Applied Mathematics and Computation 324 (2018) 105–118

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A symbolic computation framework for constitutive modelling based on entropy principles

A.F. Cheviakov[a], J. Heß[b,*]

[a] Department of Mathematics and Statistics, University of Saskatchewan, Saskatoon, Canada
[b] Department of Fluid Dynamics, Technical University of Darmstadt, Darmstadt, Germany

ARTICLE INFO

Keywords:
Constitutive modelling
Entropy principle
Symbolic computations

ABSTRACT

The entropy principle in the formulation of Müller and Liu is a common tool used in constitutive modelling for the development of restrictions on the unknown constitutive functions describing material properties of various physical continua.

In the current work, a symbolic software implementation of the Liu algorithm, based on `Maple` software and the `GeM` package, is presented. The computational framework is used to algorithmically perform technically demanding symbolic computations related to the entropy principle, to simplify and reduce Liu identities, and ultimately to derive explicit formulas describing classes of constitutive functions that do not violate the entropy principle. Detailed physical examples are presented and discussed.

1. Introduction

Entropy principles are used in continuum mechanics in order to investigate the material behavior. For a given model, the universal balance laws, such as those for mass, momentum and energy, are commonly given by a system of partial differential equations (PDEs). The specific material behavior is defined by a *constitutive model*, through the specification of *constitutive functions* present in the system. While the set of balance equations holds for a wide class of physical settings, for example, gases, ideal and non-ideal fluids, elastic and plastic solids, etc., the constitutive functions prescribe individual material behavior. From the mathematical point of view, they provide closure conditions for the system of balance equations, so that its fields can be uniquely determined. For an overview of constitutive modelling in the context of continuum mechanics, see, e.g., Hutter and Jöhnk's extensive work [1].

The principles of constitutive modelling may vary depending on the application; they can be based on theoretical considerations, experimental data, and/or heuristic assumptions. Fundamental theoretical principles for the formulation of material models include the requirements of *material objectivity*, *material symmetry* and *thermodynamic consistency*. The first requirement determines that material behavior, and therefore the constitutive equations, must be independent of the observer, that is, Galilei-invariant. The second rule points towards the fact that the material laws must also satisfy the symmetric properties of a body, such as invariance under rotations, translations, etc.

In this work, following the ideas of Müller [2,3] and Liu [4], we focus on the third requirement, which demands that the constitutive functions are restricted in such a way that an *entropy principle* holds for all solutions of the model, thus are consistent with thermodynamics. The entropy principle is formulated in terms of an entropy inequality expressing the

* Corresponding author.
E-mail addresses: shevyakov@math.usask.ca (A.F. Cheviakov), hess@fdy.tu-darmstadt.de (J. Heß).

https://doi.org/10.1016/j.amc.2017.12.004
0096-3003/

second law of thermodynamics (SL). The requirement that the entropy production inequality holds for every solution yields a set of constraints on the model's constitutive functions. Müller's approach is based on the fact that the generalized entropy inequality is linear with respect to a set of independent higher derivatives of field variables; the constraints arise as coefficients of such derivatives set to zero.

Multiple alternative approaches to constitutive modelling exist. For example, constitutive modelling approaches can also be based on the descriptions of material behavior by balance equations. For this, an additional class of fields is considered, called the *internal variables*, accounting for an internal state of the material. Such variables represent the microstructure and physical mechanisms within the body, and their evolution is described by respective balance equations. An example is given by the volume fraction (e.g., [5]), which accounts for the microstructure and the distribution of a granular material. In such approaches, the entropy principle of Müller and Liu provides an additional useful insight into the possible interdependence between those newly introduced internal variables and the general constitutive functions of a system.

Another approach used in micromechanical modelling is based on scale differences. There, microscale models are used to derive macroscale constitutive functions through the method of homogenization. This approach was employed, for example, in [6,7], to describe anistropic micropolar continua, i.e., structured solids, in the context of continuum mechanics, but without a reference to the entropy principle.

As there exist a wide range of definitions of entropy, there are also multiple entropy principles. With the *Maximum Entropy Principle*, coming from statistical mechanics, the parameter distribution for problems, in particular, in gas dynamics, can be derived. An example of an application of this principle to fluid dynamics is given in Ref. [8], while its connection with the Extended Thermodynamics for gases, established by Müller and Liu in [9], is discussed in Ref. [10]. In the context of continuum mechanics, it is important to mention the approach of Coleman and Noll [11] which provides an entropy principle based on the Clausius–Duhem inequality. Wang and Hutter [12] pointed out, however, that for mixtures, for structured continua and for polar continua like solids or liquid crystals, the entropy principle of Müller and Liu is to be favored. To the present day, the entropy principle of Müller and Liu is applied in many different fields of modelling, with applications ranging from chemical processes [13] to granular flows [14].

Liu [4] (also see [15]) systematized Müller's procedure, applying the method of Lagrange multipliers. Liu algorithm is significantly more general, and can be applied to a wide range of models, without the requirement that external supply terms be related. It can also be used for models that do not involve the physical entropy. In Section 2, we review the details of the problem of constitutive modelling based on entropy principles, and the main steps of Müller's approach and the Liu algorithm, illustrating them for a specific example of an anisotropic heat-conducting fluid.

The Liu algorithm [4] is based on the following lemma, formulated for linear algebraic equations and a linear inequality (see also [15] and [16]).

Lemma 1 (Liu). *Let $z \in \mathbb{R}^p$, and let M be a $p \times n$ real matrix. Consider a linear system $MY + z = 0$ of p equations on the components of the unknown vector $Y \in \mathbb{R}^n$, with a non-empty solution set S. Let also $\mu \in \mathbb{R}^n$, $\mu \neq 0$, and $\zeta \in \mathbb{R}$ be given. Then the following statements are equivalent:*

1. $\forall Y \in S$, $\mu^T Y + \zeta \geq 0$;
2. $\exists \lambda \in \mathbb{R}^p$ *such that* $\forall Y \in \mathbb{R}^n$, $\mu^T Y + \zeta - \lambda^T (MY + z) \geq 0$;
3. $\exists \lambda \in \mathbb{R}^p$ *such that* $\mu = M^T \lambda$, *and* $\zeta \geq \lambda^T z$.

As pointed out in [16], Lemma 1 is related to the fundamental inequality lemma of Farkas and Minkowski. The latter plays a key role in linear programming, and is in turn related to the general Hahn–Banach separation theorem.

Lemma 1 has a flavor different from the Lagrange multiplier approach to constrained optimization of nonlinear functions. In particular, both the equations and the inequality in the problem are linear. Moreover, the geometrical meaning of the Lemma can be understood as follows: since the inequality $F = \mu^T Y + \zeta \geq 0$ must hold for *all* points Y in the set defined by $MY + z = 0$, that set (a line, a hyperplane, etc.) must be, in a certain sense, *parallel* to isosurfaces $F = \text{const}$.

The technical computations related to the execution of the Müller–Liu procedure can be time consuming, and equations that arise tend to be quite lengthy. While simplest examples can be carried out within minutes by an experienced researcher, generally, the derivation of constraints on the constitutive functions can become error-prone, especially for complicated settings, such as mixture models involving multiple phases and internal variables, governed by additional balance equations, and/or having complex material behavior. One of the goals of the current paper is the application of modern symbolic software to facilitate computations related to the Liu algorithm, in particular, lengthy chain rule differentiations, computation of coefficients at higher-order derivatives, and efficient reduction and solution of overdetermined systems of partial differential equations for the unknown constitutive functions. The computations are based on a symbolic package `GeM` for `Maple`, developed in [17–19] for symmetry and conservation law computations. Significant similarities between the nature of those problems and the algorithm of Liu, and the capabilities of `GeM` software to efficiently handle linear and nonlinear PDEs, partial derivatives of field variables, and constitutive functions that may involve derivatives, make `Maple` and `GeM` a natural computational platform choice. It should be noted that this kind of treatment can be transferred to other applications in the context of constitutive modeling, for example to the aforementioned entropy principle of Coleman and Noll.

The rest of the paper is organized as follows. In Section 2, we review the main steps of the original Müller method and the Liu algorithm involving Lagrange multipliers to review their suggested principles of constitutive modelling. A running example of a model of a simple heat-conducting compressible anisotropic fluid (Section 2.2) is used to illustrate the stages.

