $\pm$  on the mass flow rates indicate that they have been limited to a positive or negative epsilon flow using a  $C^2$  regularization.

The residual equations involving  $\lambda$  are as follows.

$$\mathbf{F}_{th}(\mathbf{z}_{hyd}, \mathbf{z}_{th}, \mathbf{a}_{th}, \lambda) = (1 - \lambda) h_{pT}(\mathbf{a}_{hyd}, \mathbf{a}_{th}) \quad (8)$$

$$+ \lambda \frac{\sum_{inlets} \dot{m}_i^+ \cdot h_i}{\sum_{inlets} \dot{m}_i^+}$$

$$- h_{pT}(\mathbf{z}_{hyd}, \mathbf{z}_{th})$$

The homotopy map has been established in terms of the connection set equations. Optionally, one may create embeddings in the device models. For wall friction correlations, a convex combination of a secant approximation through some operating point and the actual wall friction correlation was successfully tested. Heat transfer may be established equally based on secant approximations or even zero heat transfer at  $\lambda = 0$ .

In order to substantiate that the thermo-fluid homotopy is globally convergent, theorem 1 (Watson's Theorem) is applied. The arguments are as follows.

- The homotopy map based on components (6) and (7) is twice continuously differentiable if and only if the device models used to assemble the residual equations in nodal form  $\mathbf{F}(\mathbf{z})$  are sufficiently smooth. It is assumed that this is fulfilled.
- The homotopy map  $\rho$  is transversal to zero as  $\partial\rho/\partial\mathbf{a}$  with  $\mathbf{a} = [\mathbf{a}_{hyd}; \mathbf{a}_{th}]$  in (1) contains a diagonal matrix with entries  $-(1 - \lambda) \cdot G$  with  $G = [G_{hyd}; G_{th}]$  if a conductance is used. If the conductance is not used, i.e.,  $G_{th} = 0$ , then  $\partial\rho/\partial\mathbf{a}$  contains  $-(1 - \lambda) \cdot G_{hyd}$  for the hydraulic part. For the thermal part,  $\partial\rho/\partial\mathbf{a}$  contains  $(1 - \lambda) c_p$ . In any case  $\partial\rho/\partial\mathbf{a}$  and the Jacobian (1) have full rank for  $\lambda < 1$ .
- The homotopy map  $\rho_{\mathbf{a}}(\mathbf{z}, 0)$  has a unique non-singular solution, because for  $\lambda = 0$  the circuit consists of adiabatic linear pressure loss models and boundary conditions only. Such a problem has a unique non-singular solution.
- $\rho_{\mathbf{a}}(\mathbf{z}, 1) = \mathbf{F}(\mathbf{z})$  because the balance equations are restored completely at  $\lambda = 1$  and each device model exposes the actual behavior.
- For the hydraulic part, the zero set  $\rho_{\mathbf{a}}^{-1}(\mathbf{0})$  is bounded due to the no-gain property of the pressure loss correlations. See [18] for further details. For the thermal part, the zero set is bounded due to the Second Law of Thermodynamics<sup>4</sup>.

<sup>4</sup>At a first glance, one could argue that going from  $\lambda = 0$  to  $\lambda = 1$  is not necessarily "forward" in time. However, the Second Law is used here on a set of steady-state problems. Therefore, no issues arise from the "direction" of time.

The code for a model class to be instantiated in each connection set is given in listing 5. This node model implements the homotopy map on the thermodynamic balance equations of mass and energy, in particular, equation (6) in lines 30 and 31 and equation (7) in lines 42 to 47. The implementation of the device models is straight-forward. As an example, in listing 6, the steady-state part of a simple dynamic pipe model is presented (the transient equations do not matter for initialization and are thus omitted for readability).

### 4.3.2 Results

Figure 3 shows a robustness profile for the resulting unidirectional thermo-fluid dynamics probability-one homotopy. The results illustrate that the proposed homotopy map and the probability-one homotopy method provide robust convergence, even in light of large variations of the start iterate and random vector.

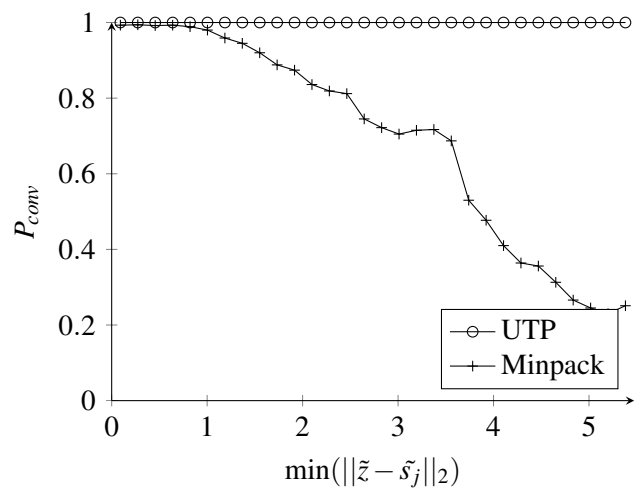


Figure 3: Robustness profiles for Air Distribution (60/1000 samples per bin)

## 5 Conclusions

The key result is that the theoretically predicted global convergence of probability-one homotopy can be realized in practice. This can be inferred from figures 1 to 3.

