NUMERICAL INVESTIGATION OF SECOND SOUND IN LIQUID HELIUM

by

Udo Piram

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
In the Graduate College
THE UNIVERSITY OF ARIZONA
In Partial Fulfillment of the Requirements
For the Degree of
DIPLOMINGENIEUR
at the
UNIVERSITY STUTTGART

1991

ACKNOWLEDGMENTS

I would like to thank Dr. François Cellier for guidance and assistance during my studies at the University of Arizona and the preparation of this paper.

I would also like to thank Prof. Dr. Voß at the University Stuttgart, who organized the exchange program, which made it possible for me to attend school at the University of Arizona. I would like to thank the German Academic Exchange Service for the financial support of the exchange program.

Finally, I would like to express my gratitude to my parents for their support of my studies in Germany and the USA.

Contents

List of Figures

Abstract

At temperatures below 2.17K the phenomenon of thermal waves has been observed in liquid Helium. This paper describes the reproduction of these waves, called “Second Sound,” by means of simulation. Two different models for the heat transport in Helium are tried. First the classical heat diffusion equation is used with temperature dependent thermal parameters. The second model for heat transport is based on the more complex Two Fluid Model which was introduced for Helium close to the absolute zero point of temperature. Simulations with various initial and boundary conditions show that the first model is not sufficient to describe heat transport in Helium, whereas the second model gives satisfactory results.

Abstract

Das Phänomen von thermischen Wellen wurde in flüssigem Helium unter 2.17K beobachtet. Diese Arbeit beschreibt die Reproduktion dieser Wellen, die ,,Second Sound" genannt werden, durch Simulation. Zwei verschiedene Modelle für den Wärmetransport wurden angewandt. Als erstes wurde die klassische Wärmeleitgleichung mit temperaturabhängigen thermischen Parametern verwendet. Das zweite Modell für den Wärmetransport basiert auf dem komplizierteren ,,Zwei–Flüssigkeiten"–Modell, das für Helium nahe dem absoluten Nullpunkt eingeführt wurde. Simulationen mit verschiedenen Anfangs– und Randbedingungen zeigen, daß das erste Modell den Wärmetransport in Helium nicht zufriedenstellend beschreibt, während das zweite Modell zufriedenstellende Ergebnisse liefert.

Chapter 1

Introduction

1.1 Mechanical and Electrical Systems

Systems can often be decomposed into several interconnected subsystems. In order to perform linear oscillations, a linear system must contain several elements that can store energy. A typical example is the mechanical mass–spring oscillator with the displacement velocity v used as state variable [Fig.1.1]. The moving mass can store energy in the form of kinetic energy, and the deformed spring stores potential energy. The damper is a different kind of element, since it is not able to store energy. It "consumes" energy in the sense that it takes energy out of the mechanical system. During oscillation, energy is alternately stored in the mass or the spring. If we removed the spring in Fig.1.1 and gave the mass an initial velocity v_0, the mass would move for a while before coming to a

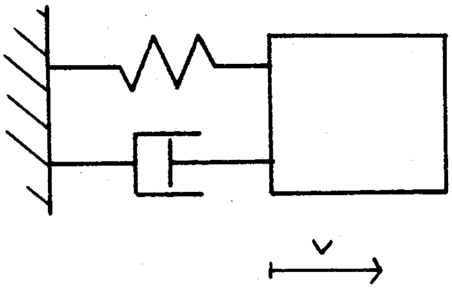

Figure 1.1: Mass–Spring Oscillator

stop, but would never reverse its direction, no oscillation is possible. For all times, the energy of the system is stored in the mass. If we built a chain consisting of dampers and masses and gave every mass an initial velocity, again the same behavior could be observed; the masses coming to a stop somewhere, but never changing their direction. Hence, two *different kinds* of storage elements are needed for linear oscillations to occur.

The same experiments can also be made with electrical systems, where the capacitor and the inductance are storage elements and the resistor is the dissipative element.

1.2 Thermodynamical Systems

Thermodynamical Systems are commonly described in terms of temperature and heat or entropy currents. The thermal capacitance of a material describes the capability of storing energy in the medium. The overall flow of entropy into an element determines the increase of temperature, so heat is stored. The relation of entropy flow between two elements with different temperatures is described as thermal conduction. This flow is a resistive effect.

We have now one storage element which can be classified as capacitive. There is no inductive storage element in thermal systems. Let us assume that such an element exists, this would mean that we can store entropy flow. Therefore, there might be an entropy flow even without a temperature difference. This is in contradiction to the second law of thermodynamics which states that a spontaneous heat or entropy flow is always directed from a point of higher temperature to a point of lower temperature. Hence, if there is an entropy flow, there must be a temperature difference. Since the previously made assumption is wrong, there is no thermodynamic inductive element.

In 1.1, we found that there have to be two different kind of storage elements in a system in order for it to oscillate. Since thermodynamical systems only have a capacitive storage element, we conclude that they cannot oscillate. However, in 1944 Peshkov was able to detect temperature waves which he called *Second Sound* [1, 2]. A tube filled with Helium is cooled down to a value below a critical temperature $T_\lambda = 2.17\text{K}$, where it is in a liquid state. Periodical heating of one end of the tube generates heat waves which can be detected with a thermometer.

This phenomenon seems to contradict the results of the previous paragraph on the first view. But we must remember that all derivations in 1.1 were done under the assumption of linear time–invariant systems. The physical properties of Helium in this low temperature range depend very much on the temperature so that we no longer deal with time invariant systems. Additionally the equations describing the storage and dissipative elements are nonlinear.

Chapter 2

Helium II and Second Sound

The phase diagram of Helium [Fig. 2.1] shows some unique properties. First we see that we have to put Helium under a pressure of about 25 bar to obtain the solid form even at lowest temperatures. Helium is the only element to stay liquid at the absolute 0-point under its own vapor pressure. We also see that there are two areas of liquid state called Helium I and Helium II, the curve separating them is called λ–line. Helium I is

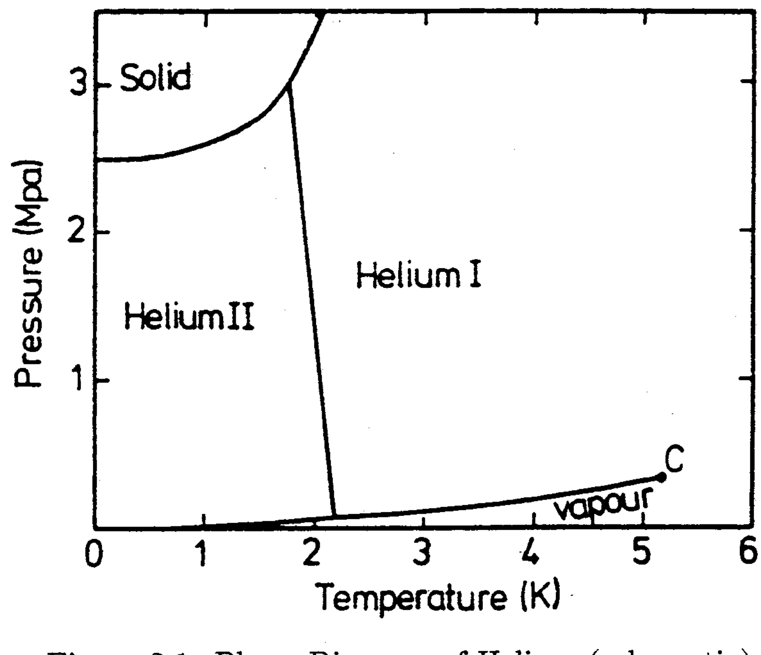

Figure 2.1: Phase Diagram of Helium (schematic)

an ordinary liquid in the sense that the laws describing its thermal and fluiddynamic behavior are those for classical liquids. Helium II is also often referred to as Superfluid Helium or as a "fourth" state of aggregation. It is able to flow through narrow channels with virtually no friction and it is a very good heat conductor. The lower end of the λ-line is at $T_\lambda = 2.173$K and is called λ-point.

The heat capacity c_p shows at T_λ a singularity [Fig. 2.2]. The curve is shaped like the Greek letter λ, what gave name to the λ-line respectively λ-point. In contradiction to ordinary phase transitions the so called λ-transition from Helium I to Helium II is not accompanied by a latent heat. The entropy over temperature curve doesn't resemble a jump but rather an edge at T_λ. There are other examples for such a type of specific heat anomalies, e.g. the binary alloy β-brass or ammonium chloride [3].

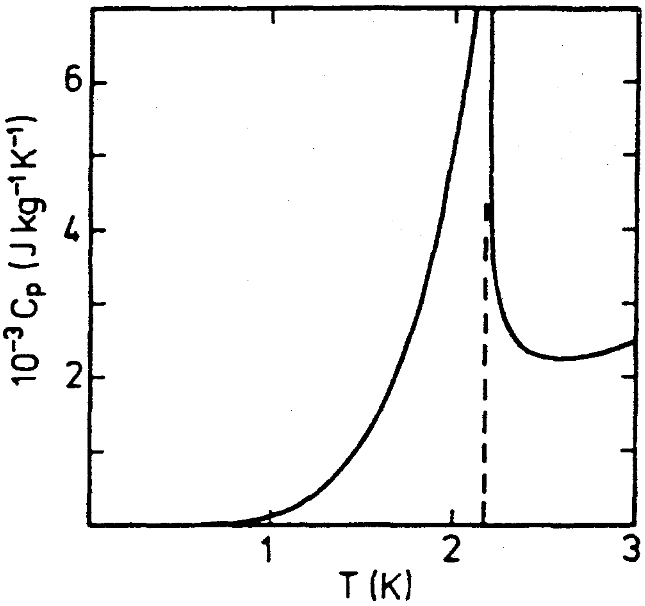

Figure 2.2: Specific Heat Capacity of Helium

A thorough description of Helium at lowest temperatures is found in the books of London [4] and Atkins [5]. They describe the phenomena and introduce the Two Fluid Model, which was developed for Helium II. Being published in the late 50's they are fundamentals of the new research field of Superfluid Helium. The latter can be proposed as a good source of information if somebody is newly confronted with this area. A paper published by Vinen describes the basic physical properties of Superfluid Helium also suitable for newcomers [6]. A more recent work is Puttermann's *Superfluid Hydrodynamics*, which covers both theory and effects in a complete form [7].

The research in Low Temperature Physics grew almost exponentially, so that there is now a huge number of papers. Therefore my selection is not at all comprehensive but rather represents those parts of research I was interested in, namely phenomenological and quantitative description of thermal behavior and numerical solution of heat transport in Helium II. Unfortunately I was not able to find a single source which provided me with all numerical values for physical properties over the Temperature range of interest. So I had to gather my data from various papers, each describing the measurement of a parameter in a small temperature range and approximating it through a function. For temperatures sufficiently smaller than T_λ, I used papers by Kramers [8] and Vinen [9]. Around T_λ the behavior of the specific heat curve is well documented by Buckingham [10] and Ferrell [11]. For the parameters describing the thermal conductance, several equations can be found for either just below or above the critical temperature [12, 13, 14, 15, 16]. I was able to find papers about numerical analysis of heat propagation in Helium II from two authors, namely Fiszdon [17] and Gentile [18, 19]. I will later use Gentile's results for comparison with my simulations.