Section 3 outlines the main stages of the symbolic computation algorithm. Two examples of symbolic computations of entropy principle constraints are subsequently considered: an elementary example of one-dimensional gas dynamics model (Section 3.1), and a model of simple heat-conducting compressible anisotropic fluid (Section 3.2).

The paper is concluded with a discussion Section 4 containing remarks about theoretical and computational aspects of entropy principles and research directions.

2. The entropy principle of Müller and Liu for constitutive modelling

2.1. The problem of constitutive modelling

Consider the motion of a physical continuum within the domain $\Omega \in \mathbb{R}^n$, $n \geq 1$. Specific instances include but are not limited to elastic media, ideal and non-ideal fluids, gases, and plasmas, as well as various kinds of mixtures, including those that undergo interactions and chemical reactions. The independent variables of the model are given by the time t and the spatial variables (commonly Cartesian coordinates) $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ (see also Section 4). Depending on the application, the spatial independent variables may be Eulerian or Lagrangian coordinates, the former choice being common in gas, fluid and plasma dynamics, and the latter choice in solid mechanics. In one, two and three spatial dimensions, the Cartesian spatial variables may be denoted, without confusion, by x, (x, y), and (x, y, z), respectively.

The physical parameters describing the continuum depend on time and spatial coordinates; they are given by the fields

$$\phi = (\phi_1(t, x), \dots, \phi_m(t, x)), \tag{2.1}$$

which are the dependent variables of the model. The evolution of these quantities is described by the PDEs

$$\Pi^{\phi} = \left\{ \Pi^{\phi_1}, \Pi^{\phi_2}, .., \Pi^{\phi_m} \right\} \tag{2.2}$$

which include the balance equations and possibly additional constraints (such as, for example, the condition of a fluid being irrotational, or a plasma being field-aligned, etc.).

In addition to independent and dependent variables, mathematical models of continua commonly involve arbitrary constitutive functions, which must be specified for any particular model through constitutive equations. Such equations provide the mathematical closure of the governing equations, describing the material behavior of a specific medium within the general class of materials whose dynamics obeys the balance equations Π^{ϕ} (2.2). As remarked in Müller's work [2], constitutive equations may be nonlocal in time, for example, they may depend on the history of the process. Similarly, constitutive models may involve spatial integral or delay-type terms, etc. In the simplest, most commonly considered situations, the constitutive functions ψ are given by local expressions

$$\psi = \psi(\phi_C), \tag{2.3}$$

depending on the set of variables ϕ_C that may include independent and dependent variables of the model, as well as possibly the derivatives of the dependent variables. The choice of the *constitutive class* ϕ_C is determined by the specific physical situation and the required material behavior. For example, the incorporation of temperature or density describes a material that is dependent on the distribution of mass and thermal energy; if the constitutive class involves the gradients of temperature or density, then the constitutive functions may depend on mass and thermal energy fluxes, etc. Furthermore, quantities like the velocity or the strain tensor components present in the constitutive class account for the mechanical, possibly anisotropic, behavior. In particular, for fluids, the dependence of constitutive functions on the strain tensor is related with viscous effects.

The main problem of constitutive modelling consists in a description of classes of dependencies (2.3), which affirm, or do not contradict, certain physical, mathematical, or philosophical principles, as well as possibly the available experimental data. The principles of constitutive modelling include:

- Coordinate invariance, i.e., invariance with respect to coordinate transformations. This property is also referred to as the principle of *material objectivity*; see, e.g., [1]. It compromises first of all the choice of the variables for ϕ_C, which, in order to be independent of the reference system, should not contain velocities, but only velocity differences.
- Material symmetry (e.g., [1]).
- Physical postulates and simplifying assumptions of physical and mathematical nature.

2.2. A Running example

Throughout the current section, as a running example, we will use the model of a simple heat-conducting compressible anisotropic fluid, following Müller [2] and Liu [4]. The main physical flow parameters are the density $\rho = \rho(x, t)$, the velocity $v = (v_1(x, t), v_2(x, t), v_3(x, t))$, and the temperature $\theta = \theta(x, t)$. The internal energy per unit mass, and the entropy per unit mass, are denoted $\epsilon = \epsilon(x, t)$ and $\eta = \eta(x, t)$. The components of $x = (x_1, x_2, x_3)$ are the Cartesian coordinates within a

physical domain $\mathcal{V} \subset \mathbb{R}^3$. The field equations express the balance of mass, momentum and internal energy, and are given by

$$\Pi^{\rho}: \ \frac{\partial \rho}{\partial t} + \frac{\partial \rho v_i}{\partial x_i} = 0, \tag{2.4a}$$

$$\Pi^{v}_{i}: \ \rho D_t v_i - \frac{\partial T_{ij}}{\partial x_j} - \rho g_i = 0, \qquad i = 1\ldots, 3, \tag{2.4b}$$

$$\Pi^{\epsilon}: \ \rho D_t \epsilon + \frac{\partial q_i}{\partial x_i} - T_{ij}\frac{\partial v_i}{\partial x_j} = \rho r, \tag{2.4c}$$

where ρr denotes the external energy supply density, and ρg_i is the body force due to gravity. Where convenient, for the sake of brevity, we will denote time and space partial derivatives by

$$\frac{\partial \rho}{\partial t} = \partial_t \rho = \rho_t, \qquad \frac{\partial \rho}{\partial x_j} = \partial_j \rho,$$

etc., and use the Einstein summation convention where appropriate. The material derivative operator is given by

$$D_t = \frac{\partial}{\partial t} + v_j \frac{\partial}{\partial x_j}. \tag{2.5}$$

In (2.4), $T_{ij} = T_{ij}(x, t)$ denotes the fluid stress tensor, and $q_i = q_i(x, t)$, $i = 1, \ldots, n$ represent the components of the outgoing energy flux $q(x, t)$. For certain classes of models given by Eq. (2.4), ρ, v and θ may play the role of dependent variables, and ϵ, T_{ij}, q_i the role of constitutive functions.

As a basis for the derivation of a set of mathematical constraints on admissible forms of the constitutive functions (2.3) of the given problem, as it will be explained below, one considers an *entropy inequality* expressing the Second Law of Thermodynamics (SL). For instance, for a medium described by the equations of motion and energy (2.4), it may be written in the local form

$$\Pi^{\eta}: \ \rho D_t \eta + \frac{\partial \Phi_i}{\partial x_i} - \rho s \geq 0, \tag{2.6}$$

where $\eta = \eta(x, t)$ and $s = s(x, t)$ denote respectively the entropy of the medium per unit mass, and the entropy supply (from external sources) per unit mass, and $\Phi = (\Phi_1(x, t), \ldots, \Phi_n(x, t))$ is the outgoing entropy flux vector.

As remarked in [16], in rational thermodynamics, a constitutive model (2.3) is considered flawed if there exists a solution of the field equations Π^{ϕ} that violates the SL. Condition (2.6), according to Müller [2], has to be interpreted as a restriction on the constitutive functions rather than as restriction on the processes that a body can possibly undergo.

The local entropy inequality (2.6) is related with a global formulation of the SL as follows. Consider a material domain $\mathcal{V}(t)$ moving with the fluid within the physical domain Ω. Suppose $\mathcal{V}(t)$ has a piecewise smooth boundary $\partial\,\mathcal{V}(t)$; it is defined by

$$\partial\,\mathcal{V}(t) = \{X(x, t) \mid D_t X(x, t) = 0\},$$

where $X(x, t)$ are macroscopic fluid particle labels (i.e., material, or Lagrangian coordinates) of the boundary points. The rate of change of the total entropy of the fluid in $\mathcal{V}(t)$ is given by

$$\frac{d}{dt}\int_{\mathcal{V}(t)} \rho\eta \, dV = \int_{\mathcal{V}(t)} \rho s \, dV - \oint_{\partial\,\mathcal{V}(t)} \Phi \cdot dA + Q_{\mathcal{V}}(t). \tag{2.7}$$

In (2.7), $Q_{\mathcal{V}}(t)$ denotes the additional rate of change of the entropy within the given volume due to all other physical effects that influence the amount of order in the system. Then the local form of the entropy inequality (2.6) provides a sufficient condition that for every material domain $\mathcal{V}(t)$, the additional rate of change $Q_{\mathcal{V}}(t)$ of the entropy is non-negative.

