The Dissertation Committee for John William Peterson
certifies that this is the approved version of the following dissertation:

Parallel Adaptive Finite Element Methods for Problems in Natural Convection

Committee:

Graham F. Carey, Supervisor

David S. Dolling

David B. Goldstein

Clint N. Dawson

Harry L. Swinney

Parallel Adaptive Finite Element Methods for Problems in Natural Convection

by

John William Peterson, B.S., M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2008

For Lorielle. Again.

Start a large, foolish project

—Rumi

Acknowledgments

I wish to thank all of my colleagues in the CFDLab, most notably Benjamin Kirk, Bill Barth, and Roy Stogner for many useful discussions at the white board and for moral support. I would also like to thank my advisor Graham F. Carey for his continued support and advice, and Professor Harry L. Swinney for his helpful suggestions in the final editing stages of the manuscript. This work was made possible in part by the NASA Graduate Student Researchers Program (GSRP) grant number NGT5-139, as well as the Thrust 2000 endowed fellowship program of the University of Texas College of Engineering.

Parallel Adaptive Finite Element Methods for Problems in Natural Convection

Publication No. _____

John William Peterson, Ph.D.
The University of Texas at Austin, 2008

Supervisor: Graham F. Carey

Numerical simulations of combined buoyant and surface tension driven flow, also known as Rayleigh-Bénard-Marangoni (RBM) convection are conducted for heated fluid layers of small aspect ratio (defined as the ratio of the horizontal extent of the domain divided by the depth of the fluid) in square cross-section containers. A particular non-dimensionalization of the governing equations is developed in which the aspect ratio of the domain appears as a continuous parameter. The simulations extend and enhance existing experimental studies of the RBM convection phenomenon by mapping continuous solution branches in aspect ratio and Marangoni number parameter space. Key implementation aspects of the development of the adaptive mesh refinement (AMR) library `libMesh` are discussed, and a series of simulations of the RBM problem with a stick-slip boundary condition demonstrate the suitability of AMR for computing these flows.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Contributions

In this initial chapter, we introduce the general problem class under investigation, list several contributions to different research areas which are contained in this work, and finally articulate the hypothesis which led to the present avenue of investigation. Rayleigh-Bénard-Marangoni (RBM) convection, the main physical phenomenon under investigation in this work, falls under the more general classification of spontaneous spatio-temporal pattern formation, an ubiquitous topic in the physical sciences.

The variety of physical systems which exhibit spatio-temporal pattern formation is truly remarkable. Well-known examples include the Taylor vortices which form between concentric, rotating cylinders [9, 56, 111], chemical traveling waves, and reaction-diffusion systems [116, 117, 142], and oscillating granular layers [119]. The study of such systems is important not only from the perspective of forming a general physical understanding of the underlying processes at work, but also because these otherwise unrelated phenomena can often be shown to share universal characteristics.

Mathematical modeling and numerical analysis are playing increasingly important roles in science and engineering, not only for basic design and anal-

ysis, but also as a means for studying fundamental processes and natural phenomena such as these. Numerical simulation is a key component of the present RBM convection research, as well as other studies conducted by the author during the dissertation preparation but which are not included here [150].

The remainder of this chapter is divided up in the following way: in §1.1 the specific contributions of the present research are listed and briefly discussed. Then, in §1.2, we give a brief introduction to the physical RBM system, from the earliest known works to more or less the present state of understanding. Finally, in §1.3 we detail the hypothesis which motivated this research.

1.1 Research Contributions

Here we list, in order of relative importance, the research contributions made by the present study.

Major Contributions

- A non-standard scaling (nondimensionalization) of the equations governing RBM convection was developed in which the aspect ratio of the fluid layer appears as an independent, continuous parameter. A robust, efficient Galerkin finite element formulation was developed and implemented to numerically solve the equations.

- A pseudo-arclength constraint for the RBM convection equations was developed and implemented in an efficient continuation algorithm suitable for tracing out branches of steady RBM solutions. Novel aspects include a "hybrid" technique for tracing non-trivial solution branches involving the use of both the unsteady and steady forms of the governing equations.

- Our numerical calculations verified and expanded the existing knowledge of spontaneous spatio-temporal pattern formation in RBM convection in small aspect ratio containers. In particular, we found that multiple stable, steady states exist for certain parameter values, and that common nonlinear-dynamics affects such as hysteresis, symmetry-breaking, and bifurcation are present in the RBM convection system.

- New solution states have been characterized that add to our understanding of the behavior and may influence future physical experimental work.

- Our numerical simulations repeated several of Koschmieder's convection patterns, however no patterns were observed below the theoretical minimum Ma of 79.6 predicted by Pearson (see §1.2 for details) and confirmed via careful experiment by Schatz [176]. This helps to verify Koschmieder's experimental uncertainty claims while still being in agreement with the universally-accepted linear stability theory.

- We computed stable configurations having 1 through 8 convection cells, and confirmed the general trend that the number of convection cells

scales roughly with container aspect ratio. We also confirmed the initial hypothesis (see §1.3) that these convection patterns would exist independently of the other problem parameters, so long as they were supercritical.

- The pseudo-arclength continuation techniques developed in this dissertation have been added as a general, reusable software module in the sophisticated, parallel, adaptive open-source finite element analysis library known as `LibMesh` [108].

- These numerical and algorithmic contributions apply more generally than to the present problem class of RBM convection: as a non-intrusive component of a general finite element library, they can also be applied to the growing collection of application codes already built using `LibMesh`'s application programming interface (API).

- The pseudo-arclength calculations were applied in high-resolution, multi-scale, multi-physics simulations over a wide parameter regime.

- The numerical simulations yield a more complete archive of spatio-temporal behavior both in terms of the standard parameters but also more significantly here with respect to the aspect ratio parametrization.

Other Specific Contributions

- We applied an explicit, second-order Adams-Bashforth predictor approach for use in our pseudo-arclength continuation scheme. Other features have been developed to enhance efficiency and reliability in the simulations, including adaptive arclength stepsize selection and the solution of bordered linear systems for use in large scale, parallel simulations of three-dimensional fluid flows.

- The standard shift-invert spectral transformation was employed along with the Arnoldi iteration for numerically computing the eigenvalues of steady solutions. This technique, combined with knowledge from unsteady solves provides a simple technique for assessing the stability of solution branches.

- The thesis is multidisciplinary, it involves contributions to C++ software frameworks/toolkits and the open source library LibMesh [108] which impact the finite element analysis community.

- Miscellaneous contributions to the `LibMesh` [108] library:

  - Implemented conical product (Jacobi-Gauss) quadrature rules for arbitrarily high-order integration over tetrahedra and triangles.

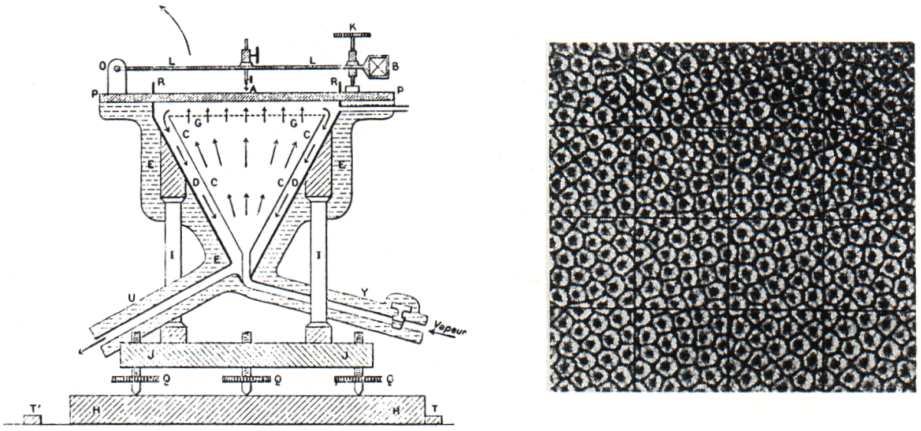
  - Implemented interface with the Delaunay triangulation library `Triangle`, written by J. R. Shewchuk.

  - Implemented restart ability on adaptively-refined grids.

- Wrote generic, extensible C++-style iterators for looping over different subsets (e.g. active, local) of elements and nodes contained within LibMesh's mesh data structure.
- Developed the `EigenTimeSolver` class to allow for the solution of generalized eigenvalue problems for systems of equations derived from the `FEMSystem` superclass.
- Wrote a timestep selector (again, applicable to all `FEMSystem` derived subclasses) which automatically halves the timestep upon failure of a Newton iteration and retries the solve.
- Wrote a mesh reader capability for two-dimensional meshes generated by Matlab's PDE toolkit.
- Wrote restart capability for GMV-formatted solution files.
- Developed an element-flagging capability for adaptive mesh refinement based on the mean and standard deviation of the population of element errors.
- Wrote an extension to the standard vector class implementing several common statistics-related procedures including a histogram function, in support of the statistical AMR element flagging routine.
- Wrote a mesh reader for `ExodusII` style meshes generated by Sandia National Labs' `cubit` mesh generator.
- Implemented a series of commonly-used cell quality metrics for quadrilateral, hexahedral, and triangular elements.

1.2 Rayleigh-Bénard-Marangoni Convection

Rayleigh-Bénard-Marangoni convection (sometimes referred to as Bénard-Marangoni convection or Rayleigh-Bénard convection) was first observed by Henri Bénard [31, 32] near the turn of the 20$^{\text{th}}$ century. His experimental apparatus was designed to study the behavior of a thin layer of spermaceti (i.e. whale oil, a solid substance at room temperature and a fluid above 46°C with very small thermal conductivity) on a metallic plate suspended above a boiling water bath at approximately 100°C. A schematic drawing of Bénard's test apparatus is given in Figure 1.1a.



(a) H. Bénard's apparatus, ca. 1900 (b) Bénard cells

Figure 1.1: Henri Bénard's original test apparatus (1.1a, reproduced in [111]) and one of his photographs (1.1b). A metallic plate of length L covered with a layer of spermaceti of depth $d \ll L$ is suspended over a container of approximately 100°C water vapor. The original experiment is described in [31].

In this experiment, the horizontal extent of the spermaceti surface was

much greater than its depth, and Bénard observed the ordered formation of hexagonal cells away from the boundaries of the container. He determined correctly that warm fluid was flowing up in the centers of the convection cells, and down the hexagonal boundaries. Furthermore, the height of the surface itself varied on the order of 1 micron over the cells themselves. Bénard (incorrectly) attributed the striking patterns shown in Figure 1.1b to buoyant forces acting in the bulk of the fluid. Rayleigh [164] and Jeffreys [97, 98] later confirmed, through linear stability analyses, that buoyant forces do in fact generate convective cells. Their findings, however, did not apply to the configurations originally investigated by Bénardbecause only buoyancy was considered.

Bénard's incorrect conclusion was not categorically dismissed for at least another fifty years, at which time it was shown experimentally by Block [35] in 1956, and theoretically by Pearson [148] and Nield [136] in 1958 and 1964 respectively, that changes in surface-tension, due to minor variations in the temperature of the fluid surface, were actually causing the convective cells noted in these thin layer problems. The ability for convection to exist due to variations in surface tension had been discovered and analyzed by C. Marangoni [121, 182] approximately one hundred years earlier. Since the 1950's, the linear [37, 52, 73, 123, 136] and nonlinear [58, 63, 64, 90, 165, 170, 174, 187] stability of such configurations has been well-analyzed, especially in the case of a layer of infinite horizontal extent (or, as we shall define it subsequently, container aspect ratio approaching infinity).

These works have led to a deeper understanding of the fundamen-

tal physics at work in the problem, and to categorizations of so-called critical and threshold parameters important to convection. Two important non-dimensional parameters are the Rayleigh number, Ra, and the Marangoni number, Ma. Their precise forms are determined by the choice of a particular scaling for the governing differential equations; this scaling is discussed in more detail in §2.6. The ratio $\frac{Ma}{Ra} > 1$ for surface tension effects to dominate (the situation originally investigated by Bénard) and all the configurations studied in the present work satisfy this condition. In fact, in nearly all of the cases subsequently reported we have $\frac{Ma}{Ra} \approx 3.07$, i.e. we are well within the regime where the surface-tension-driven instability dominates.

Holding other physical parameters constant, the ratio

$$\frac{Ma}{Ra} \propto \frac{1}{d^2} \tag{1.1}$$

where d is the depth of the layer. This implies that, all other things being equal, surface tension effects are dominant only in relatively thin fluid layers. In the related (but fundamentally different) case of a fluid layer bounded from above and below (surface tension inactive) the critical Rayleigh number for the onset of convection is approximately 1708. In the other limiting case, when there is a free surface present and buoyancy effects are negligible, Pearson [148] showed that the critical Marangoni number for the onset of convection is $Ma_c \approx 79.6$. Pearson assumed an infinite fluid layer with a perfectly insulating air layer above; finite aspect ratio effects and non-zero surface thermal fluxes can change the onset value significantly.

A large body of additional literature on large aspect ratio containers including the study of the long wavelength instability [195, 196, 202], the effect of adding surfactants to the fluid [65, 203, 204], and pattern transition dynamics [50, 75, 177] has been produced in the latter part of the 20^{th} century. The natural convection problem class in general has also proven to be ideal for benchmarking and validating large-scale computational fluid dynamics codes [27, 41, 47, 118, 173, 186], and thus has a prominent place in the arena of parallel, distributed high-performance computing and finite element analysis.

1.2.1 RBM Convection in Small Aspect Ratio Containers

In the present work, we focus on the opposite end of the RBM convection spectrum by considering containers of small aspect ratio. Here we define a "small aspect ratio" container as one having a horizontal length scale, L, on the order of 1 to 10 times the depth, d. We denote the aspect ratio by the non-dimensional quantity

$$\Gamma := \frac{L}{d} \quad \text{or} \quad \frac{\sqrt{A}}{d} \tag{1.2}$$

the second definition being more useful for arbitrarily-shaped containers having planform area A rather than a single obvious length scale L. In order to observe the effects of surface-tension-driven convection in small aspect ratio containers experimentally, one must consider extremely thin fluid layers, otherwise buoyancy effects tend to dominate. Another option for reducing

convection effects due to buoyancy is to perform the experiments in a microgravity environment such as the international space station [57], or even during the re-entry phase of a sounding rocket trajectory [180, 181].

Analytical linear stability analysis of RBM convection in finite aspect ratio containers is much more challenging than the infinite layer problem due to the effects of the no-slip sidewalls on pattern formation. Numerical linear stability analysis in three-dimensional small aspect ratio containers was also not feasible until about the mid-1990s due to the need to solve numerically a relatively large system of equations, as in the work of Dijkstra [73]. Pioneering experimental work in 1990 by Koschmieder and Prahl [113] on surface-tension-driven flows in extremely thin (small aspect ratio) layers produced diverse cell patterns implying complex interactions at different aspect ratios. Photographs from their experiments are reproduced here in Figs. 1.2 and 1.3, and show a number of different non-hexagonal convection cells, a phenomenon not typically observed in large aspect ratio cases.

Other, more recent experimental work [100, 120, 125, 140, 141, 146, 161–163] and numerical simulations [12, 33, 61, 62, 126, 149] in small aspect ratio containers have confirmed at least the qualitative aspects of several of Koschmieder's results. Quantitative comparisons with his work remain difficult due to the extremely sensitive nature of the experiments and the large experimental uncertainties present (e.g. in the working fluid properties, as discussed extensively by Schatz [175]).

