ABSTRACT

BAIRD, LORIN RICHARDS. Linear Hydrodynamic Response in a Uniform Fermi Gas. (Under the direction of John Thomas).

In this dissertation, we have developed a new probe for hydrodynamic response in ultracold atomic gases, made possible by the use of digital micromirror technology. As a first step, we directly measure the density response δn of a unitary Fermi gas confined in a box potential $U_0(z)$ and subject to a spatially periodic optical potential $\delta U(z, t)$. We create a hydrodynamic linear response model, and fit it to the data to extract transport properties, namely the thermal conductivity κ_T. We use the previously measured shear viscosity, and measured equation of state as inputs to our model.

The box trap $U_0(z)$ and optical potential $\delta U(z, t)$ are created with digital micromirror devices (DMDs), giving us independent control over the wavelength λ and frequency f of $\delta U(z, t)$. This allows us to translate $\delta U(z, t)$ into the cloud at speeds ranging from subsonic to supersonic.

We show that the time-dependent change of the density profile is sensitive to the thermal conductivity κ_T, which controls the relaxation rate γ_κ of the temperature gradients and hence the responses arising from adiabatic and isothermal compression. The thermal conductivity extracted from the density response profiles is found to be $\kappa_T = 1.14(17) \times (15/4) \, (k_B/m) \, \hbar \, n_0$, at a reduced temperature $\theta_0 = 0.50$.

We compare our measurement of the thermal conductivity with variational calculations for a unitary gas in the high temperature 2-body Boltzmann limit. These calculations determine the ratio of the thermal conductivity κ_T to shear viscosity η_0 to be $\kappa_T/\eta_0 = (15/4) \, (k_B/m)$. Using the measured shear viscosity η_0, we find that the ratio of the thermal conductivity to the shear viscosity is $(\kappa_T/\eta_0) = 0.93(14) \times (15/4) \, (k_B/m)$.

Linear Hydrodynamic Response in a Uniform Fermi Gas

by
Lorin Richards Baird

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Physics

Raleigh, North Carolina

2019

APPROVED BY:

_____
Thomas Schäfer

_____
Kenan Gundogdu

_____
Michael Kudenov

_____
John Thomas
Chair of Advisory Committee

DEDICATION

For Sarah, Miranda, and Madeline.

BIOGRAPHY

Lorin Baird was born in 1988 in Cedar City, Utah. He received his B.S. in physics with a minor in mathematics from Brigham Young University-Idaho in April of 2012. He joined Dr. John Thomas's group at North Carolina State University in August of 2013 and conducted experiments on hydrodynamics in ultracold Fermi gases.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

x

Chapter 1

Introduction

This dissertation describes the measurement of hydrodynamic linear response in a unitary Fermi gas. We utilize a sample of fermionic ^{6}Li in an optical box trap $U_0(z)$ that is nearly uniform in density and covers a smaller region in phase space than a typical harmonic trap. The uniform region is hydrodynamic and allows us observe and measure transport properties. The thermal conductivity κ_T is measured by moving a periodic optical potential $\delta U(z,t)$ through the sample with varying the wavelength λ and speed v. The decay rate $\gamma_\kappa \propto \kappa_T/\lambda^2$ of temperature gradients created by the periodic compression changes the timescale that the temperature gradients are allowed to relax. The effects from these relaxing temperature gradients are observed in the density response δn which is recorded as a function of this periodic optical potential's speed v and wavelength λ. We create a linear hydrodynamic model to extract the thermal conductivity κ_T from this density response.

Strongly interacting fermions play a central role in the structure of matter. Unitary Fermi gases have connections to condensed matter, high energy physics, astrophysics, nuclear physics, and string theory. Our tabletop system provides a clean, controllable
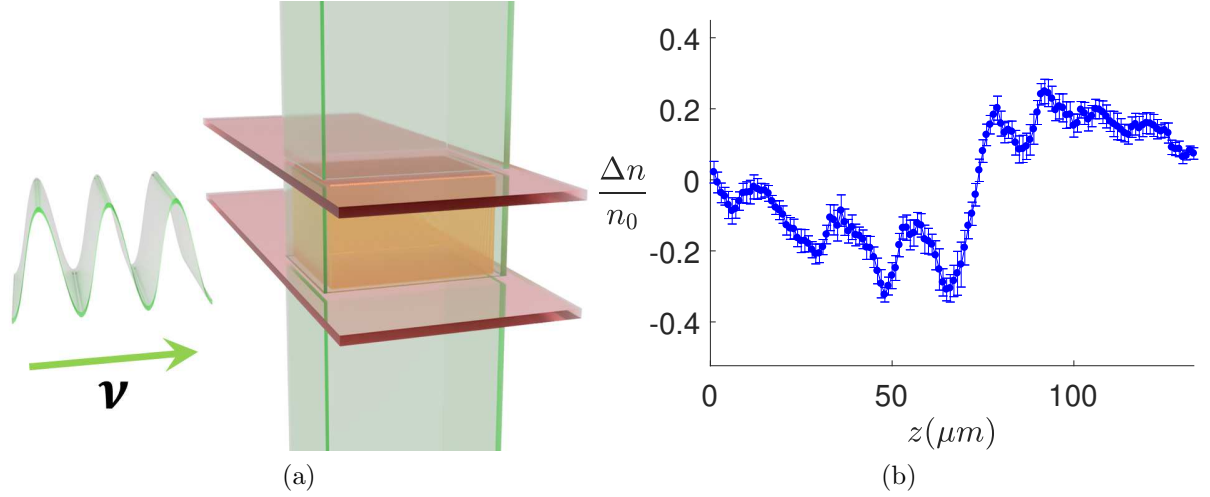
Figure 1.1: (a) Pictorial representation of the experiment. (b) Typical density response.

way for quantitative comparison between experiment and many body theory. The unitary Fermi gas shares universal behavior with other strongly interacting Fermi systems [1, 3, 13, 22, 27] such as the strongly coupled electrons in high T_C superconductors, neutron matter, and the quark-gluon plasma (QGP), a state of matter that existed moments after the big bang.

This condition for this universal behavior arises from the two-body scattering process of our quantum particles which we will briefly cover here. We will consider a two body scattering process where our incoming particle with momentum $\mathbf{p} = \hbar \mathbf{k}$ can be described by a plane wave e^{ikz}. A scattering event from a spherically symmetric scattering potential in the far field can be approximated as a spherical wave. The information from the scattering event is contained in the scattering amplitude $f(\theta, k)$, where θ is the angle between the incoming and outgoing wave in spherical coordinates. We can describe the area where a two body scattering even occurs as the collision cross section σ_c.

If we take the ratio of the probability of observing a scattered particle at the differential solid angle $d\Omega$ compared to the incident probability perpendicular to the incoming wave e^{ikz} we can obtain the equation

$$\frac{d\sigma_c}{d\Omega} = |f(\theta, k)|^2. \tag{1.1}$$

The scattering amplitude in the far field is

$$f(\theta) = \sum_{l=0}^{\infty} (2l+1)\, a_l(k)\, P_l(\cos\theta), \tag{1.2}$$

where $P_l(cos(\theta))$ are Legendre polynomials. The coefficients for each polynomial are found from the radial Schrödinger equation,

$$a_l(k) = \frac{e^{2i\delta_l} - 1}{2ik} = e^{i\delta_l} \frac{\sin\delta_l}{k}, \tag{1.3}$$

where l is the angular momentum and δ_l is the phase shift in a scattered wave compared to an unscattered wave. In our low temperature gas we have s-wave scattering retaining only the $l = 0$ term. Integrating equation 1.1 over the solid angle $d\Omega = \sin\theta\, d\theta\, d\phi$ gives us the collision cross sectional area

$$\sigma_c = \frac{4\pi}{k^2} \sin^2\delta. \tag{1.4}$$

3

If the scattering potential is spherically symmetric, the collision cross section can be described by a one dimensional length called the scattering length a_s. If we solve the radial Schrödinger equation $d^2u/dr^2 + 2\mu_m/\hbar^2 Eu = 0$ in the far field, away from the range of the potential we obtain

$$u(r) = A \, \sin(k\,r + \delta). \tag{1.5}$$

The linearized solution for low energy collisions crosses the r axis when

$$0 = A\left(1 + \frac{k}{\tan\delta}r\right)\sin\delta. \tag{1.6}$$

If we solve equation 1.6 for r we obtain our scattering length

$$a_s \equiv -\lim_{k\to 0}\frac{\tan\delta}{k}. \tag{1.7}$$

Using trig. identities gives us our collision cross section in terms of the scattering length

$$\sigma_c = \frac{4\pi}{k^2}\sin^2\delta = \frac{4\pi}{k^2}\frac{\tan^2\delta}{1+\tan^2\delta} = \frac{4\pi a_s^2}{1+(ka_s)^2}. \tag{1.8}$$

For diverging scattering length our collision cross section is simply

$$\sigma_c = \frac{4\pi}{k^2}. \tag{1.9}$$

When the s-wave scattering length a_s is much greater than the range R of the collisional potential, both parameters vanish from the problem. The gas can be completely characterized by the de Broglie wavelength λ_T and the interparticle distance, or inverse

Fermi wavenumber $1/k_F$. A rescaling of the spatial coordinates will simply rescale the Hamiltonian. Dimensional arguments then allow us to write all measurable quantities in universal form in terms of λ_T and $1/k_F$. This strongly interacting scale invariant system is a unitary Fermi gas [22].

Working at unitarity, we have a parameter free comparison of our ultracold dilute gas to different strongly interacting fermionic systems such as the QGP which is about 20 orders of magnitude higher in temperature. Neutron matter in neutron stars, which is about 25 orders of magnitude more dense than our dilute gas, is naturally strongly interacting because the neutron-neutron interactions have a large scattering length $a_s \approx -18.5\ f$m compared to the magnitude of the interaction range $R \approx 1\ f$m. The unitary condition is not naturally occurring in our dilute ultracold sample, and must be created by operating near a magnetically tunable collisional resonance.

A magnetically tunable collisional resonance, called a Feshbach resonance, allows us to change the scattering length between two fermionic particles by applying a magnetic field. To understand how this Feshbach resonance works, we need to look at the two particle collisions that occur in our gas. Our sample is made up of the two lowest energy hyperfine ground states of ^{6}Li. These two states can approach each other along two different channels, a shallow triplet potential where the electronic spin states are parallel, or a much deeper singlet potential where the spins are anti-parallel. When a magnetic field is applied, the difference in magnetic moments shifts the energy of the triplet collisional channel downward with respect to the singlet collisional channel. The triplet channel is more energetically favorable and atoms will scatter along this potential. The singlet potential, while energetically closed in a magnetic field, has bound states that are now near the energy of the open channel. A Feshbach resonance occurs when the scattering continuum of the open channel is tuned with a magnetic field near a bound state of

the closed channel. This equates to scattering from an attractive well with the scattered wavefunction having a phase shift δ which changes the scattering length a_s as seen in equation 1.7. Near a Feshbach resonance, the scattering length as a function of magnetic field, shown in Fig. 1.2, can be approximated as,

$$a = a_b \left(1 - \frac{\Delta}{B - B_0} \right), \tag{1.10}$$

where $a_b = -1582(1)a_0$ is the unmodified scattering length, $\Delta = -262.3(3)$ is the resonance width, B is the magnetic field, and $B_0 = 832.18(8)$ is the resonant magnetic field [33]. When the continuum energy is tuned to be degenerate to a bound state in the closed channel, the scattering length diverges. We obtain our universal system by operating near a Feshbach resonance where the diverging scattering length is much greater than the interaction range.
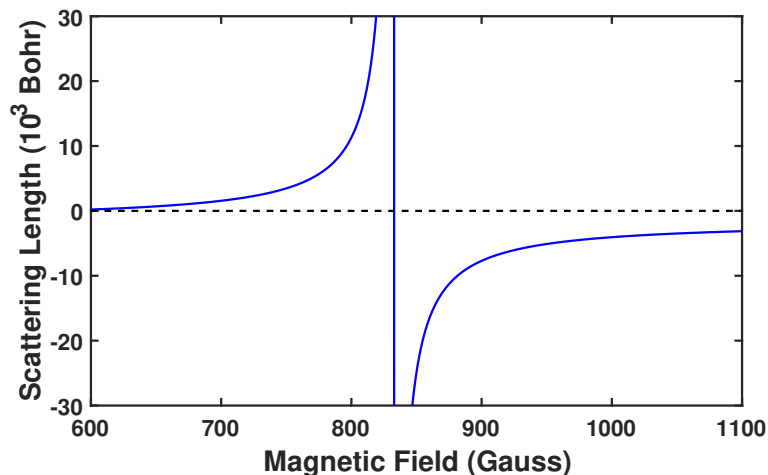
Figure 1.2: Tuning the magnetic field below the Feshbach resonance results in a molecular Bose Einstein condensate (BEC). Above the resonance is described by the Bardeen-Cooper-Schrieffer theory of a Fermionic system. The crossover region where the scattering length diverges is a unitary Fermi gas.

1.1 Thermodynamics and transport measurement in the unitary Fermi gas

A great deal of work has gone into measuring the universal numbers that describe the unitary gas and its thermodynamic properties. Dilute atomic gases held in equilibrium have provided the thermodynamic properties over a wide range of temperatures using fundamental thermodynamic relations [24, 25]. We use the equation of state obtained from these experiments as an input to our hydrodynamic model.

Transport measurements for coefficients such as the shear viscosity η, bulk viscosity ζ, and thermal conductivity κ_T require dynamical experiments. Bulk viscosity in the unitary limit is predicted to vanish in the normal phase [15, 28]. Releasing a gas from an anisotropic trap and observing the hydrodynamic expansion gives measurements of the bulk viscosity consistent with 0 in the normal fluid regime [10]. Shear viscosity is also determined by releasing the gas from an anisotropic trap and tracking the hydrodynamic expansion as a function of the reduced temperature at the center of the cloud [19]. When released from an anisotropic trap, the pressure gradients are much larger in the narrow direction. As the gas expands, the aspect ratio will reach a maximum value at large time t which is determined by the shear viscosity η. An analysis [4] of the expansion data which takes into account the non-hydrodynamic modes of the less dense edge of the cloud gives the shear viscosity η in the normal fluid regime as a function of temperature and density

$$\eta = \eta_0 \frac{(mk_BT)^{3/2}}{\hbar^2} \left[1 + \eta_2(n\lambda_T^3) + ... \right],\qquad(1.11)$$

where $\lambda_T = h/\sqrt{2\pi\, m\, k_B\, T}$ is the de Broglie wavelength and n is the total density for a balanced two-component mixture. The first two terms in the expansion, $\eta_0 = 0.265(20)$

and $\eta_2 = 0.060(20)$, fit the expansion data from temperatures just above the super-fluid transition T_C to the high temperature limit where variational two-body Boltzmann equation for a unitary Fermi gas gives $\eta_0 = 15/(32\sqrt{\pi}) = 0.26446$ [4]. The thermal conductivity κ_T has a negligible effect on expansion experiments, because the clouds are in thermal equilibrium.

Thermal conductivity is a coefficient that describes the ability of a substance to conduct heat between areas of unequal temperature. In this equation, we can see the relationship between the heat current $\mathbf{J_T}$, thermal conductivity κ_T and the temperature T

$$\mathbf{J_T} = -\kappa_T \, \nabla T. \tag{1.12}$$

In order to measure the thermal conductivity, we need a cloud of non-uniform temperature.

1.1.1 Primary results

My experiment, pictorially shown in Fig. 1.1, shows a repulsive periodic optical potential which is moved through a uniform sample, creating spatially varying temperature profiles. With control over the speed and wavelength of this potential, we can change the pressure and temperature gradients, and vary the timescales over which these gradients are allowed to relax. The density response δn has contributions from both the shear viscosity and the thermal conductivity. With this shear viscosity known, and vanishing bulk viscosity in the normal fluid regime, the only transport coefficient that remains is the thermal conductivity.

The measured density response, observed as a function of λ and v of the periodic optical potential gives us regimes where the temperature gradients relax slowly or quickly.

When the decay rate for the temperature change $\gamma_\kappa \propto \kappa_T/\lambda^2$ is larger than the wave frequency ω of the periodic optical potential, $\gamma_\kappa >> \omega$, the compression is isothermal. When $\gamma_\kappa << \omega$, the temperature gradients relax slowly and the compression is adiabatic. Our model has good sensitivity to measure the thermal conductivity when the periodic potential is moved through the gas at subsonic speeds. Moving $\delta U(z,t)$ through the gas at supersonic speed results in nonlinear effects such as the creating of weak shockwaves. The thermal conductivity is impossible to extract with a linear hydrodynamic model when the density response is nonlinear and sound waves are propagating adiabatically.

Variational calculations for a unitary gas in the 2-body Boltzmann high temperature limit [5,6] determines the ratio of the thermal conductivity κ_T to shear viscosity η to be

$$\frac{\kappa_T}{\eta_0} = \frac{15}{4}\frac{k_B}{m}. \tag{1.13}$$

Comparing the experimental measurement of the thermal conductivity at $\theta_0 = 0.50$ with the high temperature ratio, we have

$$\frac{\kappa_T}{\eta_0} = 0.93(14) \times \frac{15}{4}\frac{k_B}{m}. \tag{1.14}$$

where we have used the previously measured shear viscosity $\eta_0 = 1.23\,\hbar n_0$ [4,19].

Chapter 2

Experimental Setup

This chapter summarizes the cooling and trapping techniques used in the experiment and the addition of micro-mirror technology used to create boxlike traps $U_0(z)$ and project moving periodic optical potentials $\delta U(z, t)$, shown in Fig. 2.1. Some techniques and equipment have been covered in great detail in previous theses [7, 9, 12, 18, 21], but will be described briefly in this chapter to provide background for the current work.

Our apparatus uses two physical mechanisms for cooling and trapping the atomic sample. The first mechanism slows and traps atoms through the repeated absorption and emission of photons near the 671 nm transition of ^{6}Li. Photon absorption and emission are used to create velocity and spatially dependent radiation pressure in the Zeeman slower and magneto-optical trap or MOT, which provides a source of cold atoms.

The second cooling and trapping mechanism uses an electric dipole polarizability interaction where the frequency of a focused electric field is far away the atomic resonance to avoid optical scattering and heating. The electric dipole force, previously used only to create a harmonic trap, is now also used to create the six sheet beams that make up the box trap $U_0(z)$ and the perturbative optical potential $\delta U(z, t)$.
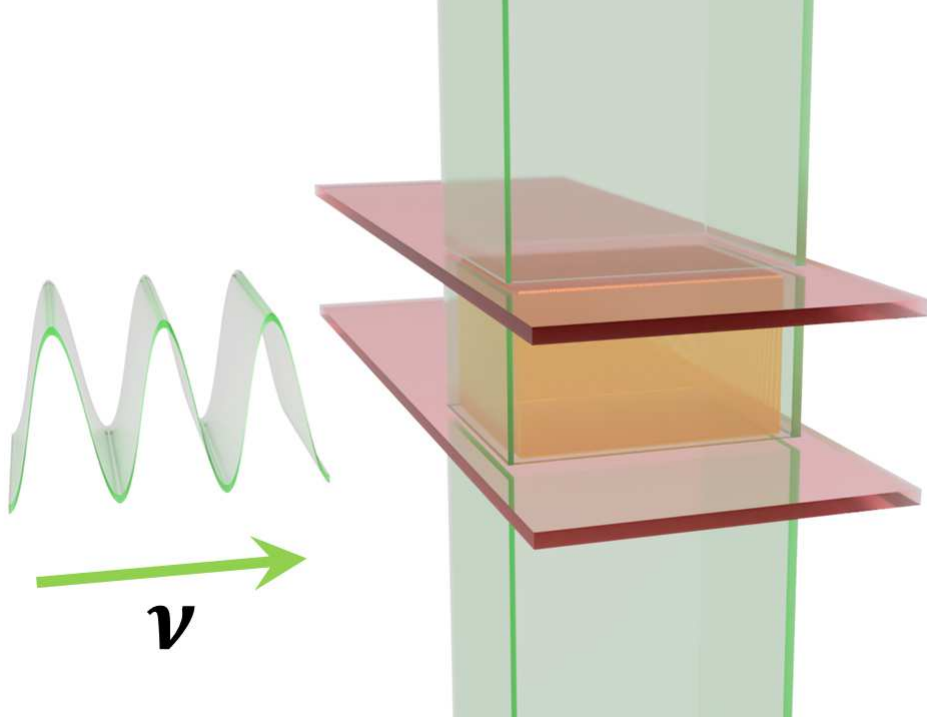
Figure 2.1: Pictorial representation of the box trap $U_0(z)$ shown on the right contains a uniform sample (orange) with a perturbative optical potential $\delta U(z, t)$ shown on the left.

We use absorption imaging to extract the density profile of the cloud. This technique uses a pulse of resonant light to image a shadow of the density distribution from the atomic cloud on a CCD camera. This imaging process destroys the atomic sample, and another one must be created.

2.0.1 Velocity dependent radiation pressure

The first cooling mechanism uses the momentum of photons $p = \hbar k = h/\lambda$ to reduce the velocity of the atoms. A photon absorbed by an atom will transfer momentum to the atom in the direction that the photon was traveling. When an atom in an excited state spontaneously emits a photon in a random direction, it will recoil in the opposite

direction. A laser beam tuned near resonance with an atomic transition will provide a net momentum transfer in the direction of the laser propagation, while the net momentum transfer from the spontaneous emission of photons will average to zero. Atoms moving with a velocity v_a in the direction of a laser will have a shifted resonance frequency

$$f = \left(1 - \frac{v_a}{c}\right) f_0, \tag{2.1}$$

from the Doppler effect, where f is the frequency observed by the atom, f_0 is the laser frequency and c is the speed of light. An atom co-propagating along a laser beam, $v_a < 0$, that is tuned below resonance will absorb upshifted photons and decrease in velocity. An atom propagating in the same direction as a laser beam tuned below resonance it will not absorb any photons as the downshifted photons will not be in resonance with the atomic transition. In one dimension, two overlapping co-propagating laser beams that have been tuned below the atomic resonance will slow atoms moving in either direction. This velocity dependent radiation pressure, also called optical molasses, is used to slow down atoms in three dimensions by overlapping three orthogonal co-propagating laser beams pairs. Atoms slowed by this velocity dependent radiation pressure will gradually drift out of the laser beam path without spatial confinement.

2.0.2 Spatially dependent radiation pressure

A spatially varying magnetic field will create a spatially dependent energy shift that we use to trap atoms. A magnetic field will energetically shift, or Zeeman shift, the $m_F \pm 1$ excited state sublevels in ^{6}Li. The red-detuned laser beams used to create a velocity dependent radiation pressure are cicrularly polarized to provide a spatially dependent radiation pressure to atoms in a spatially varying magnetic field. Atoms in a region of

positive magnetic field will have $m_F = -1$ excited state sublevel shifted down in energy and will absorb circularly polarized σ_- photons. Atoms in negative magnetic field will have the $m_F = 1$ excited sublevel downshifted and will absorb σ_+ polarized photons.

Consider a one dimensional case where atoms are free to move in a magnetic field that increases linearly in the positive direction about a zero point. A beam propagating in the positive direction with σ_+ polarization is overlapped with a beam propagating in the negative direction with σ_- polarization. Atoms in regions of negative magnetic field absorb σ_+ photons and receive momentum kicks in the positive direction. Atoms in regions of positive magnetic field absorb σ_- photons and receive momentum kicks in the negative direction. The combined result of the magnetic and optical fields creates a spatially dependent radiation pressure that confines atoms near the zero point of the magnetic field.

The spatially dependent radiation pressure, in addition to the velocity dependent radiation pressure is used to trap and cool atoms in three dimensions. One dimensional radiation pressure is used in a Zeeman slower to slow an atomic beam of vapor with thermal velocities of over 1000 m/s down to a few m/s over a distance of approximately 30 cm.

2.0.3 Zeeman slower

A constant flux of atoms is provided as a collimated atomic vapor traveling from the source oven, heated to approximately 400° C, through a small hole at the end of a pipe. The atomic beam enters the Zeeman slower where it is overlapped with a co-propagating red-detuned laser in a spatially varying magnetic field used to produce a Zeeman shift. The magnitude of the magnetic field, created by a series of electromagnetic coils, is set to

spatially decrease as the atoms move along the Zeeman slower. Atoms with large velocities are slowed by absorbing co-propagating photons that are brought into resonance by the Doppler effect and Zeeman shift from the magnetic field. As the atoms moving down the Zeeman slower are decelerated by the repeated absorption and emission of photons, the magnetic field is spatially decreased to keep the atoms close to resonance with the laser beam.

2.0.4 Magneto-optical trap

After exiting the Zeeman slower, atoms are trapped and cooled by three retro reflected orthogonal beams overlapping near the point of zero magnetic field gradient of a quadrupole magnetic field. The combination of velocity and spatially dependent radiation pressure used to cool and confine atoms, shown in Fig. 2.2, is called a magneto-optical trap or MOT. The MOT at full intensity spatially confines the atoms and cools them to $\sim 1 \, m$K.