Chapter 3

Heat Conduction Modeled With Time–Variant R–C–Networks

3.1 R–C–Networks

As I mentioned in the introduction, thermodynamical systems have only one energy storage element. The electrical analog of this type of systems is a network consisting merely of resistors and capacitors. For simplicity, from now on I shall call every system of this type R–C–network. Since heat transfer is described by a partial differential equation, we are confronted with a distributed parameter system, whereas classical mechanical or electrical systems are lumped parameter systems. By dividing the continous space into finite elements, we obtain ordinary differential equations for the heat conduction, which have the same form as the equations for mechanical and electrical R–C–networks. As for Helium at low temperatures the specific heat capacity and the thermal conductance are functions of the temperature itself, the system is time–variant, even though the temperature doesn't explicitly appear in the equations.

To demonstrate that there are differences between time–variant and time–invariant systems let me give an example. Consider the second–order system

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} \tag{3.1}$$

where

$$A = \begin{pmatrix} 4a & -3ae^{8at} \\ ae^{-8at} & 0 \end{pmatrix}. \tag{3.2}$$

The eigenvalues of the system can be easily found to be time–independent. They are

$$\begin{aligned} \lambda_1 &= a && (3.3) \\ \lambda_2 &= 3a && (3.4) \end{aligned}$$

The solution of this linear time–variant system is

$$\begin{aligned} x_1 &= 1.5(x_{10} + x_{20})e^{5at} - 0.5(x_{10} + 3x_{20})e^{7at} && (3.5) \\ x_2 &= 0.5(x_{10} + 3x_{20})e^{-at} - 0.5(x_{10} + x_{20})e^{-3at} && (3.6) \end{aligned}$$

with the initial condition as

$$\mathbf{x} = \begin{pmatrix} x_{10} \\ x_{20} \end{pmatrix} \tag{3.7}$$

If we choose $a < 0$, both eigenvalues are in the left half of the λ–plane. A linear time–invariant system is stable if and only if all eigenvalues have negative real parts. However, from (3.6) we see that the solution is unstable although the eigenvalues of the system are all negative. We see that we can't simply transfer a theorem from linear time–invariant system theory to time–variant systems. If we take the theorem stating that linear time–invariant systems can only oscillate if they have complex eigenvalues, we can make again no conclusion about the ability of a nonlinear and time–variant system to oscillate. This observation encourages us to try to find a model for the heat transfer in Helium II which has only real eigenvalues but which can be nonlinear and time–variant. By simulating this system with an experiment it can be detected if the system is capable of oscillating.

We model the heat propagation in Helium II with the normally used heat diffusion equation letting the specific heat capacity and the thermal conductance be functions of the temperature. Without reducing generality, we can consider only the one–dimensional case, i.e. heat conduction through a Helium filled channel with constant variables over every cross–section.

$$\rho c_p(T)\frac{\partial T}{\partial t} = \lambda(T)\frac{\partial^2 T}{\partial x^2} \tag{3.8}$$

$$\begin{aligned} \lambda &= \text{thermal conductivity} \\ \rho &= \text{density} \\ c_p &= \text{specific heat capacity} \end{aligned}$$

If we discretize the space into finite elements of length Δx and if we use the central–difference formula [21] for the second spatial derivative of T, we get a set of ordinary differential equations.

$$\begin{aligned} \frac{dT_i}{dt} &= \frac{\lambda}{\rho c_p}\frac{T_{i-1} - 2T_i + T_{i-1}}{(\Delta x)^2} && (3.9) \\ &= \frac{1}{A\Delta x \rho c_p}\left(\lambda A\frac{T_{i-1} - T_i}{\Delta x} - \lambda A\frac{T_i - T_{i+1}}{\Delta x}\right) && (3.10) \\ &= \frac{1}{A\Delta x \rho c_p}(\dot{Q}_{in} - \dot{Q}_{out}) && (3.11) \end{aligned}$$

where

$$\begin{aligned} A &= \text{cross section of Helium channel} \\ \Delta x &= \text{length of an element} \\ \dot{Q}_{in} &= \text{heat current into the element} \\ \dot{Q}_{out} &= \text{heat current out of element} \end{aligned}$$

3.2 Modeling in Terms of Bond Graphs

The bond graph technique is a tool to graphically represent a physical system. The part of the graphs giving name to the technique are the bonds that simultaneously connect two variables of two elements. The two variables that are assigned to each element are called adjugate variables. One of them is called the “effort” e and the other the “flow” f. The product of e and f has the dimension of a power, i.e., the bond basically represents a flow of power. Figure 3.1 shows the bond.

Other basic elements are the junctions of which two different kinds exist, the so–called “0–junction” and the “1–junction”. At the 0–junction all effort variables are equal, and the sum of all flow variables adds up to 0. At the 1–junction all flow variables are equal, and the sum of all effort variables adds up to 0 [Fig 3.2].

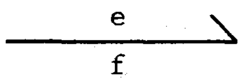

Figure 3.1: Bond

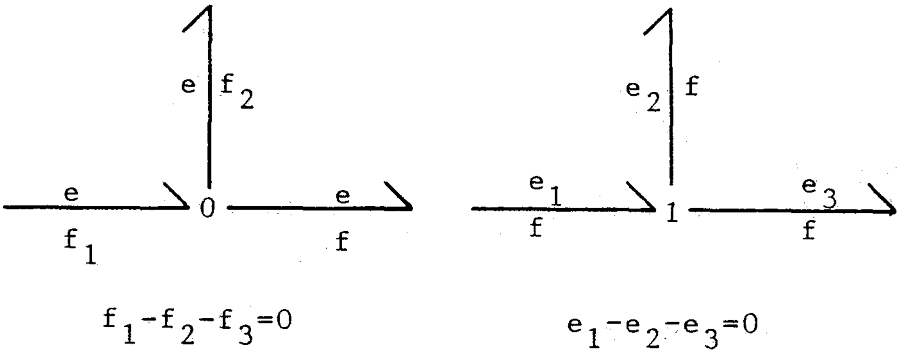

Figure 3.2: 0– and 1–Junction

Attached to the junctions by bonds are the other elements of the system like resistor and capacitor, which relate the two variables by an algebraic or differential equation. If we want to hold a variable constant at one place, we do this by attaching a source of effort or a source of flow, respectively.

3.3 Helium II Modeled in Terms of Bond Graphs

For thermodynamical systems the two adjugate variables are the temperature T used as effort and the entropy flow $\dot{S}$ used as flow. The assignment which variable represents the effort and which the flow is arbitrary, but the above choice seems a little more naturally than the alternative. The description of a system in terms of entropy and entropy flow rather than heat and heat flow is not very common in thermodynamics, but it is necessary in order to obtain a pair of adjugate state variables, the product of which is power. The model for the heat conduction is shown in Fig 3.3.

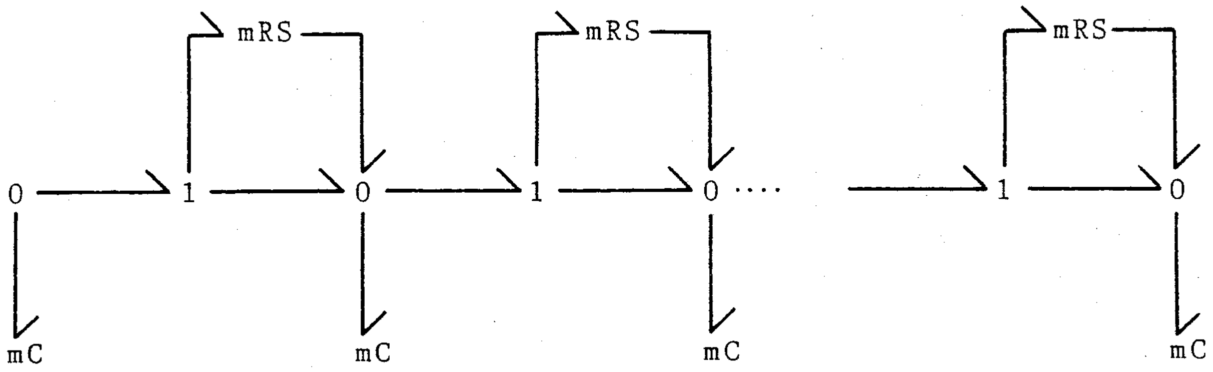

Figure 3.3: Bond Representation of Heat Conduction

The *mC* stands for "modulated capacitance." It describes the capability of Helium in the volume element to store entropy by increasing the temperature. The relation between T and $\Delta\dot{S}$ is given as a differential equation:

$$\frac{dT}{dt} = f(\Delta\dot{S}, T) \tag{3.12}$$

The *mRS* stands for "modulated resistive source." It contains the equation for the heat conduction:

$$\dot{S} = f(\Delta T, T) \tag{3.13}$$

The bond going from *mRS* to the 0–junction is necessary to satisfy the energy conservation law. The power $\Delta T \cdot \dot{S}$ which flows into the resistor dissipates and is reintroduced into the system as another entropy flow at the temperature level T. That this flow goes into the 0–junction to the right is an arbitrary assumption. Splitting it up into two flows would probably seem more reasonable, but would cause computational problems in the form of algebraic loops.

If we define a subsystem called "cell" as Fig 3.4, we can represent our Helium channel by a chain of such cells [Fig 3.5]. On the left side I added another capacitance since the cell contains only a capacitance at the right side. Furthermore I added a flow source at the left side. The simulation will run with an entropy flow being constant at the

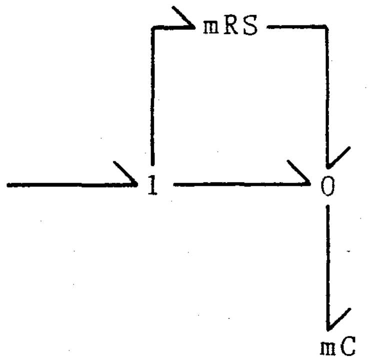

Figure 3.4: Subsystem Cell of the Heat Conduction

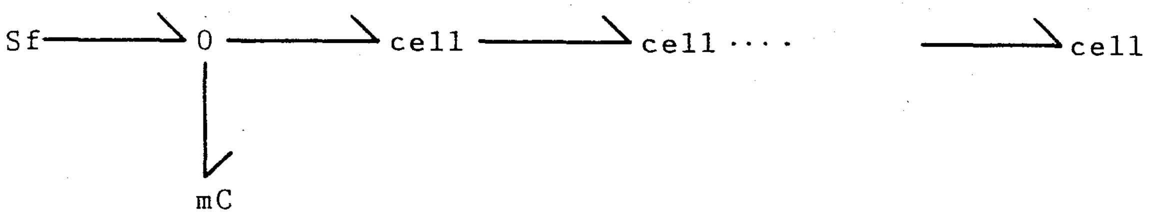

Figure 3.5: Bond Representation of Heat Conduction with Subsystems

beginning for a certain time τ and being 0 thereafter. This is equal to a heat pulse being introduced at the left end of the channel, the propagation of which can then be observed.