Koschmieder himself states [113] that there is an "experimental uncer-

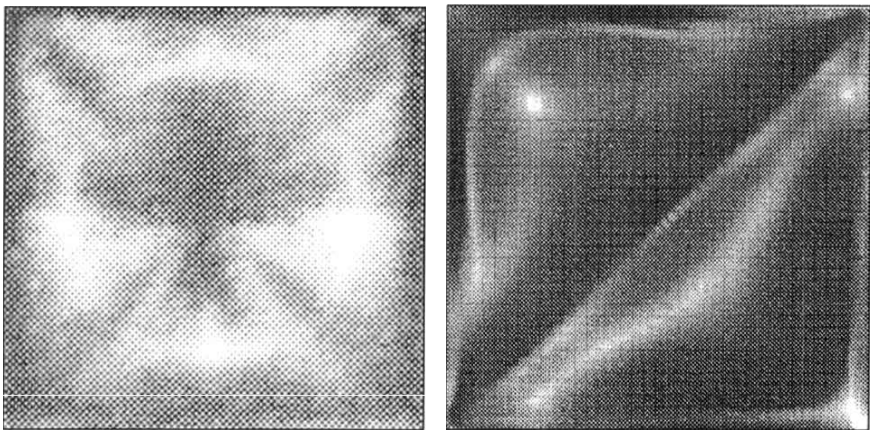
tainty of 15% in the determination of Ma." He attributes this uncertainty to several sources, such as the variation of the surface tension coefficient with respect variations in temperature ($\pm 5\%$), the viscosity and thermal diffusivity of the experimental fluid ($\pm 1\%$), and the temperature difference across the fluid layer ($\pm 5\%$). According to Schatz [175], it is likely that these uncertainties are actually even higher. In the present research, we can therefore compare our numerical results to Koschmieder and Prahl's experiments only qualitatively, bearing always in mind the issue of experimental uncertainty.

In what may be Koschmieder's most troubling finding, he reports obtaining asymptotic critical Ma values of around 60 (and below) in both his 1990 work [113] and in previous work with Biggerstaff [112] using a similar experimental setup. The theory developed by Pearson predicts a theoretical *minimum* Ma value of around 80 required for convection, clearly significantly higher than that reported by Koschmieder. In their 2002 paper, Medale and Cerisier [126] appear to actually report *quantitative* comparisons with Koschmieder's work. However, it is also possible (the paper is unclear on this point) they were simply showing Koschmieder's original results for reference and omitting the results of their own numerical calculations. The focus of the paper was not in fact the physics, but rather the numerical method being employed. In any case, neither an extensive parametric study for a given container type (e.g. squares) nor the possibility for other stable steady states, nor the difficulties in quantitative comparison with Koschmieder's work were discussed in any detail in the article.

There are also sources of experimental uncertainty beyond those in the fluid properties already mentioned. For example, it is possible that the common early practice of adding aluminum flakes to the fluid for visualization purposes had a surfactant effect (discussed in [203, 204]) on the experimental fluid. Other experimentalists (working with large aspect ratio containers) who used the shadowgraph method [177] to visualize the flow did not report subcritical convection. Another possible source of uncertainty arises in the method employed for heating the base of the experimental apparatus. Unless great care is taken, radial temperature gradients in the copper plate can induce early transitions to convective instability. A final important consideration is the surface deflection arising near a finite container boundary due to the presence of a fluid meniscus. Later experimentalists [196] employed special techniques to allow filling of the experimental container to the brim while simultaneously preventing the fluid from wetting the outside.

These experimental and numerical results raise several basic questions. For example: Is there something special about the particular Ra, Ma, Γ combinations reported by the experimentalists, or would patterns with the same number of cells exist over a range of different (say) aspect ratios? Why are asymmetric patterns (e.g. the three-cell pattern) sometimes observed while at other times symmetric patterns predominate? Which configurations transfer heat most effectively? Does a configuration with more cells always have a higher heat transfer capability than one with fewer cells? Why was there no configuration reported having seven convection cells? We tackle several of

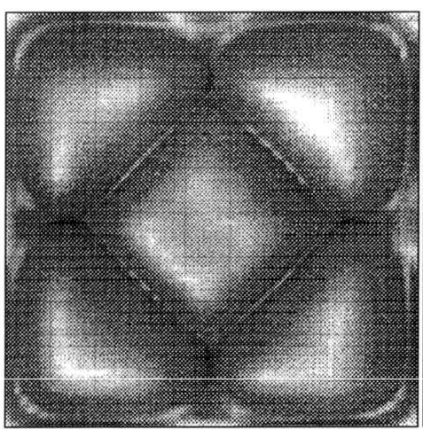
these questions in this work, and in particular attempt to verify the hypothesis stated in the following section.



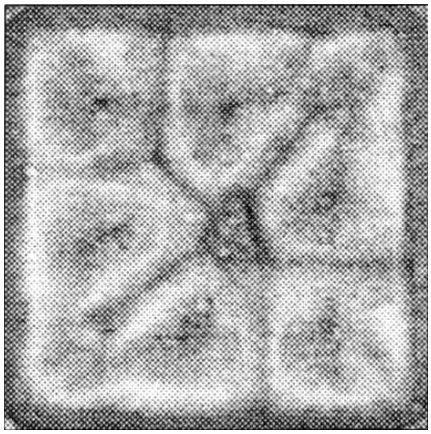
$Ra = 228,\ Ma = 380,\ \Gamma = 1.82$ $Ra = 33,\ Ma = 64,\ \Gamma = 5.68$

$Ra = 42,\ Ma = 96,\ \Gamma = 6.18$ $Ra = 38,\ Ma = 93,\ \Gamma = 6.36$

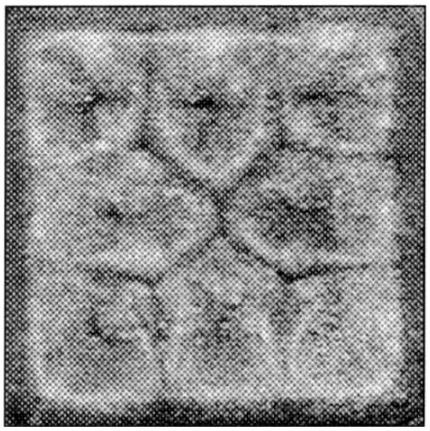
Figure 1.2: Experimental results for small aspect ratio (small horizontal extent, relative to depth) reported by Koschmieder [113] and Medale and Cerisier [126]. Ra and Ma are non-dimensional parameters which, together with the aspect ratio of the container Γ, may be used to characterize the state of the RBM system. The Rayleigh number, Ra, measures the relative importance of buoyant and dissipative effects. The Marangoni number, Ma is a measure of the relative importance of surface tension and dissipative effects.



$Ra = 19,\ Ma = 80,\ \Gamma = 8.4$



$Ra = 22,\ Ma = 86,\ \Gamma = 8.08$



$Ra = 16,\ Ma = 63,\ \Gamma = 8.75$

Figure 1.3: Experimental results for small aspect ratio (small horizontal extent, relative to depth) reported by Koschmieder [113] and Medale and Cerisier [126].

1.3 Hypothesis

Consider the following thought experiment: suppose we have a container of fixed depth whose horizontal extent can be changed continuously at our discretion. In this case, we could vary only a single parameter (the aspect ratio, Γ) while keeping the others constant and observe the behavior of the convection cells. A similar thought experiment could also be applied to the fluid: we imagine a fluid whose surface tension coefficient can be dynamically varied independently of external influences such as temperature. In this case, the Ma could be changed while the other parameters (Ra, Γ) were held fixed.

Such a fictitious container or fluid is difficult or impossible to realize in practice: previous experimentalists have generally conducted their experiments by changing the depth of the fluid layer or by changing the applied temperature gradient. These procedures have the undesirable side effect of changing all the problem parameters simultaneously. Therefore, in an experiment, it becomes difficult to say which of the independent parameters has the strongest influence on the number and shape of the cells in the configuration. These thought experiments reveal the importance of numerical simulation as a tool for gaining better insight into physical systems and for driving understanding about the critical parameters of the system. The potential may now be better-realized due to the rapid evolution of economic large-scale parallel computation and the associated software advances.

Our hypothesis in undertaking the present study is as follows: *In a square cross-section container, for given Ma and Ra (so long as they are*

above the critical values necessary for convection) there exist steady state configurations having one or more convection cells that depend only on the aspect ratio Γ *of the container.* Numerical simulation is the most straightforward way to test this hypothesis, since via simulation it is possible to conduct precisely the thought experiments described previously. In turn, this opportunity leads to new theoretical results that may stimulate future experimental studies.

The remainder of this dissertation is arranged as follows: in Ch. 2 we discuss the equations governing RBM convection and their discretization by the finite element method. In Ch. 3 we discuss a series of different unsteady simulation results obtained for different aspect ratio containers. In Ch. 4 we discuss the details of our pseudo-arclength continuation algorithm. Then in Ch. 5 we employ these continuation methods to trace non-trivial solution branches in Γ-parameter space. In Ch. 6 we tackle the related task of performing continuation in Ma space. In Ch. 7 we conduct adaptive mesh refinement simulations of a related RBM convection problem with a singular "stick-slip" boundary condition. Finally, we draw conclusions and point out promising avenues of future research in Ch. 8. Additional information regarding the RBM equations can be found in Appendix 1, while some information about the numerical methods used is given in Appendix 2.

Chapter 2

Rayleigh-Bénard-Marangoni Convection

In this chapter, we develop the partial differential equations (PDEs) which provide the underlying mathematical framework for simulating RBM convection. This chapter is arranged in the following manner: in §2.1 we begin by discussing several of the practical (from an engineering point of view) applications of RBM convection. Then, in §2.2 the equations for the conservation of momentum, mass, and energy are given, and the appropriate modeling assumptions and approximations for RBM convection are discussed. In §2.3, we consider the topic of defining a "conducting" (i.e. quiescent, initial) solution to the governing equations, and then in §2.4 we discuss the variational statement.

The variational statement naturally gives rise to the boundary conditions, which are discussed in detail §2.5. We discuss the classical non-dimensionalization of the variational form of the equations in §2.6, and a related non-dimensionalization which brings in the aspect ratio of the container as an independent, continuous parameter in §2.7. In §2.8 we discuss the finite element (spatial) discretization, followed by the time discretization in §2.9, and the linearized (Jacobian) system in §2.10.

2.1 Practical Applications

The practical applications of surface-tension-driven flow are many and varied, and help to explain the sustained interest in the topic since Bénard's work at the turn of the century. In the realm of chemical engineering applications, Dijkstra [72] mentions the operation of distillation columns [147, 222] as a particular case in which mass transfer across an interface is strongly influenced by surface tension effects.

Another application where surface tension is important is silicon crystal growth by the float-zone technique [187]. In this process, solid silicon crystals are "grown" from a molten silicon solution suspended between two moving furnace coils [60]. An advantage of this technique is that the purity of the resulting crystal is improved because the silicon does not come into contact with exterior containers. A drawback is surface tension forces alone are required to support the total weight of the silicon solution contained in the melt zone.

Optimization of the surface tension field through the use of heating devices, as well as the utilization the float-zone crystal growth technique in microgravity environments, may be able to produce higher quality, more economical silicon wafers in the semiconductor industry. Brunet et al. [38] discuss various techniques for controlling thermal oscillations in a related combined buoyant and thermocapillary instability problem using a local heating feedback process. A robust, reliable CFD code obviously provides an essential link in a design/optimization loop such as this.

In a microfluidics environments, polymerase chain reaction (PCR) amplification via thermocycling [114] is another application of thermocapillary natural convection flows. Buoyancy-driven convection in closed-loop geometries allows researchers to construct tunable pumping and mixing systems for miniaturized biomechanical analysis. Such flow devices can be scaled to nanoliter reagent volumes and used as low-power DNA analysis systems.

These applications are certainly not all of the industrial processes which rely on surface tension as an important physical aspect. Others, including the quality of metallic welds, the creation of compact disc media by heating the surface of the disc with a laser, and other microfluidics/nanomechanics applications are under active investigation as well. The author has also considered some aspects of the numerical simulation of this class of flows in [149]. Clearly combined buoyant and surface-tension driven flows are of considerable engineering importance.

2.2 Governing Equations

The flow of an incompressible, heated, Newtonian fluid with a gravitational body force is modeled by the Navier-Stokes momentum, continuity, and energy conservation equations

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right) - \nabla\cdot\boldsymbol{\sigma} = \rho\boldsymbol{g} \tag{2.1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{2.2}$$

$$\rho c_p\left(\frac{\partial T}{\partial t} + \boldsymbol{u}\cdot\nabla T\right) = \nabla\cdot(k\nabla T) \tag{2.3}$$

where $\boldsymbol{u} = (u_1, u_2, u_3)$ is the velocity vector, T is the temperature, ρ is the fluid density, $\boldsymbol{g}$ is the gravity vector, c_p is the specific heat, and k is the thermal conductivity. The stress tensor $\boldsymbol{\sigma}$, for a Newtonian fluid, is given by

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{2.4}$$

where p is the pressure, μ is the molecular viscosity coefficient, and $\boldsymbol{\varepsilon}(\boldsymbol{u})$ is *twice* the symmetric part of the velocity gradient tensor

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) := \left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^t\right) \tag{2.5}$$

and where the velocity gradient tensor is

$$(\nabla\boldsymbol{u})_{ij} := \frac{\partial u_i}{\partial x_j} \tag{2.6}$$

Eqns. (2.1)–(2.3) plus suitable initial and boundary conditions, are the basis for the numerical study of Rayleigh-Bénard-Marangoni convection considered in the present work. The development of these equations is standard. See Appendix §1.1 for notes on the derivation of the incompressible energy conservation equation in terms of temperature.