The mechanism for trapping and cooling in the MOT relies on the momentum transfer from absorbed photons. The recoil momentum from photons emitted from atoms in the MOT puts a lower bound on the temperatures that can be achieved in the MOT which is $\sim 150 \, \mu$K when the intensity is lowered by a factor of ~ 100 and the frequency is brought closer to resonance. The dipole force trapping mechanism, which doesn't rely on the momentum of photons for trapping and cooling, can achieve lower temperatures by evaporatively cooling atoms held in a dipole trap.

2.0.5 Far off resonance dipole trapping

Atoms polarized in the electric field of a laser detuned far from an atomic resonance, will experience a force from the dipole interaction with the electric field gradient $\mathbf{F} = -\nabla V$,
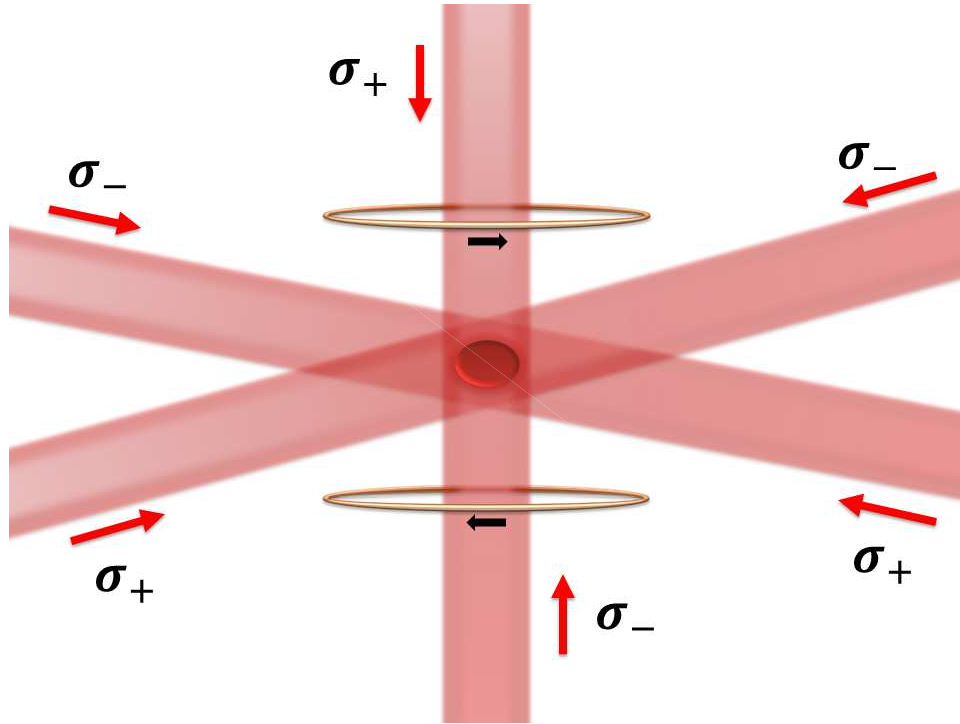
Figure 2.2: The magneto-optical trap (MOT). Three orthogonal retro-reflected laser beams along with the quadrupole magnetic field generated by two electromagnetic coils in an anti-Helmholtz configuration trap and cools the atoms exiting the Zeeman slower.

where the cycle-averaged potential of the electric field is

$$V = -\frac{1}{2}\alpha\,\overline{\mathbf{E^2}} = \frac{-2\pi}{c}\,\alpha(\omega)\,I(x,y,z), \tag{2.2}$$

where c is the speed of light, $\alpha(\omega)$ the polarizability of the sample, and $I(x,y,z)$ the intensity of the laser beam. The dipole polarizability of ^{6}Li

$$\alpha = \frac{\alpha_0\,\omega_0^2}{\omega_0^2 - \omega^2}, \tag{2.3}$$

is determined by the frequency of the laser light ω, where $\alpha_0 = 24.310^{-24}$ cm is the static polarizability [30] and ω_0 is the resonant frequency. Atoms will be attracted to regions of high laser intensity when the polarizability is positive or repelled when the polarizability is negative.

2.0.6 FORT

When the MOT beams are extinguished, the focused beam of a CO_2 laser, $\lambda_{CO_2} = 10.6\,\mu$m, produces an attractive far off resonance trap (FORT) for further cooling of the atoms through evaporation. The laser is tuned far below resonance, $\omega << \omega_0$, to provide frictionless confinement while the most high energy atoms escape from the trap. To aid in efficient evaporative cooling, a magnetic field is applied near a broad Feshbach resonance occurring at 832.2 Gauss [33] giving us a diverging scattering length a_s and strong interactions. Evaporation in the full power FORT cools the atoms to a temperature of $\sim 50\,\mu$K.

Further cooling of the sample is done by forced evaporation. The CO_2 beam power is lowered with an acousto-optic modulator (AOM), a device that uses an RF signal

to generate acoustic waves in a material to diffract a laser beam, altering the intensity and frequency of diffracted modes. An artificial waveform generator provides the control voltage $V(t)$ for the AOM, giving us full control over the trap depth $U(t)$ as a function of time. A lowering curve is programmed into the artificial waveform generator with user defined inputs including the lowering time and final trap depth. Lowering the CO_2 power over 4 seconds with a shallow final trap depth, $U_{final} \approx U_0 \times 10^{-3}$, produces an atomic sample with temperatures as low as ~ 10 nK. Cooling the gas below the superfluid critical temperature T_C will result in some portion of the gas becoming a superfluid. A typical absorption image of an atomic sample at the end of forced evaporation is shown in Fig 2.3.
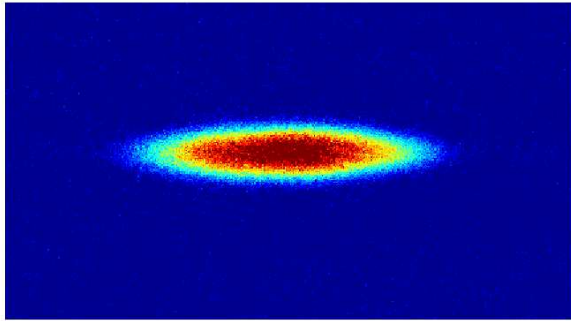
Figure 2.3: Atomic sample in the approximately harmonic FORT after forced evaporation.

The instruments controlling the trap depth $U(t)$ of the attractive dipole trap need to have low electronic noise in order to transfer the atoms from the FORT to the uniform box trap at shallow trap depths. The control voltage provided $V(t)$ to the AOM by the previous artificial waveform generators (Agilent 33250A) was electronically noisy and needed to be integrated by external electronics to quiet the signal. Integrating the control voltage signal resulted in a non-zero trap depth when the CO_2 laser beam was extinguished during the transfer to the box trap. Extinguishing the beams when the trap depth still has finite value results in an abrupt release of the atoms from a harmonic trap to a box trap causing violent collisions with the box walls and unwanted collective oscillations. To obtain a more responsive control voltage for the AOM at low trap depths, we use artificial waveform generators (Keysight 33500B) with low electronic noise which eliminates the need to integrate the signal. Lowering the shallow FORT to zero trap depth over 10 ms before extinguishing the CO_2 beams now provides a gentle release into the box trap without setting up collective oscillations.

2.1 Dynamical and spatial control of repulsive optical potentials

Digital light processing (DLP) technology is emerging as a versatile tool for creating dynamically controlled optical potentials, creating disorder, and imprinting patterns in dilute quantum gases [8,16,26]. Initially, we used a DLP chipset to improve on a previous shockwave experiment [20]. The technology gave us control over the depth and width of a separation between two fermionic atomic clouds in a harmonic potential. When the optical potential creating the separation between the two clouds was extinguished, we

would observe the collision and resulting shockwave in the gas. The DLP technology proved to be a versatile tool in the shockwave experiments and we added a second chipset. The new optical system used to project the images from two digital micromirror devices (DMD) with blue-detuned laser beams is shown in Fig. 2.4.
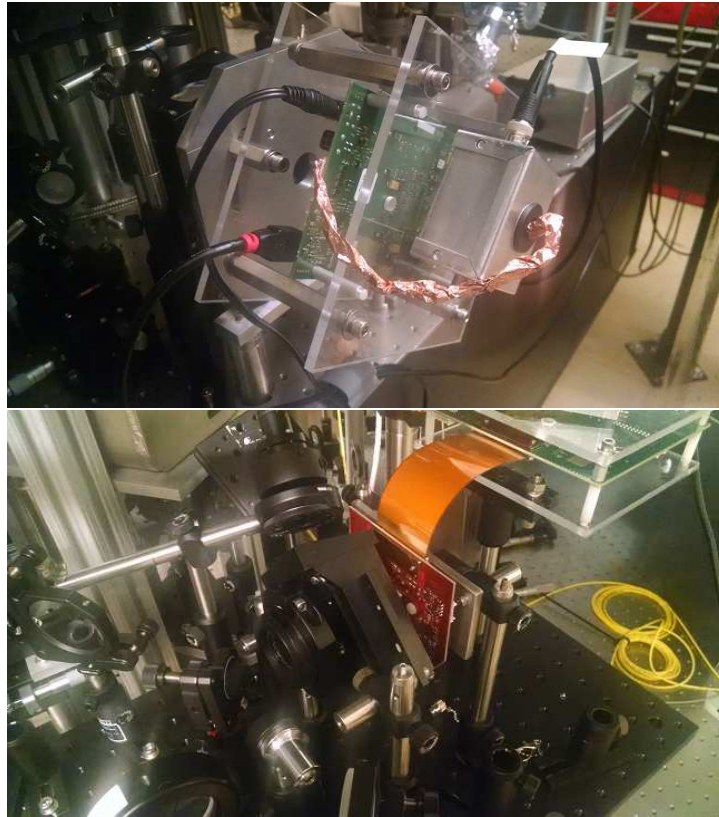
Figure 2.4: The DMDs, control boards, and trigger boxes are mounted on custom machined mounts. This apparatus couples a blue-detuned Gaussian beam from a high power fiber to a DMD. The spatially modulated beam from the DMD is demagnified and projected onto the atom sample to form the optical box potential $U_0(z)$ and perturbation $\delta U(z, t)$.

Most movie theaters across the world use digital light processing (DLP) for video projection. The technology is based on reflecting light from an array of microscopic mirrors

where each individual mirror can be rapidly repositioned to ON or OFF corresponding to a ±12 degree tilt from the flat state. A 1920×1080 resolution array has over two million mirrors, each mirror corresponding to a pixel of an image. The illuminated surface of the micromirror array can be magnified to form the images of a movie at the theater, or demagnified onto the atom sample to create the images of a dynamical optical potential $U_0(z)$ and $\delta U(z,t)$. The surface of the digital micromirror device (DMD) acts like an intensity mask on the Gaussian shaped blue-detuned beam with each mirror tilted to the ON position directing part of the beam to the atoms and mirrors in the OFF position directing part of the beam to a beam dump.

2.1.1 Optical box potential $U_0(z)$

The six sides of the optical box potential $U_0(z)$ are created by a vertically propagating green beam, $\lambda_{green} = 532$ nm, and a horizontally propagating red beam, $\lambda_{red} = 669$ nm. Using laser light with a higher frequency than the atomic transition, $\omega > \omega_0$, gives a negative dipole polarizability α, which results in a repulsive force on the atomic sample.

The green laser beam first passes through an AO for smoothly controlling the power. The output of the AO is coupled into a high-power fiber to create a clean Gaussian profile with good pointing stability at the DMD. At the output of the fiber, shown schematically in Fig. 2.5, we use a combination of two prisms, called a total internal reflection prism (TIR), to correctly angle the laser light onto the DMD and provide the best power efficiency. The array of mirrors on the DMD behaves like a 2D diffraction grating. Illuminating the array from the proper angle yields the most power into the primary diffracted beam. The blue-detuned beam is then demagnified with a microscope objective. This increases the intensity on the atoms compared to a 1:1 projection from
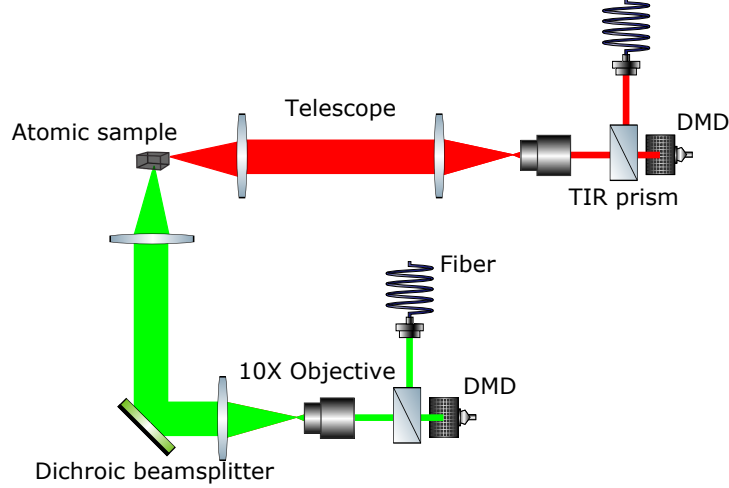
Figure 2.5: Red and green laser light is modulated by DMDs to form our repulsive optical box potential $U_0(z)$ and perturbative potential $\delta U(z,t)$. A blue-detuned repulsive Gaussian beam exits a high power fiber and is coupled to the DMD with a total internal reflection (TIR) prism. The image from the DMD is demagnified by a $10\times$ microscope objective. A telescope projects the demagnified image onto the atomic sample.

the DMD. The demagnified beam from the microscope is then reflected onto the atomic sample with a dichroic beamsplitter and an approximately 1:1 telescope. The dichroic allows the laser light for the optical potential $U_0(z)$ to share the large windows of the imaging beam path by reflecting the green laser light and allows the red imaging beam to pass through the beamsplitter to the CCD camera. The horizontally propagating red laser beam uses an optical system similar to the vertically propagating green beam shown in Fig. 2.5.

The vertically propagating green beam modulated by the DMD creates four sides of the uniform box trap $U_0(z)$ and the perturbative optical potential $\delta U(z,t)$. The spatially shaped beam, shown in Fig. 2.6, made from the 532 nm laser light has a trap depth of approximately 5 times the Fermi energy $E_F = 0.16$ μK.

<center>(a)   (b)</center>
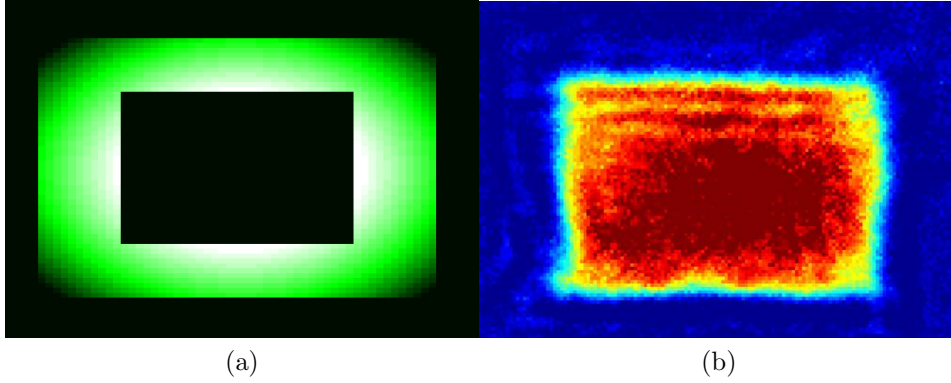
Figure 2.6: (a) The vertically propagating beam, $\lambda_{green} = 532$ nm, spatially modulated by a DMD forms four walls of the optical box potential $U_0(z)$. The effect of the Gaussian profile on the projected intensity can be seen. (b) The optical box potential gives a uniform sample, imaged from the vertical direction, with dimensions $129 \times 84 \times 58$ μm.

The horizontally propagating red beam, $\lambda_{red} = 699$ nm, spatially modulated by a second DMD creates the top and bottom of the optical box trap $U_0(z)$, shown in Fig. 2.7. Using a wavelength closer to the atomic resonance of ^{6}Li at $\lambda = 671$ nm gives dipole polarizability that is approximately 60 times larger than for 532 nm laser light. This allows us to have a much greater force on the atoms with much less laser power at the DMD. Resonant light resulting from amplified spontaneous emission from our red diode laser is eliminated with an Optigrate Bragg grating notch filter.

Gently releasing the atomic sample from the FORT into the optical box trap $U_0(z)$ gives a uniform sample where the peak density is only 10% more than the average density $n_0 = N/V \approx 2.6 \times 10^{11}$ atoms/cm^3 where N is the total atom number, and V is the volume of the box, which has dimensions $(129 \times 84 \times 58)$ μm. The slow variation of the density $n_0(z)$, shown in the horizontal and vertical directions in figures 2.6 and 2.7, is

<center>22</center>
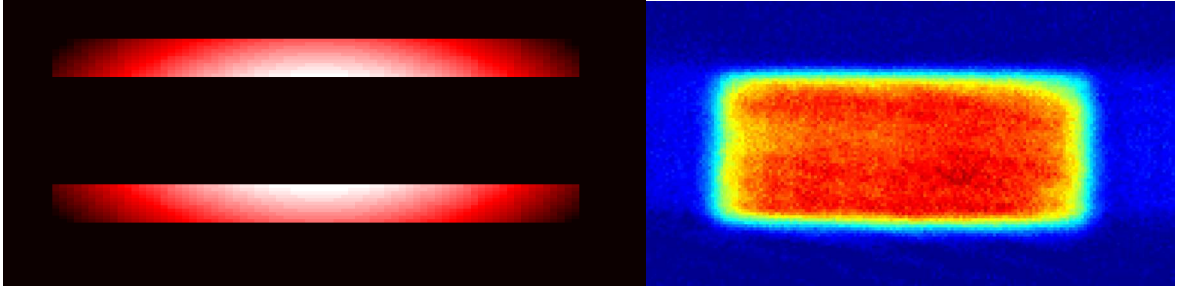
Figure 2.7: (a) The vertically propagating beam, $\lambda_{red} = 669$ nm 532 nm, spatially modulated by a DMD forms the top and bottom of the optical box potential $U_0(z)$. (b) The uniform atomic sample imaged from the horizontal direction.

due to the curvature of the bias magnetic field potential

$$U_{mag} = \frac{1}{2} m \, \omega^2_{xmag} \, x^2 + \frac{1}{2} m \, \omega^2_{ymag} \, y^2 + \frac{1}{2} m \, \omega^2_{zmag} \, z^2. \tag{2.4}$$

The spin states we trap are high field seeking and have a potential from the magnetic dipole moment $U_{mag} = -\boldsymbol{\mu} \cdot \mathbf{B}$ [21], where $\boldsymbol{\mu}$ is the magnetic dipole moment of an atom. The magnetic dipole moment will align along the direction of the magnetic field $\hat{y}$ which is created by electromagnetic coils. Using $\nabla^2 \cdot \mathbf{B} = 0$ we have

$$\nabla^2 \cdot \mathbf{B} = \nabla^2 B \hat{y} = \nabla^2 \left(\frac{-U_{mag}}{\mu} \right) \hat{y} = 0, \tag{2.5}$$

which gives us

$$\omega^2_{xmag} + \omega^2_{ymag} + \omega^2_{zmag} = 0. \tag{2.6}$$

23

Because the atoms are high field seeking, the atoms will experience a repulsive force along the direction of the electromagnet field coils bore $\hat{y}$ and an attractive force in the $\hat{x}$ and $\hat{z}$ direction. The harmonic frequencies

$$-\omega_{ymag}^2 = 2\,\omega_{xmag}^2 = 2\,\omega_{zmag}^2 \qquad (2.7)$$

are obtained from 2.6, where the cylindrical symmetry of the electromagnetic coil demands that $\omega_{ymag}^2 = \omega_{zmag}^2$. In our system, the measured frequency of the magnetic potential is $\omega_{zmag} = 2\,\pi \times 22.7$ Hz.

2.1.2 Perturbative potential $\delta U(z,t)$

An optical perturbation $\delta U(z,t)$ with a depth of $\delta U_0 \approx 0.25 E_F$ is created by modulating the vertically propagating green laser beam that makes up four of the box walls, shown in Fig. 2.8.
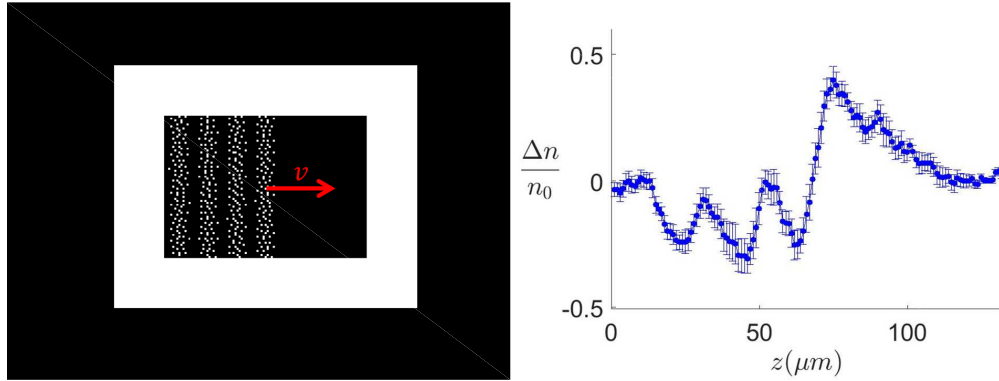
Figure 2.8: The DMD illuminated by green laser light forms four walls for the box trap and creates the perturbative potential $\delta U(z,t)$. The walls $U_0(z)$ remain stationary, while the perturbative potential $\delta U(z,t)$ moves through the uniform atomic sample with velocity v.

The perturbative optical potential, programmed into the DMD [32],

$$\delta U(z,t) = \delta U_0 \left[1 - \epsilon \, \cos(qz - qvt)\right] \phi(vt - z), \qquad (2.8)$$

has velocity v and $q = 2\pi/\lambda$ as programmable input parameters. A smooth step function $\phi(vt - z)$ gives a vanishing perturbation inside the box at $t = 0$ and provides a leading edge for the periodic perturbation moving into the uniform density.

To create a dynamical optical potential, a series of images is loaded into the DLP device memory which is triggered with digital pulses to advance through the images. Each full period of the perturbative potential is made up of ten images. The wave frequency of the spatially periodic perturbation

$$\omega = 2\pi f = 2\pi v / \lambda, \qquad (2.9)$$

is determined by the time between pulses $\Delta t = (1/10)(1/f)$ triggering the image sequence, where λ is the wavelength and v is the velocity. An absorption image is taken when $\delta U(z,t)$ has traveled a fixed number of periods into the sample. For a wavelength of $\lambda = 19 \ \mu$m, the image is taken after four periods, when the leading edge is at 76 μm, as shown in Fig. 2.8. The leading edge of the $\lambda = 30 \ \mu$m perturbation is at 90 μm, after three periods, when the image is taken. The density response δn is recorded over the frequency range of 100 Hz to 1200 Hz for these two wavelengths.

In the experiment, we want to project sinusoidal periodic optical potentials from the DMD onto the uniform sample, but each mirror can only set to a binary state of ON or OFF. We create binary images that simulate smooth images using a halftoning technique that varies the spacing of ON and OFF mirrors shown in Fig. 2.9. A 7.56 μm

mirror demagnified by a $10\times$ microscope objective has an image size that is smaller than the 2.9 μm airy radius, determined from the limited aperture of the vacuum chamber windows which have a diameter $D = 4$ cm. The blurred binary image provides a smooth sinusoidal perturbative potential.

Desired intensity   **Binary image**   **Blurred binary image**
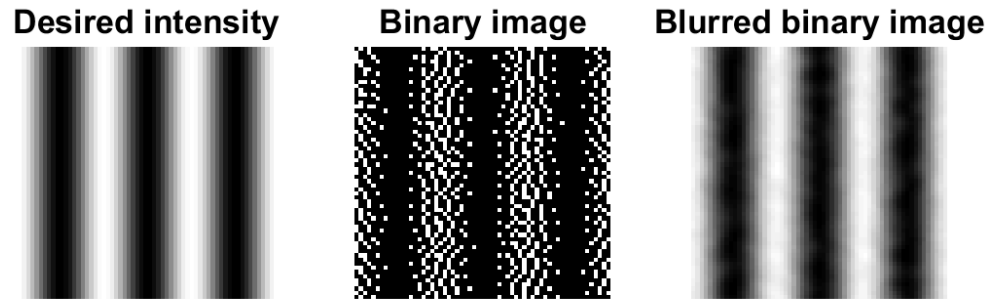
Figure 2.9: Binary images projected from the surface of the DMD are blurred through the finite aperture of our imaging system.

Chapter 3

Thermodynamic Preliminaries

The density response δn to a perturbing optical potential $\delta U(z,t)$ with variable wave frequency $\omega = 2\pi v / \lambda$ is a new probe for measuring transport properties. The measured density response δn of a nearly uniform normal unitary Fermi gas to such a perturbation is characterized by the density n_0, temperature T, equation of state, and the transport coefficients η and κ_T. To develop a hydrodynamic model for our experiment, we need a mathematical description of the change in pressure δP, change in temperature δT, and thermodynamic quantities such as the heat capacity. Fortunately, the equation of state for the unitary Fermi gas was recently measured by a group at MIT [25] which greatly aids in our analysis and allows us to determine the temperature of our gas in-situ from the measured sound speed. In this chapter we will cover the thermodynamic preliminaries needed to build a fluid dynamics description in the next chapter.