The requirement of Müller's principle is that the inequality (2.6) should hold for every admissible thermodynamic process, i.e., for every solution ρ, v_i, θ of (2.4). This requirement yields restrictions on the forms of the constitutive functions (2.3) of the problem. The specific entropy η and the entropy flux components Φ_i are regarded as additional unknown constitutive functions.

2.3. Müller's approach

In the context of continuum mechanics, Truesdell [20] proposed a set of heuristic postulates on the interdependence and mathematical specification of the distinct phases within a mixture, known as *mixture theory*. In particular, it is assumed that every point in space is simultaneously occupied by every phase, and that each phase is governed by the same balance laws as the mixture, amended by additional terms that account for interchanges between the phases. Müller [2], generalizing the previous work of Coleman and Noll [11], complemented these postulates by suggesting that for every process, i.e. the solutions of the balance laws, the constitutive functions of such a mixture (and its constituents) are restricted by the need

to obey the second law of thermodynamics. The latter was posed in terms of an *entropy inequality*, introduced in [21], expressing the fact that the entropy production in the system is nonnegative.

We now outline the main points of Müller [2], in order to summarize his approach, and to compare with the more general and systematic Müller-Liu algorithm that will follow. Please not that, for the sake of brevity, not mixtures but single phase systems are considered in the following, although the entropy principle is capable of dealing with multi-phase systems. For a recent example, see [14].

1. A mixture of substances involving several constituents is considered.
2. The classical PDEs describing the dynamics of the mixture in Eulerian coordinates (balance of mass, momentum, and internal energy, analogs of (2.4)) are written for each constituent, and for the mixture as a whole. The momentum equations involve the additional source term responsible for inter-species interactions; the energy supply term in the energy equation (2.4c) includes a radiation part ρr_R.
3. An entropy production inequality is written, analogously to Eq. (2.6) accounting for the entropy advection, spatial fluxes, and an external supply term. It is stated that physically relevant thermodynamic processes satisfy the entropy inequality.
4. The external entropy supply term in (2.6) is assumed to equal the external radiative energy supply divided by the temperature:
$$s = r_R/\theta. \tag{2.8}$$
(Condition (2.8) holds exactly for perfect gases, and is not generally true for other substances.)
5. The constitutive functions of the model, to be determined, include the energy and entropy densities (in terms of the specific free energy function ψ) and energy and entropy fluxes (encoded in the flux vectors k_i). Additionally, the stress tensor of the medium, the entropy and energy fluxes, and constituent mass production terms are also treated as unknown constitutive functions.
6. In the entropy production inequality, the external entropy supply term s is replaced by the terms of the energy balance equation (2.4c) using (2.8). In the energy balance equation, in turn, the inter-species interaction term is substituted through a similar term in the momentum balance equation. As a result, the entropy inequality becomes a linear combination involving essential parts of the energy and momentum equations. This form may be referred to as *the extended form of the entropy inequality*.
7. A form of constitutive functions, involving dependent variables of the problem and their specific partial derivatives, is assumed, and simplified according to Noll's principle of material objectivity.
8. Several additional simplifying assumptions on the form of constitutive functions are made, including linear dependence on certain higher derivatives of the dependent variables.
9. The constitutive functions are substituted in the entropy inequality in its extended form. In the result, which is linear in certain highest derivatives of the field variables, the corresponding terms are collected.
10. Since the extended entropy production inequality is required to hold for *all* solutions of the dynamic balance equations, the independent partial derivatives can assume any value. This allows one to set the corresponding coefficients to zero.
11. One consequently obtains an underdetermined set of partial differential equations on the unknown constitutive functions, providing restrictions on the previously posed forms of these constitutive functions.

2.4. The Liu algorithm

Müller's procedure (Section 2.3) has been extended first by the original author in [3], and then by Liu [4,15] through the consideration of a constrained entropy inequality and the use of Lagrange multipliers, leading to what is generally referred to as *the entropy principle of Müller and Liu*, or *the Liu algorithm*. In this approach, external supply terms are neglected; it is argued that they do not affect the material behavior. Instead of a sequence of substitutions that yields a linear combination of source-free energy and momentum equations within the entropy inequality, the Müller-Liu procedure yields a similar extended entropy inequality by adding to it a linear combination of the dynamic equations of the model. The application of Liu's lemma to the extended entropy inequality leads to the constraints on constitutive function forms. Even though formulated for algebraic equations, Lemma 1 is traditionally used to analyze entropy-type inequalities, provided that the model of interest is linear in some *parametric derivatives*. Since the subsequent work of Müller [3] is written in German, for further information a reader may be referred to the work of Hutter [22], which contains both a general overview of the method and a comparison with other methods.

The Liu algorithm can be utilized for models that do not necessarily have the "entropy" defined. Instead, one generally considers an inequality formulated for a scalar, additive and objective thermodynamic constitutive quantity η, which we still refer to as "entropy" below [16].

We now outline the main steps of the Liu algorithm, following the works of Liu [4,15] and Hausner & Kirchner [16], and illustrate them using the physical example of Section 2.2. The notation of Section 2.1 is used.

1. For a given physical model, define the fields of interest ϕ, and the dynamic PDEs Π^ϕ. In the running example, we have the dependent variables
$$\phi = (\rho, v, \theta),$$
and the governing equations (2.4) with zero source terms ($g_i, r = 0$).

2. Define the entropy inequality Π^η. For our example, it is given by Eq. (2.6) with zero source terms ($s = 0$).
3. For each of the governing scalar PDEs Π^ϕ, define a scalar Lagrange multiplier Λ^ϕ. Write down the extended entropy inequality

$$\widetilde{\Pi}^\eta : \ \Pi^\eta - \Lambda^\phi \Pi^\phi \geq 0. \tag{2.9}$$

In the running example, $\widetilde{\Pi}^\eta$ is given by

$$\begin{aligned} &\Pi^\eta - \Lambda^\rho \Pi^\rho - \Lambda_i^v \Pi_i^v - \Lambda^\epsilon \Pi^\epsilon \\ &= \rho \partial_t \eta + \rho v_i \partial_i \eta + \partial_i \Phi_i \\ &\quad - \Lambda^\rho (\partial_t \rho + \partial_i (\partial \rho v_i)) - \Lambda_i^v \left(\rho \, \partial_t v_i + \rho v_j \partial_j v_i - \partial_j T_{ij} \right) \\ &\quad - \Lambda^\epsilon \left(\rho \, \partial_t \epsilon + \rho v_i \partial_i \epsilon + \partial_i q_i - T_{ij} \partial_j v_i \right) \ \geq 0. \end{aligned} \tag{2.10}$$

4. Define the full set of constitutive functions ψ (2.3) of the model. This set includes the natural (physical) constitutive functions, additional entropy-related constitutive functions (entropy density and fluxes), and the multipliers Λ^ϕ. In the example, following Sections 7.2, 7.3 of Ref. [15], we let

$$\psi = (\epsilon, q_i, T_{ij}, \eta, \Phi_i, \Lambda^\rho, \Lambda_i^v, \Lambda^\epsilon). \tag{2.11}$$

5. Postulate the dependence (2.3) of the constitutive functions on the local variables, including certain field variables, as well as, possibly, their derivatives, and/or independent variables. (Depending on the application, there may be a substantial freedom of choice of the fields to be placed into ψ and ϕ_C.) In the running example, we choose a simplified ansatz

$$\phi_C = (\rho, \theta, \partial_i \theta). \tag{2.12}$$

6. Substitute the chosen constitutive function forms (2.3) into the extended entropy inequality $\widetilde{\Pi}^\eta$, carrying out the chain rule of differentiation for every constitutive function:

$$\frac{\partial \psi}{\partial t} = \sum_{\phi_C} \frac{\partial \psi}{\partial \phi_C} \frac{\partial \phi_C}{\partial t}, \qquad \frac{\partial \psi}{\partial x_i} = \sum_{\phi_C} \frac{\partial \psi}{\partial \phi_C} \frac{\partial \phi_C}{\partial x_i}. \tag{2.13}$$

Denote the obtained extended entropy inequality $\widetilde{\Pi}^\eta_{(1)}$.