3.4 Numerical Values for the Thermal Parameters

As I mentioned before, I had to compile my data or equations for the specific heat capacity and the heat conduction extracted from various references of different authors. I made two experiments with different initial temperatures, for which I had to use different approximations of the thermal parameters.

3.4.1 Model 1: Initial Temperature 1.4K

The formulas used in this text for the specific heat capacity at constant pressure, c_p, are actually formulas for the specific heat under saturated vapor pressure c_{sat}. However, for $T < 2.5\text{K}$ the difference is less than 1%, which can be neglected here [5]. Kramers verified through experiments an empiric formula found earlier by Hull [8, 20]. They approximated c_{sat} between 0.6 and 1.4K as

$$c_p = c_{sat} = 100\, T^{6.2} \frac{\text{J}}{\text{kgK}}. \tag{3.14}$$

The density of Helium can be considered constant in the temperature range according to [5]

$$\rho = 145.5\, \frac{\text{kg}}{\text{m}^3}. \tag{3.15}$$

The length of the elements and the crosssection of the channel were chosen to be

$$\begin{aligned} \Delta x &= 1 \cdot 10^{-4}\text{m} & (3.16)\\ F &= 1 \cdot 10^{-4}\text{m}^2. & (3.17) \end{aligned}$$

If we define the entropy flow $\dot{S}$ of a body as the heat flow $\dot{Q}$ divided by its temperature T, which is not the original definition of entropy, but which is convenient and usual in the bond graph technique [21],

$$\dot{S} = \frac{\dot{Q}}{T} \tag{3.18}$$

we find from

$$\begin{aligned} \Delta\dot{Q} &= \rho F \Delta x c_p \frac{dT}{dt} & (3.19)\\ \Delta\dot{S} &= \frac{\rho F \Delta x c_p}{T} \frac{dT}{dt}. & (3.20) \end{aligned}$$

If we plug in (3.14 – 3.17) we obtain (in SI–Units)

$$\Delta\dot{S} = 1.455 \cdot 10^{-4}\, T^{5.2} \frac{dT}{dt}. \tag{3.21}$$

This is the equation describing the modulated capacitance, mC, of the system.

For the heat current as a function of T and dT, London gives an approximation deduced from his Two Fluid Model [4]. The equation is valid for large heat currents or large slits as in our case. The formula was later verified through measurements by Vinen [9]. The heat current density $\dot{q}$ is given as

$$\dot{q} = -\left(\frac{dT}{dx}\right)^{\frac{1}{3}} \left(\frac{\rho^2 s_\lambda^4}{A}\right)^{\frac{1}{3}} T \left(1 - \left(\frac{T}{T_\lambda}\right)^{5.6}\right) \left(\frac{T}{T_\lambda}\right)^{5.6} \tag{3.22}$$

where

$$T_\lambda = 2.173\text{K}. \tag{3.23}$$

s_λ is the specific entropy at $T = T_\lambda$, which can be found to be [8]

$$s_\lambda = 1.57 \cdot 10^3 \frac{\text{J}}{\text{kgK}}. \tag{3.24}$$

A is the so–called Gorter–Mellink constant, which is a function of temperature but which we assume to be constant in our case [9, 5].

$$A = 350 \frac{\text{ms}}{\text{kg}} \tag{3.25}$$

If we replace dT/dx through $\Delta T/\Delta x$ and rewrite the equation for $\dot{S}$, we obtain

$$\dot{S} = -F \left(\frac{\rho^2 s_\lambda^4}{A\Delta x}\right)^{\frac{1}{3}} \left(1 - \left(\frac{T}{T_\lambda}\right)^{5.6}\right) \left(\frac{T}{T_\lambda}\right)^{5.6} (\Delta T)^{\frac{1}{3}}. \tag{3.26}$$

Plugging in numerical values we obtain finally (in SI–Units) an equation describing the conduction of heat in Helium II

$$\dot{S} = -154.32 \left(1 - \left(\frac{T}{2.173}\right)^{5.6}\right) \left(\frac{T}{2.173}\right)^{5.6} (\Delta T)^{\frac{1}{3}}. \tag{3.27}$$

3.4.2 Model 2: Initial Temperature 2.1728K

As a second case, I investigated the heat conduction in the region around T_λ. The specific heat capacity in the vicinity of this point is described by Ferrell through the empiric formula [11]:

$$\begin{aligned} c_p &= 0.626 \left(\ln \frac{2}{|\tau|} - 2\text{sign}(\tau)\right) \frac{\text{J}}{\text{gK}} \qquad (3.28) \\ \tau &= \frac{T - T_\lambda}{T_\lambda} \\ |\tau| &< 5 \cdot 10^{-2} \end{aligned}$$

For this second case we choose

$$\begin{aligned}\Delta x &= 1 \cdot 10^{-3}\mathrm{m} && (3.29)\\ F &= 1 \cdot 10^{-4}\mathrm{m}^2 && (3.30)\\ \rho &= 146.57\frac{\mathrm{kg}}{\mathrm{m}^3}. && (3.31)\end{aligned}$$

If we use (3.18) and plug in 3.28 we obtain (in SI–Units)

$$\Delta\dot{S} = \frac{0.19076}{T}\left(\ln\frac{2T_\lambda}{T - T_\lambda} - 2\mathrm{sign}(T - T_\lambda)\right)\frac{dT}{dt} \tag{3.32}$$

as equation for the capacitance of the system.

For the heat conduction we must consider two different cases, depending on whether the temperature is above or below T_λ. Below T_λ the heat conduction is described by Crooks, who gives for the heat flow density $\dot{q}$ the formula (all in SI–Units) [16]:

$$\dot{q} = -\rho_s s_\lambda T\left(\frac{s_\lambda \mathrm{grad}T}{263\rho_n}\right)^{\frac{1}{3.5}}(T_\lambda - T)^{-\frac{1}{5.83}} \tag{3.33}$$

with

$$\begin{aligned}\rho_s &= 1.43(T_\lambda - T)^{\frac{2}{3}}\rho && (3.34)\\ s_\lambda &= 1.57 \cdot 10^3\frac{\mathrm{J}}{\mathrm{kgK}} && (3.35)\end{aligned}$$

Using the numerical values (3.29 – 3.31) we obtain for the entropy flow as function of the temperature difference

$$\dot{S} = -94.9(T_\lambda - T)^{0.838}(\Delta T)^{\frac{1}{3.5}}. \tag{3.36}$$

For temperatures above T_λ, Ferrell gives the thermal conductivity of Helium as

$$\lambda = \lambda_0\left(\frac{2}{\tau}\right)^{\frac{1}{3}}\left(\ln\frac{2}{\tau}\right)^{\frac{1}{2}} \tag{3.37}$$

with

$$\begin{aligned}\tau &= \frac{T - T_\lambda}{T_\lambda}\\ \lambda_0 &= 1.36 \cdot 10^{-3}\frac{\mathrm{W}}{\mathrm{mK}} && (3.38)\end{aligned}$$

Using the relation

$$\dot{S} = -\frac{F}{T}\lambda\frac{\Delta T}{\Delta x} \tag{3.39}$$

we obtain

$$\dot{S} = -\frac{-1.36 \cdot 10^{-4}}{T}\left(\frac{2T_\lambda}{T - T_\lambda}\right)^{\frac{1}{3}}\sqrt{\ln\frac{2T_\lambda}{T - T_\lambda}}\Delta T. \tag{3.40}$$

The heat conduction for $T < T_\lambda$ decreases with increasing temperature below T_λ until it reaches 0 at T_λ. Above that point the conduction increases again but stays by many orders smaller than below T_λ, which is not surprising since Helium II is known to be superconductive for heat.

3.5 Simulation and Results

For the simulation I coded the problem in DYMOLA [21, 22]. DYMOLA is not a simulation language but a program generator which supports different simulation languages. It handles very easily several hierarchically structured layers of subsystems and is also well suited for systems modeled in bond graphs since the built in “nodes” are similar to the junctions in bond graphing. I used the VAX/VMS version of DYMOLA, which can generate programs in DESIRE [23]. DESIRE, which stands for “Direct–Executing–SImulation in REal time,” is a high–speed direct–executing interactive simulation program with an on–time output of colored graphs on the screen. It supports different Runge–Kutta methods and Adams and Gear rules as integration algorithms. The DESIRE programs were executed on VMS/VAX and on a PC–compatible computer. The integration algorithm used was Euler forward with a fixed step size of $\Delta t = 0.5 \cdot 10^{-7}$s for the first model, respectively $\Delta t = 1.0 \cdot 10^{-5}$s for the second model. Appendix A shows the used DYMOLA programs.

As heat source I chose for the first model a value of $\dot{Q} = 0.2$W for $\tau = 2 \cdot 10^{-5}$s, which equals an entropy flow of $\dot{S} = 0.143$W/K. Fig 3.6 shows the temperature of the first 4 space elements over time with an initial temperature of $T_0 = 1.4$K. The heat propagates in the medium in a dissipative manner, the temperature of an element never exceeds the temperature of its neighbor to the left. Fig 3.7 shows the results of the second model with an initial temperature of $T_0 = 2.1728$K just below T_λ. Again the temperatures of the first 4 elements are shown. The heat flow is $\dot{Q} = 0.1\text{W}$ and the resulting entropy flow $\dot{S} = 0.046\text{W/K}$ for $\tau = 1 \cdot 10^{-3}\text{s}$. The heat propagation through Helium in this case is even slower as before and only the first element exceeds T_λ. For both cases I conducted simulations with various entropy sources in respect to heat flow and time. I used also space elements of different lengths between $10^{-5}\text{m} \leq \Delta x \leq 10^{-2}\text{m}$. In none of all these combinations, a behavior different from the shown diffusive behavior could be observed, especially did no solution resemble a wave–like heat propagation.