3.1 Thermodynamic derivations

We begin with derivations of the relationships between the heat capacity per particle at constant volume $c_V \equiv \left(\frac{\partial Q}{\partial T}\right)_V$, the heat capacity per particle at constant pressure $c_P \equiv \left(\frac{\partial Q}{\partial T}\right)_P$, and the expansivity $\beta \equiv \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_P$, where the volume per particle is $V = 1/n$.

Starting with $dQ = TdS$, where S is the entropy per particle, and the definition of heat capacity, it follows that

$$c_V = T\left(\frac{\partial S}{\partial T}\right)_V \tag{3.1}$$

and

$$c_P = T\left(\frac{\partial S}{\partial T}\right)_P. \tag{3.2}$$

We can use 3.1, 3.2, and

$$dS = \left(\frac{\partial S}{\partial P}\right)_T dP + \left(\frac{\partial S}{\partial T}\right)_P dT \tag{3.3}$$

to obtain

$$c_V = c_P + T\left(\frac{\partial S}{\partial P}\right)_T\left(\frac{\partial P}{\partial T}\right)_V. \tag{3.4}$$

We eliminate $\left(\frac{\partial S}{\partial P}\right)_T$ using the Maxwell relation $-\left(\frac{\partial S}{\partial P}\right)_T = \left(\frac{\partial V}{\partial T}\right)_P$ obtained from the Gibbs free energy $dG = -SdT + VdP$ and

$$\left(\frac{\partial^2 G}{\partial P\,\partial T}\right) = -\left(\frac{\partial S}{\partial P}\right)_T$$
$$\left(\frac{\partial^2 G}{\partial T\,\partial P}\right) = \left(\frac{\partial V}{\partial T}\right)_P.$$

Defining the thermal expansivity β,

$$\frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_P \equiv \beta, \tag{3.5}$$

we have from the Maxwell relation,

$$\left(\frac{\partial S}{\partial P}\right)_T = -\beta V. \tag{3.6}$$

Now we can write $\left(\frac{\partial S}{\partial P}\right)_T \left(\frac{\partial P}{\partial T}\right)_V$ in 3.4 in terms of β and the isothermal sound speed $c_T^2 \equiv \frac{1}{m}\left(\frac{\partial P}{\partial n}\right)_T$. Taking $dP = 0$, we start with

$$dP = \left(\frac{\partial P}{\partial V}\right)_T dV + \left(\frac{\partial P}{\partial T}\right)_V dT$$
$$\left(\frac{\partial P}{\partial T}\right)_V = -\left(\frac{\partial P}{\partial V}\right)_T \left(\frac{\partial V}{\partial T}\right)_P. \tag{3.7}$$

Using the volume per particle $V = 1/n$, we have

$$\frac{\partial n}{\partial V} = -\frac{1}{V^2}. \tag{3.8}$$

We rewrite $\left(\frac{\partial P}{\partial V}\right)_T$ to get

$$\left(\frac{\partial P}{\partial V}\right)_T = \left(\frac{\partial P}{\partial n}\right)_T \frac{\partial n}{\partial V} = -\frac{1}{V^2}\left(\frac{\partial P}{\partial n}\right)_T. \tag{3.9}$$

Using this in 3.7 we have

$$\left(\frac{\partial P}{\partial T}\right)_V = -\left(\frac{\partial P}{\partial V}\right)_T \left(\frac{\partial V}{\partial T}\right)_P = \frac{1}{V^2}\left(\frac{\partial V}{\partial T}\right)_P \left(\frac{\partial P}{\partial n}\right)_T. \tag{3.10}$$

Substituting in 3.9 and 3.6 into 3.4 we obtain

$$c_V = c_P - \beta^2 \, T \left(\frac{\partial P}{\partial n} \right)_T.$$

(3.11)

3.1.1 Sound speed

One quantity that is easily measured is the speed of sound. The adiabatic sound speed c_0 can be determined from the fit of our hydrodynamic model to the time dependent density profiles in our experiment.

The isothermal sound speed c_T and adiabatic sound speed c_0 are given by

$$c_T^2 \equiv \frac{1}{m} \left(\frac{\partial P}{\partial n} \right)_T$$

(3.12)

$$c_0^2 \equiv \frac{1}{m} \left(\frac{\partial P}{\partial n} \right)_S.$$

(3.13)

We can use the sound speed to get the thermal expansivity β in terms of measurable quantities. We begin with $P(n,T)$, writing $P(n,S) = P[n, T(n,S)]$ so that

$$\left(\frac{\partial P}{\partial n} \right)_S = \left(\frac{\partial P}{\partial n} \right)_T + \left(\frac{\partial P}{\partial T} \right)_n \left(\frac{\partial T}{\partial n} \right)_S.$$

(3.14)

Taking $dS(P,T) = 0$, we can find $\left(\frac{\partial T}{\partial n} \right)_S$ from

$$0 = \left(\frac{\partial S}{\partial P} \right)_T \left(\frac{\partial P}{\partial n} \right)_S + \left(\frac{\partial S}{\partial T} \right)_P \left(\frac{\partial T}{\partial n} \right)_S$$

(3.15)

Using 3.6 and 3.1 in the equation above, it follows that

$$\left(\frac{\partial T}{\partial n} \right)_S = \frac{V \, \beta \, T}{c_P} \left(\frac{\partial P}{\partial n} \right)_S$$

(3.16)

30

Using 3.10 and 3.5 while noting $n = \frac{1}{V}$ so that $\left(\frac{\partial P}{\partial T}\right)_V = \left(\frac{\partial P}{\partial T}\right)_n$, we have

$$\left(\frac{\partial P}{\partial T}\right)_n = \beta\, n \left(\frac{\partial P}{\partial n}\right)_T. \tag{3.17}$$

Using 3.16 and 3.17 in 3.14 gives us

$$\left(\frac{\partial P}{\partial n}\right)_S = \left(\frac{\partial P}{\partial n}\right)_T + \frac{\beta^2\, T}{c_P}\left(\frac{\partial P}{\partial n}\right)_T \left(\frac{\partial P}{\partial n}\right)_S. \tag{3.18}$$

Using 3.11 in the equation above yields

$$\left(\frac{\partial P}{\partial n}\right)_S = \left(\frac{\partial P}{\partial n}\right)_T + \frac{c_P - c_V}{c_P}\left(\frac{\partial P}{\partial n}\right)_S. \tag{3.19}$$

Using 3.19 along with $\left(\frac{\partial P}{\partial n}\right)_T = m c_T^2$ from 3.12 and $\left(\frac{\partial P}{\partial n}\right)_S = m c_0^2$ from 3.13 gives us the relationship between the isothermal sound speed and the adiabatic sound speed

$$c_T^2 = \frac{c_V}{c_P}\, c_0^2. \tag{3.20}$$

Substituting $\left(\frac{\partial P}{\partial n}\right)_T = m c_T^2$ into 3.11 and using 3.20 gives the thermal expansivity in terms of measurable quantities

$$\beta^2 = \frac{c_P}{c_V}\frac{c_P - c_V}{m\, c_0^2\, T}. \tag{3.21}$$

3.1.2 Change in pressure and temperature

Sweeping a repulsive periodic optical potential into a uniform gas changes the pressure $P(n, S) = P[n, T(n, S)]$ and the temperature $T(S, n)$. We can use the results from the previous section to write down the change in pressure δP and temperature δT. The

31

change in pressure is

$$
\begin{aligned}
\delta P &= \left(\frac{\partial P}{\partial n}\right)_S \delta n + \left(\frac{\partial P}{\partial S}\right)_n \delta S \\
&= m\, c_0^2 \, \delta n + \left(\frac{\partial P}{\partial T}\right)_n \left(\frac{\partial T}{\partial S}\right)_n \delta S,
\end{aligned}
\tag{3.22}
$$

where 3.13 has been used. Using the heat capacity per particle at constant volume $c_V = T(\partial S/\partial T)_n$, 3.12, 3.17, and 3.20 we obtain

$$
\delta P = m\, c_0^2 \, \delta n + m\, c_0^2 \, \beta \, \frac{T}{c_P} \, n \, \delta S.
\tag{3.23}
$$

The change in temperature $T(n,s)$ is

$$
\delta T = \left(\frac{\partial T}{\partial S}\right)_n \delta S + \left(\frac{\partial T}{\partial n}\right)_S \delta n.
\tag{3.24}
$$

Using 3.16 as $\left(\frac{\partial T}{\partial n}\right)_S = \frac{1}{n}\frac{\beta T}{c_P}\left(\frac{\partial P}{\partial n}\right)_S$ and 3.1, we write 3.24 as

$$
\delta T = \frac{T}{c_V}\, \delta S + m\, c_0^2 \, \beta \frac{T}{c_P}\frac{\delta n}{n}.
\tag{3.25}
$$

3.1.3   Unitary Fermi gas thermodynamics

Thermodynamic quantities for a unitary Fermi gas can be determined as universal functions of density n and the reduced temperature $\theta = T/T_F$ using the equation of state. In 2012 Ku et al. [25] measured the universal function $f_E(\theta)$ that completely determines the equation of state for a unitary Fermi gas. In our experiments, the density is determined from absorption images and the reduced temperature is determined by measuring the adiabatic sound speed $c_0(\theta)$. The measured equation of state, density and temperature

then allows us to calculate all thermodynamic quantities needed for our hydrodynamic model.

The energy density ε, chemical potential μ, and entropy per particle S in a unitary Fermi gas are given by

$$\varepsilon = \frac{3}{5} n \, \epsilon_F(n) \, f_E(\theta) \tag{3.26}$$

$$\mu = \epsilon_F(n) \, f_\mu(\theta) \tag{3.27}$$

$$S = k_B \, f_S(\theta), \tag{3.28}$$

where the local Fermi energy $\epsilon_F(n)$ and Fermi temperature T_F are

$$k_B \, T_F = \epsilon_F(n) = \frac{\hbar^2}{2m} \left(3\pi^2 n\right)^{2/3}, \tag{3.29}$$

with n being the total density for a 50-50 mixture of two spin states.

If the density and temperature are known, any thermodynamic quantity such as the heat capacity c_V can be determined using the measured equation of state,

$$c_V = T \left(\frac{\partial S}{\partial T}\right)_V = T \frac{\partial f_S(\theta)}{\partial T} k_B, \tag{3.30}$$

or equivalently using $dE = T \, dS - P \, dV$

$$c_V = \frac{1}{n} \left(\frac{\partial \varepsilon}{\partial T}\right)_n = k_B \frac{3}{5} f_E'(\theta), \tag{3.31}$$

where S is the entropy per particle and the energy per particle is $E = \varepsilon/n$. The heat capacity c_V calculated from the equation of state is shown in Fig 3.1.

In a harmonically trapped unitary Fermi gas, the shape of the density profile with the
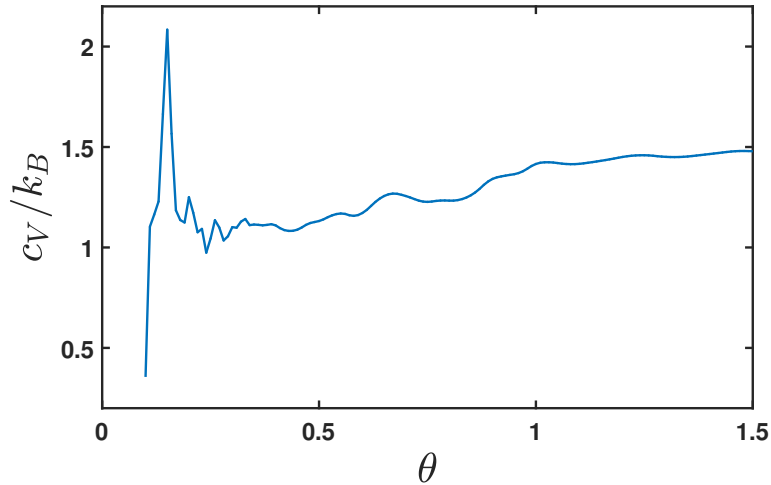
Figure 3.1: The heat capacity per particle at constant volume C_V and heat capacity at constant pressure C_P as a function of the reduced temperature θ are determined by the measured universal function $f_E(\theta)$.

equation of state gives the temperature. Parametric resonance characterizes the harmonic potential and the virial theorem gives the total energy of the cloud. In a uniform box trap, this technique is more difficult to implement. We instead measure the temperature from the adiabatic speed of sound $c_0(\theta_0)$ extracted from density profiles in-situ. We compare the in-situ measurement of c_0 with the speed of a density perturbation in a uniform sample of equivalent density for a cross check.

The adiabatic sound speed is given by

$$c_0^2 \equiv \frac{1}{m}\left(\frac{\partial P}{\partial n}\right)_S = \frac{1}{m}\left(\frac{\partial P}{\partial n}\right)_\theta, \tag{3.32}$$

where we can see from 3.28 that constant S is equivalent to constant θ. In a unitary Fermi gas the pressure P and energy density ε are related by $P = (2/3)\,\varepsilon$ [14, 29].

Using 3.32, 3.26, and $P = (2/5)\, n\, \epsilon_F(n)\, f_E(\theta)$ we obtain

$$c_0 = v_F\,\sqrt{\frac{f_E(\theta)}{3}}, \tag{3.33}$$

where $\epsilon_F(n_0) = \frac{1}{2}\, m\, v_F^2$ is given by 3.29 for $n = n_0$.

The measured sound speed c_0 combined with the measured universal function $f_E(\theta)$ enables thermometry in our experiment. The temperature as a function of the sound speed, shown in Fig 3.2, is a monotonic function of c_0/v_F for $\theta > 0.25$.
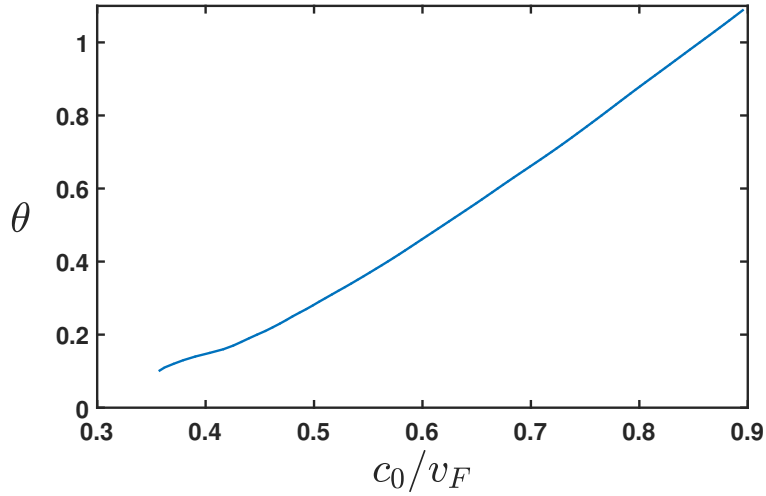
Figure 3.2: In our experiments we find $c_0 = 1.3$ cm/s for our density $n_0 = 2.6 \times 10^{11}$ and Fermi velocity $v_F = 2.11$ cm/s. The ratio $c_0/v_F = 0.62$ gives a reduced temperature $\theta = 0.50$ which we use to determine the heat capacity per particle at constant volume c_v and heat capacity per particle at constant volume c_P, as well as the shear viscosity η discussed later in chapter 4.

Chapter 4

Transport in a normal unitary Fermi gas

A unitary Fermi gas above the superfluid critical temperature T_C is a normal fluid with only two transport parameters, the shear viscosity η and the thermal conductivity κ_T. In normal fluid regime, a unitary Fermi gas has vanishing bulk viscosity ζ as a consequence of scale invariance [15, 28]. Measurements of the transport properties can be compared with variational calculations for a unitary gas in the 2-body Boltzmann limit. In the high temperature limit, the predicted ratio [5,6] of the thermal conductivity to the shear viscosity is

$$\frac{\kappa_T}{\eta} = \frac{15}{4} \frac{k_B}{m},$$ (4.1)

with the shear viscosity

$$\eta = \frac{15}{32\sqrt{\pi}} \frac{(m\,k_B\,T)^{3/2}}{\hbar^2}.$$ (4.2)

These relations are valid for a unitary Fermi gas in the Bolztmann limit, where k_B is the Boltzmann constant, T is temperature, and m is the atom mass.

4.0.1 Shear Viscosity η

The internal friction between parts of a fluid traveling with different velocities causes a transfer of momentum from regions of high velocity to low velocity. The equations of motion that describe the unitary Fermi gas as a viscous fluid include the force from the viscous stress tensor [23]

$$\partial_j \left(\eta \, \sigma_{ij} \right) = \partial_j \left[\eta \left(\frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i} - \frac{2}{3} \delta_{ik} \frac{\partial v_l}{\partial x_l} \right) \right]. \tag{4.3}$$

The shear viscosity η in a unitary Fermi gas was previously measured in the normal fluid regime [4, 19]. The shear viscosity is extracted by observing the hydrodynamic expansion of a uniform temperature cloud released from an anisotropic trap at a fixed time t. The unequal pressure gradients of an anisotropic trap lead to internal friction from the transverse flow during expansion after release. The expanding cloud aspect ratio of the recorded at a fixed time t is determined by the shear viscosity. An analysis of this data precisely determines the shear viscosity $\eta(\theta)$ in the normal fluid regime by modeling the hydrodynamic flow near the cloud center and the non-hydrodynamic modes of the less dense edges [4]. The effect of the thermal conductivity in the expanding cloud experiment can be neglected as the clouds maintain uniform temperature due to the uniform scale transformation that governs the expansion.

Our analysis of the hydrodynamic density response δn in a unitary Fermi gas to determine the thermal conductivity benefits from the above recent work in determining the shear viscosity from expanding clouds released from anisotropic traps [19]. In the analysis, the shear viscosity η in the normal fluid regime as a function of temperature

and density it taken to be an expansion in powers of the diluteness $n\lambda_T^3$

$$\eta(n, T) = \eta_0 \frac{(m\, k_B\, T)^{3/2}}{\hbar^2} \left[1 + \eta_2 \left(n\, \lambda_T^3\right) + ...\right], \tag{4.4}$$

where $\lambda_T = h/\sqrt{2\,\pi\, m\, k_B\, T}$ is the thermal de Broglie wavelength and n is the total density. The leading term $\propto \eta_0$ is a function of temperature only. η_0 has been calculated using the two-body Boltzmann equation for a unitary Fermi gas, $\eta_0 = 15/(32\sqrt{\pi}) = 0.26446$ [4]. This is in excellent agreement with the fits to the expansion data, $\eta_0 = 0.265(20)$, showing that the analysis properly reproduces the high temperature limit. The first order of the
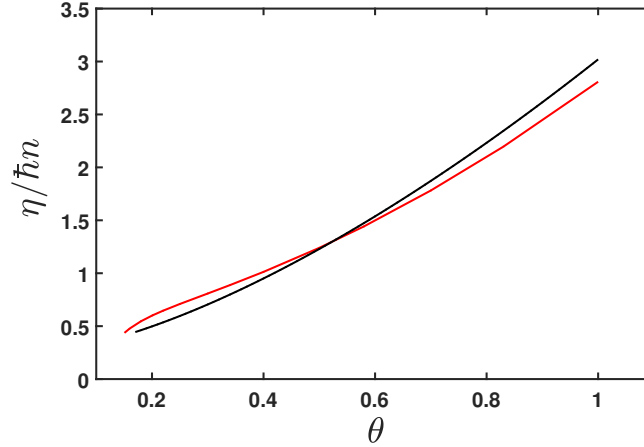
Figure 4.1: The dimensionless shear viscosity $\eta/\hbar n$ from equation 4.5 shown in black with $\hbar$ and m set to unity [4]. Shear viscosity calculated by Enss et al. [11] shown in red for comparison.

expansion is a function of the density only with $\eta_2 = 0.060(20)$. Using only the first two terms in the expansion, the dimensionless shear viscosity takes the form

$$\alpha \equiv \frac{\eta}{\hbar n} = \frac{3\pi^2}{\sqrt{8}}\, \eta_0\, \theta^{3/2} + (2\pi)^{3/2}\, \eta_0\, \eta_2 = 2.77(21)\, \theta^{3/2} + 0.25(8). \tag{4.5}$$

Equation 4.5 is shown in Fig. 4.1, is comparable to calculations from Enss et al. [11] and Wlazowski et al. [31]. We use the shear viscosity determined from the expanding cloud data [4, 19] as an input to our hydrodynamic model, leaving the thermal conductivity κ_T as the only transport property to be determined.

4.1 Hydrodynamic linear response for a normal fluid

We measure the density response δn of a uniform Fermi gas to a perturbing repulsive optical potential $\delta U(z,t)$. As noted above, the response contains information about the transport coefficients η and κ_T [32]. To extract the thermal conductivity κ_T in a unitary Fermi gas, we have created a linear hydrodynamic response model for the δn response of a uniform gas confined in a box potential $U_0(z)$ perturbed by the periodic potential

$$\delta U(z,t) = \delta U_0 \left[1 - \epsilon \, \cos(qz - qvt) \right] \, \phi(vt - z) \, e^{-(z-z_0)^2/\sigma^2}. \tag{4.6}$$

A smoothed step function $\phi(vt - z)$ gives a vanishing perturbation inside the box at $t = 0$ and provides a leading edge for the periodic perturbation moving into the uniform density. The last term in 4.6 accounts for the gaussian profile of the beam illuminating the DMD.

We begin the construction of our hydrodynamic model by requiring that the density $n(z,t)$ of the fluid satisfies the continuity equation

$$\partial_t \, n + \nabla(n\mathbf{v}) = 0. \tag{4.7}$$

The equations of motion for a viscous fluid are obtained writing the acceleration of a particle moving in a fluid

$$\frac{d\mathbf{v}}{dt} = \frac{\partial \mathbf{v}}{\partial t} + \frac{\partial \mathbf{v}}{\partial x}\frac{dx}{dt} + \frac{\partial \mathbf{v}}{\partial y}\frac{dy}{dt} + \frac{\partial \mathbf{v}}{\partial z}\frac{dz}{dt}. \tag{4.8}$$

and equating the force per unit volume

$$n\,m\,\frac{d\mathbf{v}}{dt} = n\,m\,\left(\partial_t\,\mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}\right) \tag{4.9}$$

to the forces per unit volume arising from the pressure $-\nabla P$, applied potential $-n\nabla U$, and the viscous stress tensor $\partial_i\left(\eta\,\sigma_{ij}\right)$.

We write our equations of motion as Euler's equation in one dimension for a viscous fluid

$$n\,m\,\left(\partial_t + v_z\partial_z\right)v_z = -\partial_z P - n\partial_z U + \partial_z(\eta\,\sigma_{zz}), \tag{4.10}$$

and the continuity equation

$$\partial_t\,n + \partial_z(n\,v_z) = 0. \tag{4.11}$$

The viscous stress tensor $\eta\sigma_{ij}$ in one dimension is

$$\eta\,\sigma_{zz} = \eta\left(\partial_z\,v_z + \partial_z\,v_z - \frac{2}{3}\partial_z\,v_z\right) = \eta\,\frac{4}{3}\partial_z\,v_z. \tag{4.12}$$

A perturbative external potential $\delta U(z, t)$ produces first order change in the density, pressure, and velocity

$$n = n_0(z) + \delta n(z, t)$$

$$P = P_0 + \delta P$$

$$U = U_0(z) + \delta U(z, t)$$

$$v_z = 0 + v_z.$$

Substituting this into 4.10 and 4.11 gives us

$$(n_0 + \delta n) \, m \, (\partial_t + v_z \, \partial_z) v_z = -\partial_z (P_0 + \delta P) - (n_0 + \delta n) \, \partial_z \, (U_0 + \delta U) + \partial_z \, \eta \, \frac{4}{3} \partial_z \, v_z, \quad (4.13)$$

and

$$\partial_t \, (n_0 + \delta n) + \partial_z \, [(n_0 + \delta n) v_z] = 0. \tag{4.14}$$

The force from the pressure and box potential cancel each other in equilibrium, as shown in Fig. 4.2, which requires

$$-\partial_z \, P_0 - n_0 \, \partial_z \, U_0 = 0. \tag{4.15}$$

Using 4.15 and keeping only terms up to first order we write 4.13 and 4.14 as

$$-n_0 \, \partial_t \, v_z = \frac{1}{m} \left(\partial_z \, \delta P + n_0 \, \partial_z \, \delta U + \delta n \, \partial_z \, U_0 - \partial_z \, \eta \, \frac{4}{3} \partial_z \, v_z \right) \tag{4.16}$$

and

$$\partial_t \, \delta n = -\partial_z \, n_0 \, v_z \approx -n_0 \, \partial_z \, v_z. \tag{4.17}$$

41
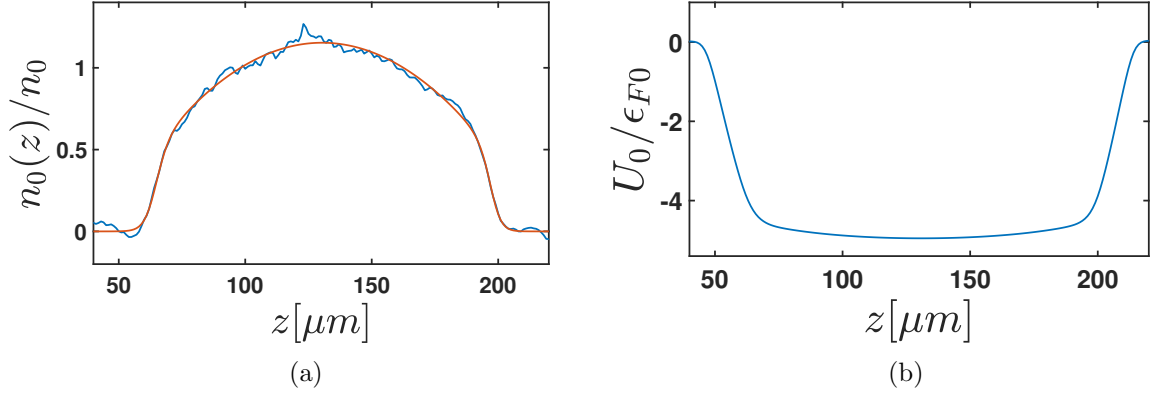
Figure 4.2: (a) Background density $\tilde{n}_0(z)$ (blue) fit with an analytic function (red). (b) The box potential $U_0(z)/\varepsilon_{F0}$ is obtained by integrating $\partial_z U_0(z)$. The gradient of the box potential $\partial_z U_0(z)$ is determined from the background density $n_0(z)$.