2.2.1 Boussinesq Approximation

The hydrostatic ($\boldsymbol{u} = 0$) solution to Eqns. (2.1)–(2.3) is given by

$$\nabla p_0 = \rho_0 \boldsymbol{g} \tag{2.7}$$

where p_0 and ρ_0 are the hydrostatic pressure and density distributions, respectively. For an incompressible liquid, ρ_0 is constant, but for a gas (e.g. the

atmosphere) ρ_0 can of course vary greatly with altitude. Subtracting Eqn. (2.7) from Eqn. (2.1) yields

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)-\nabla\cdot\left(\mu\boldsymbol{\varepsilon}\left(\boldsymbol{u}\right)\right)=-\nabla\left(p-p_0\right)+\left(\rho-\rho_0\right)\boldsymbol{g} \tag{2.8}$$

Boussinesq's approximation, originally developed in the early 1900s [36] (and based on earlier work by Oberbeck [138]), was to assume that the density ρ did not vary much from the hydrostatic density profile ρ_0, i.e. $\rho \approx \rho_0$. This turns out to be a very reasonable assumption for atmospheric flows, and is also valid for the natural convection type problems considered here. The essence of the approximation is that, if one were to divide Eqn. (2.8) by ρ and obtain

$$\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}-\frac{1}{\rho}\nabla\cdot\left(\mu\boldsymbol{\varepsilon}\left(\boldsymbol{u}\right)\right)=-\frac{1}{\rho}\nabla\left(p-p_0\right)+\frac{\left(\rho-\rho_0\right)}{\rho}\boldsymbol{g} \tag{2.9}$$

then difference between Eqn. (2.9) and the same equation with the density in the denominator replaced by the hydrostatic density ρ_0 is not very great, so long as the approximation $\rho \approx \rho_0$ remains valid. Thus, under the Boussinesq approximation, Eqn. (2.9) is transformed to

$$\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}-\frac{1}{\rho_0}\nabla\cdot\left(\mu\boldsymbol{\varepsilon}\left(\boldsymbol{u}\right)\right)=-\frac{1}{\rho_0}\nabla\left(p-p_0\right)+\frac{\left(\rho-\rho_0\right)}{\rho_0}\boldsymbol{g} \tag{2.10}$$

It can be shown via scaling arguments that the last term in Eqn. (2.10) is not negligible. We next consider, for an equation of state of the form $\rho = \rho(T)$ a Taylor-series type expansion about the hydrostatic density profile of the form

$$\frac{\rho-\rho_0}{\rho_0}\approx\frac{1}{\rho_0}\left.\frac{\partial\rho}{\partial T}\right|_{T^*}\left(T-T_0\right) \tag{2.11}$$

where now T_0 is the hydrostatic temperature distribution. We would like to emphasize that T_0 is *not* a constant, but rather the hydrostatic (linear) temperature profile obtained by solving the hydrostatic energy equation subject to appropriate boundary conditions. The quantity

$$\beta := \frac{1}{\rho_0} \left. \frac{\partial \rho}{\partial T} \right|_{T^*} \tag{2.12}$$

is the coefficient of thermal expansion of the fluid. In general, β is not a strong function of T for most fluids (thus it is not important which value T^* we decide to "evaluate" β at) and we will treat it as a constant in this work. Furthermore, in most practical situations, the density of a fluid decreases with increasing temperature, and thus $\beta < 0$. To avoid confusion with sign conventions, we will simply denote it here as $\beta := -|\beta|$. Plugging the Boussinesq approximation into the momentum equations finally yields

$$\rho_0 \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right) - \nabla \cdot \boldsymbol{\sigma} = -\rho_0 |\beta| (T - T_0)\boldsymbol{g} \tag{2.13}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.14}$$

$$\rho_0 c_p \left(\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T \right) = \nabla \cdot (k \nabla T) \tag{2.15}$$

Note that the pressure in the $\boldsymbol{\sigma}$ term of Eqn. (2.13) has been suitably redefined to include the hydrostatic pressure distribution as well. It is also customary to define a new temperature variable

$$\tilde{T} := T - T_0 \tag{2.16}$$

Since T_0 satisfies Laplace's equation, it is often a fairly simple, e.g. linear, function. Defining $\tilde{T}$ allows us to rewrite Eqns. (2.13)–(2.15) as

$$\rho_0 \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right) - \nabla \cdot \boldsymbol{\sigma} = -\rho_0 |\beta| \tilde{T} \boldsymbol{g} \tag{2.17}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.18}$$

$$\rho_0 c_p \left(\frac{\partial \tilde{T}}{\partial t} + \boldsymbol{u} \cdot \nabla \tilde{T} + \boldsymbol{u} \cdot \nabla T_0 \right) = \nabla \cdot (k \nabla \tilde{T}) \tag{2.19}$$

Note the additional convective term which appears in the energy equation due to this rescaling of the temperature. Henceforth, we will treat Eqns. (2.17)–(2.19) as *the* strong form governing equations, though we have said nothing up to this point regarding the boundary or initial conditions.

2.3 Conducting Solutions

We now more carefully consider the question of finding a hydrostatic (quiescent, conducting) solution, i.e. a solution in which the fluid is motionless ($\boldsymbol{u} = 0$), and acts as a conducting material. This is an important topic since the conducting solution will serve as initial data for the simulation. The physical situation is typically that the surface of the fluid is "free" in the sense that it is not in contact with any solid object, but it is cooled by virtue of a nearby glass (or artificial sapphire) window which is held at a fixed temperature with recirculating cold water and still permits viewing of the free liquid surface. An air gap exists between this fixed-temperature glass window and the free surface of the liquid. The lower plate is usually made of a conducting metal such as copper.

Remark. In some experimental studies of RBM convection, including those of Bénard himself, there is an unconfined air layer above the fluid. This configuration is more difficult to control experimentally, and the definition of a characteristic temperature scale is more ambiguous. In intermediate cases where the height of the air layer is large (relative to the depth of the liquid layer) but still controlled by the experimenter, the critical parameters for onset of convection can change considerably [143], and buoyancy-induced convection in the air layer may be present. ❖

We shall assume the upper glass window and lower metallic plate are maintained at fixed temperatures T_c and T_h, respectively, as shown in Figure 2.1. The air layer is assumed to have thermal conductivity k_{air} and depth d_{air} while the liquid layer is assumed to have thermal conductivity k and depth d. The conducting temperature profile (which is a function only of the vertical coordinate, z) is composed of two linear segments with (potentially) different slopes meeting at the point $z = d$. As before, we denote by $T_0(z)$ the linear temperature profile in the liquid layer, and we introduce $T_{\text{air}}(z)$ to represent the linear conducting profile in the air layer. We shall use the notation $T_0|_{z=d}$ to denote the common interface value.

The compatibility condition for the thermal flux

$$k\frac{\partial T_0}{\partial z} = k_{\text{air}}\frac{\partial T_{\text{air}}}{\partial z} \tag{2.20}$$

must be satisfied by this profile. Eqn. (2.20) essentially states that the thermal flux through the liquid layer is exactly balanced by the thermal flux through

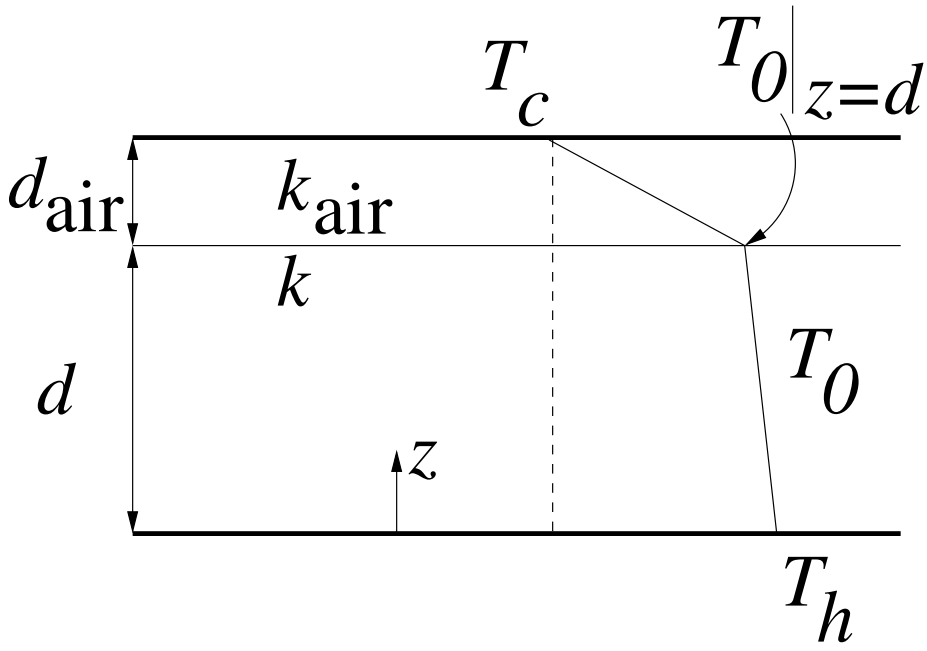


Figure 2.1: Doubly-conducing liquid-gas interface solution.

the air layer, i.e. the thermal flux is conserved. Since the temperature profiles are both linear, we can replace the temperature gradient terms to obtain

$$k\frac{(T_0|_{z=d} - T_h)}{d} = k_{\text{air}}\frac{(T_c - T_0|_{z=d})}{d_{\text{air}}} \tag{2.21}$$

Gathering terms, we can solve for the interface temperature as

$$T_0|_{z=d} = \frac{T_h + \left(\frac{d}{k}\frac{k_{\text{air}}}{d_{\text{air}}}\right)T_c}{1 + \left(\frac{d}{k}\frac{k_{\text{air}}}{d_{\text{air}}}\right)} \tag{2.22}$$

The non-dimensional parameter appearing in the numerator and denominator of Eqn. (2.22) is the "conduction" Biot number which we will denote by

$$Bi := \frac{d}{k}\frac{k_{\text{air}}}{d_{\text{air}}} \tag{2.23}$$

Based on values reported in Koschmeider's classic paper [113] on Bénard-Marangoni convection and some reasonable assumptions on the thermal conductivities of air and silicone oils ($d \approx 1$ mm, $d_{\text{air}} \approx 0.5$ mm, $k \approx 0.29$ W/m-K, and $k_{\text{air}} \approx 0.026$ W/m-K) we have

$$Bi \approx 0.18 \tag{2.24}$$

for the configurations of interest here. Thus, for the conducting thermal profile T_0, the interface temperature $T_0|_{z=d}$ is given explicitly in terms of the Biot number and the fixed hot and cold temperatures of the plates. In the limit as $Bi \to \infty$ (i.e. a very thin air gap relative to the liquid layer depth) $T_0|_{z=d} \to T_c$ from above.

2.4 Variational Statement

Returning to the full coupled Navier-Stokes and thermal transport equations, we form the weak statement by multiplying Eqn. (2.17) by a vector test function $\boldsymbol{v} \in V$, Eqn (2.18) by a scalar test function $q \in Q$, and Eqn (2.19) by $w \in W$, integrating over the domain of interest Ω, and employing the divergence theorem. Our task is then to find $\{\boldsymbol{u}, p, \tilde{T}\}$ satisfying

$$\int_\Omega \rho_0 \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right) \cdot \boldsymbol{v} + \boldsymbol{\sigma} : \nabla \boldsymbol{v} + \rho_0 |\beta| \tilde{T} \boldsymbol{g} \cdot \boldsymbol{v} \, dx - \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{v} \, ds = 0 \tag{2.25}$$

$$\int_\Omega (\nabla \cdot \boldsymbol{u}) q \, dx = 0 \tag{2.26}$$

$$\int_\Omega \rho_0 c_p \left(\frac{\partial \tilde{T}}{\partial t} + \boldsymbol{u} \cdot \nabla \tilde{T} + \boldsymbol{u} \cdot \nabla T_0 \right) w + k(\nabla \tilde{T} \cdot \nabla w) \, dx - \int_{\partial\Omega} (k \nabla \tilde{T} \cdot \boldsymbol{n}) w \, ds = 0 \tag{2.27}$$

for every $\boldsymbol{v} \in V$, $q \in Q$, and $w \in W$, where $\boldsymbol{n}$ is the outward unit normal vector. We have again used the dyadic product notation (defined in Appendix §1.2.1) between the stress tensor $\boldsymbol{\sigma}$ and the gradient of the velocity test function.

2.5 Boundary Conditions

Of particular interest in the thermocapillary surface tension problem is the surface tension boundary condition which is applied on the free surface. In fact, the only way that surface tension forces enter the problem is through this boundary term. In Appendix 1, we give a brief description (using the equations for Stokes flow as an example) of the different boundary conditions obtained when the so-called "Laplace form" of the viscous diffusion operator is used in the governing equations, as compared to the full "traction form" of this operator.

2.5.1 Momentum Equation BCs

We will be particularly interested in applying standard "no-slip" and free-surface traction thermocapillary boundary conditions in the momentum equations. For the standard no-slip condition, $\boldsymbol{v}|_{\partial\Omega} = \mathbf{0}$ is imposed, and the boundary condition is enforced directly via a penalty term. For more details on the penalty boundary condition formulation and its relation to Navier's slip condition, see Appendix §1.3.

For the free-surface traction thermocapillary condition, recall that the boundary term is

$$\int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{v} \, ds \tag{2.28}$$

with $\boldsymbol{v}$ an arbitrary test function. Let us first consider the case of an arbitrary curved surface, and then later specialize to the case of a planar surface. At a given point on the surface, define an orthogonal coordinate system in $\mathbb{R}^3$ by the unit basis vectors $\boldsymbol{t}_1$, $\boldsymbol{t}_2$, and $\boldsymbol{n}$, where $\boldsymbol{t}_{1,2}$ define the (local) tangent plane of the fluid surface, and $\boldsymbol{n}$ is the outward normal vector. Then clearly we can express an arbitrary vector $\boldsymbol{v}$ as

$$\boldsymbol{v} = (\boldsymbol{v} \cdot \boldsymbol{t}_1)\boldsymbol{t}_1 + (\boldsymbol{v} \cdot \boldsymbol{t}_2)\boldsymbol{t}_2 + (\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{n} \tag{2.29}$$

In general, $\boldsymbol{t}_1$, $\boldsymbol{t}_2$, and $\boldsymbol{n}$ depend on both the location on the free surface and time, since the free surface may be evolving in time. The boundary term may

thus be written as

$$\begin{aligned}\int_{\partial\Omega} \boldsymbol{\sigma}\cdot\boldsymbol{n}\cdot\boldsymbol{v}\; ds &= \int_{\partial\Omega} (\boldsymbol{v}\cdot\boldsymbol{t}_1)\,\boldsymbol{\sigma}\cdot\boldsymbol{n}\cdot\boldsymbol{t}_1\; ds \\ &+ \int_{\partial\Omega} (\boldsymbol{v}\cdot\boldsymbol{t}_2)\,\boldsymbol{\sigma}\cdot\boldsymbol{n}\cdot\boldsymbol{t}_2\; ds \\ &+ \int_{\partial\Omega} (\boldsymbol{v}\cdot\boldsymbol{n})\,\boldsymbol{\sigma}\cdot\boldsymbol{n}\cdot\boldsymbol{n}\; ds \end{aligned} \tag{2.30}$$

Due to kinematic and mechanical consistency conditions at the interface between two fluids where surface tension is active (see e.g. White [205] p. 235 or Cliffe and Tavener [57]) we have

$$\boldsymbol{\sigma}\cdot\boldsymbol{n}\cdot\boldsymbol{t}_i = \nabla\gamma\cdot\boldsymbol{t}_i \tag{2.31}$$

for $i = 1, 2$ in the plane of the surface, and

$$\boldsymbol{\sigma}\cdot\boldsymbol{n}\cdot\boldsymbol{n} = K\gamma \tag{2.32}$$

normal to the fluid surface, where γ is the surface tension coefficient, and K is twice the mean curvature of the surface, defined as

$$K := \left(\frac{1}{R_1} + \frac{1}{R_2}\right) \tag{2.33}$$

where R_1 and R_2 are the principal radii of curvature for the surface. We note that for a flat surface, $K = 0$. The conditions of Eqns. (2.31) and (2.32) arise from requiring a balance of stresses at the fluid/fluid (liquid/gas or liquid/liquid) interface.