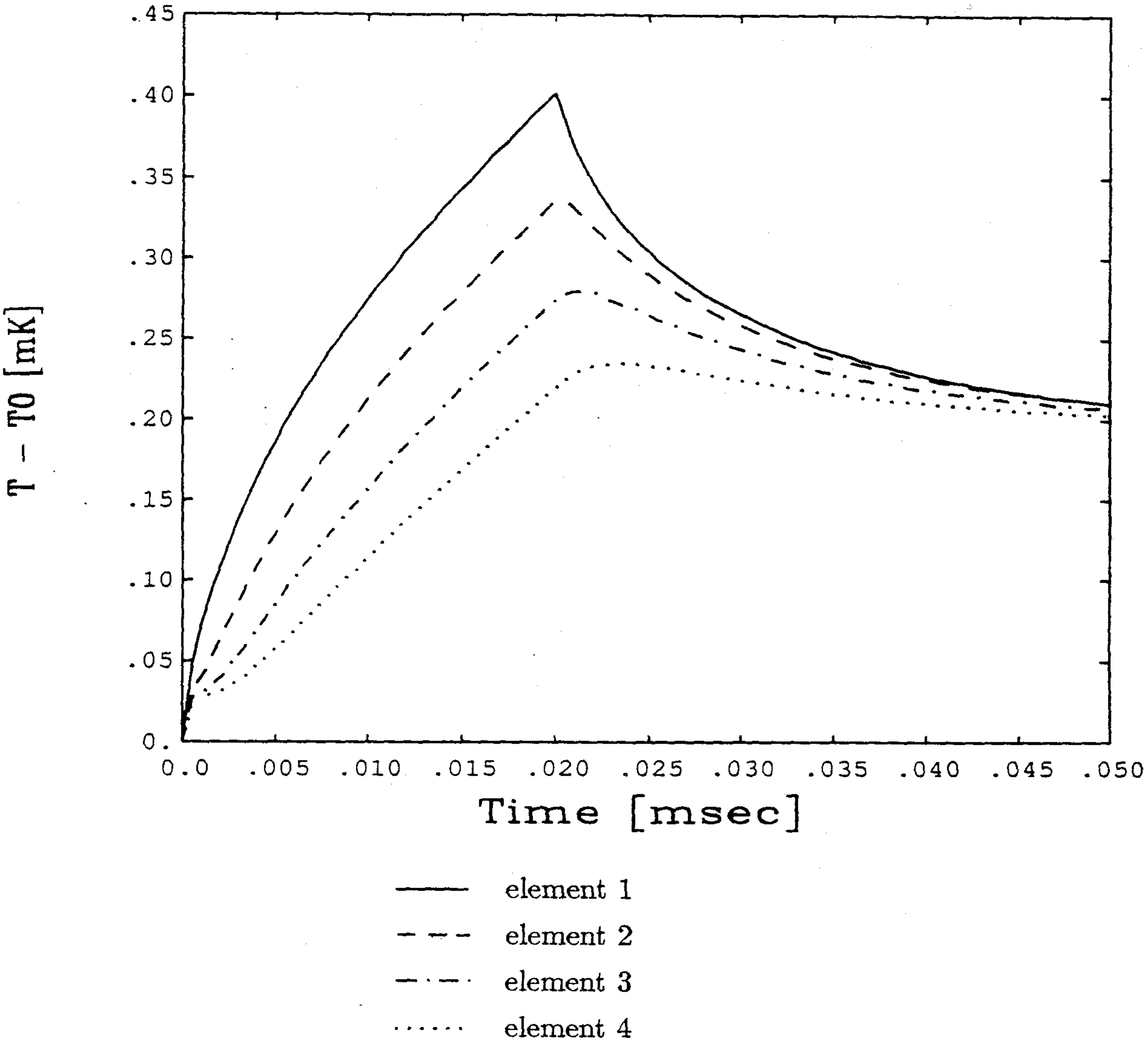

Figure 3.6: Heat Propagation in Model 1

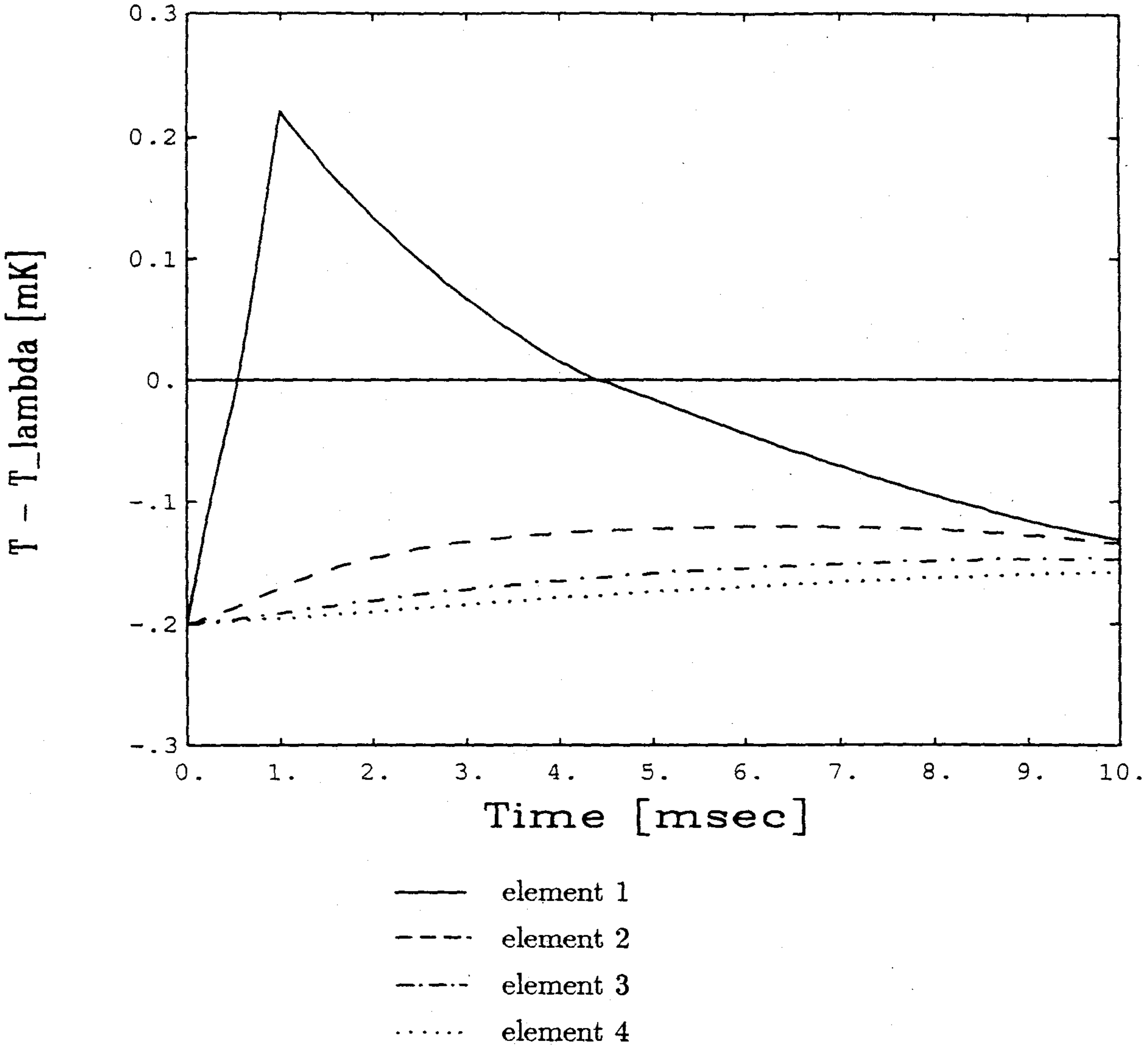

Figure 3.7: Heat Propagation in Model 2

Chapter 4

Heat Transport and Two Fluid Model

4.1 Two Fluid Model

In an attempt to explain the remarkable properties of Helium II a model was presented by two different authors in 1941 [24, 25]. The basic idea of the model is that below T_λ Helium consists of two different liquids which are ideally mixed. One of them is called the "superfluid component" with the density ρ_s, the other the "normal component" with the density ρ_n. Both partial densities add up to the total density:

$$\rho_s + \rho_n = \rho \tag{4.1}$$

The fraction of ρ_s/ρ_n is a function of the temperature; at T_λ there is no superfluid component present whereas for $T = 0\text{K}$ the normal component vanishes. Fig 4.1 shows the two densities plotted over the temperature. Another assumption of the model is that at any point in space each component has its own velocity called $\mathbf{v}_s$ respectively $\mathbf{v}_n$. Therefore we can write for the mass current density

$$\mathbf{J} = \rho_s \mathbf{v}_s + \rho_n \mathbf{v}_n \tag{4.2}$$

The normal component behaves like an ordinary liquid, it has the viscosity ν_n. The superfluid component has a viscosity coefficient equal to zero, it flows without friction.

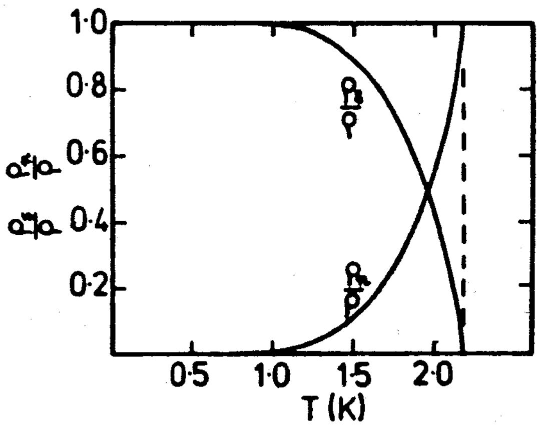

Figure 4.1: Normal and Superfluid Fractions of Helium II

This component also carries no thermal energy. All of the entropy is transported by the normal component. Using these assumptions we can write for the heat flow density:

$$\dot{\mathbf{q}} = \rho s T \mathbf{v}_n \tag{4.3}$$

where s is the entropy per mass unit. Equation (4.3) shows the transport of heat in a convective manner, i.e. the moving mass of the normal component transports the heat. We shall see later that this effect is very important for the phenomenon of Second Sound. The most interesting result of the assumption of two interpenetrating fluids is the existence of two different forms of wave propagation.

"First sound" is the analog of ordinary sound, which is the wave propagation observable in all liquids or gases. The two components move in phase with the same velocity that causes the density to change. The pressure gradient exerts a restoring force on the liquid. The result is a propagating pressure wave in the medium.

If $\mathbf{v}_n$ and $\mathbf{v}_s$ are moving in antiphase and if they are of such magnitude that the total mass flow vanishes

$$\mathbf{J} = \rho_s \mathbf{v}_s + \rho_n \mathbf{v}_n = 0 \tag{4.4}$$

there is no change in the total density of Helium. But there are variations of ρ_s/ρ and ρ_n/ρ which are equal to a variation of temperature according to Fig. 4.1. The restoring force now arises from the temperature gradient. This phenomenon is the so–called Second

Sound. It can be produced by an oscillating heat source that causes the two components to oscillate in antiphase.

Let us consider the case of constant heat transfer in a Helium II filled channel. We apply at one end of the channel a heat source that transforms the superfluid component into the normal component. This normal component flows away from the source to a heat sink at the other end of the channel, where it is transformed back to the superfluid component again. This component flows then back in the other direction. So we have a steady flow of the two components flowing in opposite directions. This mechanism of heat transport called superconduction of heat can be interpreted as Second Sound with the frequency 0.