7. Observe that the extended entropy inequality $\widetilde{\Pi}^\eta_{(1)}$ is linear with respect to a set of higher derivatives of the field variables ϕ. Take the widest set of such derivatives, excluding ones present in the constitutive dependencies ϕ_C. In the running example, this set of "arbitrary elements" is given by

$$Y = (\partial_t \rho, \partial_i \rho, \partial_t v_i, \partial_j v_i, \partial_t \theta, \partial_{t,i} \theta, \partial_{i,j} \theta). \tag{2.14}$$

8. Collect terms at $\widetilde{\Pi}^\eta_{(1)}$ with respect to the set Y of higher derivatives. In the example, one obtains

$$\begin{aligned} \widetilde{\Pi}^\eta_{(1)} : \quad & \frac{\partial \rho}{\partial t} \left(\rho \frac{\partial \eta}{\partial \rho} - \rho \Lambda^\epsilon \frac{\partial \epsilon}{\partial \rho} - \Lambda^\rho \right) \\ & + \frac{\partial \rho}{\partial x_i} \left(\rho v_i \frac{\partial \eta}{\partial \rho} - \Lambda^\epsilon \rho v_i \frac{\partial \epsilon}{\partial \rho} - \Lambda^\epsilon \frac{\partial q_i}{\partial \rho} + \Lambda_j^v \frac{\partial T_{ij}}{\partial \rho} + \frac{\partial \Phi_i}{\partial \rho} - \Lambda^\rho v_i \right) \\ & - \frac{\partial v_i}{\partial t} \Lambda_i^v \rho + \frac{\partial v_i}{\partial x_j} \left(\Lambda^\epsilon T_{ij} - \Lambda_i^v v_j \rho - \Lambda^\rho \rho \delta_{ij} \right) \\ & + \rho \frac{\partial \theta}{\partial t} \left(\frac{\partial \eta}{\partial \theta} - \Lambda^\epsilon \frac{\partial \epsilon}{\partial \theta} \right) \\ & + \rho \frac{\partial^2 \theta}{\partial t \, \partial x_i} \left(\frac{\partial \eta}{\partial (\partial_i \theta)} - \Lambda^\epsilon \frac{\partial \epsilon}{\partial (\partial_i \theta)} \right) \\ & + \frac{\partial^2 \theta}{\partial x_i \partial x_j} \left(\rho v_j \frac{\partial \eta}{\partial (\partial_i \theta)} - \Lambda^\epsilon \rho v_j \frac{\partial \epsilon}{\partial (\partial_i \theta)} - \Lambda^\epsilon \frac{\partial q_j}{\partial (\partial_i \theta)} + \Lambda_k^v \frac{\partial T_{jk}}{\partial (\partial_i \theta)} + \frac{\partial \Phi_j}{\partial (\partial_i \theta)} \right) \\ & + \frac{\partial \theta}{\partial x_i} \left(\rho v_i \frac{\partial \eta}{\partial \theta} - \Lambda^\epsilon \rho v_i \frac{\partial \epsilon}{\partial \theta} - \Lambda^\epsilon \frac{\partial q_i}{\partial \theta} + \Lambda_j^v \frac{\partial T_{ij}}{\partial \theta} + \frac{\partial \Phi_i}{\partial \theta} \right) \geq 0. \end{aligned} \tag{2.15}$$

9. In the spirit of Lemma 1, set to zero coefficients at the arbitrary elements Y. Obtain a set of *Liu identities* and a *residual inequality*. For the running example, the Liu identities are given by

$$\begin{aligned}
&\rho\frac{\partial\eta}{\partial\rho}-\rho\Lambda^{\epsilon}\frac{\partial\epsilon}{\partial\rho}-\Lambda^{\rho}=0,\\
&\rho v_i\frac{\partial\eta}{\partial\rho}-\Lambda^{\epsilon}\rho v_i\frac{\partial\epsilon}{\partial\rho}-\Lambda^{\epsilon}\frac{\partial q_i}{\partial\rho}+\Lambda^{v}_{j}\frac{\partial T_{ij}}{\partial\rho}+\frac{\partial\Phi_i}{\partial\rho}-\Lambda^{\rho}v_i=0;\\
&\Lambda^{v}_{i}\rho=0,\qquad \Lambda^{\epsilon}T_{ij}-\Lambda^{v}_{i}v_j\rho-\Lambda^{\rho}\rho\delta_{ij}=0,\\
&\frac{\partial\eta}{\partial\theta}-\Lambda^{\epsilon}\frac{\partial\epsilon}{\partial\theta}=0,\qquad \frac{\partial\eta}{\partial(\partial_i\theta)}-\Lambda^{\epsilon}\frac{\partial\epsilon}{\partial(\partial_i\theta)}=0,\\
&\rho v_j\frac{\partial\eta}{\partial(\partial_i\theta)}-\Lambda^{\epsilon}\rho v_j\frac{\partial\epsilon}{\partial(\partial_i\theta)}-\Lambda^{\epsilon}\frac{\partial q_j}{\partial(\partial_i\theta)}+\Lambda^{v}_{k}\frac{\partial T_{jk}}{\partial(\partial_i\theta)}+\frac{\partial\Phi_j}{\partial(\partial_i\theta)}=0,
\end{aligned}\tag{2.16}$$

and the residual inequality is given by

$$\frac{\partial\theta}{\partial x_i}\left(\rho v_i\frac{\partial\eta}{\partial\theta}-\Lambda^{\epsilon}\rho v_i\frac{\partial\epsilon}{\partial\theta}-\Lambda^{\epsilon}\frac{\partial q_i}{\partial\theta}+\Lambda^{v}_{j}\frac{\partial T_{ij}}{\partial\theta}+\frac{\partial\Phi_i}{\partial\theta}\right)\geq 0.\tag{2.17}$$

10. Solve the Liu identities to obtain constraints on the constitutive functions. For the example of a simple fluid, one obtains a set of constraints consistent with those presented in Ref. [15]. In particular, one has the following.
    - The requirement of isotropy: $T_{ij}=-p\delta_{ij}$, where $p=p(\rho,\theta)$ is the thermodynamic pressure.
    - The vanishing Lagrange multipliers $\Lambda^{v}_{i}=0$.
    - The classical form of the energy Lagrange multiplier $\Lambda^{\epsilon}=F(\theta)$. In particular, in many works including that of Müller, one has $\Lambda^{\epsilon}=1/\theta$ (see, e.g., [23]).
    - The form of the entropy mass density:

$$\eta=\frac{F^2}{F'}\int\frac{p}{\rho^2}\,d\rho+K,\tag{2.18}$$

      where $F=F(\theta)$, $K=K(\theta)$ are arbitrary functions, following from the total differential that is derived from (2.16).
    - The form of the energy mass density:

$$\epsilon=\frac{1}{F'}\int\frac{(pF)_\theta}{\rho^2}\,d\rho.\tag{2.19}$$

    - Relation between energy and entropy fluxes, which depend on the components of the temperature gradient.

    Here the functions p, F, K are arbitrary functions of the indicated arguments.

3. Constitutive modelling using the Liu algorithm: A symbolic implementation

The symbolic software package GeM for Maple contains routines for the computation of local conservation laws and Lie point and higher-order local symmetries of ordinary and partial differential equations and ODE/PDE systems [17,19,24]. The package can also be efficiently used for other computations that involve symbolic manipulation of differential equations (DE), their differential consequences, and related expressions involving independent and dependent variables, partial derivatives, arbitrary (constitutive) parameters, and arbitrary (constitutive) functions. In particular, equivalence transformations of DE families can be studied [25]. In conjunction with the standard Maple routine `rifsimp` for the reduction of overdetermined systems and case splitting, the GeM package has been successfully applied to many previously intractable problems of symmetry and conservation law analysis and classification.

The routines of the GeM package employ a computationally efficient representation of differential equations, through the conversion of the dependent variables and their derivatives to Maple symbols instead of functions or expressions. Then the chain rule for differential functions of dependent variables is simplified to standard partial derivative operations. For example, the Maple representation of a function $F(x, y, z)$ and its partial derivative by F_x in GeM routines is respectively `F` and `Fx`. Consequently, for constitutive functions that depend on functions (dependent variables and their derivatives), the differentiations significantly simplify. For example, if $H=H(F(x,y,z),F_x(x,y,z))$, then in the GeM representation, one has `H = H(F, Fx)`. Its partial derivative, for example, the one with respect to y, is represented as

$$\frac{\partial}{\partial y}H(F(x,y,z),F_x(x,y,z))=\texttt{diff(H(F, Fx), F)}*\texttt{Fy}+\texttt{diff(H(F, Fx), Fx)}*\texttt{Fxy},$$

and does not involve functions of functions. For details of data representation, routines, options, methods, and examples, see [17,19,25].