In general, the surface tension of a fluid is a function of many variables, such as temperature and surfactant concentration. Here we will consider only

the variation of surface tension with respect to temperature, for which a reasonable approximation in many fluids is (see e.g. [60, 91])

$$\gamma \approx \gamma_0 - \left|\frac{\partial \gamma}{\partial T}\right|_{T^*} (T - T_0) \tag{2.34}$$

where, as in the case of the Boussinesq description, we have used the absolute value to prevent ambiguity about the sign of $\frac{\partial \gamma}{\partial T}$, which we assume to always be negative. Using the model of surface tension given by Eqn. (2.34), we observe that Eqns. (2.31) and (2.32) become

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{t}_i = -\left|\frac{\partial \gamma}{\partial T}\right|_{T^*} \nabla \tilde{T} \cdot \boldsymbol{t}_i \tag{2.35}$$

and

$$\begin{aligned} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{n} &= K\gamma \\ &\approx K\gamma_0 \end{aligned} \tag{2.36}$$

where the approximation in Eqn. (2.36) is justified by the same arguments used for the Boussinesq-type approximation discussed earlier, see also [60]. The full boundary term defined in Eqn. (2.30) under the preceding substitutions is therefore

$$\begin{aligned} \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{v}\, ds &= -\int_{\partial\Omega} (\boldsymbol{v} \cdot \boldsymbol{t}_1) \left|\frac{\partial \gamma}{\partial T}\right| \nabla \tilde{T} \cdot \boldsymbol{t}_1\, ds \\ &\quad -\int_{\partial\Omega} (\boldsymbol{v} \cdot \boldsymbol{t}_2) \left|\frac{\partial \gamma}{\partial T}\right| \nabla \tilde{T} \cdot \boldsymbol{t}_2\, ds \\ &\quad +\int_{\partial\Omega} (\boldsymbol{v} \cdot \boldsymbol{n})\, K\gamma_0\, ds \end{aligned} \tag{2.37}$$

The boundary condition of Eqn. (2.37) is difficult to impose in the general case: the surface position is an evolving unknown (hence $\boldsymbol{t}_1$, $\boldsymbol{t}_2$, $\boldsymbol{n}$ are unknown)

which must be solved by an auxiliary surface equation which is in turn coupled to the other governing equations. The curvature K is a highly-nonlinear function of the unknown surface position, and a decoupled (staggered) approach with a moving mesh would likely be needed to solve the entire system. In addition, the curved surface must in general satisfy a contact angle condition which depends on the container and introduces further problem parameters.

A powerful simplifying assumption frequently imposed on these problems is that the fluid/fluid interface is flat (or nearly so). This implies the curvature $K \approx 0$. Of course the question arises of how "small" K must be relative to other parameters of the problem for this approximation to remain valid. A rigorous scaling analysis shows that

$$\frac{\gamma_0 d}{\mu\alpha} Kd := \frac{Kd}{Ca} \ll 1 \tag{2.38}$$

is required, where d is the depth of the liquid layer, α is the thermal diffusivity of the fluid, and Ca is a non-dimensional parameter known as the capillary number. For the experiments conducted by Koschmeider and Prahl [113] we can in fact compute the capillary number for the apparatus discussed in the paper ($\gamma_0 \approx 14$ dyn/cm, $d \approx 1$ mm, $\mu \approx 0.986$ g/cm-s, $\alpha \approx 0.001$ cm^2/s) as $Ca \approx 7 \times 10^{-4}$. We are unaware of any experimental studies in which accurate measurements of the surface curvature are given, but for this work we will assume the condition of Eqn. (2.38) holds. Numerical solution of the fully-deforming three-dimensional surface remains as an interesting avenue of future study.

Assuming a nearly flat surface allows us to define our (x_1, x_2, x_3) coordinate system such that x_1 and x_2 lie in the plane of the free surface. This means that, for $i = 1, 2$, we have $\boldsymbol{t}_i = \hat{e}_i$ (the unit vector in the x_i direction) and consequently the boundary term of Eqn. (2.37) becomes

$$\int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{v} \, ds \;\; = \;\; -\int_{\partial\Omega} \left| \frac{\partial \gamma}{\partial T} \right| \nabla \tilde{T} \cdot \boldsymbol{v} \, ds \tag{2.39}$$

We shall use the boundary condition of Eqn. (2.39) in the remainder of this work. We note that the assumption of a planar or nearly planar free surface also implies a zero vertical velocity (no penetration) boundary condition at the free surface. This is enforced in the usual way by (formally) considering only test functions $\boldsymbol{v}$ whose third component vanishes at the free surface. In practice, the zero vertical velocity is enforced via a mixed-type penalty boundary term.

Remark. The definitive nonlinear stability analysis of RBM convection with a deforming free surface was performed by Davis and Homsy [64] in 1980. Their results, which assumed an infinite, periodic fluid layer with a prescribed heat flux through a free, possibly deforming, surface, involved a great deal of algebraic manipulation in an era before the use of computer algebra systems (CAS) had become widespread. Using energy stability methods, they were able to show that the undeformed interface satisfied the governing equations (to first-order in Ca) in agreement with previous theoreticians [63, 183] who had *a priori* assumed an undeformed surface.

In addition, they showed (in both nonlinear and linear stability analyses with vanishing Ca) that, at least for small Ma, a deforming surface has a stabilizing effect on the fluid layer (suppresses the convective instability). Although Davis and Homsy's results are somewhat limited by various simplifying assumptions, they do suggest that the simplifying assumption of a non-deforming layer is a reasonable one, and the smallest critical Ma will still be found for a non-deforming infinite layer. We are presently unaware of any related stability analyses which apply to surface deflections in the small aspect ratio layer case considered here. ❖

2.5.2 Energy Equation BCs

In the energy equation, we will consider fixed (Dirichlet) temperature conditions, insulating, and passively-cooled boundary conditions. For fixed-temperature boundary conditions, we impose $w|_{\partial\Omega} = 0$ (similar to the no-slip BCs) and the boundary condition in Eqn. (2.27) is imposed directly via a penalty term. Note that for the reduced temperature equations, the proper Dirichlet boundary condition applied to the bottom of the domain is $\tilde{T} = 0$, i.e. the variation from the linear profile is required to be zero at the Dirichlet boundary.

The thermally-insulating ($-k\nabla\tilde{T} \cdot \boldsymbol{n} = 0$) condition is typically applied on the "side walls" of the domain. In this case, the boundary integral in Eqn. (2.27) drops out entirely. On the free surface, a variety of thermal boundary conditions can be imposed. Perhaps the most common free surface

boundary condition, sometimes known as Newton's Law of cooling, imposes (at $z = d$)

$$-k\nabla T \cdot \boldsymbol{n} = h_c \left(T|_{z=d} - T_a \right) \tag{2.40}$$

where h_c is known variously as the heat transfer coefficient, film coefficient, or unit thermal conductance [53] and depends in a complex manner on the properties of the layer of gas which is presumed to be cooling the liquid layer. In the relation above, $T|_{z=d}$ is the (unknown) temperature of the liquid-gas interface, and T_a is a "characteristic" temperature of the upper air layer. In general, h_c depends on many factors such as the material properties of the air layer itself, and whether or not the air layer above the fluid is in motion or at rest.

In this work, we will assume that the air layer above the liquid is *conducting* only, in which case we have a simplified form of Eqn. (2.40) given by

$$-k\nabla T \cdot \boldsymbol{n} = \frac{k_{\text{air}}}{d_{\text{air}}} \left(T|_{z=d} - T_c \right) \tag{2.41}$$

where T_c is the fixed temperature of the upper cooling plate. The boundary condition given in Eqn. (2.41) posed in terms of the temperature with the linear profile subtracted ($\tilde{T} := T - T_0$) is

$$-k\frac{\partial \tilde{T}}{\partial z}\bigg|_{z=d} = \frac{k_{\text{air}}}{d_{\text{air}}} \tilde{T}\Big|_{z=d} \tag{2.42}$$

A more detailed derivation of Eqn. (2.42) is given in Appendix 1.4.

Remark. While the simplifying assumption of a conducting air layer is a common and reasonable one for the present configuration (the thermal diffusivity

of the air is about two orders of magnitude larger than that of the liquid, see also the stability analyses by Regnier et al. [165]) other authors including Johnson [101] and Juel et al. [102] have also considered the case of instabilities in a liquid-liquid layer heated from either above or below. In the heated from below case, thermocapillary forces are still destabilizing in the bottom liquid layer, but in the top liquid layer they are actually *stabilizing*. The variety of convective mechanisms is greater in the bi-liquid layer case, and the associated numerical calculations are more complex. ❖

Returning to Eqns. (2.25)–(2.27), we can expand the dyadic product and make the appropriate boundary condition substitutions to arrive at

$$\int_\Omega \Big[\rho_0 \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)\cdot\boldsymbol{v} - p(\nabla\cdot\boldsymbol{v}) + \mu\boldsymbol{\varepsilon}(\boldsymbol{u}):\nabla\boldsymbol{v} + \rho_0|\beta|\tilde{T}\boldsymbol{g}\cdot\boldsymbol{v}\Big]\,dx + \int_{\partial\Omega_s}\left|\frac{\partial\gamma}{\partial T}\right|\nabla\tilde{T}\cdot\boldsymbol{v}\,ds = 0 \tag{2.43}$$

$$\int_\Omega (\nabla\cdot\boldsymbol{u})q\,dx = 0 \tag{2.44}$$

$$\int_\Omega \rho_0 c_p\left(\frac{\partial\tilde{T}}{\partial t} + \boldsymbol{u}\cdot\nabla\tilde{T} + \boldsymbol{u}\cdot\nabla T_0\right)w + k(\nabla\tilde{T}\cdot\nabla w)\,dx + \frac{k_{\text{air}}}{d_{\text{air}}}\int_{\partial\Omega_s}\tilde{T}w\,ds = 0 \tag{2.45}$$

In Eqns. (2.43)–(2.45), we have used the notation $\partial\Omega_s$ to denote that part of the boundary which is the "free" surface subject to thermocapillary surface-tension

forces. Dirichlet conditions are imposed by the addition of an appropriate penalty term (essentially a mixed boundary condition with a very large penalty coefficient that enforces the Dirichlet value to within some precision dependent on the size of the penalty) to Eqns. (2.43)–(2.45) which is fairly standard and hence not shown here.

2.6 Classical Non-Dimensionalization

A generic non-dimensionalization of Eqns. (2.43)–(2.45) may be accomplished by assuming appropriate length (l_s), time (t_s), velocity (u_s), and temperature (T_s) scales for the problem exist, and by defining the non-dimensional "starred" variables

$$\nabla^* = \frac{1}{l_s}\nabla \qquad \tilde{T}^* = \tfrac{\tilde{T}}{T_s} \qquad t^* = \frac{t}{t_s}$$
$$\boldsymbol{u}^* = \tfrac{1}{u_s}\boldsymbol{u} = \tfrac{t_s}{l_s}\boldsymbol{u}$$
$$p^* = \frac{t_s}{\mu}p \qquad dx = l_s^3 dx^* \qquad ds = l_s^2 ds^*$$

Under these substitutions, Eqns. (2.43)–(2.45) become

$$\int_{\Omega^*}\left[\frac{\rho_0 u_s l_s}{\mu}\left(\frac{\partial \boldsymbol{u}^*}{\partial t^*} + (\boldsymbol{u}^*\cdot\nabla^*)\boldsymbol{u}^*\right)\cdot\boldsymbol{v} + \boldsymbol{\varepsilon}^*(\boldsymbol{u}^*):\nabla^*\boldsymbol{v} - p^*(\nabla^*\cdot\boldsymbol{v})\right]dx^*$$
$$+\int_{\Omega^*}\frac{\rho_0|\beta||\boldsymbol{g}|T_s l_s t_s}{\mu}\tilde{T}^*\hat{\boldsymbol{g}}\cdot\boldsymbol{v}\;dx^*$$
$$+\int_{\partial\Omega_s^*}\left|\frac{\partial\gamma}{\partial T}\right|\frac{t_s T_s}{\mu l_s}\nabla^*\tilde{T}^*\cdot\boldsymbol{v}\;ds^* = 0 \tag{2.46}$$

$$\int_{\Omega^*}(\nabla^*\cdot\boldsymbol{u}^*)q\;dx^* = 0 \tag{2.47}$$

$$\int_{\Omega^*} \frac{\rho_0 c_p l_s^2}{t_s k} \left(\frac{\partial \tilde{T}^*}{\partial t^*} + \boldsymbol{u}^* \cdot \nabla^* \tilde{T}^* + \boldsymbol{u}^* \cdot \nabla^* T_0^* \right) w + (\nabla^* \tilde{T}^* \cdot \nabla^* w) \, dx^* + \frac{k_{\text{air}}}{d_{\text{air}}} \frac{l_s}{k} \int_{\partial \Omega_s^*} \tilde{T}^* w \, ds^* = 0 \quad (2.48)$$

In Eqn. (2.46), we have replaced $\boldsymbol{g}$ by $|\boldsymbol{g}|\hat{\boldsymbol{g}}$, where $|\boldsymbol{g}|$ is the magnitude of $\boldsymbol{g}$ and $\hat{\boldsymbol{g}}$ is a unit vector in the direction of gravity. We note that Eqns. (2.46)–(2.48) define the following groupings of non-dimensional parameters

$$\text{Group A} = \frac{\rho_0 u_s l_s}{\mu} \quad (2.49)$$

$$\text{Group B} = \frac{\rho_0 |\beta| |\boldsymbol{g}| T_s l_s t_s}{\mu} \quad (2.50)$$

$$\text{Group C} = \left| \frac{\partial \gamma}{\partial T} \right| \frac{t_s T_s}{\mu l_s} \quad (2.51)$$

$$\text{Group D} = \frac{\rho_0 c_p l_s^2}{t_s k} = \frac{l_s^2}{\alpha t_s} \quad (2.52)$$

$$\text{Group E} = \frac{k_{\text{air}}}{d_{\text{air}}} \frac{l_s}{k} \quad (2.53)$$

where we have introduced the thermal diffusivity $\alpha = k/\rho_0 c_p$ in Group D.

It is classical to choose the characteristic length scale l_s of the problem equal to the depth d of the fluid layer, and the characteristic time scale t_s as the thermal diffusion time scale d^2/α for the problem. There are at least two possible ways of defining the temperature scale T_s. For Rayleigh-Bénard flow (no free surface) it is most convenient to select the temperature difference across the fixed-temperature plates

$$T_{s,RB} := \Delta T = T_h - T_c \quad (2.54)$$

as the temperature scale. For surface-tension-driven flows, however, this turns out be less useful than taking the temperature difference across *just the liquid layer* in the conducting state

$$T_s := T_h - T_0|_{z=d} \tag{2.55}$$

as the relevant temperature scale. Note that this is also the temperature scaling used to report the results in Koschmeider's classic paper [113]. The conducting profile T_0 can be given (see definitions in §2.3 for details) in terms of the fixed plate temperatures and the Bi number as

$$T_0 = T_h - \frac{z}{d}\left(\frac{Bi}{1+Bi}\left(T_h - T_c\right)\right) \tag{2.56}$$

Substituting Eqn. (2.56) into Eqn. (2.55), we obtain

$$T_s := \frac{Bi}{1+Bi}\Delta T \tag{2.57}$$

and the connection to the Rayleigh-Bénard temperature scaling is immediately obvious. Some authors [72] have called Rayleigh and Marangoni numbers defined in terms of the temperature scale in Eqn. (2.57) "reduced Rayleigh" and "reduced Marangoni" numbers, since for any $Bi > 0$, this temperature scale will be smaller than that used for the classical Rayleigh-Bénard problem. We also observe that, in the limit as $Bi \to \infty$, our temperature scaling T_s approaches the temperature scale $T_{s,RB}$ used for the Rayleigh-Bénard problem.