4.2 Hydrodynamical Equations

The dynamics of Helium II are described by the following conservation equations: conservation of mass, conservation of momentum for both components and conservation of energy. A general form of these equations can be found in [26]. In our case we can simplify the model in two ways. Without loss of generality we will consider only the one–dimensional case. Assuming heat transport with a zero net mass flow, we obtain from (4.4)

$$v_s = -\frac{\rho_n}{\rho_s} v_n. \tag{4.5}$$

Then the normal velocity v_n and temperature T describe completely the behavior of Helium II, and the following equations can be obtained [18]:

$$\begin{aligned}
\frac{\partial T}{\partial t} &= \frac{K}{\rho c_p}\frac{\partial^2 T}{\partial x^2} - v_n \frac{\partial T}{\partial x} - \frac{s}{c_p}\frac{\partial v_n}{\partial x} T + \frac{A(T)\rho_n}{c_p}\left(\frac{\rho}{\rho_s}\right)^3 v_n^4 + \frac{A_0}{c_p}\left(\frac{\partial v_n}{\partial x}\right)^2 && (4.6)\\
\frac{\partial v_n}{\partial t} &= A_0 \frac{\rho_s}{\rho_n}\frac{\partial^2 v_n}{\partial x^2} - v_n\left(3\frac{\partial v_n}{\partial x} + A(T)\left(\frac{\rho}{\rho_s}\right)^2 \rho v_n^2\right) - s\frac{\rho_s}{\rho_n}\frac{\partial T}{\partial x} && (4.7)
\end{aligned}$$

with the thermodynamic parameters:

$$\begin{aligned}
K &= \text{dissipation}\\
c_p &= \text{specific heat capacity}
\end{aligned}$$

$$\begin{aligned} s &= \text{specific entropy} \\ A(T) &= \text{Gorter–Mellink constant} \\ A_0 &= \text{viscosity} \end{aligned}$$

The two equations are strongly coupled by the presence of $\partial v_n/\partial x$ in the equation for T and of $\partial T/\partial x$ in the equation for v_n. Additionally s, $A(T)$, c_p, ρ_s and ρ_n depend on the temperature and contribute to the coupling of the equations.

4.3 Numerical Values for the Thermal Parameters

For the simulation I followed for a great part the description of Gentile's numerical investigations on heat transport in Helium II [18]. The initial temperature and the point around which the parameters are approximated is $T_0 = 1.8\text{K}$. The parameters ρ, K and A_0 are considered to be constant in the used temperature range.

$$\rho = 145.3 \frac{\text{kg}}{\text{m}^3} \tag{4.8}$$

$$K = 5 \cdot 10^{-4} \frac{\text{W}}{\text{K}} \tag{4.9}$$

$$A_0 = 2.6 \cdot 10^{-8} \text{m} \tag{4.10}$$

For the entropy s, the heat capacity c_p and the density of the normal component we find for $T > 1.4\text{K}$ [18, 4]:

$$s = s_\lambda \left(\frac{T}{T_\lambda}\right)^{5.6} \tag{4.11}$$

$$s_\lambda = 1.57 \cdot 10^3 \frac{\text{J}}{\text{kg K}} \tag{4.12}$$

$$\rho_n = \rho \left(\frac{T}{T_\lambda}\right)^{5.6} \tag{4.13}$$

$$c_p = 5.6 s_\lambda \left(\frac{T}{T_\lambda}\right)^{5.6} \tag{4.14}$$

For the Gorter–Mellink constant $A(T)$ the formula of Crooks was used, which gave good results in comparison with measured data [16, 5].

$$A(T) = 263(T_\lambda - T)^{-0.74} \text{ (in MKS units)} \tag{4.15}$$

4.4 Numerical Method

The solution of the partial differential equations can be divided into two parts. By using the finite–difference method to approximate the spatial derivatives, a set of ordinary differential equations is obtained. These are solved using the simulation language ACSL.

For efficient numerical convergence, space nodes separated by $1/2\Delta x$ were used for v_n and T (Fig. 4.2).

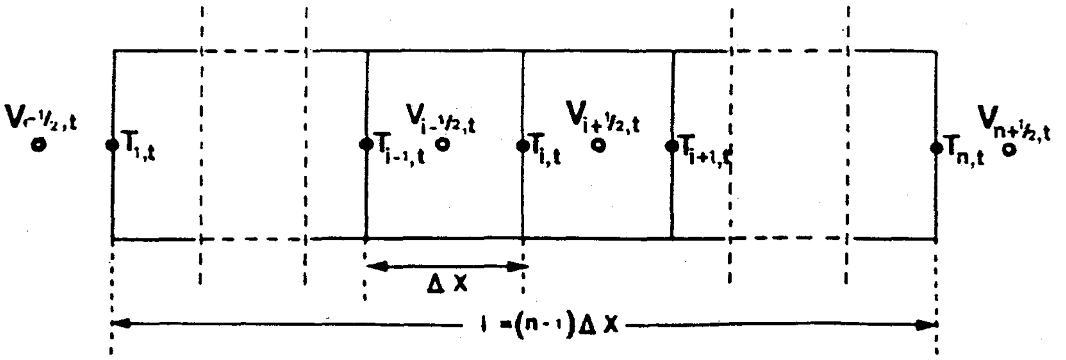

Figure 4.2: Discretization of the Helium Channel

For simplicity, from now on I shall call v_n simply v. The index $_i$ refers to the number of the space node. The following finite–difference formulas have been used for the temperature equation [21]:

$$\left(\frac{\partial T_i}{\partial x}\right)_t = \frac{T_{i+1,t} - T_{i-1,t}}{2\Delta x} \tag{4.16}$$

$$\left(\frac{\partial v_i}{\partial x}\right)_t = \frac{v_{i+1/2,t} - v_{i-1/2,t}}{\Delta x} \tag{4.17}$$

$$v_{i,t} = \frac{v_{i+1/2,t} + v_{i-1/2,t}}{2} \tag{4.18}$$

$$\left(\frac{\partial^2 T}{\partial x^2}\right)_t = \frac{T_{i+1,t} - 2T_{i,t} + T_{i-1,t}}{2\Delta x^2} \tag{4.19}$$

For the velocity we use:

$$\left(\frac{\partial v_{i+1/2}}{\partial x}\right)_t = \frac{v_{i+3/2,t} - v_{i-1/2,t}}{2\Delta x} \tag{4.20}$$

$$\left(\frac{\partial T_{i+1/2}}{\partial x}\right)_t = \frac{T_{i+1,t} - T_{i,t}}{\Delta x} \tag{4.21}$$

$$T_{i+1/2,t} = \frac{T_{i+1,t} + T_{i,t}}{2} \tag{4.22}$$

$$\left(\frac{\partial^2 v_{i+1/2}}{\partial x^2}\right)_t = \frac{v_{i+3/2,t} - 2v_{i+1/2,t} + v_{i-1/2,t}}{2\Delta x^2} \tag{4.23}$$

4.5 Boundary Conditions

At the left end of the channel, at $x = 0$, the velocity of the superfluid component is assumed to be zero.

$$v_1 = 0 \tag{4.24}$$

In the differential equation for $\partial v_1/\partial t$ forward–difference formulas have to be used for the space derivatives instead of central–difference formulas [21]. Using (4.24) we obtain:

$$\left(\frac{\partial v_{3/2}}{\partial x}\right)_t = \frac{3v_{3/2,t} + 3v_{2,t} - v_{5/2,t}}{6\Delta x} \tag{4.25}$$

$$\left(\frac{\partial^2 v_{3/2}}{\partial x^2}\right)_t = \frac{-2v_{3/2,t} + v_{2,t}}{\Delta x^2} \tag{4.26}$$

At the same end, the heat flow $\dot{q}_0$ is introduced into the channel. To obtain an equation for the temperature T_1 of the first space element, an energy balance in the first node can be made [18]. For this element the dissipation K and the viscosity A_0 have been neglected. Using (4.24) we obtain:

$$\frac{\partial T_{3/2}}{\partial T} = \frac{\dot{q}_0}{\rho c_p 2\Delta x} - \frac{s}{c_p 2\Delta x} T_2 v_2 \tag{4.27}$$

$$T_1 = 2T_{3/2} - T_2 \tag{4.28}$$

At the right end of the channel, $x = l$, the boundary conditions are not so important since the simulation ends as soon as the temperature front reaches this end. We assume symmetry conditions, so that we can write:

$$\left(\frac{\partial v}{\partial x}\right)_{x=l} = 0 \tag{4.29}$$

$$\left(\frac{\partial T}{\partial x}\right)_{x=l} = 0 \tag{4.30}$$

For the second derivatives we will use the same difference formulas with the symmetry condition:

$$T_{n+i} = T_{n-i} \tag{4.31}$$

$$v_{n+i} = v_{n-i} \tag{4.32}$$

4.6 Simulation and Results

The obtained set of ordinary differential equations was solved using ACSL (Advanced Continuous Simulation Language) [27]. ACSL is a simulation program for nonlinear differential equations. It allows a free form of input of the equations since it sorts them into the required order. It also features a possibility to use submodels which are called MACROS. As integration algorithms it supports Runge–Kutta, Adams Bashforth, and Gear formulas.

In order to have only nodes with integer numbers, I renumbered the nodes so that for space nodes with odd numbers the temperature and for even numbers the velocity of the normal component is evaluated by simulation. The values in between are calculated as median of neighbouring nodes. The distance Δx between two such new nodes was chosen as

$$\Delta x = 2.5 \cdot 10^{-3}\text{m}. \tag{4.33}$$

The heat input $\dot{q}_0$ has the shape of a rectangular pulse with a value of

$$\dot{q}_0 = 3 \cdot 10^4 \frac{\text{W}}{\text{m}^2} \tag{4.34}$$

and a duration of

$$\tau = 0.5 \cdot 10^{-3}\text{s}. \tag{4.35}$$

Appendix B shows the ACSL program. I used the Heun algorithm as integration method with a fixed step size of $\Delta t = 1 \cdot 10^{-6}$s. The simulation was run for $t_{max} = 4 \cdot 10^{-3}$s.

Fig. 4.3 shows the results of the simulation. Shown is the temperature difference of the nodes to the initial temperature T_0 over time. The distance between two plotted nodes is $5\Delta x = 12.5 \cdot 10^{-3}$m. It can be seen that the heat propagates in the medium as a wave since