In the current work, the GeM package is uses to automate computations within the Liu algorithm (Section 2.4). The sequence of steps is outlined below, starting from an elementary example of one-dimensional gas flow, and continuing with the heat-conducting compressible anisotropic fluid (see Section 2.2).

3.1. Computational example 1: One-dimensional gas dynamics equations

As a first example, we consider a one-dimensional compressible gas flow, following the notation of Grossman [26]. The continuity, momentum, and energy equations, in the absence of external supply terms (i.e., under the adiabatic process assumption), are given by

$$\Pi^{\rho}: \ \rho_t + (\rho v)_x = 0, \tag{3.1a}$$

$$\Pi^{v}: \ \rho(v_t + v v_x) + p_x = 0, \tag{3.1b}$$

$$\Pi^{\epsilon}: \ \rho(\epsilon_t + v\epsilon_x) + p v_x = 0, \tag{3.1c}$$

where $\rho(t, x)$ is the mass density, $v(t, x)$ is the scalar spatial velocity in the x-direction, $p(t, x)$ is the scalar pressure, and $\epsilon(t, x)$ is the thermal energy per unit mass. An additional "internal" dependent variable not explicitly present in the model is the temperature $\theta(t, x)$.

Model (3.1) involves four unknowns and three equations, and thus requires a closure, a constitutive relation, which will be determined through an entropy principle. For simplicity, let us assume the gas to be calorically perfect, see [26], that is, satisfying

$$\epsilon = C_v \theta,$$

where $C_v = \text{const}$ is the specific heat at constant volume. We note the relation

$$C_v = \frac{i}{2}\tilde{R} = \frac{\tilde{R}}{\gamma - 1},$$

where i is the number of degrees of freedom of a gas molecule, $\gamma = (i+2)/i$ is the adiabatic exponent, $\tilde{R} = R/M$ is the specific gas constant, and R and M are the universal gas constant and the molar mass of the gas.

Following the Liu algorithm, we define the entropy inequality by

$$\Pi^{\eta}: \ \rho(\eta_t + v\eta_x) \geq 0, \tag{3.2}$$

with $\eta(t, x)$ denoting the mass density of entropy. As seen in [26,27], for an inviscid non-heat-conducting substance, the external supply terms in the energy and entropy equations are proportional, so it is valid to assume that they vanish simultaneously. The extended entropy inequality is given by

$$\widetilde{\Pi}^{\eta}: \ \Pi^{\eta} - \Lambda^{\rho}\Pi^{\rho} - \Lambda^{v}\Pi^{v} - \Lambda^{\epsilon}\Pi^{\epsilon} \geq 0. \tag{3.3}$$

We let ρ, v, ϵ play the role of dependent variables, and allow the five constitutive functions (2.11)

$$\psi = (p, \eta, \Lambda^{\rho}, \Lambda^{v}, \Lambda^{\epsilon}) \tag{3.4}$$

to depend on

$$\phi_C = (\rho, \epsilon). \tag{3.5}$$

The computations following the Liu algorithm proceed as follows.

Step A. Initialize. Clear the variables. Initialize the package.

```
restart :
read("d : /gem32_12.mpl") :
```
(3.6)

Step B. Declare variables and constitutive functions.

```
ind:=t,x; dep:=R(ind), V(ind), E(ind);
Constit_Dependence:=R(ind), E(ind);
Constit_F:=P(Constit_Dependence), S(Constit_Dependence),
           LR(Constit_Dependence), LV(Constit_Dependence),
           LE(Constit_Dependence) ;

gem_decl_vars(indeps=[ind], deps=[dep], freefunc=[Constit_F]);
```

Here we used the Maple notation $\rho = \mathtt{R}$, $p = \mathtt{P}$, $v = \mathtt{V}$, $\epsilon = \mathtt{E}$, $\eta = \mathtt{S}$ for the fields, and $\Lambda^{\rho} = \mathtt{LR}$ $\Lambda^{v} = \mathtt{LV}$, $\Lambda^{\epsilon} = \mathtt{LE}$ for the Lagrange multipliers. It is our common convention to use small letters for independent variables and capitals for dependent variables and constitutive functions.

Step C. Declare the model equations. PDEs (3.1) are defined as follows.

```
Pi_Rho:=diff(R(t, x), t) + diff(R(t, x)*V(t, x), x) = 0;
Pi_V:=R(t, x)*(diff(V(t, x), t) + V(t, x)*diff(V(t, x), x)
          + diff(P(Constit_Dependence),x)= 0;
Pi_E:=R(t, x)*(diff(E(t, x), t) + V(t, x)*diff(E(t, x), x)
          + P(Constit_Dependence)*diff(V(t, x),x)= 0;

gem_decl_eqs([Pi_Rho, Pi_V, Pi_E]);
```

The last command declares the given PDEs in terms of Maple symbols. The symbolic representation of the left-hand sides of the PDEs (without "= 0") is stored in the internal variables GEM_ALL_EQ_AN; they can be extracted as follows:

```
Eq_R_Symb:=GEM_ALL_EQ_AN[1];
Eq_U_Symb:=GEM_ALL_EQ_AN[2];
Eq_E_Symb:=GEM_ALL_EQ_AN[3];
```

For example, the density equation has the form Eq_R_Symb = Rt + R * Vx + Rx * V, with all derivatives of dependent variables replaced by Maple symbols.

Step D. The entropy inequality; the extended entropy inequality. The left-hand side of the entropy inequality (3.2) is defined as

```
Pi_S:= R(ind)*diff(S(Constit_Dependence), t)
     + R(ind)*V(ind)*diff(S(Constit_Dependence), x)
     + diff(Phi1(Constit_Dependence),x);
```

It is subsequently converted into a Maple expression using the GeM command

```
Eq_S_Symb:=gem_analyze(Pi_S);
```

The left-hand side of the extended entropy inequality (3.3) is obtained:

```
Entropy_Inequality:= Eq_S_Symb - LR(R, E)* Eq_R_Symb
                     - LV(R, E)* Eq_U_Symb - LE(R, E)* Eq_E_Symb;
```

this expression is linear in terms of the higher derivatives

$$Y = (\rho_t, \rho_x, v_t, v_x, \epsilon_t, \epsilon_x),$$

which are not parts of the constitutive dependence ϕ_C (3.5).

Step E. Obtain a system of constraints. According to the Liu's lemma, one now sets to zero coefficients at elements of Y, and obtains a split set of constraints, and a residual entropy inequality.

The independent terms in the extended entropy inequality can be collected as follows:

```
Y:= [ Rt, Rx, Vt, Vx, Et, Ex ];
collect(Entropy_Inequality, Y);
```

the output is

```
((diff(S(R, E), R))*R-LR(R, E))*Rt
+((diff(S(R, E), R))*R*V-LR(R, E)*V-LV(R, E)*(diff(P(R, E), R)))*Rx
-LV(R, E)*Vt*R+(-LV(R, E)*R*V-LR(R, E)*R-LE(R, E)*P(R, E))*Vx
+((diff(S(R, E), E))*R-LE(R, E)*R)*Et
+((diff(S(R, E), E))*R*V-LE(R, E)*R*V-LV(R, E)*(diff(P(R, E), E)))*Ex
```

The collect step is not necessary – it simply formats the extended entropy inequality as a sum of terms proportional to the elements of Y. Moreover, one observes that the residual entropy inequality is zero, since every term in the output above is proportional to an arbitrary element contained in Y.