Under these choices, and the induced velocity scale $u_s = l_s/t_s = \alpha/d$,

we arrive at the following standard parameters for the non-dimensional groups

$$\text{Group A} = \frac{\alpha\rho_0}{\mu} = \frac{1}{Pr} \tag{2.58}$$

$$\text{Group B} = \frac{\rho_0|\beta||\boldsymbol{g}|T_s d^3}{\alpha\mu} = Ra \text{ (reduced)} \tag{2.59}$$

$$\text{Group C} = \left|\frac{\partial\gamma}{\partial T}\right| \frac{dT_s}{\mu\alpha} = Ma \text{ (reduced)} \tag{2.60}$$

$$\text{Group D} = \frac{d^2}{\alpha(d^2/\alpha)} = 1 \tag{2.61}$$

$$\text{Group E} = \frac{k_{\text{air}}}{d_{\text{air}}}\frac{d}{k} = Bi \tag{2.62}$$

where Pr is the Prandtl number, Ra is the Rayleigh number, Ma is the Marangoni number, Bi is the Biot number, and T_s is the reduced temperature scale, as defined in Eqn. (2.57). Substituting these parameters into Eqns. (2.46)–(2.48) we arrive at the following non-dimensionalized set of equations

$$\int_\Omega \left[\frac{1}{Pr}\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)\cdot\boldsymbol{v} + \boldsymbol{\varepsilon}(\boldsymbol{u}):\nabla\boldsymbol{v} + Ra\ \tilde{T}\hat{\boldsymbol{g}}\cdot\boldsymbol{v}\right] dx - \int_\Omega p(\nabla\cdot\boldsymbol{v})\ dx + Ma\int_{\partial\Omega_s}\nabla\tilde{T}\cdot\boldsymbol{v}\ ds = 0 \tag{2.63}$$

$$\int_\Omega (\nabla\cdot\boldsymbol{u})q\ dx = 0 \tag{2.64}$$

$$\int_\Omega \left[\left(\frac{\partial\tilde{T}}{\partial t} + \boldsymbol{u}\cdot\nabla\tilde{T} + \boldsymbol{u}\cdot\nabla T_0\right)w + (\nabla\tilde{T}\cdot\nabla w)\right] dx + Bi\int_{\partial\Omega_s}\tilde{T}w\ ds = 0 \tag{2.65}$$

where we have dropped the asterisks for convenience, and it is understood that we are referring to the non-dimensionalized variables from here on.

2.7 Non-Dimensionalization with Aspect Ratio as a Continuous Parameter

In previous work by the author [149], it was observed that the aspect ratio $\Gamma := \frac{\sqrt{A}}{d}$ of the container, where A is the free surface area and d is the container depth, appeared to be an important parameter in determining the flow structure and convection patterns which are selected. The aspect ratio does not, however, appear explicitly in the governing equations — it changes implicitly when the computational domain used for the computation is changed. This insight provided the motivation to formulate a related set of governing equations with the aspect ratio as another non-dimensional parameter in the equations.

In fact, purely geometric non-dimensional parameters (aspect ratios) are quite common and have been investigated as bifurcation parameters in the context of several different types of flows. Seydel [184, 185] has investigated Marangoni convection in cylindrical containers using $Q := A^3 Re$ (where $A := R/L$ is the aspect ratio, R is the cylinder radius, L is the cylinder length, and Re is the Reynolds number) as a bifurcation parameter. His governing equations are formulated such that Q appears as an explicit parameter, and thus continuation algorithms in Q are possible.

In the context of the classical Taylor vortices problem, Cliffe [54–56]

and in conjunction with coworkers Pfister [151] and Anson [10] has investigated the aspect ratio $\Gamma := l/d$ of the concentric cylinders as an important bifurcation parameter. Here, l is the height of the annulus and $d := r_2 - r_1$ is the difference between the concentric cylinder radii. Their work combines both experimental observation and numerical calculations employing the finite element method, and like Seydel, they have formulated the Navier-Stokes equations in cylindrical polar coordinates in such a way that the aspect ratio appears as an independent parameter or in conjunction with other problem parameters.

Riley and Winters [168, 169] have investigated Lapwood convection (i.e. natural convection in a heated porous medium) with the aspect ratio $h := W/H$ (where W is the width and H is its height) of the two-dimensional container as an explicit parameter in the governing equations. Employing a finite element method, they solved standard bifurcation systems to obtain critical values in the h, Ra parameter space. Winters [208] numerically investigated laminar flows in tubes with rectangular cross-section, using a set of axisymmetric equations similar to those of Cliffe, where the aspect ratio of the tube appears as a parameter. These examples are only a few of those available in the fluid mechanics literature, and they show that it is often helpful to formulate the governing equations so that the aspect ratio of the container can be investigated as an independent parameter using continuation methods.

Here, we define a non-dimensionalization of the equations governing RBM flow by considering two representative length scales: $L \approx \sqrt{A}$ for the

horizontal extent of the container and d for the vertical depth of the container, and we non-dimensionalize the independent spatial variables x_1, x_2, and x_3 individually as

$$x_1^* := \frac{x_1}{L} \qquad x_2^* := \frac{x_2}{L} \qquad x_3^* := \frac{x_3}{d} \tag{2.66}$$

This allows us to transform the dimensional gradient operator as

$$\begin{aligned}
\nabla &= \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}\right) \\
&= \left(\frac{1}{L}\frac{\partial}{\partial x_1^*}, \frac{1}{L}\frac{\partial}{\partial x_2^*}, \frac{1}{d}\frac{\partial}{\partial x_3^*}\right) \\
&= \frac{1}{L}\left(\frac{\partial}{\partial x_1^*}, \frac{\partial}{\partial x_2^*}, \frac{L}{d}\frac{\partial}{\partial x_3^*}\right) \\
&= \frac{1}{L}\left(\frac{\partial}{\partial x_1^*}, \frac{\partial}{\partial x_2^*}, \Gamma\frac{\partial}{\partial x_3^*}\right) \\
&:= \frac{1}{L}\nabla_\Gamma^*
\end{aligned}$$

where the subscript has been used to specify that the third component of the vector is scaled by the aspect ratio $\Gamma := L/d$. The infinitesimal volumes dx and ds (when taken over an x_1, x_2 surface) are non-dimensionalized as

$$dx = (L^2 d)dx^* \qquad ds = L^2 ds^* \tag{2.67}$$

We can substitute these scalings into Eqns. (2.46)–(2.48) to obtain

$$\begin{aligned}
\int_\Omega \left[\frac{\rho_0 u_s L}{\mu}\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla_\Gamma)\boldsymbol{u}\right)\cdot\boldsymbol{v} + \boldsymbol{\varepsilon}_\Gamma(\boldsymbol{u}):\nabla_\Gamma\boldsymbol{v} - p\left(\nabla_\Gamma\cdot\boldsymbol{v}\right)\right]dx \\
+ \int_\Omega \frac{\rho_0|\beta||\boldsymbol{g}|T_s L^2}{\mu u_s}\tilde{T}\hat{\boldsymbol{g}}\cdot\boldsymbol{v}\,dx \\
+ \Gamma\int_{\partial\Omega_s}\left|\frac{\partial\gamma}{\partial T}\right|\frac{T_s}{\mu u_s}\left(\nabla\tilde{T}\cdot\boldsymbol{v}\right)ds = 0
\end{aligned} \tag{2.68}$$

$$\int_\Omega (\nabla_\Gamma \cdot \boldsymbol{u}) q \, dx = 0 \tag{2.69}$$

$$\int_\Omega \frac{\rho_0 c_p L u_s}{k} \left(\frac{\partial \tilde{T}}{\partial t} + \boldsymbol{u} \cdot \nabla_\Gamma \tilde{T} + \boldsymbol{u} \cdot \nabla_\Gamma T_0 \right) w + (\nabla_\Gamma \tilde{T} \cdot \nabla_\Gamma w) \, dx$$
$$- \Gamma \int_{\partial \Omega_s} (\nabla_\Gamma \tilde{T} \cdot \boldsymbol{n}) w \, ds = 0 \tag{2.70}$$

where we have now used $u_s := L/t_s$ for the velocity scale and dropped the asterisks which denote the non-dimensionalized variables and operators. We also define

$$\boldsymbol{\varepsilon}_\Gamma(\boldsymbol{u}) := \left(\nabla_\Gamma \boldsymbol{u} + (\nabla_\Gamma \boldsymbol{u})^t \right) \tag{2.71}$$

as twice the symmetric part of the non-dimensional velocity gradient tensor with suitably-scaled x_3 component. We note that Γ already appears explicitly in the boundary integrals due to the fact that we have divided through by the representative volume $L^2 d$.

It remains still to choose a representative time scale t_s for this case. Recall that in the classical equations, the so-called "diffusion time scale" d^2/α is used. Suppose we now take

$$t_s := \Gamma^Q \frac{d^2}{\alpha} \tag{2.72}$$

for some arbitrary power Q. Setting, for example, $Q = 0$ then leads to the classical diffusion timescale in terms of layer depth d, while taking e.g. $Q = 2$

would yield a time scale purely in terms of the horizontal length scale L. Substituting the velocity scaling

$$u_s := \frac{\alpha L}{\Gamma^Q d^2} \tag{2.73}$$

into Eqns. (2.68)–(2.70) yields

$$\begin{aligned}\int_\Omega \left[\frac{\Gamma^{2-Q}}{Pr}\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla_\Gamma)\boldsymbol{u}\right)\cdot\boldsymbol{v} + \boldsymbol{\varepsilon}_\Gamma(\boldsymbol{u}):\nabla_\Gamma\boldsymbol{v} - p\,(\nabla_\Gamma\cdot\boldsymbol{v})\right] dx \\ + \Gamma^{Q+1}Ra\int_\Omega \tilde{T}\hat{\boldsymbol{g}}\cdot\boldsymbol{v}\,dx \\ + \Gamma^Q Ma\int_{\partial\Omega_s}\left(\nabla\tilde{T}\cdot\boldsymbol{v}\right) ds = 0\end{aligned} \tag{2.74}$$

$$\int_\Omega (\nabla_\Gamma\cdot\boldsymbol{u})q\,dx = 0 \tag{2.75}$$

$$\begin{aligned}\int_\Omega \Gamma^{2-Q}\left(\frac{\partial\tilde{T}}{\partial t} + \boldsymbol{u}\cdot\nabla_\Gamma\tilde{T} + \boldsymbol{u}\cdot\nabla_\Gamma T_0\right) w + (\nabla_\Gamma\tilde{T}\cdot\nabla_\Gamma w)\,dx \\ - \Gamma\int_{\partial\Omega_s}(\nabla_\Gamma\tilde{T}\cdot\boldsymbol{n})w\,ds = 0\end{aligned} \tag{2.76}$$

where Ra, Ma, and Pr are the *same* non-dimensional parameters (defined in terms of the layer depth d) as before. While it is possible to choose any value of Q, it appears that selecting $Q := 2$, which removes the Γ-dependence from the convective terms in the momentum and energy conservation equations, is a reasonable one. This is equivalent to selecting a time scale based on the horizontal length scale, L. The advantages of using Eqns. (2.74)–(2.76) are:

- The dependence of the flow on the enclosure aspect ratio Γ is now explicit. Continuation methods in Γ are now possible since the equations are differentiable with respect to Γ.

- The meshes for box-type enclosures are now perfect cubes, whereas before they could have had significant stretching in the horizontal directions.

- The classical non-dimensional equations can be recovered by setting $\Gamma = 1$, in which case $\nabla_\Gamma = \nabla$ as well.

Finally, we note that by employing the standard conductive cooling boundary condition in the energy conservation Eqn. (2.76), we obtain the boundary term

$$-\Gamma \int_{\partial\Omega_s} (\nabla_\Gamma \tilde{T} \cdot \boldsymbol{n}) w \, ds = \Gamma^2 Bi \int_{\partial\Omega_s} \tilde{T} w \, ds \tag{2.77}$$

on the free surface. The final form of the governing equations under this non-dimensionalization with $Q = 2$ is thus

$$\begin{aligned} \int_\Omega \left[\frac{1}{Pr} \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla_\Gamma)\boldsymbol{u} \right) \cdot \boldsymbol{v} + \boldsymbol{\varepsilon}_\Gamma(\boldsymbol{u}) : \nabla_\Gamma \boldsymbol{v} - p \left(\nabla_\Gamma \cdot \boldsymbol{v} \right) \right] dx \\ + \Gamma^3 Ra \int_\Omega \tilde{T} \hat{\boldsymbol{g}} \cdot \boldsymbol{v} \, dx \\ + \Gamma^2 Ma \int_{\partial\Omega_s} \left(\nabla \tilde{T} \cdot \boldsymbol{v} \right) ds = 0 \end{aligned} \tag{2.78}$$

$$\int_\Omega (\nabla_\Gamma \cdot \boldsymbol{u}) q \, dx = 0 \tag{2.79}$$

$$\int_\Omega \left(\frac{\partial \tilde{T}}{\partial t} + \boldsymbol{u} \cdot \nabla_\Gamma \tilde{T} + \boldsymbol{u} \cdot \nabla_\Gamma T_0 \right) w + (\nabla_\Gamma \tilde{T} \cdot \nabla_\Gamma w)\, dx + \Gamma^2 Bi \int_{\partial\Omega_s} \tilde{T} w \, ds = 0 \tag{2.80}$$

Note that we have presented the entire non-dimensionalization discussion in terms of the variational form of the governing equations, but similar results apply to the strong-form equations (2.17)–(2.19). The non-dimensional strong-form equations (treating Γ as an independent parameter) along with the appropriate non-dimensional boundary conditions are given as Eqns. (2.81)–(2.83) below. The non-dimensional parameters are all defined in the same manner as before.

$$\frac{1}{Pr}\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla_\Gamma)\boldsymbol{u} \right) - \nabla_\Gamma \cdot \boldsymbol{\sigma}_\Gamma = -\Gamma^3 Ra\, \tilde{T} \hat{\boldsymbol{g}} \tag{2.81}$$

$$\nabla_\Gamma \cdot \boldsymbol{u} = 0 \tag{2.82}$$

$$\frac{\partial \tilde{T}}{\partial t} + \boldsymbol{u} \cdot \nabla_\Gamma \tilde{T} + \boldsymbol{u} \cdot \nabla_\Gamma T_0 = \nabla_\Gamma^2 \tilde{T} \tag{2.83}$$

where $\boldsymbol{\sigma}_\Gamma := -p\boldsymbol{I} + \boldsymbol{\varepsilon}_\Gamma(\boldsymbol{u})$. The boundary conditions for standard RBM convection in a container (as described previously in this section) are

$$\begin{aligned} \boldsymbol{u} &= 0 && \text{bottom, sides} \\ \boldsymbol{\sigma}_\Gamma \cdot \boldsymbol{n} &= -\Gamma Ma\, \nabla_\Gamma \tilde{T} && \text{top (free surface)} \end{aligned}$$

for the momentum/mass conservation equations, and

$$\begin{aligned} \tilde{T} &= 0 \qquad && \text{bottom} \\ -\nabla_\Gamma \tilde{T} \cdot \boldsymbol{n} &= 0 \qquad && \text{sides} \\ -\nabla_\Gamma \tilde{T} \cdot \boldsymbol{n} &= \Gamma Bi\ \tilde{T} \qquad && \text{top (free surface)} \end{aligned}$$

for the energy conservation equation.