Taking the derivative ∂_z of equation 4.16 and ∂_t of equation 4.17 we obtain

$$-\partial_z \, \partial_t \, n_0 \, v_z = \frac{1}{m} \, \partial_z \left(\partial_z \, \delta P + n_0 \, \partial_z \, \delta U + \delta n \, \partial_z \, U_0 - \partial_z \, \eta \, \frac{4}{3} \, \partial_z \, v_z \right) \tag{4.18}$$

and

$$\partial_t^2 \, \delta n = -\partial_t \, \partial_z \, n_0 \, v_z. \tag{4.19}$$

Substituting equation 4.19 into equation 4.18, we have

$$\partial_t^2 \, \delta n = \frac{1}{m} \, \partial_z \left(\partial_z \, \delta P + n_0 \, \partial_z \, \delta U + \delta n \, \partial_z \, U_0 - \partial_z \, \eta \, \frac{4}{3} \, \partial_z \, v_z \right). \tag{4.20}$$

We use

$$\delta P = m \, c_0^2 \, \delta n + m \, c_0^2 \, \beta \, \frac{T}{c_P} \, n \, \delta S \tag{4.21}$$

from chapter 3, where S is the entropy per particle, and $\partial_z \, v_z = \frac{-1}{n_0} \, \partial_t \, \delta n$ from 4.17 to

eliminate the velocity field to obtain

$$\partial_t^2 \, \delta n - c_0^2 \, \partial_z^2 \, \delta n - c_0^2 \, \beta \, n_0 \, \frac{T}{c_P} \, \partial_z^2 \, \delta S - \frac{4}{3} \, \frac{\eta}{n_0 \, m} \partial_z^2 \, \partial_t \, \delta n = \frac{1}{m} \, \partial_z \, (n_0 \, \partial_z \, \delta U + \delta n \, \partial_z \, U_0) \,. \quad (4.22)$$

We treat the sound speed c_0 as a constant with respect to position in equation 4.22 because c_0/v_F should be independent of density in the unitary Fermi gas [17]. In Fig. 4.3 we show c_0/v_F as a function of density for a fixed temperature θ_0.
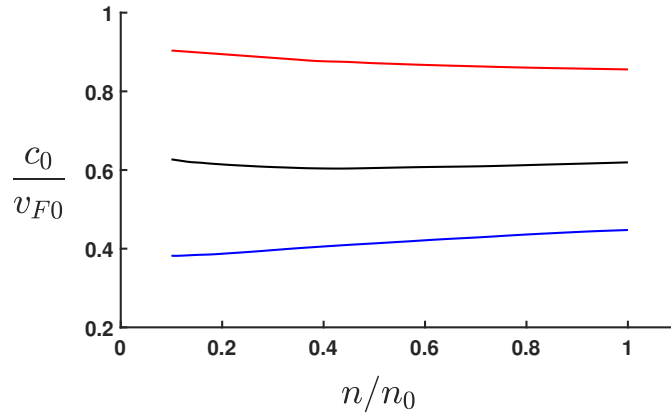
Figure 4.3: The ratio of the sound speed to the Fermi velocity c_0/v_F as a function of density n/n_0. We use $c_0/v_F = \sqrt{f(\theta)/3}$ from chapter three, where $\theta(n/n_0) = \theta_0(n/n_0)^{-2/3}$. We show the nearly constant c_0/v_F for $\theta_0 = 1$ in red, $\theta_0 = 0.5$ in black, and $\theta_0 = 0.2$ in blue.

The evolution of δS is determined by the heating rate per unit volume,

$$\frac{\delta \dot{q}}{n_0} = T_0 \, (\partial_t + v_z \, \partial_z) \, \delta S. \tag{4.23}$$

The heating rate from conduction $\dot{q}$ is obtained by substituting the heat current

$$J_Q = -\kappa_T \, \nabla T \tag{4.24}$$

into the thermal conduction equation

$$\partial_t \, \varepsilon + \nabla \cdot J_Q = 0. \tag{4.25}$$

We neglect higher order heating terms in $\partial_t \varepsilon$ arising from shear viscosity $O(v^2)$ to obtain

$$\delta \dot{q} \approx \kappa_T \, \partial_z^2 \, \delta T. \tag{4.26}$$

Substituting the temperature change

$$\delta T = \frac{T_0}{c_V} \, \delta S + m \, c_0^2 \, \beta \, \frac{T_0}{c_P} \, \frac{\delta n}{n}, \tag{4.27}$$

from chapter 3 with $n = n_0$ and $T = T_0$, and equation 4.26 into 4.23 we obtain

$$\partial_t \, \delta S - \frac{\kappa_T}{n_0 c_V} \, \partial_z^2 \, \delta S = \frac{\kappa_T \, m \, c_0^2}{n_0^2 \, c_P} \, \beta \, \partial_z^2 \, \delta n, \tag{4.28}$$

where the $v_z \, \partial_z \, \delta S$ term from 4.23 is second order.

It is convenient to use dimensionless variables

$$\delta \tilde{S} \equiv \beta \, T_0 \, n_0 \, \delta S \, / \, (c_P \, n_0)$$

$$\delta \tilde{n} \equiv \delta n / n_0$$

$$\tilde{n}_0(z) \equiv n_0(z)/n_0$$

$$\delta \tilde{U}(z,t) \equiv \delta U(z,t)/\epsilon_{F0}$$

$$\tilde{U}_0(z,t) \equiv U_0(z,t)/\epsilon_{F0}$$

in 4.22 and 4.28 to make numerical integration easier, where n_0 is the average density and the Fermi energy $\epsilon_{F0} = (1/2) \, m \, v_{F0}^2 = \hbar^2 \, (3 \, \pi \, n_0)^{2/3} \, / \, 2 \, m$. Working with dimensionless variables, we have two coupled equations which describe the response of the density $\delta \tilde{n}$ and entropy $\delta \tilde{S}$ to a perturbing potential $\delta \tilde{U}(z,t)$ in a box trap $\tilde{U}_0(z,t)$,

$$\boxed{\partial_t \, \delta \tilde{S} - \frac{\kappa_T}{n_0 \, c_V} \partial_z^2 \, \delta \tilde{S} = \frac{\kappa_T}{n_0 \, c_V} \frac{c_P - c_V}{c_P} \partial_z^2 \, \delta \tilde{n},} \tag{4.29}$$

and

$$\boxed{\partial_t^2 \, \delta \tilde{n} - c_0^2 \, (\partial_z^2 \, \delta \tilde{n} + \partial_z^2 \, \delta \tilde{S}) - \frac{4}{3} \frac{\eta}{n_0 \, m} \partial_z^2 \, \partial_t \, \delta \tilde{n} = \frac{v_{F0}^2}{2} \partial_z \left(\tilde{n}_0 \, \partial_z \, \delta \tilde{U} + \delta \tilde{n} \, \partial_z \, \tilde{U}_0 \right).} \tag{4.30}$$

The c_0^2 terms in equation 4.30 are from the change in pressure δP, the η term accounts for the viscous damping of the density response δn which is driven by the $\delta \tilde{U}$ term. The $\tilde{U}_0$ term is the force from the box potential, derived later in this chapter.

Equations 4.29 and 4.30 are solved numerically with the initial conditions

$$\delta \tilde{n} = 0,$$

$$\delta \dot{\tilde{n}} = 0,$$

$$\delta \tilde{S} = 0.$$

4.2   Damping and decay rates

We can understand the damping in equation 4.29 and 4.30 as the irreversible processes of thermal conduction and internal friction. Density perturbations created by moving $\delta U(z,t)$ through the uniform sample are damped at a rate set by the shear viscosity η and the thermal conductivity κ_T. The decay rate of temperature gradients is obtained by considering the energy flux resulting from thermal conduction in a static fluid

$$\partial_t \, \varepsilon + \nabla \cdot J_Q = 0. \tag{4.31}$$

The heat current

$$J_Q = -\kappa_T \, \nabla T \tag{4.32}$$

results from the heat transfer from points of high temperature regions to low temperature. The magnitude of the heat current is determined by the thermal conductivity κ_T.

We can write equation 4.31 as

$$\left(\frac{\partial \varepsilon}{\partial T} \right)_n \left(\frac{\partial T}{\partial t} \right) + \nabla \cdot J_Q = 0. \tag{4.33}$$

Substituting

$$\left(\frac{\partial \varepsilon}{\partial T}\right)_n = n\, c_V \tag{4.34}$$

into 4.33, we obtain

$$n\, c_V \left(\frac{\partial T}{\partial t}\right) + \nabla \cdot J_Q = 0. \tag{4.35}$$

Substituting 4.32 into 4.35, we have

$$\left(\frac{\partial T}{\partial t}\right) - \frac{\kappa_T}{nc_V} \nabla^2 T = 0. \tag{4.36}$$

Assuming we have a solution of the form

$$T(z,t) = T_0 + A(t)\,\cos(qz), \tag{4.37}$$

we can solve

$$\left[\dot{A}(t) + \frac{\kappa_T q^2}{nc_V} A(t)\right] \cos(qz) = 0 \tag{4.38}$$

to obtain

$$A(t) = A_0\, e^{-\gamma_\kappa t}. \tag{4.39}$$

This gives us the rate at which temperature gradients decay,

$$\gamma_\kappa = \frac{\kappa_T}{n\, c_V}\, q^2. \tag{4.40}$$

The decay rate γ_κ can be seen in the equation that describes the change in entropy 4.29 as a result of a change in density $\delta\tilde{n}$, assuming a sinusoidal solution for the change in entropy

$$\delta\tilde{S}(z,t) = \delta\tilde{S}_0 + A(t)\,\cos(qz). \tag{4.41}$$

The frictional or viscous forces resulting from sheer stress $\partial_z v_z$ have a damping rate

$$\gamma_\eta = \frac{4}{3} \frac{\eta}{n_0 \, m} q^2. \tag{4.42}$$

that can be obtained from equation 4.30 by assuming a solution for the change in density

$$\delta\tilde{n}(z,t) = \delta\tilde{n}_0 + A(t) \, \cos(qz). \tag{4.43}$$

Damping form shear viscosity and thermal conductivity both contribute to sound absorption. In the high frequency limit, $c_0 \, q >> \gamma_\kappa$, the damping rate [23] is

$$\Gamma = q^2 \left[\frac{4}{3} \frac{\eta}{n_0 m} + \frac{\kappa_T}{n_0 c_V} \left(1 - \frac{c_V}{c_P} \right) \right]. \tag{4.44}$$

4.2.1 Force from the box potential

An atomic sample trapped in the optical box $U_0(z)$ trap does not provide an infinite medium. We need to account for the boundary conditions to properly model the density response $\delta n(z,t)$. The force from the box potential $F = -\partial_z U_0(z)$, shown in Fig. 4.4, is obtained from the condition for equilibrium $\partial_z P_0 = n_0 \partial_z U_0(z)$, where the gradient of the pressure is

$$\partial_z \, P_0 \left[n_0(z), T_0 \right] = \left(\frac{\partial P_0}{\partial n_0} \right)_T \partial_z \, n_0(z). \tag{4.45}$$

We can write the isothermal sound speed from chapter 3 as

$$m \, c_T^2 = \left(\frac{\partial P_0}{\partial n_0} \right)_T = m \, \frac{c_V}{c_P} \, c_0^2, \tag{4.46}$$
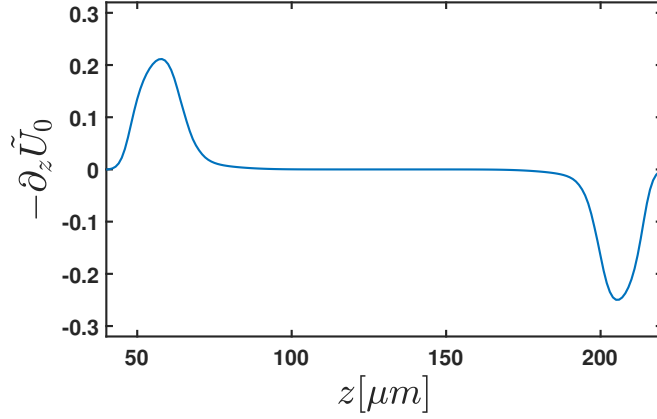
Figure 4.4: Box force profile $-\partial_z \tilde{U}_0$ in dimensionless variables.

and the adiabatic sound speed as

$$c_0^2 = \frac{2\,\epsilon_F}{m} \frac{f_E(\theta)}{3}, \tag{4.47}$$

where we have used $v_F^2 = 2\,\epsilon_F\,/\,m$. Substituting 4.47 and 4.46 into equation 4.45, we obtain

$$\partial_z\, U_0 = -\frac{2}{3}\, \epsilon_F\, f_E(\theta)\, \frac{c_V}{c_P}\, \frac{\partial_z\, n_0(z)}{n_0(z)}. \tag{4.48}$$

Using dimensionless variables, $\tilde{n}_0(z) \equiv n_0(z)/n_0$ and $\tilde{U}_0(z,t) \equiv U_0(z,t)/\epsilon_{F0}$, we have

$$\partial_z\, \tilde{U}_0(z) = -\frac{2}{3}\, f_E(\theta)\, \frac{c_V[\theta(z)]}{c_P[\theta(z)]}\, \frac{\partial_z\, \tilde{n}_0(z)}{[\tilde{n}_0(z)]^{1/3}}. \tag{4.49}$$

The thermodynamic quantities and measured equation of state,

$$\varepsilon = \frac{3}{5}\, n\, \epsilon_F(n)\, f_E(\theta), \tag{4.50}$$

from chapter 3, are universal functions only of the reduced temperature $\theta = T/T_F$. The

49

reduced temperature θ scales as $n^{-2/3}$ as seen in the equation for the local Fermi energy

$$k_B T_F = \epsilon_F(n) = \frac{\hbar^2}{2m} \left(3\,\pi^2\,n\right)^{2/3}.$$

(4.51)

In our sample,

$$\theta(z) = \theta_0 \left[\tilde{n}_0(z)\right]^{-2/3}$$

(4.52)

is calculated from the density profile $\tilde{n}_0(z)$. The reduced temperature evaluated at the mean central density, $\theta_0 = T_0 / T_F(n_0)$, is determined from the fitted adiabatic sound speed c_0. We avoid the divergence of θ as $\tilde{n}_0 \to 0$ by adding an offset $\tilde{n}_0 \times 10^{-4}$ to the density $\tilde{n}_0(z)$ in equation 4.52.

Equation 4.45 can also be used to determine the force from the box potential $\partial_z U_0(z)$ by evaluating $(\partial P/\partial n)_T$ from the pressure for a unitary gas. From chapter three, we have the pressure for a unitary Fermi gas,

$$P = \frac{2}{5}\,n\,\epsilon_F(n)\,f_E(\theta) = \frac{2}{5}\,\frac{\hbar^2}{2m}\,(3\pi^2)^{2/3}\,n^{5/3}\,f_E(\theta).$$

(4.53)

This directly gives us the force from the box potential for the unitary Fermi gas

$$\partial_z \tilde{U}_0(z) = -\frac{2}{3}\left[f_E(\theta) - \frac{2}{5}\,\theta\,f'_E(\theta)\right]\frac{\partial_z \tilde{n}_0(z)}{[\tilde{n}_0(z)]^{1/3}},$$

(4.54)

from the density profile, using the known equation of state $f(\theta)$, and the reduced temperature $\theta(z)$ from equation 4.52.

Chapter 5

Comparison of Model to Data

In this chapter we compare the hydrodynamic model to the data from the experiment. We discuss the effect each fitting parameter has on the modeled density response. We then measure the thermal conductivity κ_T from the global fit of the model to the data.

Using a DMD gives us independent control over the wavelength λ and frequency f of the periodic optical potential $\delta U(z, t)$. This allows us to move the optical perturbation through the uniform sample at speeds above and below the relaxation rate of temperature gradients

$$\gamma_\kappa = \frac{\kappa_T}{n_0\, c_V}\, q^2, \tag{5.1}$$

where $q = 2\pi/\lambda$. When the density is driven at low frequencies relative to the decay rate, $\omega < \gamma_\kappa$, the density response δn is sensitive to the thermal conductivity κ_T. In this regime, temperature gradients have time to decay and the compression is closer to isothermal.

Independently choosing the wavelength and frequency of the periodic optical potential allows us to probe regimes where the data and model are sensitive to the thermal

51

conductivity κ_T and regimes where the thermal conductivity has a negligible effect on the density response δn.

5.1   Experimental data

We observe the density response using two different wavelengths for our periodic optical potential $\delta U(z, t)$, $\lambda = 19$ μm and $\lambda = 30$ μm. The decay rate γ_κ for each wavelength $\gamma_\kappa \propto q^2$, where $q = 2\pi/\lambda$. To estimate the decay rate γ_κ, we use the thermal conductivity calculated in the high temperature limit [5,6]

$$\kappa_T = \eta_0 \frac{15}{4} \frac{k_B}{m}. \tag{5.2}$$

We calculate the decay rate γ_κ using equation 5.1 for two different wavelengths using the measured shear viscosity $\eta_0 = 1.23 \, \hbar n_0$ [4, 19] and heat capacity $c_V = 1.13 \, k_B$ from the measured equation of state [25], assuming a temperature of $\theta_0 = 0.5$, which is comparable to what we measure in our experiments.

Table 5.1:   Damping rates.

Rate	$\lambda = 19$ μm	$\lambda = 30$ μm
$c_0 \, q$	$2\pi \times 685$ Hz	$2\pi \times 433$ Hz
γ_η	$2\pi \times 301$ Hz	$2\pi \times 121$ Hz
γ_κ	$2\pi \times 750$ Hz	$2\pi \times 300$ Hz

We choose a range of frequencies to measure the density response above and below the decay rate γ_κ. The frequency range of the $\lambda = 19$ μm optical potential is 100-1200

Hz in steps of 100 Hz. The frequency range of the $\lambda = 30~\mu$m optical potential is 100-750 Hz in steps of 50Hz.

The optical potential $\delta U(z,t)$ vanishes inside the box at $t = 0$, and moves along the z axis for a fixed number of periods. The leading edge of $\delta U(z,t)$ is roughly centered in the box trap when an absorption image is taken to obtain the column density. The $\lambda = 19~\mu$m perturbation moves four periods into the sample for each frequency f, resulting in a leading edge at 76 μm when the absorption image is taken, as shown in Fig. 5.1. The $\lambda = 30~\mu$m perturbation moves only three periods into the sample, resulting in a leading edge is at 90 μm when the absorption image is taken. Five absorption images are taken for each frequency f to obtain an average column density. The background density $n_0(z)$ is recorded for an unperturbed uniform sample $\delta U_0 = 0$. The average background density is $n_0 = N/V = 2.6 \times 10^{11}$. The total atom number for both spin states is $N = 1.69 \times 10^5$ and V is the volume of the box trap $U_0(z)$, with dimensions $129 \times 84 \times 58~\mu$m. The density response profiles,

$$\delta\tilde{n} \equiv \frac{\Delta n}{n_0} = \frac{n(z) - n_0(z)}{n_0}, \tag{5.3}$$

for the optical potential with wavelength $\lambda = 19~\mu$m and $\lambda = 30~\mu$m are sown in Fig. 5.2 and 5.3 respectively.

Figures 5.2 and 5.3 show how the shape of the density response δn changes as the perturbing potential is moved through at subsonic $v < c_0$ and supersonic $v > c_0$ speeds. When the drive speed v is low, $v/c_0 < 0.5$, sound waves propagate past the leading edge of the perturbing optical potential $\delta U(z,t)$, resulting in a positive and nearly flat density response δn in the region between the leading edge of the perturbation and the box wall at 129 μm. The leading edge of the density response increases in amplitude and narrows as v approaches c_0 as the sound waves cannot propagate far past the edge of $\delta U(z,t)$.
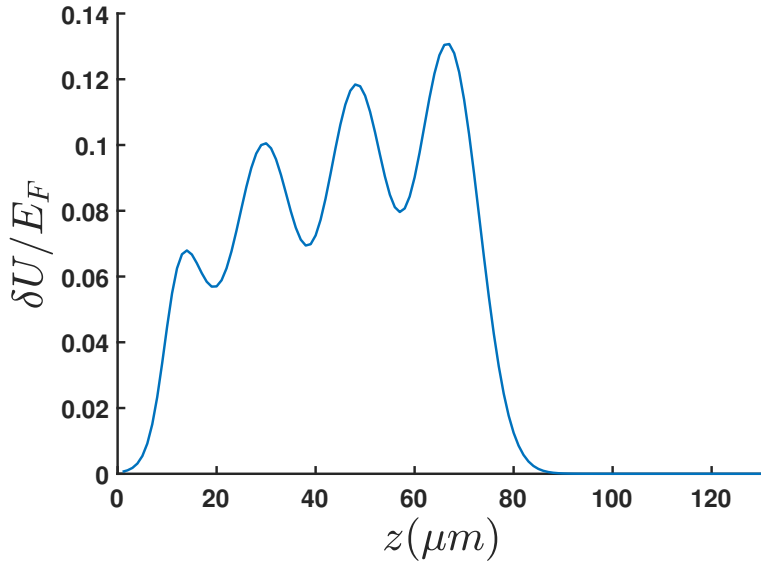
(a) δU at the final time for $\lambda = 19$ μm.
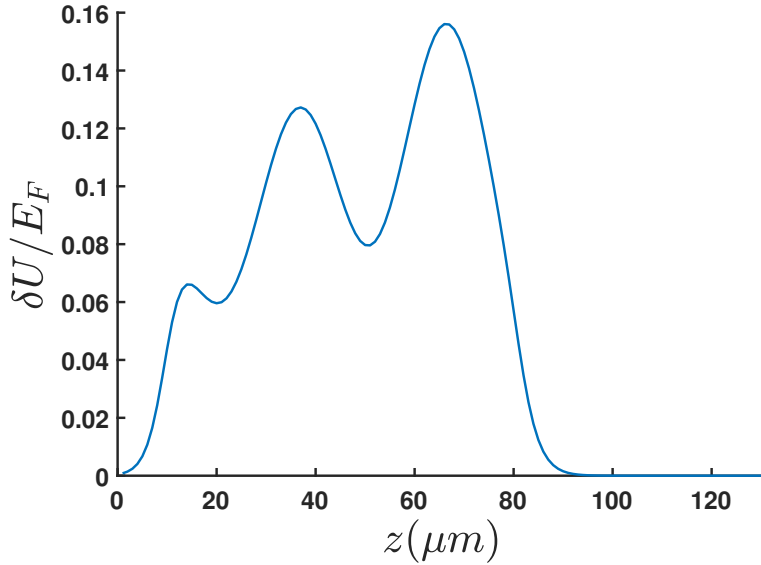
(b) δU at the final time for $\lambda = 30$ μm.