The actual split set of constraints (Liu identities) is obtained by setting to zero the coefficients of the elements of Y, as follows:

```
coeffs_constraints:=[coeffs(Entropy_Inequality, Y)];
```

The six corresponding constraints are given by

$$\rho\Lambda^{\rho} + \rho v\Lambda^{v} + p\Lambda^{\epsilon} = 0; \qquad \rho v\frac{\partial\eta}{\partial\epsilon} - \Lambda^{v}\frac{\partial p}{\partial\epsilon} - \rho v\Lambda^{\epsilon} = 0;$$

$$\rho\frac{\partial\eta}{\partial\rho} - \Lambda^{\rho} = 0; \qquad v\left(\rho\frac{\partial\eta}{\partial\rho} - \Lambda^{\rho}\right) - \Lambda^{v}\frac{\partial p}{\partial\rho} = 0;$$

$$\rho\Lambda^{v} = 0; \qquad \rho\left(\frac{\partial\eta}{\partial\epsilon} - \Lambda^{\epsilon}\right) = 0.$$

Step F. Rif-simplify the constraints. At this optional step, one can use the `Maple rifsimp` routine for the Gröbner basis-based reduction of the overdetermined linear system of determining equations obtained at the previous step.

```
all_f := [P(R, E), S(R, E), LR(R, E), LV(R, E), LE(R, E)];
simplified_eqs := DEtools[rifsimp](coeffs_constraints, all_f, mindim = 1);
```

This step is particularly important for large systems of constraint equations, since through the elimination of the redundancy, it leads to a significant simplification and reduction of the number of PDEs in the set of constraints. The output of `rifsimp` is a `Maple` table. The simplified constraints are stored in `simplified\_eqs[Solved]`.

When used with `mindim = 1` option, the `rifsimp` routine determines the dimension of solution space without solving the DEs: `simplified\_eqs[dimension]`. The infinity value corresponds to the presence of arbitrary functions. Another option, `casesplit`, may be also useful in specific settings, when the most general situation needs to be avoided (such as in the case of an *a priori* vanishing constitutive function, etc.; see [17,19,24] and `Maple` help for details).

In our case, the `rifsimp` output `simplified\_eqs[Solved]` contains a simple set of constraints

$$\Lambda^{v} = 0, \qquad \frac{\partial\eta}{\partial\rho} = \frac{\Lambda^{\rho}}{\rho}, \qquad \frac{\partial\Lambda^{\epsilon}}{\partial\rho} = \frac{1}{\rho}\frac{\partial\Lambda^{\rho}}{\partial\epsilon}, \qquad \frac{\partial\eta}{\partial\epsilon} = \Lambda^{\epsilon}, \qquad p = -\rho\frac{\Lambda^{\rho}}{\Lambda^{\epsilon}}. \tag{3.7}$$

Step G. Solve the Liu identities. PDEs (3.7) can be readily solved by hand, or using the built-in `Maple` PDE solver:

```
pdsolve(simplified_eqs[Solved],all_f);
```

The resulting solution is given by

$$\begin{aligned} &\Lambda^{\rho}(\rho,\epsilon) = \rho\int F_{\rho}\,d\epsilon + G(\rho), \quad \Lambda^{v}(\rho,\epsilon) = 0, \quad \Lambda^{\epsilon}(\rho,\epsilon) = F(\rho,\epsilon),\\ &p(\rho,\epsilon) = -\rho\frac{\Lambda^{\rho}}{\Lambda^{\epsilon}}, \quad \eta = \int F\,d\epsilon + \int\frac{G}{\rho}\,d\rho, \end{aligned} \tag{3.8}$$

where $F(\rho,\epsilon)$ and $G(\rho)$ are arbitrary functions. Alternatively, one may treat $p = p(\rho,\epsilon)$ as an arbitrary function, and express the restrictions on η and the multipliers in its terms. Solution (3.8) is in agreement with Müller and Ruggeri [27]. Moreover, PDEs (3.7) yield a well-known formula

$$p = -\rho^2\frac{\eta_{\rho}}{\eta_{\epsilon}}. \tag{3.9}$$

It is easy to verify that indeed, for any $\eta(\rho,\epsilon)$ satisfying (3.9), the original entropy inequality Π^{η} (3.2) vanishes identically as long as (3.1a), (3.1c) are satisfied, that is, on all solutions of the given model. This is consistent with the initial assumption of the adiabaticity of the flow: the entropy of the system is conserved due to the absence of external energy and entropy supply terms.

As a special case, consider an ideal (thermally perfect) gas; then one has

$$p = \rho\tilde{R}\theta, \quad \eta = C_v\ln(\rho^{1-\gamma}\theta). \tag{3.10}$$

This yields a particular case of the solution (3.8), given by

$$p(\rho,\epsilon) = (\gamma-1)\rho\epsilon, \quad \eta(\rho,\epsilon) = C_v\ln\left(\frac{\epsilon}{C_v\,\rho^{\gamma-1}}\right),$$

$$\Lambda^{\rho}(\rho,\epsilon) = -\tilde{R}, \quad \Lambda^{v}(\rho,\epsilon) = 0, \quad \Lambda^{\epsilon}(\rho,\epsilon) = \frac{C_v}{\epsilon} = \frac{1}{\theta}.$$

In particular, the form of Λ^{ϵ} is a well-known result, which commonly arises in constitutive modelling with Lagrange multipliers.

3.2. Computational example 2: Heat-conducting anisotropic fluid

We now apply the same symbolic algorithm to the running example of a two-dimensional ($n = 2$) simple heat-conducting compressible anisotropic fluid (Section 2.2). The computations presented here are the ones that lead to the formulas discussed above in Section 2.4.

Step A. Initialize. The initialization proceeds using (3.6).

Step B. Declare variables and constitutive functions.

```
ind:=t,x,y; dep:=R(ind), U(ind), V(ind), W(ind);
gem_decl_vars(indeps=[ind], deps=[dep]);
```

The independent variables are given by t, $x_1 = \mathtt{x}$, $x_2 = \mathtt{y}$; the physical fields are the density $\rho = \mathtt{R}$, the velocity in the x-direction $u = \mathtt{U}$, the velocity in the y-direction $v = \mathtt{V}$, and the temperature $\theta = \mathtt{W}$. Unlike the previous example, here the temperature is considered a field, while the internal energy ϵ is treated as a constitutive function. The constitutive dependence ϕ_C and the constitutive functions ψ are defined according to Eqs. (2.11) and (2.12), using the commands

```
Constit_Dependence:=R(ind),W(ind),diff(W(ind),x),diff(W(ind),y);
Constit_F:=T11(Constit_Dependence), T12(Constit_Dependence),
           T22(Constit_Dependence),
           Q1(Constit_Dependence), Q2(Constit_Dependence),
           E(Constit_Dependence), S(Constit_Dependence),
           Phi1(Constit_Dependence), Phi2(Constit_Dependence),
           LR(Constit_Dependence), LU(Constit_Dependence),
           LV(Constit_Dependence), LE(Constit_Dependence) ;
```

Instead of an isotropic pressure, the current example features a symmetric stress tensor, with three independent entries denoted by $T_{11} = \mathtt{T11}$, $T_{12} = T_{21} = \mathtt{T12}$, $T_{22} = \mathtt{T22}$. The heat flux is denoted by $q_1 = \mathtt{Q1}$, $q_2 = \mathtt{Q2}$, the entropy and the internal energy by $\eta = \mathtt{S}$, $\epsilon = \mathtt{E}$, and the entropy flux components by $\Phi_1 = \mathtt{Phi1}$, $\Phi_2 = \mathtt{Phi2}$. The Maple expressions LR, LU, LV, LE denote the respective Lagrange multipliers. All these quantities are declared using the command

```
gem_decl_vars(indeps=[ind], deps=[dep], freefunc=[Constit_F]);
```

Step C. Declare the model equations. Before defining the balance equations, we introduce the symbolic material derivative operator (2.5):

```
MaterialDer:=Q->diff(Q,t)+U(ind)*diff(Q,x)+V(ind)*diff(Q,y);
```

The PDEs are entered as follows.

```
Pi_Rho:=diff(R(ind), t) + diff(R(ind)*U(ind), x)
        + diff(R(ind)*V(ind), y)= 0;
Pi_U:=R(ind)*MaterialDer(U(ind)) - diff(T11(Constit_Dependence),x)
          - diff(T12(Constit_Dependence),y)= 0;
Pi_V:=R(ind)*MaterialDer(V(ind)) - diff(T12(Constit_Dependence),x)
          - diff(T22(Constit_Dependence),y)= 0;
Pi_E:=R(ind)*MaterialDer(E(Constit_Dependence))
         +diff(Q1(Constit_Dependence),x)
         +diff(Q2(Constit_Dependence),y)
           -T11(Constit_Dependence)*diff(U(ind),x)
           -T12(Constit_Dependence)*diff(U(ind),y)
           -T12(Constit_Dependence)*diff(V(ind),x)
           -T22(Constit_Dependence)*diff(V(ind),y)=0;
```