The associated coercivity proofs and the construction of underlying embeddings are rather involved however and require considerable understanding and a substantial investment in engineering time.

## References

- [1] F. Casella, L. Savoldelli, and M. Sielemann. Steady-state initialization of object-oriented thermo-fluid models by homotopy methods. In *Proceedings of Eighth International Modelica Conference*, Dresden, Germany, March 2011.
- [2] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM Classics in Applied Mathematics, 1996.
- [3] P. Deuffhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer Verlag, 2004.
- [4] P. Dimo. *Nodal analysis of power systems*. Taylor & Francis, 1975.
- [5] R. Duffin. Nonlinear networks IIa. *B. Am. Math. Soc.*, 53:963–971, 1947.
- [6] H. Elmqvist. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Lund University, Department of Automatic Control, Sweden, May 1978.
- [7] M. Green and R. Melville. Sufficient conditions for finding multiple operating points of dc circuits using continuation methods. In *IEEE International Symposium on Circuits and Systems*, pages 117–120, Seattle, 1995.
- [8] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *Acm. T. Math. Software.*, 31(3):397–423, 2005.
- [9] Y. Inoue. A practical algorithm for DC operating-point analysis of large-scale circuits. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 77(10):49–62, 1994.
- [10] C. T. Kelley. *Solving nonlinear equations with Newton's method*. SIAM Classics in Applied Mathematics, 2003.
- [11] W. Mathis, L. Trajkovic, M. Koch, and U. Feldmann. Parameter embedding methods for finding DC operating points of transistor circuits. In *Third international specialist workshop on Nonlinear Dynamics of Electronic Systems, NDES 1995*, pages 147–150, Dublin, Ireland, July 1995.
- [12] S. Mattsson, H. Elmqvist, M. Otter, and H. Olsson. Initialization of hybrid differential-algebraic equations in Modelica 2.0. In *Proceedings of the Second International Modelica Conference*, 2002.
- [13] R. Melville, S. Moinian, P. Feldmann, and L. Watson. Sframe: An efficient system for detailed DC simulation of bipolar analog integrated circuits using continuation methods. *Analog. Integr. Circ. S.*, 3(3):163–180, 1993.
- [14] R. C. Melville, L. Trajkovic, S.-C. Fang, and L. T. Watson. Artificial parameter homotopy methods for the DC operating point problem. *IEEE T. Comput. Aid. D.*, 12(6):861–877, June 1993.
- [15] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. User guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, 1980.
- [16] J. Roychowdhury and R. Melville. Delivering global DC convergence for large mixed-signal circuits via homotopy/continuation methods. *IEEE T. Comput. Aid. D.*, 25(1):66–78, January 2006.
- [17] J. S. Roychowdhury and R. C. Melville. Homotopy techniques for obtaining a DC solution of large-scale mos circuits. In *Proceedings of the 33rd Design Automation Conference*, pages 286–291, 1996.
- [18] M. Sielemann. *Device-Oriented Modeling and Simulation in Aircraft Energy Systems Design*. PhD thesis, Technical University of Hamburg-Harburg, Institute of Thermo-Fluid Dynamics, 2012.
- [19] M. Sielemann, F. Casella, M. Otter, C. Clauss, J. Eborn, S. Mattsson, and H. Olsson. Robust initialization of differential-algebraic equations using homotopy. In *Proceedings of Eighth International Modelica Conference*, Dresden, Germany, March 2011.
- [20] M. Sielemann and G. Schmitz. A quantitative metric for robustness of nonlinear algebraic equation solvers. *Math. Comput. Simulat.*, 81(12):2673–2687, 2011.