Figure 5.1: The periodic optical potential $\delta U(z,t)/E_F$ plotted at time t_f when the absorption image is taken. The form of $\delta U(z,t)$, as shown in equation 5.12, properly accounts for the blurring through the imaging system and the gaussian profile of the beam illuminating the DMD. (a) The 19 μm wavelength perturbation has moved 4 spatial periods into the uniform sample at time $t = t_f$. (b) The 30 μm wavelength perturbation has moved 3 spatial periods into the uniform sample at time $t = t_f$.
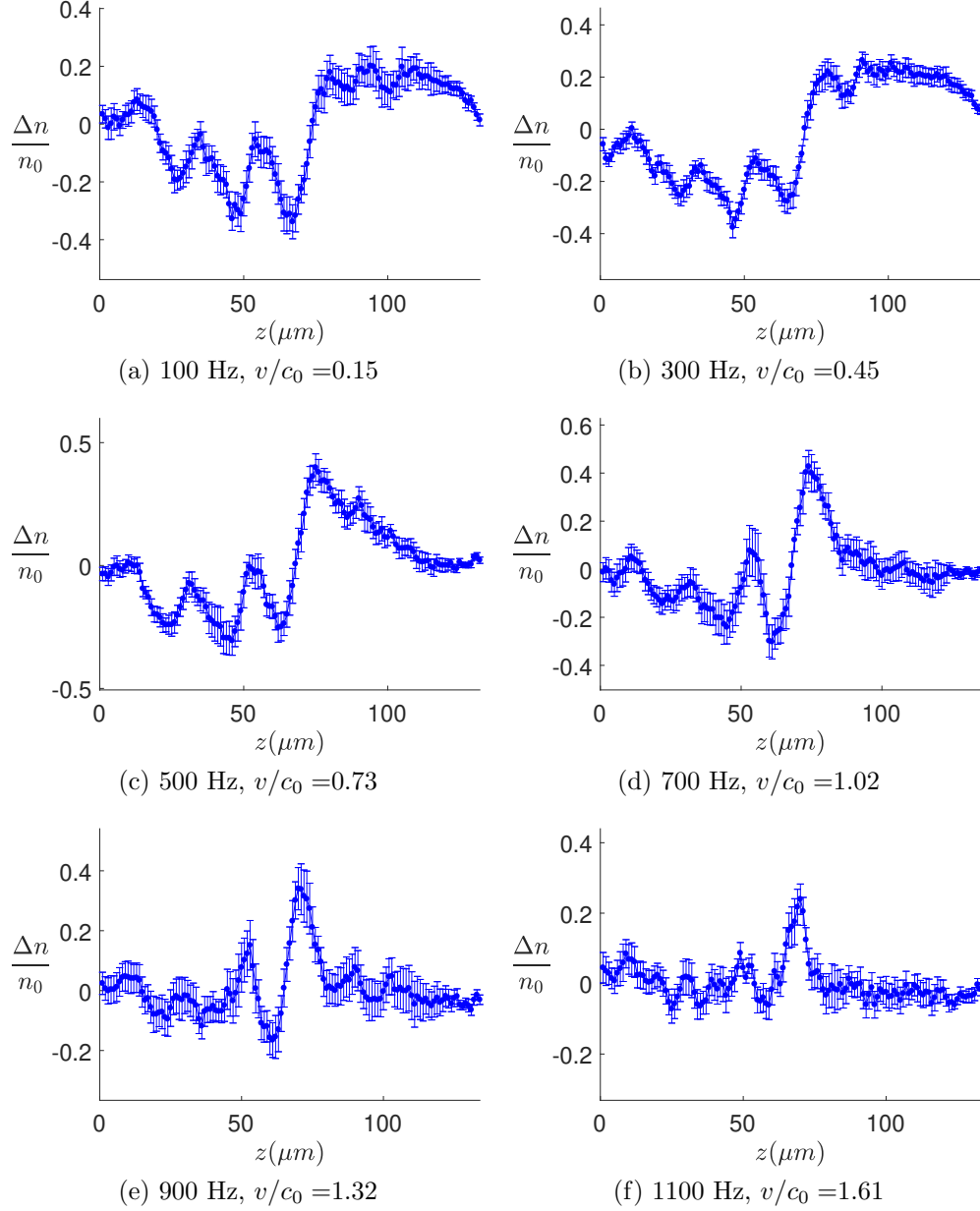
Figure 5.2: A periodic optical potential with wavelength $\lambda = 19$ μm, velocity $v = \lambda f$, and magnitude $\delta U_0 \approx 0.25 E_F$ is swept through a uniform sample which has an average density $n_0 = N/V = 2.6 \times 10^{11}$ atoms/cm^3, where N is the total atom number for both spin states. The response of the density δn is recorded as a function of the periodic potential frequency with a range f=100-1200 Hz in steps of 100 Hz.
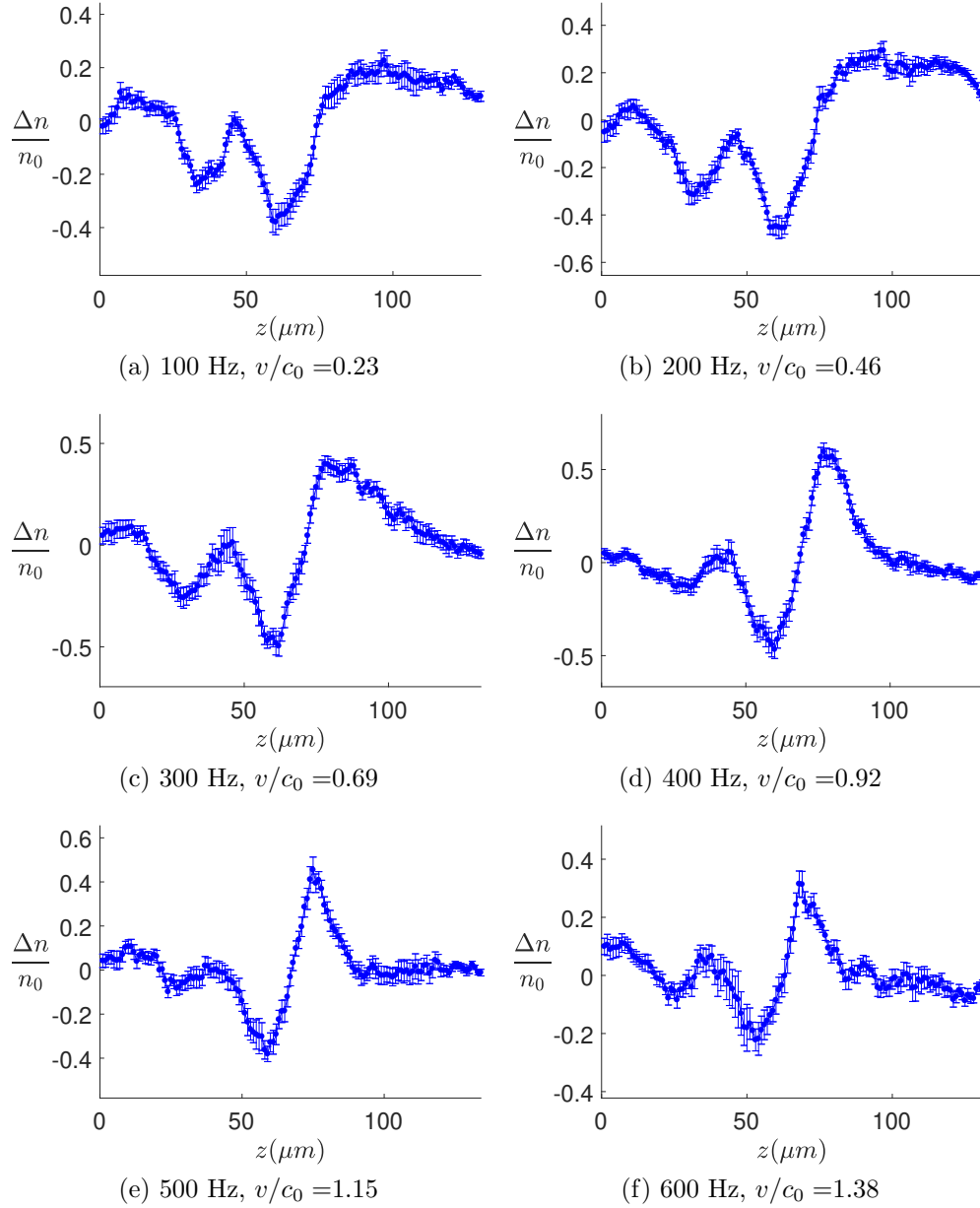
Figure 5.3: A periodic optical potential with wavelength $\lambda = 30$ μm, velocity $v = \lambda f$, and magnitude $\delta U_0 \approx 0.25 E_F$ is swept through a uniform sample which has an average density $n_0 = N/V = 2.6 \times 10^{11}$ atoms/cm^3, where N is the total atom number for both spin states. The response of the density δn is recorded as a function of the periodic potential frequency with a range f=100-750 Hz in steps of 50 Hz.

The amplitude of the density response is largest when the speed of $\delta U(z, t)$ is nearly resonant with the sound speed, $v/c_0 = 1$. In the supersonic regime, $v/c_0 > 1$, the amplitude of the density response falls as the perturbation travels faster than the speed of sound.

5.2 Preparing the hydrodynamic model for numerical integration

We use the hydrodynamic model from chapter 4 to understand the features of the density response δn and to extract the thermal conductivity κ_T. To make the extraction of transport parameters easier, we will slightly modify the form of the equations in chapter 4, which are repeated here,

$$\partial_t^2 \, \delta\tilde{n} - c_0^2 \left(\partial_z^2 \, \delta\tilde{n} + \partial_z^2 \, \delta\tilde{S}\right) - \frac{4}{3} \frac{\eta_0}{n_0 \, m} \, \partial_z^2 \, \partial_t \, \delta\tilde{n} = \frac{v_{F0}^2}{2} \, \partial_z \left(\tilde{n}_0 \, \partial_z \, \delta\tilde{U} + \delta\tilde{n} \, \partial_z \, \tilde{U}_0\right), \quad (5.4)$$

and

$$\partial_t \, \delta\tilde{S} - \frac{\kappa_T}{n_0 \, c_V} \, \partial_z^2 \, \delta\tilde{S} = \frac{\kappa_T}{n_0 \, c_V} \frac{c_P - c_V}{c_P} \, \partial_z^2 \, \delta\tilde{n}. \quad (5.5)$$

The measured shear viscosity $[4, 19]$ at temperature θ_0

$$\eta_0 = \alpha_0 \, \hbar \, n_0, \quad (5.6)$$

is used as input to equation 5.4. Using 5.6, the factor $(4/3)(\eta_0/n_0 \, m)$ in equation 5.4 becomes

$$\frac{4}{3} \frac{\eta_0}{n_0 \, m} = \frac{4}{3} \frac{\hbar}{m} \, \alpha_0. \quad (5.7)$$

We use a fitting parameter C_κ in equation 5.5 to extract the thermal conductivity κ_T. We insert the fitting parameter C_κ into our model by writing the thermal conductivity in terms of the calculated ratio of the thermal conductivity to the shear viscosity in the high temperature limit [5,6],

$$\kappa_T = C_\kappa \frac{15}{4} \frac{k_B}{m} \eta_0 = C_\kappa \frac{15}{4} \frac{k_B}{m} \alpha_0 \, \hbar \, n_0. \tag{5.8}$$

Using 5.8, the factor $\kappa_T/(n_0 \, c_V)$ in equation 5.5 becomes

$$\frac{\kappa_T}{n_0 \, c_V} = C_\kappa \frac{15}{4} \frac{k_B}{c_V} \frac{\hbar}{m} \alpha_0. \tag{5.9}$$

Using the substitutions 5.7 and 5.9, the hydrodynamic model takes the form

$$\boxed{\partial_t^2 \, \delta \tilde{n} - c_0^2 \left(\partial_z^2 \, \delta \tilde{n} + \partial_z^2 \, \delta \tilde{S} \right) - \frac{4}{3} \frac{\hbar}{m} \alpha_0 \partial_z^2 \, \partial_t \, \delta \tilde{n} = \frac{v_{F0}^2}{2} \partial_z \left(\tilde{n}_0 \, \partial_z \, \delta \tilde{U} + \delta \tilde{n} \, \partial_z \, \tilde{U}_0 \right),} \tag{5.10}$$

and

$$\boxed{\partial_t \, \delta \tilde{S} - C_\kappa \frac{15}{4} \frac{k_B}{c_V} \frac{\hbar}{m} \alpha_0 \partial_z^2 \, \delta \tilde{S} = C_\kappa \frac{15}{4} \frac{k_B}{c_V} \frac{\hbar}{m} \alpha_0 \frac{c_P - c_V}{c_P} \partial_z^2 \, \delta \tilde{n}.} \tag{5.11}$$

5.2.1 Periodic optical potential $\delta U(z,t)$

The data are modeled by numerically integrating equations 5.10, and 5.11 with the perturbing optical potential

$$\boxed{\delta U(z,t) = \delta U_0 \left[1 - \epsilon \cos(qz - qvt) \right] \phi(vt - z) \, e^{-(z-z_0)^2/\sigma^2}} \tag{5.12}$$

which is shown at $t = t_f$ in Fig. 5.1. The gaussian profile of the beam illuminating the DMD is accounted for with the gaussian function

$$e^{-(z-z_0)^2/\sigma^2}. \tag{5.13}$$

The position z_0 and width σ of the gaussian beam used to create the perturbing optical potential are easily measured and are not used as fit parameters.

The envelope function

$$\phi(vt - z) = \frac{1 + \tanh\left[z/\sigma_\phi\right]}{2} \times \frac{1 + \tanh\left[(vt - z)/\sigma_\phi\right]}{2} \tag{5.14}$$

is created with hyperbolic tangent functions. The time dependent factor

$$\frac{1 + \tanh\left[(vt - z)/\sigma_\phi\right]}{2} \tag{5.15}$$

moves with $[1 - \epsilon \cos(qz - qvt)]$ and provides a cutoff for the leading edge of the perturbation. The time independent factor of the envelope function

$$\frac{1 + \tanh\left[z/\sigma_\phi\right]}{2} \tag{5.16}$$

provides a cutoff for the perturbation $\delta U(z, t)$ at the box potential wall located at $z = 0$. The width of the envelope function edges σ_ϕ are set to 4 μm, to approximate the blurring from the finite resolution of the imaging system.

5.2.2 Fit parameter effects on the modeled density response

Global fits determine the fit parameters, ϵ, c_0, δU_0 and C_κ. Each fit parameter controls unique spatial features in the density response δn.
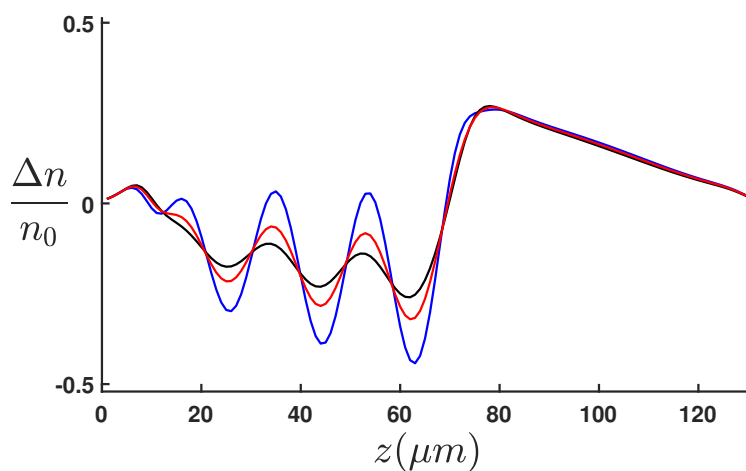
Figure 5.4: Effect of the modulation depth ϵ of equation 5.12 on the destiny response. The perturbing optical potential has wavelength $\lambda = 19$ μm and frequency f =400 Hz. The optical perturbation parameter $\epsilon = 0.23$ is shown in red, $\epsilon \times 2$ shown in blue, and $\epsilon \times 0.5$ shown in black.

Fig. 5.4 shows the effect of the perturbing potential parameter ϵ in equation 5.12. The depth of the periodic part of the perturbing potential ϵ affects the density response δn in the region where the density response shows periodic modulation.

Fig. 5.5 shows the effect of the sound speed c_0 on the modeled density response δn. As discussed in chapter 3, the fitted c_0 is used as an *in situ* thermometer to determine the reduced temperature $\theta_0 = T_0/T_F$. This temperature $\theta_0(c_0)$ determines the heat capacities per particle c_V and c_P from the measured equation of state [25], and the measured shear viscosity η_0 [4, 19] in equation 5.22.
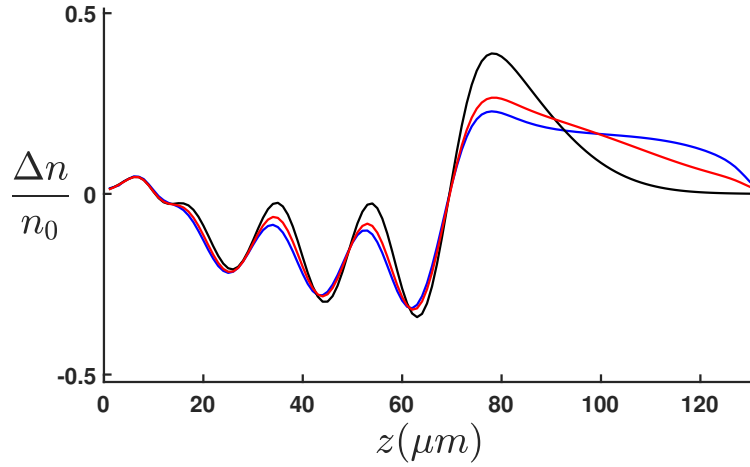
Figure 5.5: Effect of the adiabatic sound speed c_0 on the destiny response. The perturbing optical potential has wavelength $\lambda = 19$ μm and frequency f =400 Hz. The sound speed $c_0 = 1.3$ cm/s is shown in red, $c_0 = 1.6$ cm/s shown in blue, and, $c_0 = 1.0$ cm/s shown in black.

The fitted c_0 is checked by measuring the speed of a density perturbation in a uniform trap, as shown in Fig. 5.6. We create the density perturbation with a 300 μs pulse of repulsive laser light at position $z = 0$ and time $t = 0$. Tracking the position of the density peak as a function of times gives a speed of $v_p = 1.4$ cm/s, which is within 10% of the fitted sound speed from the density response, $c_0 = 1.3$ cm/s.

Fig. 5.7 shows the effect of the thermal conductivity fitting parameter C_κ on the density response for different ω/γ_κ. A larger value for the fitting parameter C_κ increases the amplitude of the leading density response peak. When $\delta U(z,t)$ moves through the sample slowly enough to allow temperature gradients to relax, $\omega \ll \gamma_\kappa$, the compression is isothermal. Moving $\delta U(z,t)$ through the sample with frequency $\omega \gg \gamma_\kappa$ results in adiabatic compression.

(a)

(b)
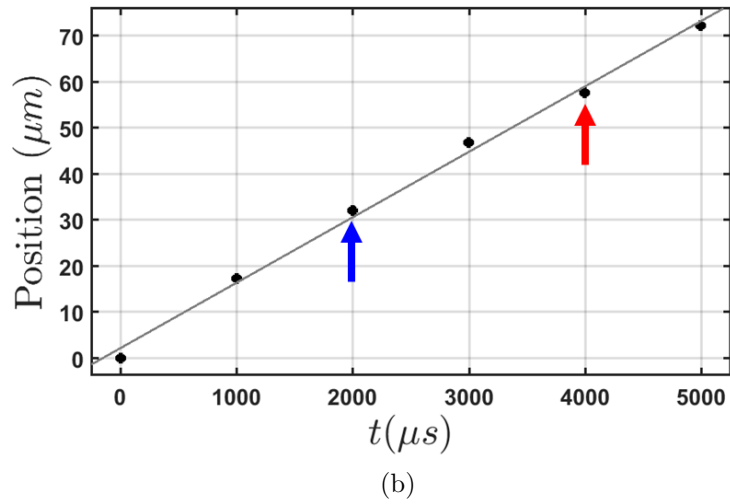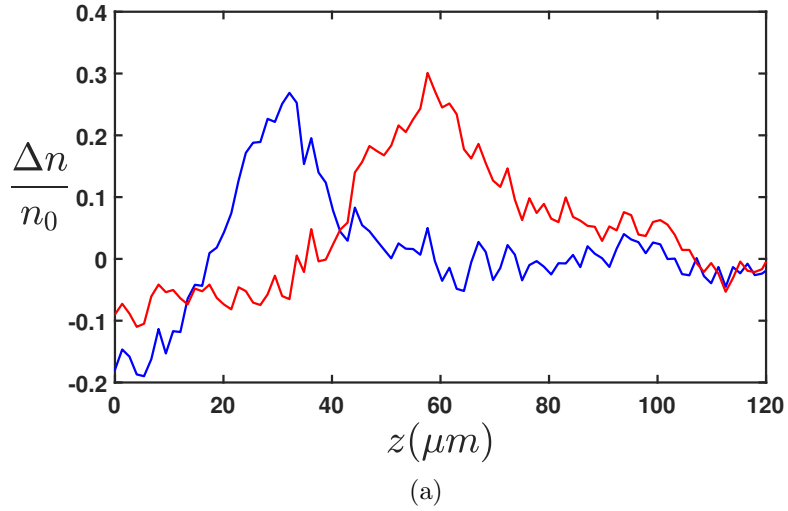
Figure 5.6: (a) Profiles of a density perturbation traveling through the uniform sample at two different times: $t = 2000$ μs in blue and $t = 4000$ μs in red. (b) Tracking the position of the density peak as a function of times gives a speed of $v_p = 1.4$ cm/s, which is within 10% of the fitted sound speed from the density response, $c_0 = 1.3$ cm/s.
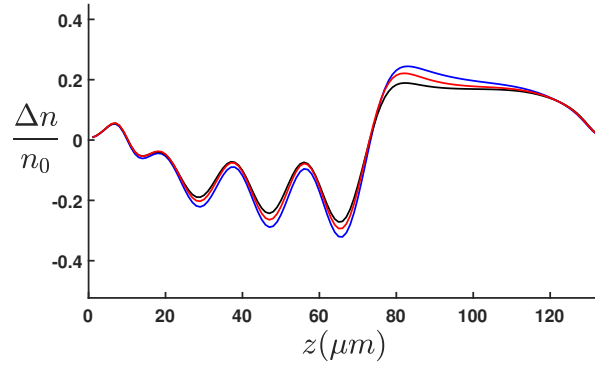
(a) 200 Hz
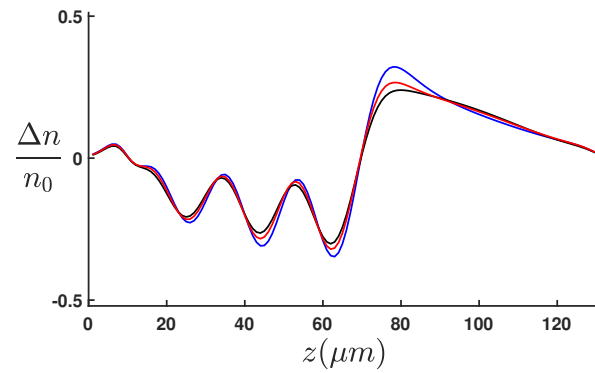
(b) 400 Hz
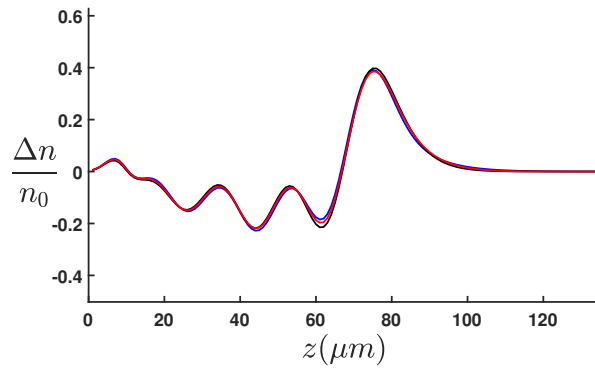
(c) 700 Hz

Figure 5.7: Effect of the thermal conductivity fitting parameter C_κ on the destiny response. The perturbing optical potential has wavelength $\lambda = 19$ μm. The modeled density response is shown for three frequencies $f = 200$ Hz, $f = 400$ Hz, and $f = 700$ Hz. As the wave frequency $\omega = 2\pi f$ of the perturbing potential approaches the decay rate γ_κ, the sensitivity to the thermal conductivity vanishes. The thermal conductivity fitting parameter $C_\kappa = 1$ is shown in red, $C_\kappa = 2$ shown in blue, and $C_\kappa = 0.5$ shown in black. (a) $\omega/\gamma_\kappa = 0.26$ (b) $\omega/\gamma_\kappa = 0.52$ (c) $\omega/\gamma_\kappa = 0.92$

5.2.3 Global fits

The thermal conductivity κ_T is extracted by varying the parameters c_0, δU_0, ϵ, and C_κ, one at a time, across all frequencies and minimizing χ^2 in the central region of the data away from the less dense edges at the box potential walls. A global best fit is obtained when varying the parameters no longer results in improvement. The uncertainty in C_κ is determined from the curvature of χ^2 in the region of the minimum [2]

$$\sigma_{\kappa_T}^2 = 2 \left(\frac{\partial^2 \chi^2}{\partial \kappa_T} \right)^{-1}, \tag{5.17}$$

where

$$\chi^2 \equiv \sum \left[\frac{1}{\sigma_i^2} (y_i - y(x_i))^2 \right]. \tag{5.18}$$

The results of the global fits are shown in Fig. 5.8 and 5.9 with the fitted parameters in the figure caption and the table 5.2. The hydrodynamic model reproduces the spatial features of the density response δn for all frequencies in the subsonic regime, $v < c_0$.

Table 5.2: Best Fit Parameters.