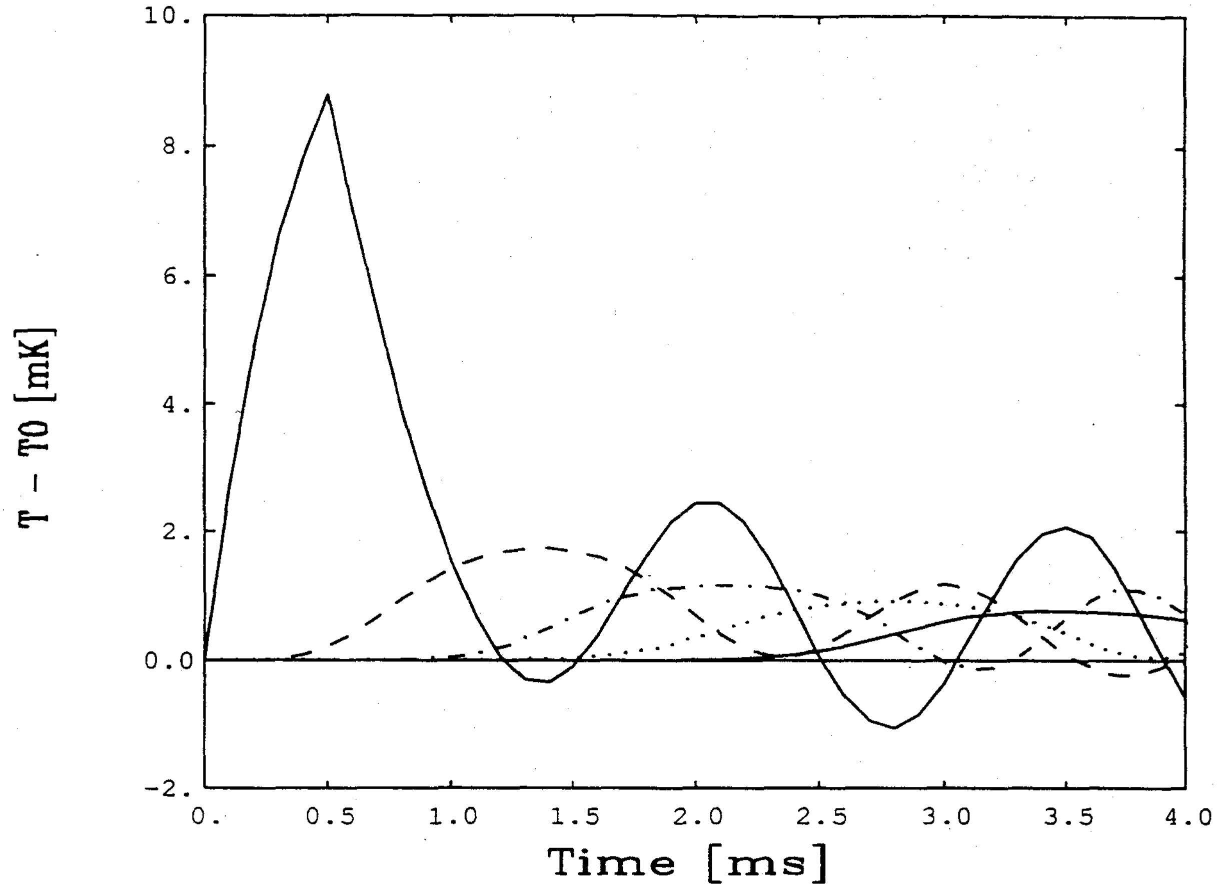

Figure 4.3: Heat Propagation in Two Fluid Model

as a wave since neighboring curves cross each other. The curves confirm the simulation results of Gentile with respect to the wave–like propagation of a heat pulse through Helium II and with respect to the amplitude of the propagating temperature pulse. Fig. 4.3 shows oscillating temperatures of the nodes after the pulse went through. This is not in accordance to the results of Gentile who used the same numerical values for initial temperature and heat pulse. The difference is probably the result of a numerical problem, which hasn't been further investigated.

From Fig 4.3 we can also obtain an approximation of the velocity of the heat pulse.

The distance between the maxima of two neighbouring curves is

$$\Delta t = 0.81 \cdot 10^{-3}\mathrm{s}. \tag{4.36}$$

As the distance between two curves is given as

$$\Delta x = 12.5 \cdot 01^{-3}\mathrm{m} \tag{4.37}$$

we obtain for the propagation velocity

$$v = \frac{\Delta x}{\Delta t} = 15.4\frac{\mathrm{m}}{\mathrm{s}} \tag{4.38}$$

Measurements show for a temperature of 1.8 K a velocity of second sound of $u = 19.5\frac{\mathrm{m}}{\mathrm{s}}$ [28]. The difference in measured and calculated velocity probably originates from the description of the thermal parameters as functions of the temperature [18], but the simulation results basically confirm the Two Fluid Model of Helium II.

Chapter 5

Conclusions

5.1 Conclusions

The purpose of this project was to investigate whether the heat propagation in Helium II can be described with the classical heat diffusion equation and nonlinear parameters. If this model of Helium had shown the capacity of oscillating, it would have been an example of a nonlinear oscillating system with only real eigenvalues, as mentioned in chapter 1. But the results of several simulations with different initial and boundary conditions show no sign of wave–like heat transport. This shows that the description with the classical but nonlinear diffusion equation is not sufficient for the heat propagating system in Helium II.

The motivation for the search for nonlinear oscillations was the temperature dependance of the specific heat capacity and heat conduction. But for temperatures below T_λ the bending of the curves for these parameters is very small. Considering the small temperature variations, the nonlinear portion of these parameters is not very important. This motivated a simulation around T_λ where the nonlinearities are significant. But these nonlinearities are such, that they damp heat transportation even more, the specific heat capacity has a singularity and goes to infinity at that point and the heat conduction goes to zero.

The simulation with the Two Fluid Model shows that this theory is sufficient to describe heat conduction in Helium II. For further research it would be interesting to describe the Two Fluid Model in terms of bond graphs. By showing the energy flow in the system this could contribute to a better understanding of the thermal behavior of Helium II.

5.2 Zusammenfassung

Das Ziel dieser Studienarbeit war zu untersuchen, ob die Wärmeleitung in Helium II mit der klassischen Wärmediffusionsgleichung mit nichtlinearen Parametern beschrieben werden kann. Wäre dieses Modell für Helium schwingungsfähig gewesen, würde es ein Beispiel darstellen für ein nichtlineares schwingungsfähiges System, das nur Eigenwerte auf der realen Achse hat, wie in Kapitel 1 erwähnt wurde. Aber die Ergebnisse von mehreren Simulationen mit verschiedenen Anfangs– und Randbedingungen zeigen keine Anzeichen einer wellenartigen Wärmeausbreitung. Dies zeigt, daß die Beschreibung mit den klassischen aber nichtlinearen Differentialgleichungen nicht genügend ist für die Wärmeleitung in Helium II.

Die Temperaturabhängigkeit der spezifischen Wärmekapazität und der Wärmeleitung war die Motivation für die Suche nach nichtlinearen Schwingungen. Aber für Temperaturen unter T_λ ist die Krümmung der Kurven dieser Parameter sehr klein. Berücksichtigt man weiter die kleinen Temperaturänderungen, so ist der nichtlineare Anteil an diesen Parametern nicht maßgeblich. Dieser Umstand regte eine Simulation um T_λ an, wo die Nichtlinearitäten signifikant sind. Aber diese Nichtlinearitäten sind gerade so, daß sie den Wärmetransport noch mehr dämpfen; die spezifische Wärmekapazität hat an dieser Stelle eine Singularität und geht gegen unendlich, während die Wärmeleitung nach null geht.

Die Simulation mit dem Zweiflüssigkeitenmodell zeigt, daß diese Theorie die Wärmeleitung in Helium II zufriedenstellend beschreibt. Für weitere Untersuchungen wäre es interressant, das Zweiflüssigkeitenmodell mit Hilfe von Bondgraphen darzustellen. Daß auf diese Weise die Energieflüsse aufgezeigt werden, kann zum Verständnis der thermischen Vorgänge beitragen.

Appendix A

DYMOLA Programs

This appendix contains the DYMOLA programs which were used to simulate the heat transport with the help of Bond Graphs. It consists each of a main program, a library file `bond.lib` and a file `exp.ctl` which describes the simulation experiment.

A.1 Model 1: Initial Temperature 1.4K

model1.dym

```
@bond.lib

model project
  submodel (SF) SF
  submodel (cell)  c1, c2, c3, c4, c5, c6, c7, c8, c9,c10, ->
                  c11,c12,c13,c14,c15,c16,c17,c18,c19,c20
  submodel (mC) mC(mgamma = 1.455E-4)
  node             n1, n2, n3, n4, n5, n6, n7, n8, n9,n10, ->
                  n11,n12,n13,n14,n15,n16,n17,n18,n19,n20,n21
  constant T0 = 1.4
  output y1, y2, y3, y4
  input time, tau

  connect    SF      at     n1
  connect    mC      at     n1
  connect    c1     from    n1      to    n2
  connect    c2     from    n2      to    n3
  connect    c3     from    n3      to    n4
  connect    c4     from    n4      to    n5
  connect    c5     from    n5      to    n6
  connect    c6     from    n6      to    n7
  connect    c7     from    n7      to    n8
  connect    c8     from    n8      to    n9
  connect    c9     from    n9      to    n10
  connect    c10    from    n10     to    n11
  connect    c11    from    n11     to    n12
  connect    c12    from    n12     to    n13
  connect    c13    from    n13    to    n14
  connect    c14    from    n14    to    n15
  connect    c15    from    n15    to    n16
  connect    c16    from    n16    to    n17
  connect    c17    from    n17    to    n18
  connect    c18    from    n18    to    n19
  connect    c19    from    n19    to    n20
  connect    c20    from    n20    to    n21

  SF.F0 = 0.2/T0
  SF.time = time
  SF.tau = tau
  y1 = (c1.e1 - T0) * 1000.
  y2 = (c2.e1 - T0) * 1000.
  y3 = (c3.e1 - T0) * 1000.
  y4 = (c4.e1 - T0) * 1000.

end
```

bond.lib

```
model type mRS
  cut A(e1/f1), B(e2/-f2)
  main cut C[A B]
  main path  P<A-B>
  parameter mtheta = 1.0
  constant Tlam = 2.173 , eps =1.E-30
  local help1, help2

  help1 = (1-(e2/Tlam)**5.6) * (e2/Tlam)**5.6
  help2 = (lim(abs(e1)-eps) + eps)**(1/3)
  f1 = mtheta * help2 * sgn(e1) * help1
  e1 * f1 = e2 * f2
end



model type mC
  main cut A(e/f)
  parameter mgamma = 1.0
  local C
  C = mgamma * (e)**5.2
  C * der(e) = f
end



model type bond
  cut A(x/y), B(y/-x)
  main cut C[A B]
  main path P<A-B>
```

```
end



model type cell
  submodel (mRS) mRS(mtheta = 154.32)
  submodel (mC) mC(mgamma = 1.455E-4)
  submodel (bond) b1, b2, b3
  node n1, n2

  cut A(e1/f1), B(e2/-f2)
  main cut C[A B]
  main path P<A-B>

  connect    b1     from     A    to    n1
  connect    b2     from     n1   to    B
  connect    b3     from     n1   to    n2
  connect    mRS    from     n2   to    B
  connect    mC     at       B

end



model type SF
  main cut A(./-f)
  terminal F0, tau, time
  f = F0 * swtch(tau - time)
end
```

exp.ctl

```
cmodel
  simutime 5.E-5
  step 0.5E-7
  commupoints 101
  input 2, time(depend,t), tau(independ,2.E-5)



ctblock
  scale = 1
  XCCC = 1
  label TRY
  drunr
if XCCC<0 then XCCC = -XCCC | scale = 2*scale | go to TRY
          else proceed
ctend

outblock
  OUT
  dispt y1, y2, y3, y4
outend

end
```