- [21] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1:146–160, 1972.
- [22] L. Trajkovic and W. Mathis. Parameter embedding methods for finding DC operating points: formulation and implementation. In *1995 International Symposium on Nonlinear Theory and its Applications, NOLTA 1995*, pages 1159–1164, Las Vegas NE, USA, December 1995.
- [23] L. Trajkovic, R. Melville, and S.-C. Fang. Passivity and no-gain properties establish global convergence of a homotopy method for DC operating points. In *IEEE International Symposium on Circuits and Systems*, volume 2, pages 914–917, May 1990.
- [24] L. Trajkovic, R. C. Melville, and S.-C. Fang. Finding DC operating points of transistor circuits using homotopy methods. In *Proc. IEEE Int Circuits and Systems Symposium*, pages 758–761, 1991.
- [25] L. Trajkovic, R. C. Melville, and S.-C. Fang. Improving DC convergence in a circuit simulator using a homotopy method. In *Proc. Custom Integrated Circuits Conf. the IEEE 1991*, 1991.
- [26] L. T. Watson. Globally convergent homotopy methods: A tutorial. *Appl. Math. Comput.*, 31:369–396, May 1989.
- [27] L. T. Watson. Probability-one homotopies in computational science. *J. Comput. Appl. Math.*, 140:785–807, 2002.
- [28] A. N. Willson Jr. The no-gain property for networks containing three-terminal elements. *IEEE T. Circuits. Syst.*, 22(8):678–687, August 1975.
- [29] K. Yamamura, T. Sekiguchi, and Y. Inoue. A fixed-point homotopy method for solving modified nodal equations. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 46(6):654–665, 1999.

```

1 model ThermoFluidDynamicsNode
2   replaceable package Medium = PartialPureSubstanceMedium;
3
4   // Connectors
5   parameter Integer nInlets = 0 "Number of inlets"
6     annotation(Evaluate=true, Dialog(connectorSizing=true));
7   parameter Integer nOutlets = 0 "Number of outlets"
8     annotation(Evaluate=true, Dialog(connectorSizing=true));
9   Modelica.Fluid.Interfaces.FluidPort_a inlet[nInlets](
10    redeclare package Medium = Medium);
11   Modelica.Fluid.Interfaces.FluidPort_b outlet[nOutlets](
12    redeclare package Medium = Medium);
13
14   // Parameters
15   parameter Medium.AbsolutePressure a_hyd "Random pressure";
16   parameter Medium.Temperature a_th "Random temperature";
17   parameter Real G_hyd "Leakage in hydraulic part"
18
19   // Variables
20   Medium.AbsolutePressure p "Pressure in node";
21   SI.MassFlowRate m_flow_plus[nInlets] "Limited inlet flow";
22 equation
23   // Hydraulic part
24   for i in 1:nInlets loop
25     inlet[i].p = p;
26   end for;
27   for i in 1:nOutlets loop
28     outlet[i].p = p;
29   end for;
30   0 = (1-lambda())*G_hyd*(a_hyd - p) +
31     sum(inlet[:].m_flow) + sum(outlet[:].m_flow);
32
33   // Thermal part, no conductance
34   for i in 1:nInlets loop
35     // Hypothetical case
36     inlet[i].h_outflow = Medium.h_default;
37   end for;
38   for i in 1:nOutlets loop
39     // Actual case
40     outlet[i].h_outflow = Medium.h_pT(p, T);
41   end for;
42   0 = ((1-lambda())*Medium.h_pT(a_hyd, a_th) +
43     lambda() * sum({
44       m_flow_plus[i]*
45       inStream(inlet[i].h_outflow) for i in 1:nInlets}
46     )/sum({m_flow_plus[i] for i in 1:nInlets}) -
47     Medium.h_pT(p, T));
48   m_flow_plus[:] = f(inlet[:].m_flow, ...);
49 end ThermoFluidDynamicsNode;

```

Listing 5: Thermo-fluid dynamics node class

```

1 model Pipe
2   replaceable package Medium = PartialPureSubstanceMedium;
3
4   // Connectors
5   Modelica.Fluid.Interfaces.FluidPort_a port_a[nInlets](
6     redeclare package Medium = Medium);
7   Modelica.Fluid.Interfaces.FluidPort_b port_b[nOutlets](
8     redeclare package Medium = Medium);
9
10  // Parameters
11  parameter SI.Length diameter "Pipe inside diameter";
12  parameter SI.Length length "Pipe length";
13  parameter SI.Length Delta "Surface roughness";
14  final parameter SI.Area heatTransferArea =
15    Modelica.Constants.pi*diameter*length;
16  parameter SI.Temperature T_amb "Ambient temperature";
17  parameter SI.Pressure dp_nominal "Nominal dp";
18
19  // Variables
20  SI.SpecificEnthalpy dh "Change of h over device"
21  SI.CoefficientOfHeatTransfer kc;
22  Real effectiveness "NTU effectiveness";
23  SI.Density rho "Upstream density";
24  SI.DynamicViscosity eta "Upstream dynamic viscosity";
25  SI.SpecificHeatCapacity cp "At constant pressure";
26  SI.ThermalConductivity lambda "Thermal conductivity";
27
28  equation
29    // Static mass balance
30    port_a.m_flow + port_b.m_flow = 0;
31
32    // Static energy balance
33    port_b.h_outflow = inStream(port_a.h_outflow) + dh;
34    port_a.h_outflow = Medium.h_default;
35
36    // Static momentum balance
37    m_flow =
38      lambda()*wallFriction_mflow_dp(dp, ...) +
39      (1-lambda())*dp/dp_nominal*
40      wallFriction_mflow_dp(dp_nominal, ...);
41
42    // Heat transfer
43    kc = heatTransfer_kc_mflow(m_flow, ...);
44    effectiveness = 1-exp(-(kc*heatTransferArea/(cp*m_flow)));
45    dh = lambda()*effectiveness*cp*(T_amb - state.T);
46
47    // Auxiliary equations for thermodynamic,
48    // transport properties
49    // ...
50  end Pipe;

```

Listing 6: Pipe model using UTP