Parameter	$\lambda = 19\ \mu m$	$\lambda = 30\ \mu m$
c_0	1.3 cm/s	1.3 cm/s
$\theta(c_0)$	0.50	0.50
ϵ	0.23	0.29
$\delta U_0/E_F$	0.23	0.26
z_0	93 μm	93 μm
σ	113 μm	113 μm
σ_ϕ	4 μm	4 μm

The fitted values of $\delta U_0/E_F \approx 0.25$ are consistent with the calculated value of

64
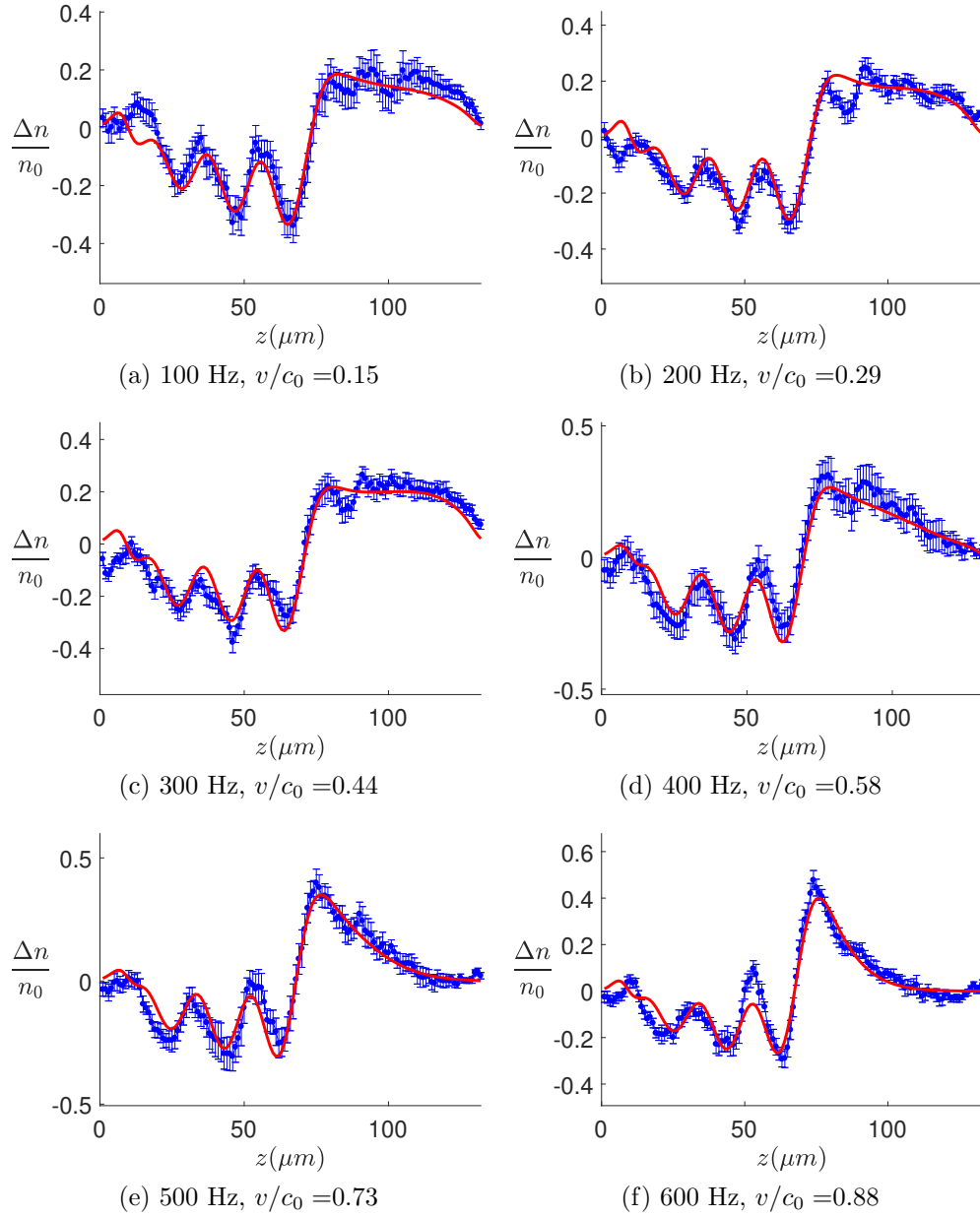
Figure 5.8: The hydrodynamic model is fit globally to the subsonic data for a periodic optical potential with wavelength $\lambda = 19$ μm and velocity $v = \lambda f$. Fit parameters are $\delta U_0 / E_F = 0.23$, $\epsilon = 0.23$, and $c_0 = 1.3$ cm/s.

65
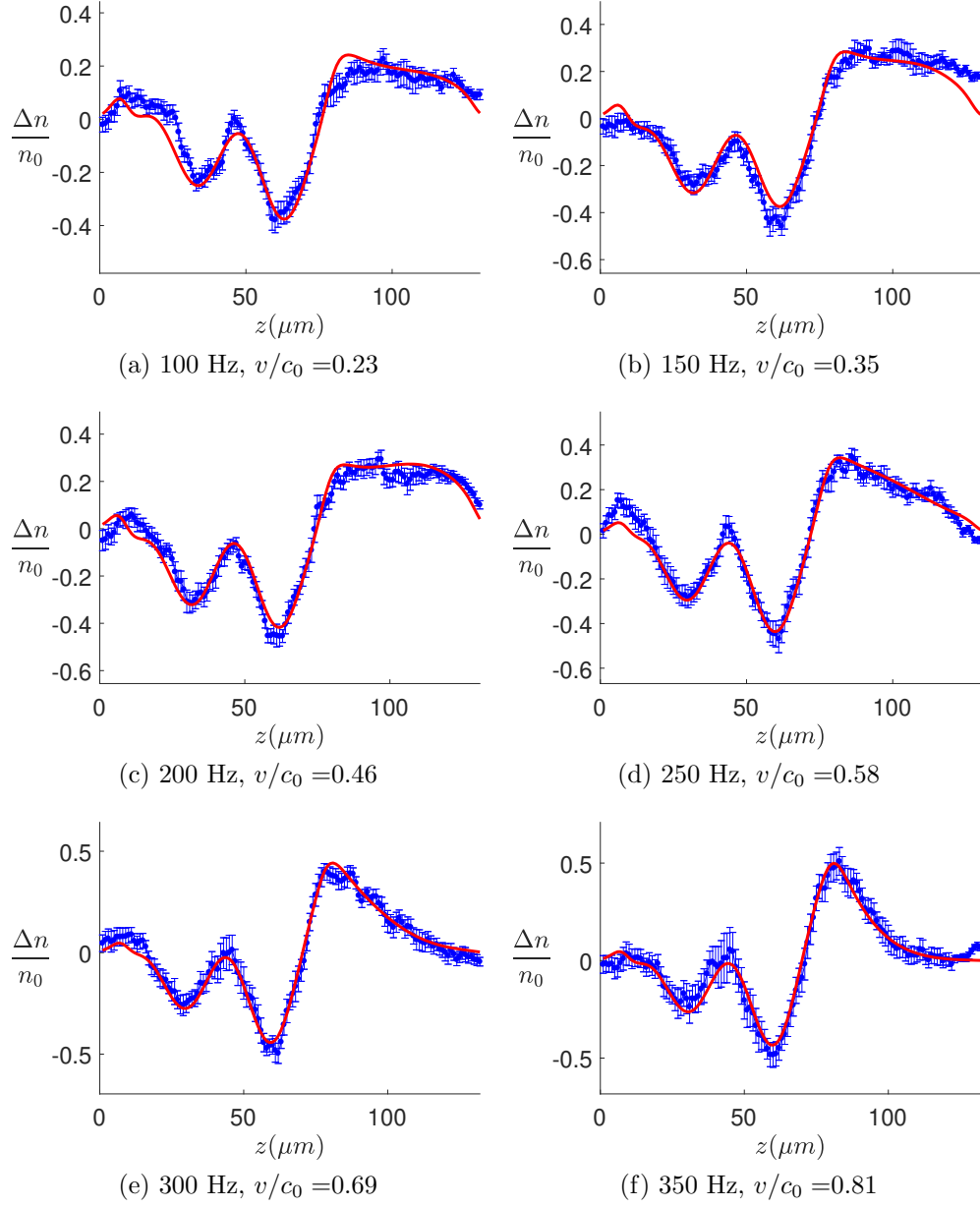
Figure 5.9: The hydrodynamic model is fit globally to the subsonic data for a periodic optical potential with wavelength $\lambda = 30$ μm and velocity $v = \lambda f$. Fit parameters are $\delta U_0 / E_F = 0.26$, $\epsilon = 0.29$, and $c_0 = 1.3$ cm/s.

66

$\delta U_0/E_F = 0.31$ which is estimated from the measured intensity of the illuminating blue-detuned beam. The fitted value of δU_0 is used to account for variations in the beam intensity and imperfect imaging. The perturbing potential parameter ϵ for $\lambda = 19$ μm is smaller than the $\lambda = 30$ μm wavelength because of the blurring from the finite resolution ≈ 3.5 μm of our imaging system. The fitted sound speed $c_0 = 1.3$ cm/s, determines the temperature to be $\theta_0(c_0) = 0.50$. In chapter 3 we discussed how the measured sound speed c_0 combined with the measured universal function $f_E(\theta)$ enables thermometry in our experiment. The temperature as a function of the sound speed, shown in Fig 3.2, is a monotonic function of c_0/v_F for $\theta > 0.25$.

5.2.4 Measurement of κ_T

The global fits to the data determine the fitting parameter in equation 5.8 to be

$$C_\kappa = 0.93(14). \tag{5.19}$$

We defined the fitting parameter C_κ in equation 5.8 to provide an easy comparison with the variational calculations in the high temperature limit [5,6]

$$\frac{\kappa_T}{\eta_0} = \frac{15}{4} \frac{k_B}{m}. \tag{5.20}$$

The fitting parameter C_κ gives a straightforward comparison with the high temperature ratio,

$$\frac{\kappa_T}{\eta_0} = 0.93(14) \times \frac{15}{4} \frac{k_B}{m}. \tag{5.21}$$

The previously measured shear viscosity is given by equation 4.5 of chapter 4,

$$\eta_0 = \alpha_0 \, \hbar \, n_0 = \left[2.77(21) \, \theta_0^{3/2} + 0.25(8) \right] \hbar \, n_0 \qquad (5.22)$$

yields $\eta_0 = 1.23 \, \hbar n_0$ at $\theta_0 = 0.50$.

Using $\eta_0 = 1.23 \, \hbar n_0$ in equation 5.21 yields the measured thermal conductivity from our global fits,

$$\kappa_T = 1.14(17) \times \frac{15}{4} \frac{k_B}{m} \hbar \, n_0. \qquad (5.23)$$

The fitting parameters in table 5.2 are fit globally and are constant for all frequencies. The hydrodynamic model loses sensitivity to the thermal conductivity as the periodic potential is moved through the sample at speeds comparable to the decay rate of temperature gradients γ_κ. The vanishing sensitivity to the thermal conductivity can be seen in Fig. 5.7. For this reason, we determine C_κ from fits to the data below the decay rate $\omega < \gamma_\kappa$.

For the optical potential with wavelength $\lambda = 19$ μm, where $\gamma_\kappa = 2\pi \times 760$ Hz, we obtain reliable measurements of C_κ in the frequency range 200-600 Hz. For the optical potential with wavelength $\lambda = 30$ μm and decay rate $\gamma_\kappa = 2\pi \times 305$ Hz, we obtain reliable measurements of C_κ in the frequency range 150-300 Hz. The extracted thermal conductivity κ_T is determined from the weighted arithmetic mean of the individually fit C_κ from both wavelengths.

When the perturbing optical potential is moved through the sample at supersonic speed, $v > c_0$, we observe a discrepancy with the model, as shown in Fig. 5.10. This discrepancy may arise from the creation of weak shock waves which are not included in our linear hydrodynamic model.
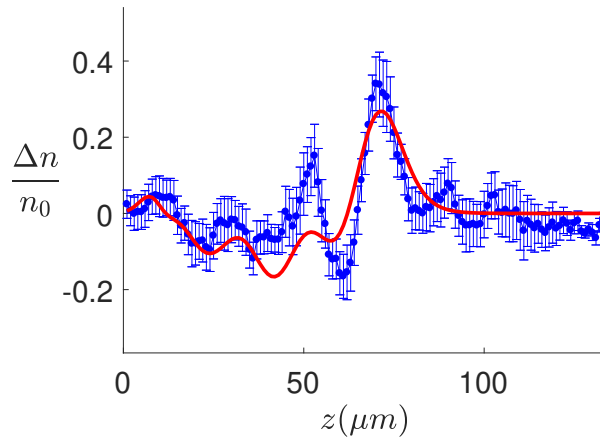
Figure 5.10: The quality of the fits decreases as the speed of $\delta U(z,t)$ surpasses the adiabatic sound speed $v > c_0$. The globally fit hydrodynamic model is plotted for a $\lambda = 19$ μm optical perturbation at 900 Hz, $v/c_0 = 1.32$.

Chapter 6

Conclusion

We have created a novel technique for measuring transport properties from the density response δn of a uniform density atomic cloud driven by a perturbative optical potential $\delta U(z,t)$. Using DMDs to create the perturbing periodic optical potential $\delta U(z,t)$ gives us independent control over the wavelength λ and frequency f of the driving potential. This control allows us to probe the density response for both adiabatic and isothermal compression. We measure the density response δn to the perturbing optical potential $\delta U(z,t)$ as a function of wavelength and frequency, and fit the data with a hydrodynamic linear response model. Excellent agreement between the data and our model validates this treatment.

The model is sensitive to the thermal conductivity κ_T when $\delta U(z,t)$ moves through the sample slowly enough to allow temperature gradients to relax, $\omega << \gamma_\kappa$. In this regime, the measured thermal conductivity is

$$\kappa_T = 1.14(17) \times \frac{15}{4} \frac{k_B}{m} \hbar \, n_0, \tag{6.1}$$

at a temperature $\theta_0 = 0.50$.

The temperature θ_0 in our experiment was not further increased because the trapping potential $U_0(z, t)$ was too weak to confine higher temperatures. Future work may be done to obtain a deeper trapping potential to measure the thermal conductivity at higher temperatures.

6.1 Outlook

The addition of DMD technology to the apparatus enable us to create designer optical potentials. The ability to arbitrarily shape dynamic optical potentials opens up the possibility for many exciting new experiments. Varying the shape of the driving optical potential $\delta U(z, t)$ is easy to implement and could allow us to probe transport parameters in a new way. Using a DMD, the damping and decay of sound can be studied by imprinting a uniform density sample with multiple wavelength patterns and observing the evolution of density perturbations.

Other possibilities for future work include trapping spin imbalanced mixtures in the box trap $U_0(z)$, which changes the transport properties, and studying the breakdown of fluid dynamics by changing the interaction strength to weakly interacting.

REFERENCES

[1] A. Adams, L. D. Carr, T. Schäfer, P. Steinberg, and J. E. Thomas. Strongly correlated quantum fluids: ultracold quantum gases, quantum chromodynamic plasmas and holographic duality. *New Journal of Physics*, 14(11):115009, nov 2012.

[2] P. R. Bevington and D. K. Robinson. *Data Reduction and Error Analysis*. McGraw-Hill Higher Education, 3rd edition, 1980.

[3] W. Bloch, Immanuel and Dalibard, Jean and Zwerger. Many-body physics with ultracold gases. *Rev. Mod. Phys.2*, 80(3):885—-964, 2008.

[4] M. Bluhm, J. Hou, and T. Schäfer. Determination of the Density and Temperature Dependence of the Shear Viscosity of a Unitary Fermi Gas Based on Hydrodynamic Flow. *Physical Review Letters*, 119(6):1–6, 2017.

[5] M. Braby, J. Chao, and T. Schäfer. Thermal conductivity and sound attenuation in dilute atomic Fermi gases. *Physical Review A - Atomic, Molecular, and Optical Physics*, 82(3), 2010.

[6] H. Bruun, G. and Smith. Shear viscosity and damping for a Fermi gas in the unitarity limit. *Physical Review Letters*, 75(January 2007), 2007.

[7] C. Cao. *Universal Quantum Viscosity in a Unitary Fermi Gas*. PhD thesis, Duke University, 2012.

[8] S. Eckel, A. Kumar, T. Jacobson, I. B. Spielman, and G. K. Campbell. A Rapidly Expanding Bose-Einstein Condensate: An Expanding Universe in the Lab. *Physical Review X*, 8(2):21021, 2018.

[9] E. Elliott. *Quantum Transport and Scale Invariance in Expanding Fermi Gases.* PhD thesis, Duke University, 2014.

[10] E. Elliott, J. A. Joseph, and J. E. Thomas. Observation of conformal symmetry breaking and scale invariance in expanding Fermi gases. *Physical Review Letters*, 112(4):1–5, 2014.

[11] T. Enss, R. Haussmann, and W. Zwerger. Viscosity and scale invariance in the unitary Fermi gas. *Annals of Physics*, 326(3):770–796, 2011.

[12] M. Gehm. *Optically-Trapped Degenerate Fermi Gas of 6Li: Finding the Route to Degeneracy.* PhD thesis, Duke University, 2003.

[13] S. Giorgini, Stefano and Pitaevskii, Lev P. and Stringari. Theory of ultracold atomic Fermi gases. *Rev. Mod. Phys.*, 80(4):1215—-1274, 2008.

[14] T.-L. Ho. Universal Thermodynamics of Degenerate Quantum Gases in the Unitarity Limit. *Physical Review Letters*, 92(March):9–12, 2004.

[15] Y. H. Hou, L. P. Pitaevskii, and S. Stringari. Scaling solutions of the two-fluid hydrodynamic equations in a harmonically trapped gas at unitarity. *Physical Review A - Atomic, Molecular, and Optical Physics*, 87(3):2–4, 2013.

[16] C. G. Jae-yoon Choi, Sebastian Hild, Johannes Zeiher, Peter Schauß, Antonio Rubio-Abadal, Tarik Yefsah, Vedika Khemani, David A. Huse, Immanuel Bloch. many-body localization transition in two dimensions. *Science*, 352(6293):1547, 2016.

[17] J. Joseph, B. Clancy, L. Luo, J. Kinast, A. Turlapov, and J. E. Thomas. Measurement of Sound Velocity in a Fermi Gas near a Feshbach Resonance. *Physical Review Letters*, 98(April):1–4, 2007.

[18] J. A. Joseph. *Precision Measurement of the Sound Velocity in an Ultracold Fermi Gas Through the BEC-BCS Crossover.* PhD thesis, Duke University, 2010.

[19] J. A. Joseph, E. Elliott, and J. E. Thomas. Shear Viscosity of a Unitary Fermi Gas Near the Superfluid Phase Transition. *Physical Review Letters*, 115(2):1–5, 2015.

[20] J. A. Joseph, J. E. Thomas, M. Kulkarni, and A. G. Abanov. Observation of shock waves in a strongly interacting fermi gas. *Physical Review Letters*, 106(15):1–4, 2011.

[21] J. Kinast. *Thermodynamics and SuperFluidity of a Strongly Interacting Fermi Gas.* PhD thesis, Duke University, 2006.

[22] K.~M.~O'Hara, S.~L.~Hemmer, M.~E.~Gehm, S.~R.~Granade, and J.~E.~Thomas. Observation of a strongly interacting degenerate Fermi gas of atoms. *Science*, 298(December):2179, 2002.

[23] L. Landau and E. Lifshitz. *Fluid Mechanics.* Reed Educational and Professional Publishing Ltd, 2nd edition, 1987.

[24] L. Luo and J. E. Thomas. Thermodynamic measurements in strongly interacting Fermi gas. *Journal of Low Temperature Physics*, 154(1-2):1–29, 2009.

[25] M. W. Z. Mark J. H. Ku, Ariel T. Sommer, Lawrence W. Cheuk. Revealing the Superfluid Lambda Transition in the Universal Thermodynamics of a Unitary Fermi Gas. *Science*, 335:563, 2012.

[26] N. Meyer, H. Proud, M. Perea-Ortiz, C. O'Neale, M. Baumert, M. Holynski, J. Kronjäger, G. Barontini, and K. Bongs. Observation of Two-Dimensional Localized Jones-Roberts Solitons in Bose-Einstein Condensates. *Physical Review Letters*, 119(15):1–6, 2017.

[27] E. Shuryak. Why does the Quark-Gluon Plasma at RHIC behave as a nearly ideal fluid ? *Progress in Particle and Nuclear Physics*, 53:273–303, 2004.

[28] D. T. Son. Vanishing bulk viscosities and conformal invariance of the unitary fermi gas. *Physical Review Letters*, 98(2):12–15, 2007.

[29] J. E. Thomas, J. Kinast, and A. Turlapov. Virial theorem and universality in a unitary fermi gas. *Physical Review Letters*, 95(12):1–4, 2005.

[30] W. L. Wiese, M. W. Smith, Glennon, and B. M. *Atomic Transition Probabilities, National Standard Reference Data Series*. National Bureau of Standards 4, 1966.

[31] G. Wlazłowski, P. Magierski, A. Bulgac, and K. J. Roche. Temperature evolution of the shear viscosity in a unitary Fermi gas. *Physical Review A*, 013639:1–6, 2013.

[32] Z. Zhang, Pengfei and Yu. Energy absorption spectroscopy of unitary Fermi gases in a uniform potential. *Phys. Rev. A*, 97(4):041601, 2018.

[33] G. Zurn, T. Lompe, A. N. Wenz, S. Jochim, P. S. Julienne, and J. M. Hutson. Precise Characterization of 6 Li Feshbach Resonances Using Trap-Sideband-Resolved RF Spectroscopy of Weakly Bound Molecules. *Physical Review Letters*, 135301(March):1–5, 2013.

APPENDICES

Appendix A

Numerical integration and best fit MATLAB code

The following MATLAB code is used simulate the density response to a periodic optical potential

$$\delta U(z,t) = \delta U_0 \left[1 - \epsilon \cos(qz - qvt)\right] \phi(vt - z) \, e^{-(z-z_0)^2/\sigma^2}. \qquad (A.1)$$

The program numerically integrates the two coupled equations,

$$\partial_t^2 \, \delta\tilde{n} - c_0^2 \left(\partial_z^2 \, \delta\tilde{n} + \partial_z^2 \, \delta\tilde{S}\right) - \frac{4}{3}\frac{\hbar}{m}\alpha_0 \, \partial_z^2 \, \partial_t \, \delta\tilde{n} = \frac{v_{F0}^2}{2} \, \partial_z \left(\tilde{n}_0 \, \partial_z \, \delta\tilde{U} + \delta\tilde{n} \, \partial_z \, \tilde{U}_0\right), \qquad (A.2)$$

and

$$\partial_t \, \delta\tilde{S} - C_\kappa \, \frac{15}{4}\frac{k_B}{c_V}\frac{\hbar}{m}\alpha_0 \, \partial_z^2 \, \delta\tilde{S} = C_\kappa \, \frac{15}{4}\frac{k_B}{c_V}\frac{\hbar}{m}\alpha_0 \, \frac{c_P - c_V}{c_P}\partial_z^2 \, \delta\tilde{n}. \qquad (A.3)$$

Global fits determine the fit parameters, ϵ, c_0, δU_0 and C_κ by minimizing χ^2 in the central region of the data away from the less dense edges at the box potential walls.