The numerical method proceeds with a spatial discretization using the Galerkin finite element method, a time discretization via the θ-scheme, and solution of the fully-discrete nonlinear differential algebraic (either steady or time-dependent form) equations via the Inexact Newton method. The details of these three parts (which by this time are fairly standard) are given in §§2.8, 2.9, and 2.10, respectively.

2.8 Spatial Discretization

Proceeding with the finite element method for Eqns. (2.78)–(2.80), we now discretize the problem domain into a set of finite elements $\{\Omega_e\}, e = 1 \ldots N_e$. Introducing the finite-dimensional subspaces $V^h \subset V$, $Q^h \subset Q$, and $W^h \subset W$, the finite element problem is then: find $\{\boldsymbol{u}^h, p^h, \tilde{T}^h\}$ satisfying the boundary conditions and such that

$$\int_\Omega \left[\frac{1}{Pr} \left(\frac{\partial \boldsymbol{u}^h}{\partial t} + (\boldsymbol{u}^h \cdot \nabla_\Gamma)\boldsymbol{u}^h \right) \cdot \boldsymbol{v}^h + \boldsymbol{\varepsilon}_\Gamma(\boldsymbol{u}^h) : \nabla_\Gamma \boldsymbol{v}^h + \Gamma^3 Ra\ \tilde{T}^h \hat{\boldsymbol{g}} \cdot \boldsymbol{v}^h \right] dx$$
$$- \int_\Omega p^h (\nabla_\Gamma \cdot \boldsymbol{v}^h)\ dx + \Gamma^2 Ma \int_{\partial\Omega_s} \nabla \tilde{T}^h \cdot \boldsymbol{v}^h\ ds = 0 \qquad (2.84)$$

$$\int_\Omega (\nabla_\Gamma \cdot \boldsymbol{u}^h) q^h \, dx = 0 \tag{2.85}$$

$$\int_\Omega \left(\frac{\partial \tilde{T}^h}{\partial t} + \boldsymbol{u}^h \cdot \nabla_\Gamma \tilde{T}^h + \boldsymbol{u}^h \cdot \nabla_\Gamma T_0 \right) w^h + (\nabla_\Gamma \tilde{T}^h \cdot \nabla_\Gamma w^h) \, dx + \Gamma^2 Bi \int_{\partial\Omega_s} \tilde{T}^h w^h \, ds = 0 \tag{2.86}$$

hold for all admissible $\boldsymbol{v}^h \in V^h$, $q^h \in Q^h$, and $w^h \in W^h$.

We now require that the velocity and temperature spaces be related in the sense that the individual components of the vector-valued functions $\boldsymbol{v}^h \in V^h$ lie in the space W^h, i.e. $\boldsymbol{v}^h := (v_1^h, v_2^h, v_3^h)$ with $v_1^h, v_2^h, v_3^h \in W^h$. We then choose a standard finite element polynomial basis $\{\varphi_j\}$, $j = 1 \ldots N$ which spans the space W^h. To satisfy the well-known LBB-stability criterion for saddle-point problems, we choose Q^h as a lower-order polynomial space for the pressure, with basis $\{\psi_j\}$, $j = 1 \ldots M$, $M < N$. The approximate solutions $\boldsymbol{u}^h, p^h, \tilde{T}^h$ are now expressed as linear combinations of the basis functions

$$u_\alpha^h(\boldsymbol{x}, t) = \sum_{j=1}^{N} u_{\alpha_j}(t) \varphi_j(\boldsymbol{x}) \quad , \quad \alpha = 1, 2, 3 \tag{2.87}$$

$$\tilde{T}^h(\boldsymbol{x}, t) = \sum_{j=1}^{N} \tilde{T}_j(t) \varphi_j(\boldsymbol{x}) \tag{2.88}$$

$$p^h(\boldsymbol{x}, t) = \sum_{j=1}^{M} p_j(t) \psi_j(\boldsymbol{x}) \tag{2.89}$$

Since $\boldsymbol{v}^h$ is arbitrary, we can "generate" the three appropriate momentum equations as in [89] by choosing the sequence of test functions $\boldsymbol{v}^h = (\varphi_i, 0, 0)$,

$(0, \varphi_i, 0)$, and $(0, 0, \varphi_i)$. For each $i = 1 \ldots N$, we then have for the x_α-momentum equation (Note: Here and in the following equations, summation over repeated indices is not implied unless an actual summation sign is present.)

$$\begin{aligned}
\int_\Omega \frac{1}{Pr} \frac{\partial u_\alpha^h}{\partial t} \phi_i \, dx = & - \int_\Omega \frac{1}{Pr} \left(\boldsymbol{u}^h \cdot \nabla_\Gamma u_\alpha^h \right) \phi_i \, dx \\
& - \int_\Omega \nabla_\Gamma u_\alpha^h \cdot \nabla_\Gamma \phi_i \, dx \\
& - \sum_{\beta=1}^{3} \int_\Omega \Gamma^{(\delta_{\alpha 3} + \delta_{\beta 3})} \frac{\partial u_\beta^h}{\partial x_\alpha} \frac{\partial \phi_i}{\partial x_\beta} \, dx \\
& - \int_\Omega \Gamma^3 Ra \, \tilde{T}^h \hat{g}_\alpha \phi_i \, dx \\
& + \int_\Omega \Gamma^{\delta_{\alpha 3}} p^h \frac{\partial \phi_i}{\partial x_\alpha} \, dx \\
& - \int_{\partial \Omega_s} \Gamma^2 Ma \, \frac{\partial \tilde{T}^h}{\partial x_\alpha} \phi_i \, ds
\end{aligned} \tag{2.90}$$

where δ_{ij} is the Kronecker delta. Note that for $\alpha = 3$, ϕ_i is required to vanish on $\partial \Omega_s$, and the no-penetration boundary condition is enforced with a penalty term. Likewise, for each $i = 1 \ldots M$, we have for the mass conservation equation

$$0 = \int_\Omega (\nabla_\Gamma \cdot \boldsymbol{u}^h) \psi_i \, dx \tag{2.91}$$

Finally, for each $i = 1 \ldots N$, we have for the energy conservation equation

$$\begin{aligned}
\int_\Omega \frac{\partial \tilde{T}^h}{\partial t} \phi_i \, dx = & - \int_\Omega \left(\boldsymbol{u}^h \cdot \nabla_\Gamma \tilde{T}^h + \boldsymbol{u}^h \cdot \nabla_\Gamma T_0 \right) \phi_i \\
& - \int_\Omega \nabla_\Gamma \tilde{T}^h \cdot \nabla_\Gamma \phi_i \, dx \\
& - \int_{\partial \Omega_s} \Gamma^2 Bi \, \tilde{T}^h \phi_i \, ds
\end{aligned} \tag{2.92}$$

We have written Eqns. (2.90)–(2.92) in the particular manner shown to emphasize a splitting of the spatial and the temporal parts of the equation. These are sometimes referred to as "semi-discrete" equations at this point, since they have been discretized in space but not yet in time, the subject of the next section.

2.9 Time Discretization

In the remainder of this section, we will drop the h-superscript as it is understood we are talking about the approximate solution. Consider, as an illustration, the left-hand side of Eqn. (2.92), which may be expanded as

$$\begin{aligned} \int_\Omega \frac{\partial \tilde{T}}{\partial t}\phi_i \, dx &= \sum_{j=1}^{N} \frac{\partial \tilde{T}_j}{\partial t} \int_\Omega \phi_j \phi_i \, dx \\ &:= \boldsymbol{M}\dot{\boldsymbol{T}} \end{aligned} \tag{2.93}$$

where

$$M_{ij} := \int_\Omega \phi_j \phi_i \, dx \quad \forall \, i,j = 1 \ldots N \tag{2.94}$$

is known as the mass matrix, and $\dot{\boldsymbol{T}}$ is a $N \times 1$ vector of finite element coefficient time derivatives for the temperature variable. More generally, we can consider the big vector

$$\dot{\boldsymbol{U}} := \begin{bmatrix} \{\dot{u}_{1_j}\}_{j=1}^{N} \\ \{\dot{u}_{2_j}\}_{j=1}^{N} \\ \{\dot{u}_{3_j}\}_{j=1}^{N} \\ \{\dot{p}_j\}_{j=1}^{M} \\ \{\dot{T}_j\}_{j=1}^{N} \end{bmatrix} \tag{2.95}$$

of *all* the finite element coefficient time derivatives, which allows us to write Eqns. (2.90)–(2.92) as

$$\boldsymbol{M}\dot{\boldsymbol{U}} = \boldsymbol{f}(\boldsymbol{U}) \tag{2.96}$$

which is precisely the standard ODE form of Eqn. (B-2.2) for the θ-method discussed in Appendix 2.1. Here, the mass matrix does not depend on the vector $\boldsymbol{U}$, but it is not exactly the "standard" mass matrix either, since some entries will be scaled by $1/Pr$ and others (for the time-independent continuity constraint equations) will be zero. We need not worry about the precise form of $\boldsymbol{f}$, it can be determined by inspection of the right-hand sides of Eqns. (2.90)–(2.92).

Following the same plan and notation as in Appendix §2.1, we arrive at the θ-method time-discretization for the i^{th} x_α-momentum equation

$$\begin{aligned}
\int_\Omega \frac{1}{Pr}\frac{u_\alpha^{n+1}-u_\alpha^n}{\Delta t}\phi_i\, dx = \; & - \int_\Omega \frac{1}{Pr}\left(\boldsymbol{u}^\theta \cdot \nabla_\Gamma u_\alpha^\theta\right)\phi_i\, dx \\
& - \int_\Omega \nabla_\Gamma u_\alpha^\theta \cdot \nabla_\Gamma \phi_i\, dx \\
& - \sum_{\beta=1}^{3} \int_\Omega \Gamma^{(\delta_{\alpha 3}+\delta_{\beta 3})} \frac{\partial u_\beta^\theta}{\partial x_\alpha}\frac{\partial \phi_i}{\partial x_\beta}\, dx \\
& - \int_\Omega \Gamma^3 Ra\, \tilde{T}^\theta \hat{g}_\alpha \phi_i\, dx \\
& + \int_\Omega \Gamma^{\delta_{\alpha 3}} p^\theta \frac{\partial \phi_i}{\partial x_\alpha}\, dx \\
& - \int_{\partial\Omega_s} \Gamma^2 Ma\, \frac{\partial \tilde{T}^\theta}{\partial x_\alpha}\phi_i\, ds
\end{aligned} \tag{2.97}$$

the i^{th} mass conservation equation

$$0 = \int_\Omega (\nabla_\Gamma \cdot \boldsymbol{u}^{n+1})\psi_i\, dx \tag{2.98}$$

and the i^{th} energy conservation equation

$$\begin{aligned}\int_\Omega \frac{\tilde{T}^{n+1}-\tilde{T}^n}{\Delta t}\phi_i\,dx = &- \int_\Omega \left(\boldsymbol{u}^\theta\cdot\nabla_\Gamma\tilde{T}^\theta + \boldsymbol{u}^\theta\cdot\nabla_\Gamma T_0\right)\phi_i \\ &- \int_\Omega \nabla_\Gamma\tilde{T}^\theta\cdot\nabla_\Gamma\phi_i\,dx \\ &- \int_{\partial\Omega_s}\Gamma^2 Bi\,\tilde{T}^\theta\phi_i\,ds \end{aligned} \tag{2.99}$$

where $\tilde{T}^{n+1}$ is the (spatially-varying) temperature variable at time level $n+1$, and $\tilde{T}^\theta := \theta\tilde{T}^{n+1} + (1-\theta)\tilde{T}^n$ is the intermediate-time "interpolated" temperature field. We assume, without loss of generality, that the "old time" value $\tilde{T}^n$ is known, either from initial data or from a previous timestep.

2.10 Linearization

Consider the time-discretized momentum, mass, and energy conservation residuals of Eqns. (2.97)–(2.99) occupying a single large vector $\boldsymbol{F}$ where

$$\boldsymbol{F} := \begin{bmatrix} F^{(u_1)} \\ F^{(u_2)} \\ F^{(u_3)} \\ F^{(p)} \\ F^{(T)} \end{bmatrix} \tag{2.100}$$

and where

$$
\begin{aligned}
F_i^{(u_\alpha)} \;:=\; & \int_\Omega \frac{1}{Pr}\frac{u_\alpha^{n+1}-u_\alpha^n}{\Delta t}\phi_i \, dx \\
+ \; & \int_\Omega \frac{1}{Pr}\left(\boldsymbol{u}^\theta \cdot \nabla_\Gamma u_\alpha^\theta\right)\phi_i \, dx \\
+ \; & \int_\Omega \nabla_\Gamma u_\alpha^\theta \cdot \nabla_\Gamma \phi_i \, dx \\
+ \; & \sum_{\beta=1}^{3} \int_\Omega \Gamma^{(\delta_{\alpha 3}+\delta_{\beta 3})} \frac{\partial u_\beta^\theta}{\partial x_\alpha}\frac{\partial \phi_i}{\partial x_\beta} \, dx \\
+ \; & \int_\Omega \Gamma^3 Ra \; \tilde{T}^\theta \hat{g}_\alpha \phi_i \, dx \\
- \; & \int_\Omega \Gamma^{\delta_{\alpha 3}} p^\theta \frac{\partial \phi_i}{\partial x_\alpha} \, dx \\
+ \; & \int_{\partial\Omega_s} \Gamma^2 Ma \; \frac{\partial \tilde{T}^\theta}{\partial x_\alpha}\phi_i \, ds
\end{aligned}
\tag{2.101}
$$

for $\alpha = 1, 2, 3$ is the i^{th} x_α-momentum equation residual ($i = 1 \dots N$)

$$
F_i^{(p)} := \int_\Omega (\nabla_\Gamma \cdot \boldsymbol{u}^{n+1})\psi_i \, dx \tag{2.102}
$$

is the i^{th} continuity equation residual ($i = 1 \dots M$) and

$$
\begin{aligned}
F_i^{(T)} \;:=\; & \int_\Omega \frac{\tilde{T}^{n+1}-\tilde{T}^n}{\Delta t}\phi_i \, dx \\
+ \; & \int_\Omega \left(\boldsymbol{u}^\theta \cdot \nabla_\Gamma \tilde{T}^\theta + \boldsymbol{u}^\theta \cdot \nabla_\Gamma T_0\right)\phi_i \, dx \\
+ \; & \int_\Omega \nabla_\Gamma \tilde{T}^\theta \cdot \nabla_\Gamma \phi_i \, dx \\
+ \; & \int_{\partial\Omega_s} \Gamma^2 Bi \; \tilde{T}^\theta \phi_i \, ds
\end{aligned}
\tag{2.103}
$$

is the i^{th} energy conservation residual ($i = 1 \dots N$). Then we are seeking a solution to the nonlinear system of equations

$$
\boldsymbol{F}(\boldsymbol{U}^{n+1}) = \boldsymbol{0} \tag{2.104}
$$

for the vector of finite element coefficients $\boldsymbol{U}^{n+1}$ at time $n+1$. Henceforth, we will drop the $n+1$ superscript for simplicity of notation.