The equations are declared and converted to Maple symbolic expressions using

```
gem_decl_eqs([Pi_Rho, Pi_U, Pi_V, Pi_E]);
```

and are copied into the variables

```
Eq_R_Symb:=GEM_ALL_EQ_AN[1];
Eq_U_Symb:=GEM_ALL_EQ_AN[2];
Eq_V_Symb:=GEM_ALL_EQ_AN[3];
Eq_E_Symb:=GEM_ALL_EQ_AN[4];
```

Step D. The entropy inequality; the extended entropy inequality. The left-hand side of the entropy inequality (2.6) ($s = 0$) is defined and converted into a Maple expression using the commands

```
Pi_S:= R(ind)*MaterialDer(S(Constit_Dependence)) +diff(Phi1(Constit_Dependence),x)
       +diff(Phi2(Constit_Dependence),y);
Eq_S_Symb:=gem_analyze(Pi_S);
```

The left-hand side of the extended entropy inequality (2.10) is given by

```
Entropy_Inequality:= Eq_S_Symb - LR(R, W, Wx, Wy)* Eq_R_Symb
                     - LU(R, W, Wx, Wy)* Eq_U_Symb
                     - LV(R, W, Wx, Wy)* Eq_V_Symb
                     - LE(R, W, Wx, Wy)* Eq_E_Symb;
```

Step E. Obtain a system of constraints. The independent terms in the extended entropy inequality can be isolated as follows:

```
Y:= [ Rt, Rx, Ry, Ut, Ux, Uy, Vt, Vx, Vy, Wt, Wxx, Wxy, Wyy, Wtx, Wty ];
collect(Entropy_Inequality, Y);
```

The split set of constraints, i.e., the Liu identities, is obtained as follows:

```
coeffs_constraints:=[coeffs(Entropy_Inequality, Y)];
```

It is given by the formulas (2.16). Note that the terms of the entropy inequality that involve no components of $|\mathrm{Y}|$ yield the residual entropy inequality.

Step F. Rif-simplify the constraints. The simplification step proceeds the same way as for the gas dynamics example. This time, following [4] we apply additional constraints in order to simplify the computations. In particular, we require that the Lagrange multiplier of the energy equation does not vanish, $\Lambda^{\epsilon} \neq 0$, and both the internal energy and entropy are functions of the temperature but not of the temperature gradient: i.e. $\eta = \eta(\theta, \rho)$ and $\epsilon = \epsilon(\epsilon, \rho)$. In order to achieve these restrictions, we append to the determining equations the conditions

$$\frac{\partial \eta}{\partial \theta_x} = \frac{\partial \eta}{\partial \theta_y} = \frac{\partial \epsilon}{\partial \theta_x} = \frac{\partial \epsilon}{\partial \theta_y} = 0, \qquad \Lambda^{\epsilon} \neq 0.$$

Then,

```
all_f := [T11(R, W, Wx, Wy), T12(R, W, Wx, Wy), T22(R, W, Wx, Wy),
          Q1(R, W, Wx, Wy), Q2(R, W, Wx, Wy),
          Phi1(R, W, Wx, Wy), Phi2(R, W, Wx, Wy),
          E(R, W, Wx, Wy), S(R, W, Wx, Wy),
          LR(R, W, Wx, Wy), LU(R, W, Wx, Wy),
          LV(R, W, Wx, Wy), LE(R, W, Wx, Wy)];
simplified_eqs := DEtools[rifsimp]([coeffs_constraints[],
   LE(R, W, Wx, Wy)<>0,
   diff(S(R, W, Wx, Wy),Wx)=0, diff(S(R, W, Wx, Wy),Wy)=0,
   diff(E(R, W, Wx, Wy),Wx)=0, diff(E(R, W, Wx, Wy),Wy)=0,
   diff(S(R, W, Wx, Wy),W)<>0, diff(E(R, W, Wx, Wy),W)<>0],
all_f, mindim=1);
simplified_eqs[Solved];
```

Step G. Solve the Liu identities. PDEs (3.7) are subsequently solved, for example, symbolically:

The output contains the results discussed in the end of Section 2.4. In particular, The Lagrange multipliers have the form

$$\Lambda_u = \Lambda_v = 0, \qquad \Lambda_\rho = H, \qquad \Lambda_\epsilon = \frac{1}{G}$$

```
pdsolve(simplified_eqs[Solved],all_f);
```

in terms of the arbitrary functions $G = G(\theta)$ and $H = H(\rho, \theta)$. For the stress components, one has

$$T_{12} = 0, \qquad T_{11} = T_{22} = \rho\, G H = -p,$$

that is, the requirement of fluid isotropy, and the form of the hydrostatic pressure $p = p(\rho, \theta)$ consistent with the entropy principle. The internal entropy and energy density forms are consequently given by Eqs. (2.18) and (2.19). The results are in agreement with [4].

4. Discussion and conclusions

In the present work, a CAS (computer algebra software) implementation of the entropy principle of Müller and Liu, a framework to derive thermodynamically consistent constraints for the constitutive functions of a system in continuum mechanics, has been presented and discussed. Two examples were analyzed: a one-dimensional gas dynamics model, and a model of a anisotropic compressible heat-conducting fluid. The entropy inequality and the extended entropy inequality were presented as `Maple` expressions in terms of the independent and dependent variables of the problem and their derivatives (treated as `Maple` symbols), and unknown constitutive functions (treated as functions of several simple scalar variables). Using routines of the `GeM` package, the coefficients at independent higher-order derivatives of field variables were automatically set to zero, yielding Liu identities. The latter were efficiently simplified and solved using standard `Maple` commands. The results obtained provided the required restrictions on the constitutive functions; these restrictions are physically meaningful and are in agreement with the literature.

The presented computational automation of the well-known and widely used Liu algorithm is particularly important since such computations, when done by hand, tend to be technically demanding, time-consuming, and prone to human error. For the examples considered in this work, the full set of computations performed on a standard desktop computer took seconds. The `Maple`-based symbolic framework developed in the current contribution can be applied with minimal changes to treat similar as well as more complicated models; the program sequence can also be naturally modified as necessary. Nonetheless, the presented scheme is not universally applicable to a model of arbitrary complexity. Steps A–E can be carried out efficiently for virtually any set of equations and modelling assumptions, however, steps F (simplification of Liu identities) and G (their solution) may not be computationally accessible if the dependency ϕ_C of the constitutive functions involves a large number of variables. Exact numbers that are symbolically tractable depends on the types of PDEs involved in the model. For example, in a similar `Maple`-based computation of local conservation laws of Euler equations [28], the unknown multipliers depended on 45 variables, including partial derivatives of the fields; however, the conservation law determining equations were substantially more overdetermined. The Liu identities in constitutive modelling are usually less overdetermined (for example, six conditions on five functions in Computational Example 1, Section 3.1; 16 conditions on 13 functions in Computational Example 2, Section 3.2). In particular, Liu identities never include conditions on the derivatives of the Lagrange multipliers Λ^{ϕ}.

Several research directions aimed at the extension of the symbolic algorithm outlined in this work can be named.