A.1 Linear response code main function

```
1  %% beginning
2
3  % choose which program to run
4  modenumber = 0;
5  %case0 plot dn vs dn experimental
6  %case1 C_\kappa fit
7  %case2 epsilon fit
8  %case3 c0 fit
9  %case4 wavelength fit
10 %case5 Adjust Starting point for \delta U
11 %case6 Multiplier to adjust parameters on the fly
12 %case7 Vary c_0 and mult at the same time for one frequency
13
14 % index correspoding to frequency for single frequency plot
15 freqindex = 6;
16 % multiplier set to 1
17 mult = 1;
18 % Dataset number
19 dataset = 14;
20 % 14->19 micron wavelength
21 % 10->25 micron wavelength
22 % 8 ->30 micron wavelength
23
```

```matlab
24 %% parameters for different datasets
25 switch dataset
26     case 14
27         filename = 'Z:\Network Harddrives for Lab Computers\
                LAB1_DOCS\NCSU Data\2019.1.8\14pi8%\LorinAnalysis\
                JohnsData1long.mat';
28         % frequency range to globally fit data
29         freqrangeStart=1;
30         freqrange =7;
31         % C_\kappa
32         KappaT = 1;
33         % epsilon for perturbing potential \delta U
34         eps = .2251;
35         % sound speed in m/s
36         c0 = .013;
37         % multiplier for calculated \delta U_0
38         % accounts for loss through optics and imperfect imaging
                optics
39         dUmult = .7277*(1/2);
40         % adjust the starting point of \delta U(z,0)
41         dUoffset = -2;
42         % wavelength
43         lam = 18.667;
44         % number of spatial periods \delta U is moved into the
```

```matlab
                sample
45              NumPeriods = 4;
46          % atom number in the box trap (both spin states)
47              AtomNumber = 76199*2;
48      case 10
49              filename = 'Z:\Network Harddrives for Lab Computers\
                    LAB1_DOCS\NCSU Data\2019.1.8\10pi8%\LorinAnalysis\
                    JohnsData1long.mat';
50              freqrangeStart=1;
51              freqrange = 4;
52              KappaT = 1;
53              eps = .3271;
54              c0 = .0142;
55              dUmult = .8275*(1/2);
56              dUoffset = -1;
57              lam = 24.8182;
58              NumPeriods = 3;
59              AtomNumber = 84922*2;
60      case 8
61              filename = 'Z:\Network Harddrives for Lab Computers\
                    LAB1_DOCS\NCSU Data\2019.1.8\8pi8%\LorinAnalysis\
                    JohnsData1long.mat';
62              freqrangeStart=1;
63              freqrange = 6;
```

```matlab
64          KappaT = 1;
65              eps = .2854;
66              c0 = .013;
67              dUmult = .8414*(1/2);
68              dUoffset = 15;
69              lam = 30;
70              NumPeriods = 3;
71              AtomNumber = 84291*2;
72          otherwise
73              disp('Invalid case')
74              filename = 'Invalid';
75  end
76
77  load(filename);
78
79  switch modenumber
80
81      %% KappaT
82      % minimize chi^2 to find best fit for thermal conductivity
                fitting
83      % parameter C_\kappa
84          case 1
85  % range to check
86  KappaT = .25:.25:4;
```

```matlab
87  freqindex = freqrangeStart:freqrange;
88  for j = 1:length(freqindex)
89  parfor i = 1:length(KappaT)
90  [A,B,BErr,Chi2,Chi2reduced(i,j),n0fit,leftside,rightside,z] =
        LR13Thesis(filename,NumPeriods,freqindex(j),KappaT(i),eps,c0,
        lam,dUoffset,dUmult,mult);
91  disp('Parameter KappaT')
92  end
93  end
94  % minimize chi^2 and find corresponding value for C_\kappa
95  v = sum(Chi2reduced,2);
96  x = KappaT;
97  finexq = linspace(min(KappaT),max(KappaT),100);
98  finevq = interp1(x,v,finexq,'spline');
99  [minval,indexmin] = min(finevq);
100 disp([num2str(finexq(indexmin)),' is the value of KT'])
101 disp([num2str(minval),' is the Chi2'])
102 plot(KappaT,v)
103 shg
104
105     %% eps
106     % minimize chi^2 to find best fit for \delta U fitting
            parameter
107     % epsilon
```

```matlab
108         case 2
109   eps = .2:.005:.23;
110   freqindex = freqrangeStart:freqrange;
111   for j = 1:length(freqindex)
112   parfor i = 1:length(eps)
113   [A,B,BErr,Chi2,Chi2reduced(i,j),n0fit,leftside,rightside,z] = ...
          LR13Thesis(filename,NumPeriods,freqindex(j),KappaT,eps(i),c0, ...
          lam,dUoffset,dUmult,mult);
114   disp('Parameter eps')
115   end
116   end
117   v = sum(Chi2reduced,2);
118   x = eps;
119   finexq = linspace(eps(1),max(eps),100);
120   finevq = interp1(x,v,finexq,'spline');
121   [minval,indexmin] = min(finevq);
122   finexq(indexmin)
123   disp([num2str(minval),' is the Chi2'])
124   plot(eps,v)
125   shg
126
127         %% c0
128         % minimize chi^2 to find best fit for the sound speed c_0
129          case 3
```

```matlab
130  c0 = .012:.0005:.016;

131  freqindex = freqrangeStart:freqrange;

132  for j = 1:length(freqindex)

133  parfor i = 1:length(c0)

134  [A,B,BErr,Chi2,Chi2reduced(i,j),n0fit,leftside,rightside,z] =
         LR13Thesis(filename,NumPeriods,freqindex(j),KappaT,eps,c0(i),
         lam,dUoffset,dUmult,mult);

135  disp('Parameter c0')

136  end

137  end

138  v = sum(Chi2reduced,2);

139  x = c0;

140  finexq = linspace(min(c0),max(c0),100);

141  finevq = interp1(x,v,finexq,'spline');

142  [minval,indexmin] = min(finevq);

143  finexq(indexmin)

144  disp([num2str(minval),' is the Chi2'])

145  plot(c0,v)

146  shg

147     %% lam

148     % minimize chi^2 to find best fit for the wavelength (known
           parameter)

149      case 4

150  lam = 17:.5:22;
```

```matlab
151  freqindex = freqrangeStart:freqrange;

152  for j = 1:length(freqindex)

153  parfor i = 1:length(lam)

154  [A,B,BErr,Chi2,Chi2reduced(i,j),n0fit,leftside,rightside,z] =
         LR13Thesis(filename,NumPeriods,freqindex(j),KappaT,eps,c0,lam
         (i),dUoffset,dUmult,mult);

155  disp('Parameter lam')

156  end

157  end

158  v = sum(Chi2reduced,2);

159  x = lam;

160  finexq = linspace(min(lam),max(lam),100);

161  finevq = interp1(x,v,finexq,'spline');

162  [minval,indexmin] = min(finevq);

163  finexq(indexmin)

164  plot(lam,v)

165  shg

166

167      %% dUoffset

168      % adjust the starting point for \delta U to correctly begin
             at the edge

169      % of the box

170        case 5

171  dUoffset = -3:1:3;
```

```matlab
172  freqindex = freqrangeStart:freqrange;

173  for j = 1:length(freqindex)

174  parfor i = 1:length(dUoffset)

175  [A,B,BErr,Chi2,Chi2reduced(i,j),n0fit,leftside,rightside,z] =
         LR13Thesis(filename,NumPeriods,freqindex(j),KappaT,eps,c0,lam
         ,dUoffset(i),dUmult,mult);

176  disp('Parameter dUoffset')

177  end

178  end

179  v = sum(Chi2reduced,2);

180  x = dUoffset;

181  finexq = linspace(min(dUoffset),max(dUoffset),100);

182  finevq = interp1(x,v,finexq,'spline');

183  [minval,indexmin] = min(finevq);

184  finexq(indexmin)

185  plot(dUoffset,v)

186  shg

187

188      %% mult

189      % multiplier for adjusting parameters on the fly

190       case 6

191  mult = .4:.05:.9;

192  % freqindex = 1:length(SumData.frequency);

193  freqindex = freqrangeStart:freqrange;
```

```matlab
194  for j = 1:length(freqindex)
195  parfor i = 1:length(mult)
196  [A,B,BErr,Chi2,Chi2reduced(i,j),n0fit,leftside,rightside,z] =
         LR13Thesis(filename,NumPeriods,freqindex(j),KappaT,eps,c0,lam
         ,dUoffset,dUmult,mult(i));
197  disp('Parameter mult')
198  end
199  end
200  v = sum(Chi2reduced,2);
201  x = mult;
202  finexq = linspace(min(mult),max(mult),100);
203  finevq = interp1(x,v,finexq,'spline');
204  [minval,indexmin] = min(finevq);
205  finexq(indexmin)
206  disp([num2str(minval),' is the Chi2'])
207  plot(mult,v)
208  shg
209
210
211      %% dUmult and c_0 at the same time
212      % adjust \delta U_0 and the sound speed for a single
             frequency
213      case 7
214  mult = .7 :.05:1.2;
```

```matlab
c0 = .012:.001:.016;
for j = 1:length(c0)
parfor i = 1:length(mult)
[A,B,BErr,Chi2,Chi2reduced(i,j),n0fit,leftside,rightside,z] = ...
    LR13Thesis(filename,NumPeriods,freqindex,KappaT,eps,c0(j),lam ...
    ,dUoffset,dUmult,mult(i));
disp('Parameter mult')
end
end
close all
imagesc(Chi2reduced)
[min_val,idx]=min(Chi2reduced(:))
[rowmin,colmin]=ind2sub(size(Chi2reduced),idx)
disp([num2str(min_val),' Chi2reduced'])
disp([num2str(c0(colmin)),' c0'])
disp([num2str(mult(rowmin)*dUmult),' dUmult'])

    otherwise
%% Just a plot of the fit with default parameters
[A,B,BErr,Chi2,Chi2reduced,n0fit,leftside,rightside,z] = ...
    LR13Thesis(filename,NumPeriods,freqindex,KappaT,eps,c0,lam, ...
    dUoffset,dUmult,mult);

% reduced chi^2
```

```matlab
235    chi2KT = Chi2reduced;
236
237 %% density plots
238 ymin = min([min(A),min(B)]);
239 ymax = max([max(A),max(B)]);
240 % plot the hydrodynamic model
241 plot(A(leftside:rightside),'r','Linewidth',2.5)
242 xlabel('$$z(\mu m)$$','interpreter','latex');
243 ylabel('$$\frac{\Delta n}{n_0}$$','interpreter','latex')
244 ylh = get(gca,'ylabel');
245 ylp = get(ylh, 'Position');
246 set(ylh, 'Rotation',0, 'Position',ylp, 'VerticalAlignment','
         middle', 'HorizontalAlignment','right')
247 xlim([0 length(A(leftside:rightside))])
248 ylim([ymin-.2 ymax+.2])
249 hold on
250 % scatter has a lr offset to match up the data
251 % plot the experimental data with errorbars
252 errorbar(z(leftside:rightside)-leftside+2,B(leftside:rightside),
         BErr(leftside:rightside),'b')
253 scatter(z(leftside:rightside)-leftside+2,B(leftside:rightside)
         ,25,'blue','filled')
254 % put the frequency in the title
255 % title([num2str(SumData.frequency(freqindex)),' Hz']);
```

```
256  shg
257
258
259  end
```

A.2 Linear response code numerical integration

```
1   %% begining
2   function [A,B,BErr,Chi2,Chi2reduced,n0fit,leftside,rightside,z]
        = LR13Thesis(filename,NumPeriods,freqindex,KappaT,eps,c0,lam,
        dUoffset,dUmult,mult)
3
4   load(filename);
5
6   % Tp and Lp are time and length scales (microseconds and microns
        ).
7   Tp = 10^6;
8   Lp = 10^6;
9
10
11  %% Constants
12  % volume of the box trap
13  vol=(129E-6)*(58E-6)*(84E-6);
14  %frequency in MHz
15  freq = SumData.frequency(freqindex)/Tp;
```

```matlab
16  q=2*pi/lam;

17  nu=2*pi*freq;

18  v = nu/q;

19  % Veff = v/c0;

20  num = 2*pi/lam;

21

22  % number of atoms for both spin states

23  N=1.686e+05;

24  % average density in the box trap

25  n0=N/vol;

26  % peak density is n0*1.116

27  % Fermi temperature

28  TF=hbar^2/(2*mLi6)*(3*pi^2*n0)^(2/3)/kB;

29  % Fermi energy

30  EF = TF*kB;

31  % Fermi velocity

32  vF = sqrt(2*EF/mLi6);

33

34

35

36  % theta0 interpolation range

37  theta0interp = .1:.01:1.5;

38  % equation of state data

39  load('MartinEEOS.mat');
```

```matlab
40  % interpolated f_E(\theta)
41  Feqinterp = interp1(FEtheta(:,1),FEtheta(:,2),theta0interp,'
        spline');
42  % c_0 corresponding to interpolated equation of state
43  c0interp = vF*sqrt(Feqinterp/3);
44  % Calculated theta0 from c0 and equation of state
45  theta0 = interp1(c0interp,theta0interp,c0,'spline');
46
47  % dimensionless shear viscosity (muliplied by hbar/m)
48  a0 = Tp*(hbar/mLi6)*(2.77*theta0^1.5+0.27);
49
50  % calculate specific heat per particle at theta_0
51  [Cv,Cp]=SpecificHeats(theta0);%from MartinEoS
52  % C1 is the substitution for the factor KT/n0/Cv multiplied by
        the fitting
53  % parameter C_\kappa to extract the thermal conductivity
54  C1 = KappaT*(15/4)*(1/Cv)*a0;
55  C2 = (Cp-Cv)/Cp;
56
57
58  %% Time Mesh
59  % The simulation is run for a little longer than the end time to
        make sure
60  % the simulation ends at the same time as the data.
```

```
61  tend=NumPeriods/freq +0.1/ freq ;

62  % number of points in time

63  numtim = 6000;

64  t=linspace (0, tend ,numtim) ;

65  dt = t (2)−t (1) ;

66

67

68  %% Space Mesh

69  % start and endpoint spatially . The data includes a range
       outside of the

70  % box potential

71  startpoint = 20;

72  endpoint = 200;

73  % pixel size for the camera in microns

74  psize = 1.34;

75  % select region of interest for data

76  % background density n_0

77  vn0data = lrn0 ( freqindex , startpoint : endpoint ) ;

78  % \delta n

79  vdndata = lrdn ( freqindex , startpoint : endpoint ) ;

80  % error in \delta n

81  vdnErrdata = lrdnErr ( freqindex , startpoint : endpoint ) ;

82  % xdata is the units of the camera

83  xdata = linspace (0 ,( endpoint−startpoint ) ,( endpoint−startpoint )
```

```matlab
            +1)*psize;
84  % number of points spatially
85  numzpoints = floor(max(xdata));
86  % z makes each bin one micron in size
87  z=linspace(0,numzpoints,numzpoints+1);
88  dz=z(2)-z(1);
89  [Z,T]=meshgrid(z,t);
90  % interpolates the data to the micron scale rather than the
        psize scale
91  vn0dataq = interp1(xdata,vn0data,z,'spline');
92  vdndataq = interp1(xdata,vdndata,z,'spline');
93  vdnErrdataq = interp1(xdata,vdnErrdata,z,'spline');
94
95  %% n0fit
96  % fits the background density n_0 with the fitfunction
97  % fitfunction = @(a0,a1,a2,b1,b2,b3,b4,x) (a0+a1*x+a2*x.^2).*...
98  %     0.25.*(1+tanh((x-b1)/b2)).*(1+tanh((-x+b3)/b4));
99  [gfit] = FitUniformGas(vn0dataq,0);
100 % left side of the box
101 n0left = gfit.b1;
102 % right side of the box
103 n0right = gfit.b3;
104 % fitted density n_0(z)
105 n0fit = gfit(z)';
```

```matlab
106
107  %%% dU0dz
108  % temperature corresponing to the max central density
109  Ttwid = theta0*(max(n0fit)^(2/3));
110  % 10^-4 offset to the density
111  bgpercent = .0001;
112  % add the offset to the density and rescale the max
113  n0offset = (1-bgpercent)*n0fit+max(n0fit)*bgpercent;
114  % reduced temperature of the gas \theta(z)
115  TTfgas = Ttwid./(n0offset.^(2/3));
116  %This data properly interpolates the equation of state in the
         high
117  %temperature regime with Martins Data.
118  %created with fEinterphigh.m
119  %Reference John's mathematica file on 9/5/2019
120  load('MartinDataHigh.mat');
121  % Feqinterp2 is the equation of state for the extended
         temperature (up to
122  % 450)
123  % TTFinterp is the extended temperature up to 450
124  % specific heat per particle as a function of position c_V(z)
         and c_p(z)
125  Cvall = interp1(TTfinterp,CVExtend,TTfgas,'spline');
126  Cpall = interp1(TTfinterp,CPExtend,TTfgas,'spline');
```

```matlab
127  % the specific heat per particle is calulated with
128  % CVExtend = (3/5)*gradient(Feqinterp2,TTfinterp);
129  % CPExtend = (3/5)*gradient(Feqinterp2,TTfinterp).*Feqinterp2...
130  %  ./(Feqinterp2-(2/5).*TTfinterp.*gradient(Feqinterp2,TTfinterp)
         );
131
132  %% Interpolate f_E(\theta) for the TTf
133  FeqfordU0dz = interp1(FEtheta(:,1),FEtheta(:,2),TTfgas,'spline')
         ;
134  GradientFeqforDU0dz = gradient(FeqfordU0dz,TTfgas);
135
136  % %For plots of c_0/v_F vs density
137  % nn0 = 0.1:.01:1;
138  % theta05 = 0.5*nn0.^(-2/3);
139  % theta10 = 1*nn0.^(-2/3);
140  % theta02 = 0.2*nn0.^(-2/3);
141  % Feq05 = interp1(TTfinterp,Feqinterp2,theta05,'spline');
142  % Feq10 = interp1(TTfinterp,Feqinterp2,theta10,'spline');
143  % Feq02 = interp1(TTfinterp,Feqinterp2,theta02,'spline');
144  % cvf05 = nn0.^(1/3).*sqrt(Feq05/3);
145  % cvf10 = nn0.^(1/3).*sqrt(Feq10/3);
146  % cvf02 = nn0.^(1/3).*sqrt(Feq02/3);
147  % close all
148  % hold on
```

```matlab
149  % plot(nn0,cvf05,'k','LineWidth',0.2)

150  % plot(nn0,cvf10,'r','LineWidth',0.2)

151  % plot(nn0,cvf02,'b','LineWidth',0.2)

152  % ylabel('$$\frac{c_0}{v_{F0}}$$','Interpreter','LaTeX')

153  % xlabel('$$n/n_0$$','Interpreter','LaTeX')

154

155  % calculate dU0dz from the equation of state and \theta(z)
156  dU0dzsmooth = (-2/3)*(FeqfordU0dz-(2/5)*TTfgas.*
         GradientFeqforDU0dz).*(gradient(n0fit,dz)./(n0offset.^(1/3)))
         ;

157

158  %% Integrate dU0dz to obtain U_0(z)
159  U0 = zeros(1,size(dU0dzsmooth,2));
160  for i=1:size(dU0dzsmooth,2)
161      U0(i)= sum(dU0dzsmooth(1:i));
162  end

163

164  % for plotting the potential U_0(z)
165  % close all
166  % plot(U0,'LineWidth',2)
167  % xlabel('$$z [\mu m]$$','Interpreter','LaTeX')
168  % ylabel('$$\tilde{U}_{0}$$','Interpreter','LaTeX')

169

170
```

```matlab
171  %%% dU
172
173  % width of edge of hyperbolic tanh
174  sig = 4;
175  % starting point adjusted to start at edge of box for each
         wavelength
176  dUstart = n0left−dUoffset;
177
178  % dUenvelope leading edge
179  dUenvelope = (1/2)*(1+tanh(−(Z−v*T−dUstart)/sig));
180  % z_0 for the gaussian illuminating beam
181  Guassmiddle = 1.4175*n0left+(n0right−n0left)/2;
182  % width of the gaussian illuminating beam
183  SigGauss = 113;
184  % parameter adjust the static envelope to the location of the
         box potential
185  % wall
186  nudgeleft = 7;
187  % gaussian profile and LHS of the envelope function
188  % put together here because they are both static
189  dUGaussandWall = exp(−(z−Guassmiddle).^2/SigGauss^2).*(.5*(1+
         tanh((z−n0left−nudgeleft)/sig)));
190
191  % sinusoidal part of the \delta U(z,t) function
```

```matlab
192  dUcos = (1-eps*cos(-num*(Z-v*T-dUstart-lam)));
193
194  % \delta U(z,t)
195  dU = dUenvelope.*dUcos.*dUGaussandWall;
196
197
198  %%% Estimate for \delta U_0 from the measured intensity of the
         green beam
199  U0GB=(2*.4/(pi*(SigGauss*10^-6)^2)/10^6)*.0627*(10^-6*kB);
200  % we use 8% modulation of the GB on the DMD
201  % dUmult accounts for loss and imperfect imaging system
202  dU0 = dUmult*0.08*U0GB;
203  % adjust the data to match the units hydrodynamic model used for
         the
204  % simulation
205  vdndataq = vdndataq/(dU0/EF);
206  vdnErrdataq = vdnErrdataq/(dU0/EF);
207
208  %%% Initialize for computational speed
209  dn = zeros(length(t),length(z));
210  dndot = zeros(length(t),length(z));
211  dUdz = zeros(length(t)-1,length(z));
212  Vforce = zeros(length(t)-1,length(z));
213  dU0dz = zeros(length(t)-1,length(z));
```

99

```matlab
214  V0force = zeros(length(t)-1,length(z));
215  ds = zeros(length(t),length(z));
216
217
218
219  %% Nmerical integration of hydrodynamic model
220  for l=3:(length(t))
221
222      % eq 1
223      % time derivative of change in density
224      dndot(l-1,:) = 1/dt*(dn(l-1,:)-dn(l-2,:));
225
226      % spatial gradients
227      dndotdz = gradient(dndot(l-1,:),dz);
228      d2ndotdz2 = gradient(dndotdz,dz);
229
230      dndz = gradient(dn(l-1,:),dz);
231      d2ndz2 = gradient(dndz,dz);
232
233      dsdz = gradient(ds(l-1,:),dz);
234      d2sdz2 = gradient(dsdz,dz);
235
236      % spatial gradient of \delta U(z,t)
237      dUdz(l-1,:) = gradient(dU(l-1,:),dz);
```

```matlab
238
239        % force  term  from  \delta  U(z,t)
240        Vforce(l-1,:) = 0.5*vF^2*gradient(n0fit.*dUdz(l-1,:),dz);
241        % gradient  of  box  potential  calculated  above
242        dU0dz(l-1,:) = dU0dzsmooth;
243        % force  term  from  U_0(z)
244        V0force(l-1,:) = 0.5*vF^2*gradient(dn(l-1,:).*dU0dz(l-1,:),
              dz);
245
246        % eq2
247        % change  in  density  from  a  perturbative  potential  in  a  box
248        dn(l,:) = 2*dn(l-1,:)-dn(l-2,:)+dt^2*(c0^2*d2ndz2+c0^2*
              d2sdz2+4/3*a0*d2ndotdz2+Vforce(l-1,:)+V0force(l-1,:));
249        % supress  density  change  in  regions  of  low  density
250        supress = dn(l,:);
251        oldsupress = dn(l-1,:);
252        supress(n0fit<.0005) = oldsupress(n0fit<.0005);
253        dn(l,:) = supress;
254
255        % eq3
256        % change  in  entropy  equation
257        ds(l,:) = ds(l-1,:)+dt*(C1*d2sdz2+C1*C2*d2ndz2);
258  end
259
```

```matlab
260 %% Time Correction
261 % make sure the simulation ends on the same time as the data
262 % this algorithm automatically matches end time of the
       simulation to match
263 % the data
264 trange = round(.5/freq/dt);
265 trangestep = round(.025/freq/dt);
266 timecorrection = numtim-trange:trangestep:numtim;
267 Chi2time = zeros(1,length(timecorrection));
268 for m=1:length(timecorrection)
269 A=dn(timecorrection(m),:);
270 B=vdndataq;
271 Chi2time(m) = sum((A-B).^2);
272 end
273 [Valuemin,Indexmin] = min(Chi2time);
274 % t index of best fit
275 tindexbestfit = timecorrection(Indexmin);
276 % time of best fit
277 t(timecorrection(Indexmin));
278
279 %% Chi^2 calculation
280 leftside = round(n0left);
281 rightside = round(n0right);
282 % A is the simulation
```

```matlab
283  A=dn( tindexbestfit ,:) *(dU0/EF);
284  % B is the data
285  B=vdndataq *(dU0/EF);
286  % Error bars in the data
287  BErr = vdnErrdataq *(dU0/EF);
288  % Calculate Chi^2 for the data and simulation
289  Chi2 = sum((A-B).^2);
290  %Chi2reduced used for calculating the actual reduced Chi2
291
292
293  %for KappaT measurement change the region of interest to the
           middle where
294  % the data and model are sensetive to the parameter.
295  % for all other parameters , go to edge of the rs = rightside.
296  % ls and rs are for calculating chi^2 away from the less dense
           edge of the
297  % sample
298  ls = leftside +30;
299  rs = rightside -10;
300  % Chi2reduced = sum((A(ls:rs)-B(ls:rs)).^2./BErr(ls:rs).^2)/(
           length(A(ls:rs))-1);
301  Chi2reduced = sum((A(ls:rs)-B(ls:rs)).^2./BErr(ls:rs).^2);
302  end
```

Appendix B

DMD control

B.1   DMD MATLAB code
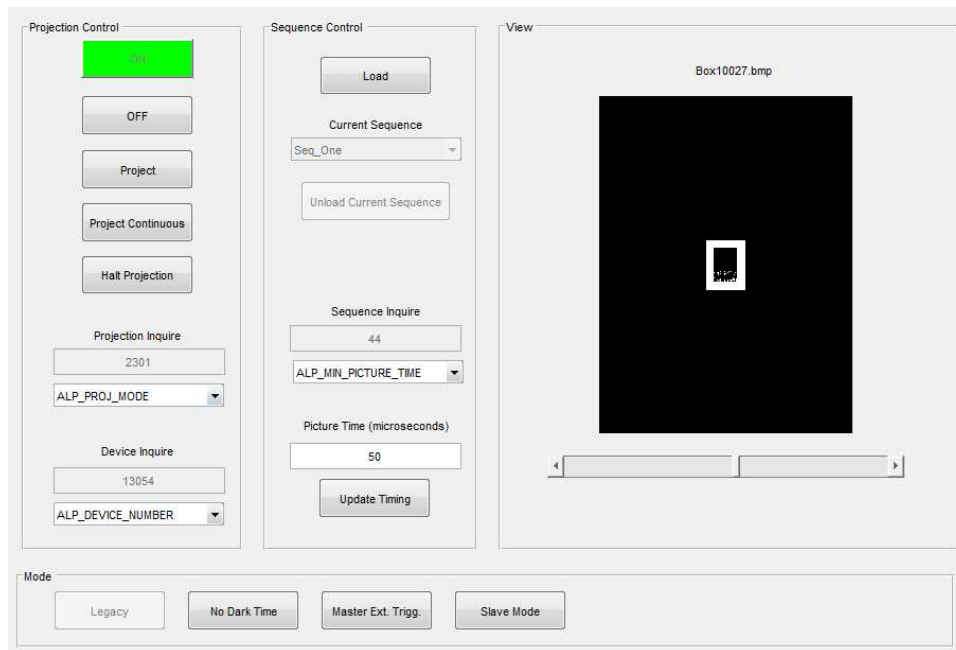
Figure B.1:   Graphical user interface for controlling the DMD.