Newton's method (or more precisely the Inexact Newton method, see Appendix 2.3 for more details) is a commonly-used technique for solving nonlinear systems of equations of this sort, due primarily to its efficiency and rapid convergence characteristics near the root. Newton's method is simply to solve (approximately, and usually via an iterative Krylov solver in the case of the Inexact Newton method) linear systems of the form

$$\boldsymbol{F}'(\boldsymbol{U}_k)\delta\boldsymbol{U}_k = -\boldsymbol{F}(\boldsymbol{U}_k) \tag{2.105}$$

for iteration $k = 1, 2, \ldots$ until one or more convergence criteria are met. For example, one possible criterion is based on the relative size of $\delta\boldsymbol{U}_k := \boldsymbol{U}_{k+1} - \boldsymbol{U}_k$. In this case, convergence is detected when

$$\frac{\|\delta\boldsymbol{U}_k\|}{\|\boldsymbol{U}_k\|} < TOL \tag{2.106}$$

for some norm $\|\cdot\|$ and tolerance TOL. In Eqn. (2.105) $\boldsymbol{F}'(\boldsymbol{U}_k)$ is a linear operator (matrix) called the Jacobian, whose i,j entry is given by

$$[\boldsymbol{F}'(\boldsymbol{U}_k)]_{ij} := \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}_j}(\boldsymbol{U}_k) \tag{2.107}$$

where i,j range over the length of the big vector $\boldsymbol{U}$ of all finite element coefficients. The Jacobian matrix for the RBM equations under consideration has

the particular form

$$\boldsymbol{F}' = \left[\begin{array}{ccc|c|c} K_{u_1u_1} & \dots & K_{u_1u_3} & K_{u_1p} & K_{u_1\tilde{T}} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ K_{u_3u_1} & \dots & K_{u_3u_3} & K_{u_3p} & K_{u_1\tilde{T}} \\ \hline K_{pu_1} & \dots & K_{pu_3} & 0 & 0 \\ \hline K_{\tilde{T}u_1} & \dots & K_{\tilde{T}u_3} & 0 & K_{\tilde{T}\tilde{T}} \end{array}\right] \tag{2.108}$$

where

$$[K_{XY}]_{ij} := \frac{\partial F_i^{(X)}}{\partial Y_j} \tag{2.109}$$

denotes the entries of the submatrix obtained by differentiating the i^{th} component of the X-equation residual with respect to the j^{th} component of variable Y. Due to the particularly simple form of the finite element expansions used, it is trivial to differentiate e.g. $\tilde{T}^\theta$ with respect to $\tilde{T}_j$ (recall that we are really differentiating with respect to $\tilde{T}_j^{n+1}$ though we have dropped the superscript by this point)

$$\begin{aligned} \frac{\partial \tilde{T}^\theta}{\partial \tilde{T}_j} &= \theta \frac{\partial}{\partial \tilde{T}_j} \sum_{j=1}^{N} \tilde{T}_j \phi_j \\ &= \theta \phi_j \end{aligned} \tag{2.110}$$

For $\alpha = 1, 2, 3$ we obtain

$$[K_{u_\alpha u_\alpha}]_{ij} = \int_\Omega \frac{1}{Pr\ \Delta t} \phi_j \phi_i \, dx \tag{2.111}$$

$$\begin{aligned} &+ \theta \int_\Omega \frac{1}{Pr} \left(\boldsymbol{u}^\theta \cdot \nabla_\Gamma \phi_j + \Gamma^{\delta_{\alpha 3}} \phi_j \frac{\partial u_\alpha^\theta}{\partial x_\alpha} \right) \phi_i \, dx \\ &+ \theta \int_\Omega \nabla_\Gamma \phi_j \cdot \nabla_\Gamma \phi_i \, dx \\ &+ \theta \int_\Omega \Gamma^{2\delta_{\alpha 3}} \frac{\partial \phi_j}{\partial x_\alpha} \frac{\partial \phi_i}{\partial x_\alpha} \, dx \end{aligned} \tag{2.112}$$

and for $\alpha \neq \beta$,

$$\begin{aligned}\left[K_{u_\alpha u_\beta}\right]_{ij} &= \theta \int_\Omega \frac{1}{Pr} \Gamma^{\delta_{\beta 3}} \frac{\partial u_\alpha^\theta}{\partial x_\beta} \phi_j \phi_i \, dx \\ &+ \theta \int_\Omega \Gamma^{(\delta_{\alpha 3}+\delta_{\beta 3})} \frac{\partial \phi_j}{\partial x_\alpha} \frac{\partial \phi_i}{\partial x_\beta} \, dx \end{aligned} \tag{2.113}$$

for the velocity-velocity coupling,

$$\left[K_{u_\alpha p}\right]_{ij} = -\theta \int_\Omega \Gamma^{\delta_{\alpha 3}} \psi_j \frac{\partial \phi_i}{\partial x_\alpha} \, dx \tag{2.114}$$

for the velocity-pressure coupling, and

$$\begin{aligned}\left[K_{u_\alpha \tilde{T}}\right]_{ij} &= \theta \int_\Omega \Gamma^3 Ra \; \hat{g}_\alpha \phi_j \phi_i \, dx \\ &+ \theta \int_{\partial\Omega_s} \Gamma^2 Ma \; \frac{\partial \phi_j}{\partial x_\alpha} \phi_i \, ds \end{aligned} \tag{2.115}$$

for the velocity-temperature coupling for the x_α-momentum equation. Similarly, for the entries of the continuity equation we obtain, for $\alpha = 1, 2, 3$

$$\left[K_{p u_\alpha}\right]_{ij} = \int_\Omega \Gamma^{\delta_{\alpha 3}} \frac{\partial \phi_j}{\partial x_\alpha} \psi_i \, dx \tag{2.116}$$

For the energy conservation equation's temperature-velocity coupling we obtain

$$\left[K_{\tilde{T} u_\alpha}\right]_{ij} = \theta \int_\Omega \Gamma^{\delta_{\alpha 3}} \left(\frac{\partial \tilde{T}^\theta}{\partial x_\alpha} + \frac{\partial T_0}{\partial x_\alpha} \right) \phi_j \phi_i \, dx \tag{2.117}$$

and finally, for the temperature-temperature coupling

$$\begin{aligned}\left[K_{\tilde{T}\tilde{T}}\right]_{ij} &= \int_\Omega \frac{1}{\Delta t} \phi_j \phi_i \, dx \\ &+ \theta \int_\Omega \left(\boldsymbol{u}^\theta \cdot \nabla_\Gamma \phi_j \right) \phi_i \, dx \\ &+ \theta \int_\Omega \nabla_\Gamma \phi_j \cdot \nabla_\Gamma \phi_i \, dx \\ &+ \theta \int_{\partial\Omega_s} \Gamma^2 Bi \; \phi_j \phi_i \, ds \end{aligned} \tag{2.118}$$

Note that the Jacobian matrices are always evaluated at the previous Newton iterate $\boldsymbol{U}_k$. Thus, in order to evaluate (say) $\boldsymbol{u}^\theta$ in Eqn. (2.118) one has $\boldsymbol{u}^\theta = \theta\boldsymbol{u}_k + (1-\theta)\boldsymbol{u}^n$, and similarly for the other variables.

Chapter 3

Unsteady Rayleigh-Bénard-Marangoni Results

The numerical results which follow in this and the next Chapter are broken into two different parts: solutions obtained with time-accurate solves of the unsteady equations having fixed parameter values (Ch. 3), and solutions obtained by following solution paths in parameter space with pseudo-arclength continuation (Ch. 3, see also Appendix §4 for the details of the arclength continuation scheme.) Solving for solution paths in parameter space is done using the steady form of the governing equations.

The reasons for using both the steady and unsteady forms of the governing equations are several. One reason (perhaps the most important) is the role played by the selection of an initial condition in a numerical method. Since the simplest initial condition, zero, is in fact a solution to the governing equations (in our formulation) it is not an ideal starting point for our evolution equations: a "zero in" yields a "zero out" result. This imposes the requirement of selecting a non-zero initial condition.

For a time-dependent calculation, one is essentially free to impose *any* initial condition, and let the equations involve in time to a steady state. (We note that in a time-dependent calculation, the initial condition selected can

influence the final steady state which is obtained. This occurs for nonlinear systems of equations with multiple *stable* steady states.) This is not the case for a continuation method, in which the initial condition is typically required to be *on* a solution branch, and thus already be a (non-trivial) steady-state solution to the governing equations. Hence, we have the first possible connection between the steady and unsteady equations: the unsteady equations with an arbitrary initial guess take us to some point on some solution branch (which we cannot precisely control of course) and this then serves as an initial condition for a continuation scheme that allows us to travel along said branch.

Another reason for employing both the steady and unsteady forms of the governing equations is related to the task of reproducing experimental results. In general it is not possible to know the true initial state of the experimental system (including random disturbances) and therefore evolving the unsteady equations from an essentially random initial state in an attempt to achieve an experimentally-observed steady state may fail, due to the existence of multiple solutions mentioned above. Continuation methods, which provide fine-grained control over small changes in the system parameters, are ideal for this purpose. However, it is also well-known that continuation methods will converge to unsteady (and therefore physically unrealizable in a laboratory) solutions as well. Determining the linear stability of solution branches computed via continuation methods typically requires access to the unsteady equations, in particular the associated mass matrix, and the solution of a generalized eigenvalue problem. Thus, we have a second important connection

between the two forms of the governing equations: the steady equations allow us to trace a solution branch in parameter space (which we arrived at in the first place by a solution to the unsteady equations) and then the unsteady equations can be used to determine the stability of the branch.

In the remainder of this chapter, we discuss time-accurate calculation of the discretized form of Eqns. (2.78)–(2.80), for varying aspect ratios Γ, and using the following fixed values for the other parameters: $Ma = 92$, $Ra = 30$, $Pr = 880$, $Bi = 0.2$. These values are representative of those investigated experimentally by Koschmeider [113] and Medale and Cerisier [126]. In general, the results (which are all for three-dimensional containers) will generally be presented as "top-down" views of the free surface with plots of either the numerically computed free surface velocity field vectors or the surface temperature contours. Where applicable, we have also included images of experimental results for similar flows obtained in the laboratory. The time-accurate simulations are computed using the adaptive "step-doubling" method described in Appendix §2.2.

3.1 Grid Resolution Study

In order to address the question of whether the selected uniform grid resolution is able to accurately represent the convection patterns for a given aspect ratio, we computed the steady-state solution for $\Gamma = 8$ (and the other parameters listed above) on both a 20×20×2 and a 30×30×2 uniform grid and compared the results to check for mesh convergence. The computed surface

temperature contours at a particular timestep are shown in Fig. 3.1. The two solutions appear to be in reasonable agreement.

In addition, we have plotted several side-by-side comparisons of the coarse and fine mesh surface velocity fields and temperature contours in Figs. 3.2–3.4. Once again, the solutions match reasonably well at each of the timesteps shown (the same fixed timestep was used for both simulations) and at these resolutions the grid does not appear to change the basic configuration of the convection cells. On the basis of this rather non-rigorous comparison, we decided to conduct the uniform grid simulations which follow on the coarser ($20\times20\times2$) of the two meshes considered. The overall problem size for the coarser mesh was much less than the fine one, yielding an appreciable savings in computation time.

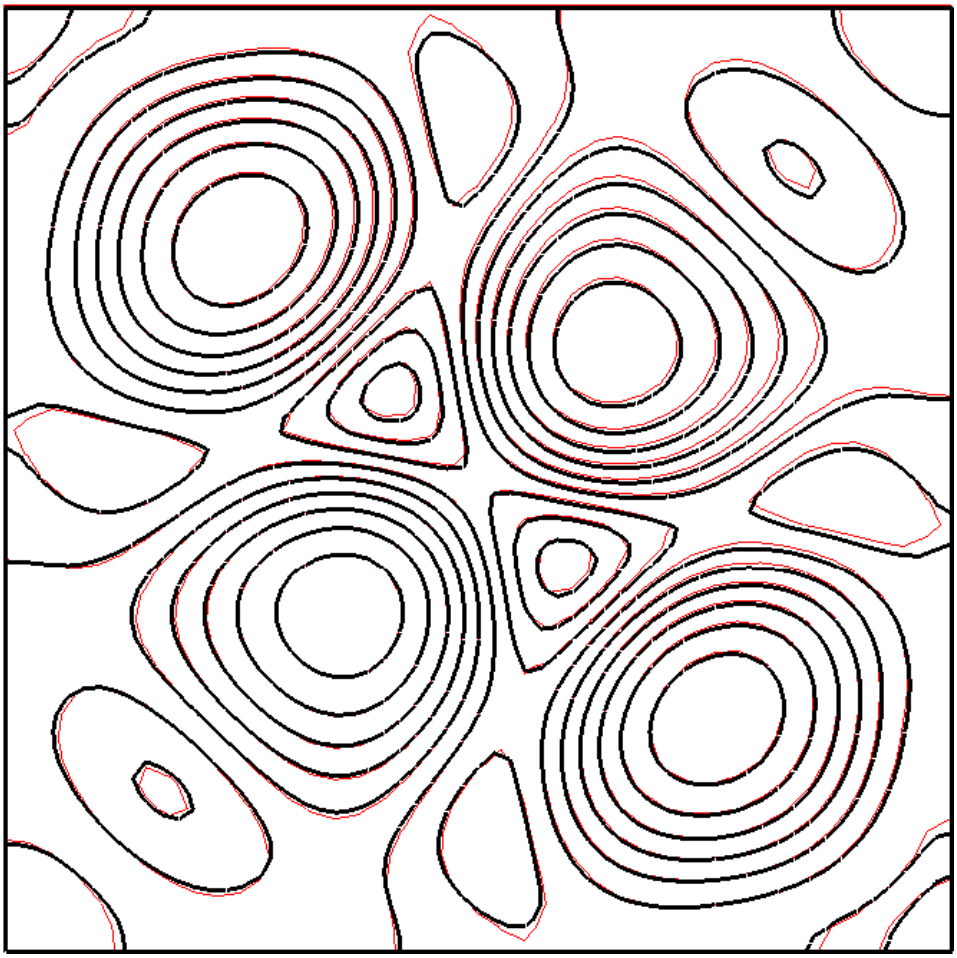


Figure 3.1: Comparison of coarse (20×20×2, thin red contours) and fine (30×30×2, thick black contours) grid solutions at a particular timestep of a time-accurate calculation.