A.2 Model 2: Initial Temperature 2.178K

model2.dym

```
@bond.lib

model project
  submodel (SF)    SF
  submodel (cell)  c1, c2, c3, c4, c5, c6, c7, c8, c9,c10
  submodel (mC)    mC(mgamma = 0.019076)
  node             n1, n2, n3, n4, n5, n6, n7, n8, n9,n10,n11
  constant         T0 = 2.1718
  output           y1, y2, y3, y4
  input            time, tau

  connect    SF      at     n1
  connect    mC      at     n1
  connect    c1     from    n1     to    n2
  connect    c2     from    n2     to    n3
  connect    c3     from    n3     to    n4
  connect    c4     from    n4     to    n5
  connect    c5     from    n5     to    n6
  connect    c6     from    n6     to    n7
  connect    c7     from    n7     to    n8
  connect    c8     from    n8     to    n9
  connect    c9     from    n9     to    n10
  connect    c10    from    n10    to    n11

  SF.F0 = 0.1/T0
  SF.time = time
  SF.tau = tau
  y1 = (c1.e1 - mC.Tlam) * 1000.
  y2 = (c2.e1 - mC.Tlam) * 1000.
  y3 = (c3.e1 - mC.Tlam) * 1000.
  y4 = (c4.e1 - mC.Tlam) * 1000.

end
```

bond.lib

```
 model type mRS
  cut A(e1/f1), B(e2/-f2)
  main cut C[A B]
  main path  P<A-B>
  parameter mtheta1 = 1.0, mtheta2 = 1.0
  constant Tlam = 2.172 , eps =1.E-37
  local   f1l, f1u, tau

  tau = (e2-Tlam)/Tlam
  f1l = mtheta1*(abs(e1)+eps)**(1/3.5) * sgn(e1) * ->
        (abs(Tlam-e2)+eps)**(0.8381)
  f1u = mtheta2 * (2/(abs(tau)+eps))**(1/3) * ->
        sqrt(ln(2/(abs(tau)+eps)))*e1/e2
  f1 = f1l * swtch(-tau) + f1u * swtch(tau)
  e1 * f1 = e2 * f2
end


model type mC
  main cut A(e/f)
  parameter mgamma = 1.0
  constant Tlam = 2.172 ,eps = 1.E-37
  local C
  C = mgamma * (ln(2*Tlam/(abs(e-Tlam)+eps)) - 2*sgn(e-Tlam))/e
  C * der(e) = f
end
```

```
model type bond
  cut A(x/y), B(y/-x)
  main cut C[A B]
  main path P<A-B>
end



model type cell
  submodel (mRS) mRS(mtheta1 = 94.897, mtheta2 = 1.36E-4)
  submodel (mC) mC(mgamma = 0.019076)
  submodel (bond) b1, b2, b3
  node n1, n2

  cut A(e1/f1), B(e2/-f2)
  main cut C[A B]
  main path P<A-B>

  connect    b1     from     A    to    n1
  connect    b2     from     n1   to    B
  connect    b3     from     n1   to    n2
  connect    mRS    from     n2   to    B
  connect    mC     at       B

end



model type SF
  main cut A(./-f)
  terminal F0, tau, time
  f = F0 * swtch(tau - time)
end
```

exp.ctl

```
cmodel
  simutime 10.E-3
  step 10.E-6
  commupoints 101
  input 2, time(depend,t), tau(independ,1.E-3)


ctblock
  scale = 1
  XCCC = 1
  label TRY
  drunr
if XCCC<0 then XCCC = -XCCC | scale = 2*scale | go to TRY
          else proceed
ctend

outblock
  OUT
  dispt y1, y2, y3, y4
outend

end
```

Appendix B

ACSL Program

This appendix contains the ACSL program used for the simulation of the heat transport in Helium II with the use of the Two Fluid Model.

helium.csl

```
PROGRAM helium
  INITIAL
    MACRO gm(out,tem) $"-----------to calculate Gorter Mellink constant"
      out = 10.**(2.1+3.12*alog10(tem)+0.0076/(1-(tem/tlam)))
    MACRO END
    MACRO temdot(out,temm2,tem,temp2,vm1,vp1)
    MACRO REDEFINE v
      v = (vm1+vp1)/2
      out = k/(rho*5.6*slam) * (tlam/tem)**5.6 *...
              (temm2-2*tem+temp2)/(4*dx**2) -...
            v*(temp2-temm2)/(4*dx) - tem/5.6 * (vp1-vm1)/(2*dx) +...
            (gm(tem)*rho*(1/(1-(tem/tlam)**5.6))*v**4 +...
             a0*(tlam/tem)**5.6*((vp1-vm1)/(2*dx))**2)/(5.6*slam)
    MACRO END
    MACRO vdot(out,vm2,v,vp2,temm1,temp1)
    MACRO REDEFINE tem
      tem = (temm1+temp1)/2
      out = ((tlam/tem)**5.6 -1) * ( a0*(vp2-2*v+vm2)/(4*dx**2) -...
             slam*(tem/tlam)**5.6*(temp1-temm1)/(2*dx)) -...
            v * (3*(vp2-vm2)/(4*dx) +...
             gm(tem) * (1/(1-(tem/tlam)**5.6))**2 * rho * v**2)
    MACRO END
  constant...
    tem0 = 1.8,  v0 = 0,  taut = 0.5E-3,  tmax = 4E-3,...
    slam = 1.57E+3, tlam = 2.173, a0 =2.6E-8, dx = 2.5E-3,...
    k = 5E-4, rho = 0.1453E+3,...
    tem2i = 1.8, tem3i = 1.8, tem5i = 1.8, tem7i = 1.8,...
    tem9i = 1.8, tem11i = 1.8, tem13i = 1.8, tem15i = 1.8,...
    tem17i = 1.8, tem19i = 1.8, tem21i = 1.8, tem23i = 1.8,...
    tem25i = 1.8, tem27i = 1.8, tem29i = 1.8, tem31i = 1.8,...
    tem33i = 1.8, tem35i = 1.8, tem37i = 1.8, tem39i = 1.8,...
    tem41i = 1.8, tem43i = 1.8, tem45i = 1.8, tem47i = 1.8,...
    tem49i = 1.8, tem51i = 1.8, tem53i = 1.8, tem55i = 1.8,...
    tem57i = 1.8, tem59i = 1.8, tem61i = 1.8, tem63i = 1.8,...
    tem65i = 1.8, tem67i = 1.8, tem69i = 1.8, tem71i = 1.8,...
    tem73i = 1.8, tem75i = 1.8, tem77i = 1.8, tem79i = 1.8,...
    tem81i = 1.8
  cinterval cint = 1E-4
  algorithm ialg = 3
  nsteps nstp = 500
  q0 = 3E+4
  schedule q0stop .at. taut
   END $"----------------------------------------------of INITIAL"
  DYNAMIC
    DERIVATIVE
      "----------------------------- first two nodes"
```

```
temd2 = (q0/(rho*5.6*slam)*(tlam/tem2)**5.6-tem3*v3/5.6)/(2*dx)
tem2 = integ(temd2,tem2i)
tem1 = 2*tem2 - tem3
v1 = 0
vdot2 = ((tlam/tem2)**5.6 -1) * (a0*(v3-2*v2)/dx**2 -...
          slam*(tem2/tlam)**5.6*(tem4-tem2)/(2*dx)) -...
          v2*( 3*(3*v2+3*v3-v4)/(6*dx) + gm(tem2)*...
          (1/(1-(tem2/tlam)**5.6))**2 * rho * v2**2)
v2 = integ(vdot2,0)
"------------------------- nodes in the middle"
tem3   = integ(temdot(tem1,tem3,tem5,v2,v4),tem3i)
v3     = (v2+v4)/2
tem4   = (tem3+tem5)/2
v4     = integ(vdot(v2,v4,v6,tem3,tem5),0)
tem5   = integ(temdot(tem3,tem5,tem7,v4,v6),tem5i)
v5     = (v4+v6)/2
tem6   = (tem5+tem7)/2
v6     = integ(vdot(v4,v6,v8,tem5,tem7),0)
tem7   = integ(temdot(tem5,tem7,tem9,v6,v8),tem7i)
v7     = (v6+v8)/2
tem8   = (tem7+tem9)/2
v8     = integ(vdot(v6,v8,v10,tem7,tem9),0)
tem9   = integ(temdot(tem7,tem9,tem11,v8,v10),tem9i)
v9     = (v8+v10)/2
tem10  = (tem9+tem11)/2
v10    = integ(vdot(v8,v10,v12,tem9,tem11),0)
tem11  = integ(temdot(tem9,tem11,tem13,v10,v12),tem11i)
v11    = (v10+v12)/2
tem12  = (tem11+tem13)/2
v12    = integ(vdot(v10,v12,v14,tem11,tem13),0)
tem13  = integ(temdot(tem11,tem13,tem15,v12,v14),tem13i)
```

```
v13    = (v12+v14)/2
tem14  = (tem13+tem15)/2
v14    = integ(vdot(v12,v14,v16,tem13,tem15),0)
tem15  = integ(temdot(tem13,tem15,tem17,v14,v16),tem15i)
v15    = (v14+v16)/2
tem16  = (tem15+tem17)/2
v16    = integ(vdot(v14,v16,v18,tem15,tem17),0)
tem17  = integ(temdot(tem15,tem17,tem19,v16,v18),tem17i)
v17    = (v16+v18)/2
tem18  = (tem17+tem19)/2
v18    = integ(vdot(v16,v18,v20,tem17,tem19),0)
tem19  = integ(temdot(tem17,tem19,tem21,v18,v20),tem19i)
v19    = (v18+v20)/2
tem20  = (tem19+tem21)/2
v20    = integ(vdot(v18,v20,v22,tem19,tem21),0)
tem21  = integ(temdot(tem19,tem21,tem23,v20,v22),tem21i)
v21    = (v20+v22)/2
tem22  = (tem21+tem23)/2
v22    = integ(vdot(v20,v22,v24,tem21,tem23),0)
tem23  = integ(temdot(tem21,tem23,tem25,v22,v24),tem23i)
v23    = (v22+v24)/2
tem24  = (tem23+tem25)/2
v24    = integ(vdot(v22,v24,v26,tem23,tem25),0)
tem25  = integ(temdot(tem23,tem25,tem27,v24,v26),tem25i)
v25    = (v24+v26)/2
tem26  = (tem25+tem27)/2
v26    = integ(vdot(v24,v26,v28,tem25,tem27),0)
tem27  = integ(temdot(tem25,tem27,tem29,v26,v28),tem27i)
v27    = (v26+v28)/2
tem28  = (tem27+tem29)/2
v28    = integ(vdot(v26,v28,v30,tem27,tem29),0)
```

```
tem29  = integ(temdot(tem27,tem29,tem31,v28,v30),tem29i)
v29    = (v28+v30)/2
tem30  = (tem29+tem31)/2
v30    = integ(vdot(v28,v30,v32,tem29,tem31),0)
tem31  = integ(temdot(tem29,tem31,tem33,v30,v32),tem31i)
v31    = (v30+v32)/2
tem32  = (tem31+tem33)/2
v32    = integ(vdot(v30,v32,v34,tem31,tem33),0)
tem33  = integ(temdot(tem31,tem33,tem35,v32,v34),tem33i)
v33    = (v32+v34)/2
tem34  = (tem33+tem35)/2
v34    = integ(vdot(v32,v34,v36,tem33,tem35),0)
tem35  = integ(temdot(tem33,tem35,tem37,v34,v36),tem35i)
v35    = (v34+v36)/2
tem36  = (tem35+tem37)/2
v36    = integ(vdot(v34,v36,v38,tem35,tem37),0)
tem37  = integ(temdot(tem35,tem37,tem39,v36,v38),tem37i)
v37    = (v36+v38)/2
tem38  = (tem37+tem39)/2
v38    = integ(vdot(v36,v38,v40,tem37,tem39),0)
tem39  = integ(temdot(tem37,tem39,tem41,v38,v40),tem39i)
v39    = (v38+v40)/2
tem40  = (tem39+tem41)/2
v40    = integ(vdot(v38,v40,v42,tem39,tem41),0)
tem41  = integ(temdot(tem39,tem41,tem43,v40,v42),tem41i)
v41    = (v40+v42)/2
tem42  = (tem41+tem43)/2
v42    = integ(vdot(v40,v42,v44,tem41,tem43),0)
tem43  = integ(temdot(tem41,tem43,tem45,v42,v44),tem43i)
v43    = (v42+v44)/2
tem44  = (tem43+tem45)/2
```

```
v44    = integ(vdot(v42,v44,v46,tem43,tem45),0)
tem45  = integ(temdot(tem43,tem45,tem47,v44,v46),tem45i)
v45    = (v44+v46)/2
tem46  = (tem45+tem47)/2
v46    = integ(vdot(v44,v46,v48,tem45,tem47),0)
tem47  = integ(temdot(tem45,tem47,tem49,v46,v48),tem47i)
v47    = (v46+v48)/2
tem48  = (tem47+tem49)/2
v48    = integ(vdot(v46,v48,v50,tem47,tem49),0)
tem49  = integ(temdot(tem47,tem49,tem51,v48,v50),tem49i)
v49    = (v48+v50)/2
tem50  = (tem49+tem51)/2
v50    = integ(vdot(v48,v50,v52,tem49,tem51),0)
tem51  = integ(temdot(tem49,tem51,tem53,v50,v52),tem51i)
v51    = (v50+v52)/2
tem52  = (tem51+tem53)/2
v52    = integ(vdot(v50,v52,v54,tem51,tem53),0)
tem53  = integ(temdot(tem51,tem53,tem55,v52,v54),tem53i)
v53    = (v52+v54)/2
tem54  = (tem53+tem55)/2
v54    = integ(vdot(v52,v54,v56,tem53,tem55),0)
tem55  = integ(temdot(tem53,tem55,tem57,v54,v56),tem55i)
v55    = (v54+v56)/2
tem56  = (tem55+tem57)/2
v56    = integ(vdot(v54,v56,v58,tem55,tem57),0)
tem57  = integ(temdot(tem55,tem57,tem59,v56,v58),tem57i)
v57    = (v56+v58)/2
tem58  = (tem57+tem59)/2
v58    = integ(vdot(v56,v58,v60,tem57,tem59),0)
tem59  = integ(temdot(tem57,tem59,tem61,v58,v60),tem59i)
v59    = (v58+v60)/2
```

```
tem60  = (tem59+tem61)/2
v60    = integ(vdot(v58,v60,v62,tem59,tem61),0)
tem61  = integ(temdot(tem59,tem61,tem63,v60,v62),tem61i)
v61    = (v60+v62)/2
tem62  = (tem61+tem63)/2
v62    = integ(vdot(v60,v62,v64,tem61,tem63),0)
tem63  = integ(temdot(tem61,tem63,tem65,v62,v64),tem63i)
v63    = (v62+v64)/2
tem64  = (tem63+tem65)/2
v64    = integ(vdot(v62,v64,v66,tem63,tem65),0)
tem65  = integ(temdot(tem63,tem65,tem67,v64,v66),tem65i)
v65    = (v64+v66)/2
tem66  = (tem65+tem67)/2
v66    = integ(vdot(v64,v66,v68,tem65,tem67),0)
tem67  = integ(temdot(tem65,tem67,tem69,v66,v68),tem67i)
v67    = (v66+v68)/2
tem68  = (tem67+tem69)/2
v68    = integ(vdot(v66,v68,v70,tem67,tem69),0)
tem69  = integ(temdot(tem67,tem69,tem71,v68,v70),tem69i)
v69    = (v68+v70)/2
tem70  = (tem69+tem71)/2
v70    = integ(vdot(v68,v70,v72,tem69,tem71),0)
tem71  = integ(temdot(tem69,tem71,tem73,v70,v72),tem71i)
v71    = (v70+v72)/2
tem72  = (tem71+tem73)/2
v72    = integ(vdot(v70,v72,v74,tem71,tem73),0)
tem73  = integ(temdot(tem71,tem73,tem75,v72,v74),tem73i)
v73    = (v72+v74)/2
tem74  = (tem73+tem75)/2
v74    = integ(vdot(v72,v74,v76,tem73,tem75),0)
tem75  = integ(temdot(tem73,tem75,tem77,v74,v76),tem75i)
```

```
      v75    = (v74+v76)/2
      tem76  = (tem75+tem77)/2
      v76    = integ(vdot(v74,v76,v78,tem75,tem77),0)
      tem77  = integ(temdot(tem75,tem77,tem79,v76,v78),tem77i)
      v77    = (v76+v78)/2
      tem78  = (tem77+tem79)/2
      v78    = integ(vdot(v76,v78,v80,tem77,tem79),0)
      tem79  = integ(temdot(tem77,tem79,tem81,v78,v80),tem79i)
      v79    = (v78+v80)/2
      tem80  = (tem79+tem81)/2
      vdot80 = ((tlam/tem80)**5.6 -1) *...
               ( a0*(v80-2*v78+v76)/(4*dx**2) -...
                slam*(tem80/tlam)**5.6*(tem81-tem79)/(2*dx)) -...
              v80 *gm(tem80)*(1/(1-(tem80/tlam)**5.6))**2 * rho * v80**2
      v80    = integ(vdot80,0)
      temd81 = k/(rho*5.6*slam) * (tlam/tem81)**5.6 *...
                   (tem77-2*tem79+tem81)/(4*dx**2) +...
               gm(tem81)*rho*(1/(1-(tem81/tlam)**5.6))*v81**4/(5.6*slam)
      tem81  = integ(temd81,tem81i)
      v81    = v79

    END $"---------------------------------------------- of DERIVATIVE"
    DISCRETE q0stop
      q0 = 0
    END $" of DISCRETE q0stop"
    termt (t.gt.tmax .or. (tem81-tem0).gt. 1E-2)
  END $" of DYNAMIC"
END $" of PROGRAM"
```