The MATLAB code used to communicate with the device and control the board. The graphic user interface shown in Fig. B.1 allows the user to probe all parameters and status of the DMD, load images to the memory of the control board, and set up the device to accept triggers from a pulse generator that controls the equipment in the lab.

```matlab
function varargout = ALP_GUI_2(varargin)
% ALP_GUI_2 MATLAB code for ALP_GUI_2.fig
%        ALP_GUI_2, by itself, creates a new ALP_GUI_2 or raises
    the existing
%        singleton*.
%
%        H = ALP_GUI_2 returns the handle to a new ALP_GUI_2 or
    the handle to
%        the existing singleton*.
%
%        ALP_GUI_2('CALLBACK',hObject,eventData,handles,...) calls
     the local
%        function named CALLBACK in ALP_GUI_2.M with the given
    input arguments.
%
%        ALP_GUI_2('Property','Value',...) creates a new ALP_GUI_2
     or raises the
%        existing singleton*.  Starting from the left, property
    value pairs are
%        applied to the GUI before ALP_GUI_2_OpeningFcn gets
```

```
       called.   An
15  %        unrecognized  property  name  or  invalid  value  makes
    property  application
16  %        stop.   All  inputs  are  passed  to  ALP_GUI_2_OpeningFcn  via
    varargin.
17  %
18  %        *See GUI Options on GUIDE's Tools menu.   Choose "GUI
    allows  only  one
19  %        instance  to  run  (singleton)".
20  %
21  % See also: GUIDE, GUIDATA, GUIHANDLES
22
23  % Edit  the  above  text  to  modify  the  response  to  help  ALP_GUI_2
24
25  % Last  Modified by GUIDE v2.5  25-Aug-2016  12:19:51
26
27  % Begin  initialization  code - DO NOT EDIT
28  gui_Singleton = 1;
29  gui_State = struct('gui_Name',          mfilename, ...
30                     'gui_Singleton',   gui_Singleton, ...
31                     'gui_OpeningFcn',  @ALP_GUI_2_OpeningFcn, ...
32                     'gui_OutputFcn',   @ALP_GUI_2_OutputFcn, ...
33                     'gui_LayoutFcn',   []  , ...
34                     'gui_Callback',    []);
```

```matlab
35   if  nargin && ischar(varargin{1})
36       gui_State.gui_Callback = str2func(varargin{1});
37   end
38
39   if  nargout
40       [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:})
            ;
41   else
42       gui_mainfcn(gui_State, varargin{:});
43   end
44   % End initialization code - DO NOT EDIT
45
46
47   % --- Executes just before ALP_GUI_2 is made visible.
48   function ALP_GUI_2_OpeningFcn(hObject, eventdata, handles,
        varargin)
49   % This function has no output args, see OutputFcn.
50   % hObject    handle to figure
51   % eventdata  reserved - to be defined in a future version of
        MATLAB
52   % handles    structure with handles and user data (see GUIDATA)
53   % varargin   command line arguments to ALP_GUI_2 (see VARARGIN)
54   % load library
55
```

```matlab
56  if ~libisloaded('alp4395')
57      hfile = 'alp.h';
58      loadlibrary('alp4395', hfile)
59  end
60
61  disp('library loaded');
62  % default device Id
63  ALP_DEFAULT = 0;
64  handles.DeviceId = ALP_DEFAULT;
65
66  % default parameters for AlpSeqTiming
67  handles.IlluminateTime = ALP_DEFAULT;
68  handles.PictureTime = 33334;
69  handles.SynchDelay = ALP_DEFAULT;
70  handles.SynchPulseWidth = ALP_DEFAULT;
71  handles.TriggerInDelay = ALP_DEFAULT;
72
73  % set picture time text box to default
74  set(handles.Picture_Time,'String',num2str(handles.PictureTime));
75
76  % default plot
77  columns = 1024;
78  rows = 768;
79  handles.BlackImage = zeros(columns, rows);
```

```matlab
80  imshow ( handles . BlackImage , [ 0 , 1 ] ) ;

81

82  % Choose default command line output for ALP_GUI_2
83  handles . output = hObject ;

84

85  % Update handles structure
86  guidata ( hObject , handles ) ;

87

88  % UIWAIT makes ALP_GUI_2 wait for user response ( see UIRESUME)
89  % uiwait ( handles . figure1 ) ;

90

91  % ---- Outputs from this function are returned to the command
        line .
92  function varargout = ALP_GUI_2_OutputFcn ( hObject , eventdata ,
        handles )
93  % varargout   cell array for returning output args ( see VARARGOUT
        ) ;
94  % hObject     handle to figure
95  % eventdata   reserved - to be defined in a future version of
        MATLAB
96  % handles     structure with handles and user data ( see GUIDATA)

97

98  % Get default command line output from handles structure
99  varargout {1} = handles . output ;
```

```matlab
100
101  % ——— Executes on slider movement.
102  function slider1_Callback(hObject, eventdata, handles)
103  % hObject      handle to slider1 (see GCBO)
104  % eventdata    reserved − to be defined in a future version of
         MATLAB
105  % handles      structure with handles and user data (see GUIDATA)
106
107  % % first we need the handles structure which we can get from
         hObject
108  %  handles = guidata(hObject);
109
110  % get the slider value and convert it to the nearest integer
         that is less
111  % than this value
112  newVal = round(get(hObject,'Value'));
113
114   % set the slider value to this integer which will be in the set
         {1,2...}
115  set(hObject,'Value',newVal);
116
117
118   % now only do something in response to the slider movement if
         the
```

```matlab
119   % new value is different from the last slider value
120   if newVal ~= handles.lastSliderVal
121       % it is different, so we have moved up or down from the
              previous integer
122       % save the new value
123       handles.lastSliderVal = newVal;
124       guidata(hObject, handles);
125   end
126
127   % set the axes name to be the first file name
128   set(handles.ImageName, 'String', handles.D(get(hObject, 'Value')).
         name)
129
130   % plot the first image
131   imshow(handles.image(:,:, get(hObject, 'Value')), [0,1])
132
133    % save the new value
134   %        handles.lastSliderVal = newVal;
135       guidata(hObject, handles);
136
137   % Hints: get(hObject, 'Value') returns position of slider
138   %         get(hObject, 'Min') and get(hObject, 'Max') to determine
         range of slider
139
```

```matlab
140  % ――― Executes during object creation , after setting all
         properties .
141  function slider1_CreateFcn(hObject, eventdata , handles )
142  % hObject     handle to slider1 ( see GCBO)
143  % eventdata   reserved − to be defined in a future version of
         MATLAB
144  % handles     empty − handles not created until after all
         CreateFcns called
145
146  % Hint : slider controls usually have a light gray background .
147  if isequal(get(hObject,'BackgroundColor'), get(0,'
         defaultUicontrolBackgroundColor'))
148      set(hObject,'BackgroundColor',[.9 .9 .9]) ;
149  end
150
151  % ――― Executes on button press in Allocate .
152  function Allocate_Callback(hObject, eventdata , handles )
153  % hObject     handle to Allocate ( see GCBO)
154  % eventdata   reserved − to be defined in a future version of
         MATLAB
155  % handles     structure with handles and user data ( see GUIDATA)
156
157  handles . DeviceId = Start_DMD() ;
158
```

```matlab
159  % enable buttons
160  set(handles.Load, 'Enable', 'on');
161  set(handles.Projection_Inquire_Pop, 'Enable', 'on');
162  set(handles.Device_Inquire_Pop, 'Enable', 'on');
163  set(handles.Projection_Inquire_Pop, 'Enable', 'on');
164  set(handles.Picture_Time, 'Enable', 'on');
165
166  % change colors
167  set(handles.Free, 'BackgroundColor', [.94,.94,.94]);
168  set(handles.Allocate, 'BackgroundColor', 'green');
169  set(handles.Allocate, 'Enable', 'off');
170
171  % update handles
172  guidata(hObject, handles);
173
174
175  % ——— Executes on button press in Free.
176  function Free_Callback(hObject, eventdata, handles)
177  % hObject      handle to Free (see GCBO)
178  % eventdata   reserved − to be defined in a future version of
        MATLAB
179  % handles      structure with handles and user data (see GUIDATA)
180
181  disp(handles.DeviceId);
```

```matlab
182  Halt_DMD( handles.DeviceId );

183  Free_DMD( handles.DeviceId );

184

185  % enable buttons

186  set( handles.Load, 'Enable', 'off' );

187  set( handles.Projection_Inquire_Pop, 'Enable', 'off' );

188  set( handles.Device_Inquire_Pop, 'Enable', 'off' );

189  set( handles.Picture_Time, 'Enable', 'off' );

190  set( handles.slider1, 'Enable', 'off' );

191  set( handles.Sequence_Inquire_Pop, 'Enable', 'off' );

192  set( handles.Project, 'Enable', 'off' );

193  set( handles.Project_Continuous, 'Enable', 'off' );

194  set( handles.No_Dark_Time, 'Enable', 'off' );

195  set( handles.Master_External_Trigger, 'Enable', 'off' );

196  set( handles.Update_Timing, 'Enable', 'off' );

197  set( handles.Slave_Mode, 'Enable', 'off' );

198

199  % change colors

200  set( handles.Free, 'BackgroundColor', 'red' );

201  set( handles.Allocate, 'BackgroundColor', [.94,.94,.94] );

202  set( handles.Allocate, 'Enable', 'on' );

203

204  % turn plot black because no images are loaded

205  imshow( handles.BlackImage, [0,1] )
```

```matlab
206  set(handles.ImageName, 'String', 'No Sequence Loaded')
207
208  % ——— Executes on button press in Project.
209  function Project_Callback(hObject, eventdata, handles)
210  % hObject    handle to Project (see GCBO)
211  % eventdata  reserved − to be defined in a future version of
         MATLAB
212  % handles    structure with handles and user data (see GUIDATA)
213
214  Proj_DMD(handles.DeviceId, handles.SequenceId)
215
216  % ——— Executes on button press in Project_Continuous.
217  function Project_Continuous_Callback(hObject, eventdata, handles
         )
218  % hObject    handle to Project_Continuous (see GCBO)
219  % eventdata  reserved − to be defined in a future version of
         MATLAB
220  % handles    structure with handles and user data (see GUIDATA)
221
222  Proj_Cont_DMD(handles.DeviceId, handles.SequenceId)
223
224  % ——— Executes on button press in Halt.
225  function Halt_Callback(hObject, eventdata, handles)
226  % hObject    handle to Halt (see GCBO)
```

115

```matlab
227  % eventdata   reserved − to be defined in a future version of
         MATLAB
228  % handles      structure with handles and user data (see GUIDATA)
229
230  Halt_DMD(handles.DeviceId);
231
232  % −−− Executes on selection change in Projection_Inquire_Pop.
233  function Projection_Inquire_Pop_Callback(hObject, eventdata,
         handles)
234  % hObject      handle to Projection_Inquire_Pop (see GCBO)
235  % eventdata   reserved − to be defined in a future version of
         MATLAB
236  % handles      structure with handles and user data (see GUIDATA)
237
238  val = get(hObject, 'Value');
239  str = get(hObject, 'String');
240
241  %This part will inquire the ALP based on the command from the
         pop up menu
242  [ret, Inq] = AlpProjInquire(handles.DeviceId, Projection_Inquire
         (str{val}));
243
244  % success of the inquiry
245  Return_Values(ret);
```

```matlab
246
247 % result of the inquiry displayed in a text box
248 set(handles.Projection_Inquire_Display, 'String', num2str(Inq))
249
250
251 % Hints: contents = cellstr(get(hObject,'String')) returns
        Projection_Inquire_Pop contents as cell array
252 %        contents{get(hObject,'Value')} returns selected item
        from Projection_Inquire_Pop
253
254
255 % ---- Executes during object creation, after setting all
        properties.
256 function Projection_Inquire_Pop_CreateFcn(hObject, eventdata,
        handles)
257 % hObject     handle to Projection_Inquire_Pop (see GCBO)
258 % eventdata   reserved - to be defined in a future version of
        MATLAB
259 % handles     empty - handles not created until after all
        CreateFcns called
260
261 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
        defaultUicontrolBackgroundColor'))
262     set(hObject,'BackgroundColor','white');
```

```matlab
263  end
264
265  function Projection_Inquire_Display_Callback(hObject, eventdata,
         handles)
266  % hObject     handle to Projection_Inquire_Display (see GCBO)
267  % eventdata   reserved − to be defined in a future version of
         MATLAB
268  % handles     structure with handles and user data (see GUIDATA)
269
270
271  % −−− Executes during object creation, after setting all
         properties.
272  function Projection_Inquire_Display_CreateFcn(hObject, eventdata
         , handles)
273  % hObject     handle to Projection_Inquire_Display (see GCBO)
274  % eventdata   reserved − to be defined in a future version of
         MATLAB
275  % handles     empty − handles not created until after all
         CreateFcns called
276
277  if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
         defaultUicontrolBackgroundColor'))
278      set(hObject,'BackgroundColor','white');
279  end
```

```matlab
280
281
282   % ——— Executes on button press in Load.
283   function Load_Callback(hObject, eventdata, handles)
284   % hObject      handle to Load (see GCBO)
285   % eventdata   reserved - to be defined in a future version of
          MATLAB
286   % handles      structure with handles and user data (see GUIDATA)
287
288   % load all the files from a folder into the dmd
289   [handles.SequenceId, handles.image, handles.D, handles.PicNum] =
          Load_DMD(handles.DeviceId, handles.PictureTime);
290
291   % plot the first image
292   imshow(handles.image(:,:,1), [0,1])
293
294   % set the axes name to be the first file name
295   set(handles.ImageName,'String',handles.D(1).name)
296
297   % set the slider range and step size
298   if handles.PicNum > 1
299    numSteps = handles.PicNum;
300    set(handles.slider1, 'Min', 1);
301    set(handles.slider1, 'Max', numSteps);
```

```matlab
302    set ( handles . slider1 , 'Value ', 1);
303    set ( handles . slider1 , 'SliderStep ', [1/( numSteps −1) , 1/(
           numSteps −1) ]) ;
304  end
305
306  % save the current/last slider value
307  handles . lastSliderVal = get ( handles . slider1 ,'Value ') ;
308
309  % enable slider and other buttons after images are loaded
310  set ( handles . slider1 , 'Enable ', 'on ') ;
311  set ( handles . Sequence_Inquire_Pop , 'Enable ', 'on ') ;
312  set ( handles . Project , 'Enable ', 'on ') ;
313  set ( handles . Project_Continuous , 'Enable ', 'on ') ;
314  set ( handles . Halt , 'Enable ', 'on ') ;
315  set ( handles . No_Dark_Time , 'Enable ', 'on ') ;
316  set ( handles . Master_External_Trigger , 'Enable ', 'on ') ;
317  set ( handles . Update_Timing , 'Enable ', 'on ') ;
318  set ( handles . Slave_Mode , 'Enable ', 'on ') ;
319
320  %We need to updata the handles data structure once again
321  guidata ( hObject , handles ) ;
322
323
324  % —— Executes on button press in pushbutton7 .
```

```matlab
325 function pushbutton7_Callback(hObject, eventdata, handles)
326 % hObject      handle to pushbutton7 (see GCBO)
327 % eventdata   reserved - to be defined in a future version of
        MATLAB
328 % handles     structure with handles and user data (see GUIDATA)
329
330 % ---- Executes on selection change in Sequence_Inquire_Pop.
331 function Sequence_Inquire_Pop_Callback(hObject, eventdata,
        handles)
332 % hObject      handle to Sequence_Inquire_Pop (see GCBO)
333 % eventdata   reserved - to be defined in a future version of
        MATLAB
334 % handles     structure with handles and user data (see GUIDATA)
335
336 val = get(hObject, 'Value');
337 str = get(hObject, 'String');
338
339 %This part will inquire the ALP based on the command from the
        pop up menu
340 [ret, Inq] = AlpSeqInquire(handles.DeviceId, handles.SequenceId,
        Sequence_Inquire(str{val}));
341
342 % success of the inquiry
343 Return_Values(ret);
```

```
344
345 % result of the inquiry displayed in a text box
346 set(handles.Sequence_Inquire_Display, 'String', num2str(Inq))
347
348 % ――― Executes during object creation, after setting all
        properties.
349 function Sequence_Inquire_Pop_CreateFcn(hObject, eventdata,
      handles)
350 % hObject     handle to Sequence_Inquire_Pop (see GCBO)
351 % eventdata   reserved − to be defined in a future version of
        MATLAB
352 % handles     empty − handles not created until after all
        CreateFcns called
353
354 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
      defaultUicontrolBackgroundColor'))
355     set(hObject,'BackgroundColor','white');
356 end
357
358
359 function Sequence_Inquire_Display_Callback(hObject, eventdata,
      handles)
360 % hObject     handle to Sequence_Inquire_Display (see GCBO)
361 % eventdata   reserved − to be defined in a future version of
```

```matlab
            MATLAB
362 % handles     structure with handles and user data (see GUIDATA)
363
364
365 % ——— Executes during object creation , after setting all
        properties .
366 function Sequence_Inquire_Display_CreateFcn(hObject, eventdata,
        handles)
367 % hObject      handle to Sequence_Inquire_Display (see GCBO)
368 % eventdata    reserved − to be defined in a future version of
        MATLAB
369 % handles      empty − handles not created until after all
        CreateFcns called
370
371 % Hint: edit controls usually have a white background on Windows
        .
372 %        See ISPC and COMPUTER.
373 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
        defaultUicontrolBackgroundColor'))
374     set(hObject,'BackgroundColor','white');
375 end
376
377
378 % ——— Executes on selection change in Current_Sequence_Pop.
```

```matlab
379  function Current_Sequence_Pop_Callback(hObject, eventdata,
         handles)
380  % hObject      handle to Current_Sequence_Pop (see GCBO)
381  % eventdata    reserved - to be defined in a future version of
         MATLAB
382  % handles      structure with handles and user data (see GUIDATA)
383
384  % --- Executes during object creation, after setting all
         properties.
385  function Current_Sequence_Pop_CreateFcn(hObject, eventdata,
         handles)
386  % hObject      handle to Current_Sequence_Pop (see GCBO)
387  % eventdata    reserved - to be defined in a future version of
         MATLAB
388  % handles      empty - handles not created until after all
         CreateFcns called
389
390  if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
         defaultUicontrolBackgroundColor'))
391      set(hObject,'BackgroundColor','white');
392  end
393
394
395
```

```matlab
396  function Picture_Time_Callback(hObject, eventdata, handles)
397  % hObject      handle to Picture_Time (see GCBO)
398  % eventdata    reserved - to be defined in a future version of
         MATLAB
399  % handles      structure with handles and user data (see GUIDATA)
400
401
402  % get the contents of Picture time editable text
403  PictureTime = uint32(str2num(get(hObject,'String')));
404
405  % picture time must be less than 10^7 microseconds
406  if PictureTime > 10000000
407      PictureTime = 10000000;
408  end
409
410  handles.PictureTime = PictureTime;
411
412  set(hObject,'String',num2str(PictureTime));
413
414
415  guidata(hObject, handles);
416
417  % ---- Executes during object creation, after setting all
         properties.
```

```matlab
418  function Picture_Time_CreateFcn(hObject, eventdata, handles)
419  % hObject      handle to Picture_Time (see GCBO)
420  % eventdata    reserved - to be defined in a future version of
         MATLAB
421  % handles      empty - handles not created until after all
         CreateFcns called
422
423  if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
         defaultUicontrolBackgroundColor'))
424      set(hObject,'BackgroundColor','white');
425  end
426
427
428  % --- Executes on selection change in Device_Inquire_Pop.
429  function Device_Inquire_Pop_Callback(hObject, eventdata, handles
         )
430  % hObject      handle to Device_Inquire_Pop (see GCBO)
431  % eventdata    reserved - to be defined in a future version of
         MATLAB
432  % handles      structure with handles and user data (see GUIDATA)
433
434  val = get(hObject, 'Value');
435  str = get(hObject, 'String');
436
```

```matlab
437 %This part will inquire the ALP based on the command from the
        pop up menu
438 [ret, Inq] = AlpDevInquire(handles.DeviceId, Device_Inquire(str{
        val}));
439
440
441 % success of the inquiry
442 Return_Values(ret);
443
444 % result of the inquiry displayed in a text box
445 set(handles.Device_Inquire_Display, 'String', num2str(Inq))
446
447 % ——— Executes during object creation, after setting all
        properties.
448 function Device_Inquire_Pop_CreateFcn(hObject, eventdata,
        handles)
449 % hObject     handle to Device_Inquire_Pop (see GCBO)
450 % eventdata   reserved − to be defined in a future version of
        MATLAB
451 % handles     empty − handles not created until after all
        CreateFcns called
452
453 % Hint: popupmenu controls usually have a white background on
        Windows.
```

```matlab
454  %           See ISPC and COMPUTER.
455  if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
         defaultUicontrolBackgroundColor'))
456      set(hObject,'BackgroundColor','white');
457  end
458
459
460
461  function Device_Inquire_Display_Callback(hObject, eventdata,
         handles)
462  % hObject      handle to Device_Inquire_Display (see GCBO)
463  % eventdata    reserved - to be defined in a future version of
         MATLAB
464  % handles      structure with handles and user data (see GUIDATA)
465
466
467  % --- Executes during object creation, after setting all
         properties.
468  function Device_Inquire_Display_CreateFcn(hObject, eventdata,
         handles)
469  % hObject      handle to Device_Inquire_Display (see GCBO)
470  % eventdata    reserved - to be defined in a future version of
         MATLAB
471  % handles      empty - handles not created until after all
```

```matlab
        CreateFcns called
472
473 % Hint: edit controls usually have a white background on Windows
        .
474 %        See ISPC and COMPUTER.
475 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
        defaultUicontrolBackgroundColor'))
476     set(hObject,'BackgroundColor','white');
477 end
478

479

480 % ——— Executes on button press in No_Dark_Time.
481 function No_Dark_Time_Callback(hObject, eventdata, handles)
482 % hObject      handle to No_Dark_Time (see GCBO)
483 % eventdata    reserved − to be defined in a future version of
        MATLAB
484 % handles      structure with handles and user data (see GUIDATA)
485

486 Fast_Switch_DMD(handles.DeviceId, handles.SequenceId, handles.
        PictureTime)
487

488 External_Trigger_Frame_Transition(handles.DeviceId, handles.
        PictureTime, handles.SequenceId);
489
```

```matlab
490
491  % ——— Executes on button press in Master_External_Trigger.
492  function Master_External_Trigger_Callback(hObject, eventdata,
         handles)
493  % hObject      handle to Master_External_Trigger (see GCBO)
494  % eventdata    reserved − to be defined in a future version of
         MATLAB
495  % handles      structure with handles and user data (see GUIDATA)
496
497
498  External_Trigger_Frame_Transition(handles.DeviceId, handles.
         PictureTime, handles.SequenceId);
499
500  % ——— Executes on button press in Slave_Mode.
501  function Slave_Mode_Callback(hObject, eventdata, handles)
502  % hObject      handle to Slave_Mode (see GCBO)
503  % eventdata    reserved − to be defined in a future version of
         MATLAB
504  % handles      structure with handles and user data (see GUIDATA)
505
506  External_Trigger_Mode(handles.DeviceId, handles.PictureTime,
         handles.SequenceId);
507
508
```

```matlab
509  % ——— Executes on button press in pushbutton11.
510  function pushbutton11_Callback(hObject, eventdata, handles)
511  % hObject      handle to pushbutton11 (see GCBO)
512  % eventdata    reserved − to be defined in a future version of
         MATLAB
513  % handles      structure with handles and user data (see GUIDATA)
514
515
516  % ——— Executes on button press in pushbutton12.
517  function pushbutton12_Callback(hObject, eventdata, handles)
518  % hObject      handle to pushbutton12 (see GCBO)
519  % eventdata    reserved − to be defined in a future version of
         MATLAB
520  % handles      structure with handles and user data (see GUIDATA)
521
522
523  % ——— Executes when user attempts to close figure1.
524  function figure1_CloseRequestFcn(hObject, eventdata, handles)
525  % hObject      handle to figure1 (see GCBO)
526  % eventdata    reserved − to be defined in a future version of
         MATLAB
527  % handles      structure with handles and user data (see GUIDATA)
528
529
```

```matlab
530  % unload library
531
532  unloadlibrary alp4395
533  disp('library unloaded');
534
535
536  % Hint: delete(hObject) closes the figure
537  delete(hObject);
538
539
540  % ——— Executes on button press in Update_Timing.
541  function Update_Timing_Callback(hObject, eventdata, handles)
542  % hObject     handle to Update_Timing (see GCBO)
543  % eventdata   reserved - to be defined in a future version of
         MATLAB
544  % handles     structure with handles and user data (see GUIDATA)
545
546  Update_Timing_DMD(handles.DeviceId, handles.SequenceId, handles.
         PictureTime)
```

B.2   DMD datasheet

Table B.1:   A datasheet for the two DMDs used in our apparatus.

Name	DLP7000	LightCrafter6500
Array Size	1024x768	1920x1080
Mirror Size	13.68 μm	7.56 μm
Mirror Width	1.4 cm	1.45 cm
Binary Rate	32,000Hz	9,523Hz
Micromirror tilt angle (degrees)	± 12	± 12
Damage threshold intensity	$20W/cm^2$ CW	$20W/cm^2$ CW
Micromirror array optical efficiency	68%	68%