1. The presented procedure can be generalized to aid in constitutive modelling for *thermodynamic equilibria*, the states of minimal (i.e., zero) entropy production. In these states, additional equations are formulated that serve as constraints for the equilibrium parts of the unknown constitutive functions [1], based on the residual entropy inequality.
2. Depending on the model, the solution of Liu identities may not lead to one set of closed-form expressions for constitutive functions. In related symbolic computations, the `rifsimp` routine used without the `casesplit` option, as done in the above example, returns the simplified equations in the most general case, when no pivot coefficient is zero. Pivoting based on zero or nonzero values of certain coefficients may lead to different cases, for example, more general forms of some constitutive functions and less general forms of the other ones. The `casesplit` option of the `Maple` `rifsimp` routine may be useful to produce investigate *trees* of such special cases. Indeed, in specific physical situations, assumptions that some coefficient (such as a Lagrange multiplier or its certain partial derivative) *does* vanish may lead to additional physical solutions. The `caseplot` routine for case tree plotting, with its ability display of zero and nonzero pivots for the cases, may then be used to visualize possible case trees that arise.
3. The choice of the constitutive dependencies (2.3) may have a significant effect on both the results themselves and the related symbolic computations. We note the importance of choosing a *coordinate-invariant* approach. Indeed, if one allows the constitutive functions $\psi = \psi(\phi_C)$ (2.3) to depend on spatial partial derivatives, the dependence should in fact involve only coordinate-invariant combinations of such derivatives, such as a divergence, norm of a gradient, etc., but not, for example, Cartesian components of a gradient. Such choices would ensure the consistency of the application of the procedure in different coordinate systems. In particular, when computations are done in curvilinear coordinates, one needs to explicitly include into ϕ_C the spatial variables, to accommodate for the scaling (Lamé) factors that will appear in coordinate-invariant differential operators. The importance of this coordinate invariance requirement is illustrated in Ref. [29]; in contrast, the original work of Liu [4,15] does not mention this aspect. The efficient computational treatment of cases in which symmetric tensor relations emerge is an important direction of the future research.

It is also of interest to investigate the mathematical details related to Liu's lemma itself, when it is applied to a set of nonlinear partial differential equations. The standard Liu algorithm described in Section 2.4 above attempts to apply Lemma 1 to move away from the solution set of the given PDE system (2.2), and consider the vector Y of higher-order partial derivatives of field variables as a set of arbitrary values. However, it is not completely clear how the linear algebra-based Lemma 1 applies to PDEs. Indeed, there is a substantial difference between linear equations and inequalities, where solutions and isosurfaces are given by linear or affine spaces, and solution manifolds of nonlinear differential equations in the jet spaces (see, e.g., [30]). It is normally assumed that the considered PDE systems are *locally solvable*, i.e., the solution set of the PDE system in jet space is actually represented by these PDEs. In addition to the model PDEs (2.2), further relations between the field variables are provided by their *differential consequences*. It follows that if one was to consistently follow through with the Liu's approach, the extended entropy inequality $\widetilde{\Pi}^{\eta}$ would possibly need to include the differential consequences of the equations Π^{ϕ} (2.2), with additional multipliers. For example, for the heat-conducting fluid model (2.4), if one extends the constitutive dependence ϕ_C (2.12) to include $\partial_t \rho$, it is clear that the time derivative $\partial_t \epsilon$ in the energy balance will include the second derivative $\partial_t^2 \rho$. It follows that the entropy condition (2.10) should include an additional term $\Lambda_1^{\rho}\, \partial_t \Pi^{\rho}$ involving a time differential consequence of the continuity equation. In a similar way, spatial differential consequences $\partial_{x_i} \Pi^{\rho}$ would be added, with appropriate Lagrange multipliers.

In a related forthcoming publication, we will present a generalization of Liu algorithm, which considers the entropy inequality *a priori* on the solution set of the dynamic equations of a given model, thus requiring no Lagrange multipliers, and respecting the nonlinear nature of the solution set of the model.

Acknowledgement

The authors thank Y. Wang and M. Oberlack for the ideas and discussions related to this project, and are also grateful for the financial support through the NSERC Discovery grant RGPIN-2014-05733, and the Deutsche Forschungsgemeinschaft (DFG) project WA 2610/3-1.

References

[1] K. Hutter, K. Jöhnk, Continuum Methods of Physical Modeling, Springer, 2004.
[2] I. Müller, A thermodynamic theory of mixtures of fluids, Arch. Ration. Mech. Anal. 28 (1) (1968) 1–39.
[3] I. Müller, Die Kältefunktion, eine universelle Funktion in der Thermodynamik viskoser wärmeleitender Flüssigkeiten, Arch. Ration. Mech. Anal. 40 (1) (1971) 1–36.
[4] I.-S. Liu, Method of Lagrange multipliers for exploitation of the entropy principle, Arch. Ration. Mech. Anal. 46 (2) (1972) 131–148.
[5] M. Goodman, S. Cowin, A continuum theory for granular materials, Arch. Ration. Mech. Anal. 44 (4) (1972) 249–266.
[6] I. Goda, M. Assidi, S. Belouettar, J. Ganghoffer, A micropolar anisotropic constitutive model of cancellous bone from discrete homogenization, J. Mech. Behav. Biomed. Mater. 16 (2012) 87–108.
[7] I. Goda, M. Assidi, J.-F. Ganghoffer, A 3D elastic micropolar model of vertebral trabecular bone from lattice homogenization of the bone microstructure, Biomech. Model. Mechanobiol. 13 (1) (2014) 53–83.
[8] R. Robert, A maximum-entropy principle for two-dimensional perfect fluid dynamics, J. Stat. Phys. 65 (3–4) (1991) 531–553.
[9] L. I-Shih, I. Müller, Extended thermodynamics of classical and degenerate gases, Arch. Ration. Mech. Anal. 83 (1983) 285.
[10] W. Dreyer, M. Kunik, Maximum entropy principle revisited, Contin. Mech. Thermodyn. 10 (6) (1998) 331–347.
[11] B.D. Coleman, W. Noll, The thermodynamics of elastic materials with heat conduction and viscosity, Arch. Ration. Mech. Anal. 13 (1) (1963) 167–178.
[12] Y. Wang, K. Hutter, Comparison of two entropy principles and their applications in granular flows with / without fluid, Arch. Mech. 51 (5) (1999) 605–632.
[13] M.C. Reis, Y. Wang, A two-fluid model for reactive dilute solid–liquid mixtures with phase changes, Contin. Mech. Thermodyn. 29 (2) (2016) 1–26.
[14] J. Heß, Y. Wang, K. Hutter, Thermodynamically consistent modeling of granular-fluid mixtures incorporating pore pressure evolution and hypoplastic behavior, Contin. Mech. Thermodyn. 29 (1) (2017) 311–343.
[15] I.S. Liu, Continuum Mechanics, Springer, 2002.
[16] R. Hauser, N. Kirchner, A historical note on the entropy principle of Müller and Liu, Contin. Mech. Thermodyn. 14 (2) (2002) 223–226.
[17] A.F. Cheviakov, GeM software package for computation of symmetries and conservation laws of differential equations, Comput. Phys. Commun. 176 (1) (2007) 48–61.
[18] A.F. Cheviakov, Computation of fluxes of conservation laws, J. Eng. Math. 66 (1–3) (2010a) 153–173.
[19] A.F. Cheviakov, Symbolic computation of local symmetries of nonlinear and linear partial and ordinary differential equations, Math. Comput. Sci. 4 (2–3) (2010b) 203–222.
[20] C. Truesdell, Mechanical basis of diffusion, J. Chem. Phys. 37 (10) (1962) 2336–2344.
[21] I. Müller, On the entropy inequality, Arch. Ration. Mech. Anal. 26 (2) (1967) 118–141.
[22] K. Hutter, The foundations of thermodynamics, its basic postulates and implications. A review of modern thermodynamics, Acta Mech. 27 (1–4) (1977) 1–54.
[23] I. Müller, I.-S. Liu, Thermodynamics of mixtures of fluids, in: C. Truesdell (Ed.), Rational Thermodynamics, Springer, 1984.
[24] GeM software package, examples, and description, (http://math.usask.ca/~shevyakov/gem/).
[25] A.F. Cheviakov, Symbolic computation of equivalence transformations and parameter reduction for nonlinear physical models, Comput. Phys. Commun. 220 (Supplement C) (2017b) 56–73, doi:10.1016/j.cpc.2017.06.013.
[26] B. Grossman, Fundamental concepts of real gas dynamics, Technical Report, Virginia Polytechnic Institute and State University, 2000.
[27] I. Müller, T. Ruggeri, Rational Extended Thermodynamics, Springer, 1998.
[28] A.F. Cheviakov, M. Oberlack, Generalized Ertel's theorem and infinite hierarchies of conserved quantities for three-dimensional time-dependent Euler and Navier–Stokes equations, J. Fluid Mech. 760 (2014) 368–386.
[29] Y. Wang, K. Hutter, Shearing flows in a Goodman–Cowin type material – theory and numerical results, Part. Sci. Technol. 17 (1) (1999) 97–124.
[30] P.J. Olver, Applications of Lie Groups to Differential Equations, Springer, 2000.