Bibliography

[1] V. Peshkov, "'Second Sound' in Helium II," *J. Phys. USSR*, **v. 8**, p. 381 (1944).

[2] V. Peshkov, "Determination of the velocity of Propagation of the Second Sound in Helium II," *J. Phys. USSR*, **v. 10**, pp. 389–398 (1946).

[3] E. S. R. Gopal, *Specific Heats at Low Temperatures*, Plenum Press, New York (1966).

[4] Fritz London, *Superfluids Vol. II*, John Wiley & Sons, New York (1954).

[5] K. R. Atkins, *Liquid Helium*, Cambridge University Press (1959).

[6] W. F. Vinen, "Physical Properties of Superfluid Helium. A General Review," *Inst. Int. Froid, Int. Inst. Refrig.*, Saclay, Refrig. Sci. Technol (1981).

[7] S. J. Putterman, *Superfluid Hydrodynamics*, North–Holland Publishing Company, Amsterdam London (1974).

[8] H. C. Kramers, J. D. Wasscher, C. J. Gorter, "The Specific Heat of Liquid Helium between 0.25 and 1.9°K," *Physica*, **v. 18**, n. 5, pp. 329–338 (1952).

[9] W. F. Vinen, "Mutual friction in a heat current in liquid helium II, I. Experiments on steady heat currents," *Proc. Roy. Soc. A*, **v. 240**, pp. 114–127 (1957).

[10] M. J. Buckingham, W. M. Fairbank, "The Nature of the λ–transition in Liquid Helium," *Progress in Low Temp. Phys.*, **v. 3**, pp. 80–112 (1961).

[11] R. A. Ferrell, N. Menyhàrd, H. Schmidt, F. Schwabl, P. Szépfalusy, "Fluctuations and Lambda Phase Transition in Liquid Helium," *Annals of Physics*, **v. 47**, pp. 565–613 (1968).

[12] M. Archibald, J. M. Mochel, L. Weaver, "Size Dependence of the Thermal Conductivity of Helium I near the λ Point," *Phys. Rev. Letters*, **v. 21**, n. 16, pp. 1156–1158 (1968).

[13] Guenter Ahlers, "Thermal Conductivity of He I near the Superfluid Transition," *Phys. Rev. Letters*, **v. 21**, n. 16, pp. 1159–1162 (1968).

[14] Paul E. Dimotakis, "Local temperature measurements in supercritical counterflow in liquid helium II," *The Physics of Fluids*, **v. 16**, n. 11, pp. 1787–1795 (1973).

[15] Paul E. Dimotakis, "Gorter–Mellink scale, and critical velocities in liquid–helium–II counterflow," *Physical Review A*, **v. 10**, n. 5, pp. 1721–1723 (1974).

[16] Michael J. Crooks, David L. Johnson, "Temperature Differences and Mutual Friction in He II Near T_λ," *Canadian Journal of Physics*, **v. 49**, pp. 1035–1039 (1971).

[17] W. Fiszdon, J. Piechna, W. Poppe, *A qualitative numerical analysis of some features of propagation of counterflow temperature waves in superfluid helium*, Max–Planck–Inst. Strömungsforschung, Göttingen, West Germany, Bericht 9/1988 (1988).

[18] D. Gentile, J. Pakleza, "Numerical solution of transient heat transport through Helium II," *Numerical Heat Transfer*, **v. 6**, pp. 317–335 (1983).

[19] D. Gentile, J. Pakleza, "An analytical and numerical investigation of heat transport in superfluid helium," *Arch. Mech.*, **v. 38**, pp. 585–593 (1986).

[20] R. A. Hull, K. R. Wilkinson, J. Wilks, *Proc. phys. Soc., London A*, **v. 64**, pp. 379 (1951).

[21] François E. Cellier, *Continuous System Modeling*, Springer–Verlag, New York (1991).

[22] Hilding Elmqvist, *A Structured Model Language for Large Continous Systems*, Ph.D. Thesis, Report CODEN: LUTFD2/(TRFT–1015), Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden (1978).

[23] Granino A. Korn, *Interactive Dynamic System Simulation*, McGraw–Hill, New York (1989).

[24] L. D. Landau, *J. Phys.*, Moscow, **v. 5**, pp. 71 (1941).

[25] P. Kapitza, *J. Phys.*, Moscow, **v. 5**, pp. 60 (1941).

[26] I. M. Khalatnikov, *Introduction to the Theory of Superfluidity*, Benjamin, New York (1965).

[27] *Advanced Continuous Simulation Language (ACSL), User Guide/Reference Manual*, Mitchell and Gauthier, Assoc., Concord, Mass. (1981).

[28] R. J. Donnely, *Experimental Superfluidity*, Univ. of Chicago Press, Chicago, Ill. (1967).