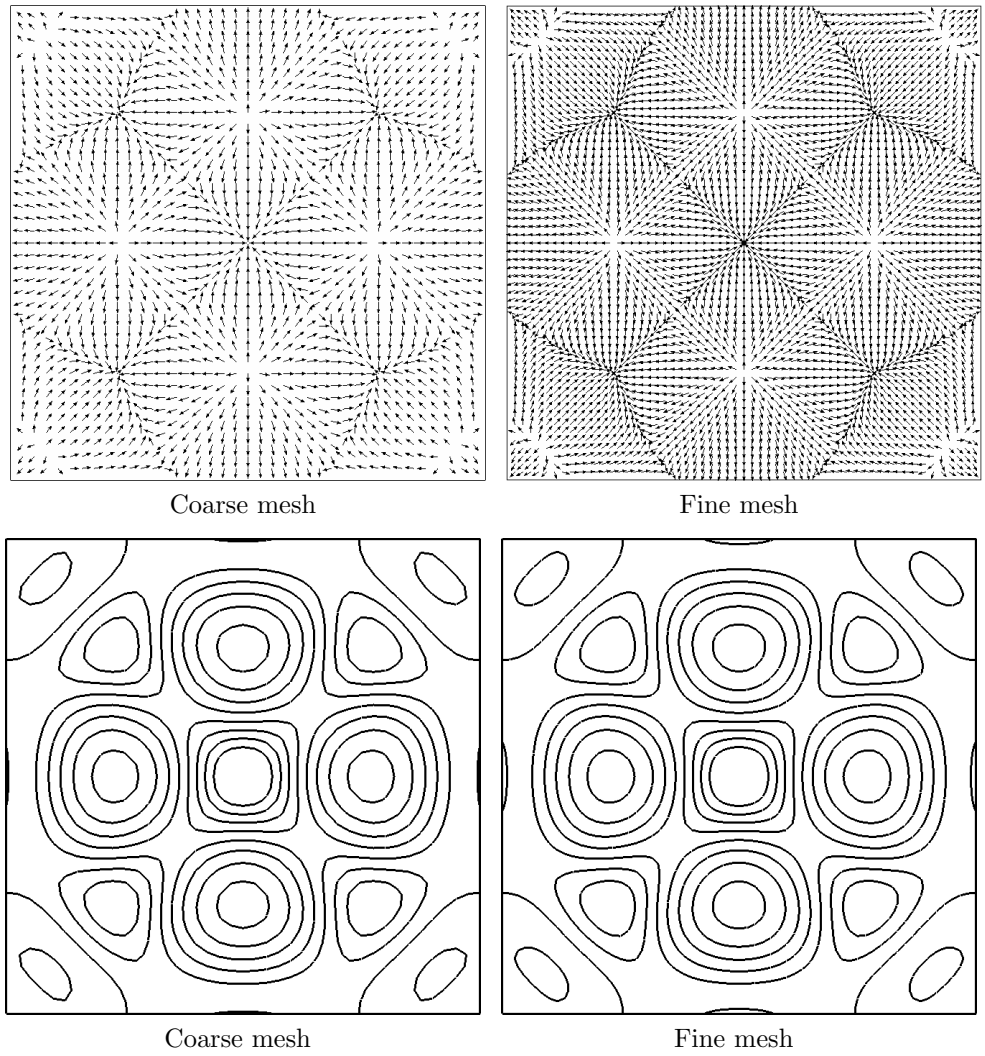


Figure 3.2: Surface velocity and temperature field comparison. There are 10 contour levels with $T \in [-0.115, 0.135]$. The largest temperature values are at the centers of the convection cells.

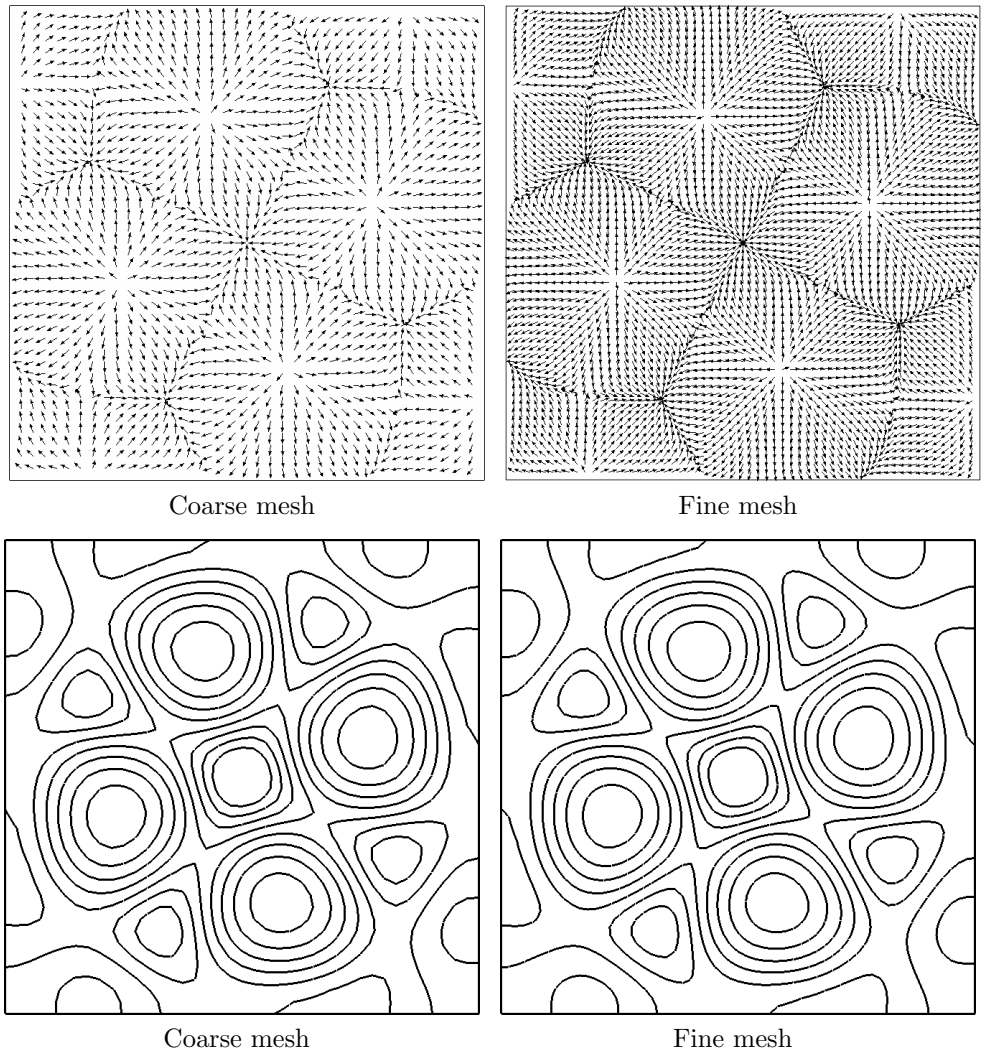


Figure 3.3: Surface velocity and temperature field comparison (at a later time than shown in Fig. 3.2.)

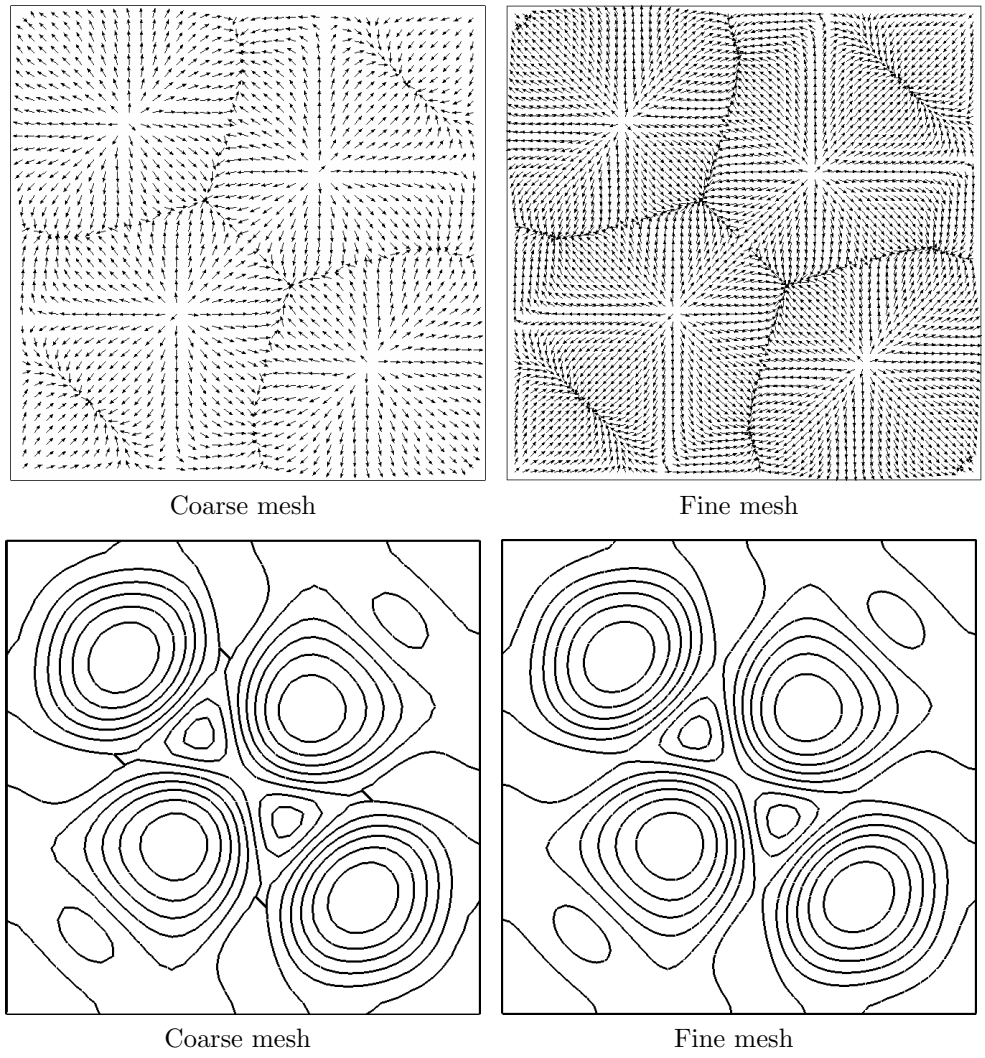


Figure 3.4: Surface velocity and temperature field comparison (at a later time than shown in Fig. 3.3.)

3.1.1 Uniform Refinement in the Vertical Direction

Note that in the previous section, there are only two elements in the vertical (x_3) direction. We employ here and in all the following simulations the classical 27-node Taylor-Hood elements (tri-quadratic velocity, tri-linear pressure bases) and therefore with only two elements through the depth it is certainly possible that (at least) the pressure may be inadequately approximated.

In order to examine the effects of having only two elements through the depth, we considered two additional levels of uniform refinement in the vertical direction, i.e. in addition to the 20×20×2 mesh considered in the previous section, we also considered 20×20×4 and 20×20×8 grids, holding the other parameters of the problem fixed. We compare contour lines of the different variables along a y cut-plane through the center of the domain, as shown in Fig. 3.5.

In Fig. 3.6, contours of the scaled temperature $\tilde{T}$ (left) and the pressure p (right) are shown. Note that this solution comes from an earlier timestep than the one considered in the preceding section (§3.1). A pseudo-stable five-cell solution exists at this early time, which eventually transitions into the stable, four-cell solution shown previously. The same qualitative behavior is present in all three of the temperature images: a warm upwelling of fluid sits in the center of the domain surrounded by two cold, downwelling regions. The contour line at the extreme left and right-hand sides of the image varies appreciably between the 20×20×2 and 20×20×4 solutions, while the 20×20×4

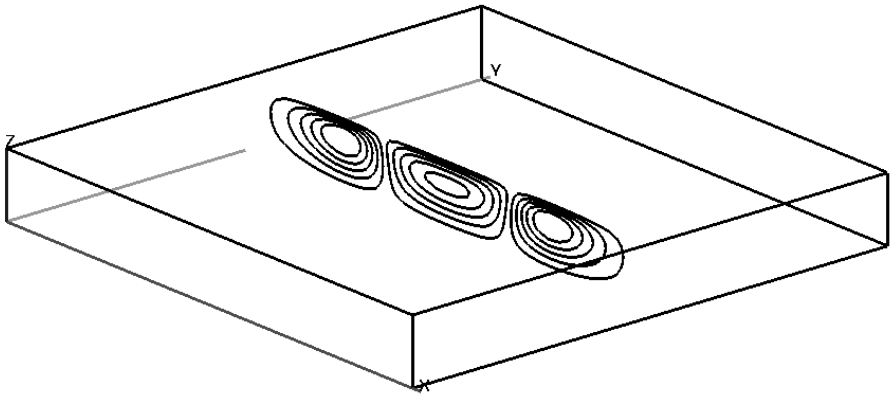


Figure 3.5: Three-dimensional view showing the location of the y cut-plane used for comparing solutions with different resolutions in the vertical direction. Comparisons of pressure and temperature contours are given in Fig. 3.6.

and 20×20×8 are in closer agreement. For the pressure contours, the largest disagreement between the different grids appears to be in the center of the domain for the 20×20×2 and 20×20×4 cases. The linear approximation of the pressure over two finite elements is clearly very coarse in this case but the qualitative behavior of all the solutions is nonetheless the same.

In Fig. 3.7, on the left-hand side we plot the velocity vector components tangent to the cut-plane, while on the right the third (vertical) component of

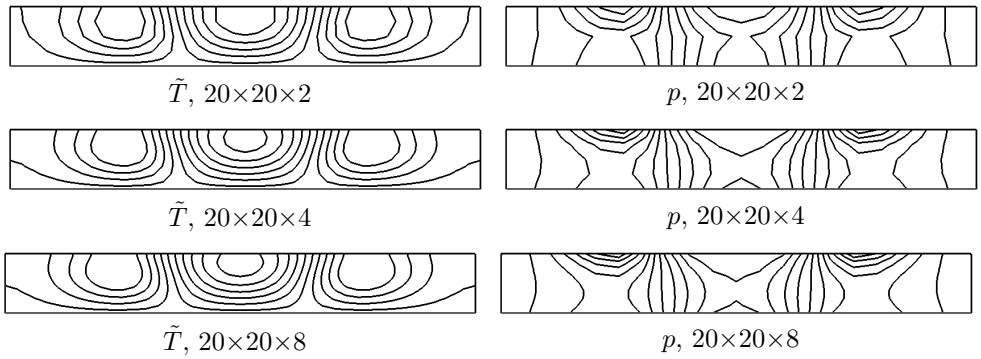


Figure 3.6: Temperature (left) and pressure (right) contours for a sequence of meshes uniformly refined in the vertical direction. The same contour levels ($\tilde{T} \in [-1.213 \times 10^{-1}, 1.479 \times 10^{-1}]$ and $p \in [-1.866 \times 10^{3}, 1.442 \times 10^{3}]$) are used for each figure.

velocity u_3 is shown. The velocity vectors in each of the three cases confirm that there is a warm upwelling of fluid in the center of the domain surrounded by two cold downwelling regions. Regions of vortical flow near the top of the domain in the center, left, and right of the container can be distinguished in each of the figures, though they are certainly better resolved in the 20×20×4 and 20×20×8 cases. The vertical component of velocity is reasonably well-captured on each of the meshes shown. The structure of this component of the flow is similar to the temperature field, and the same comments made previously apply here as well.

We next turn to the quality of some computed measures of the solution which are of interest here. Specifically, we computed the Nusselt number Nu, as defined in §5.1, and the kinetic energy KE, which is defined in the usual

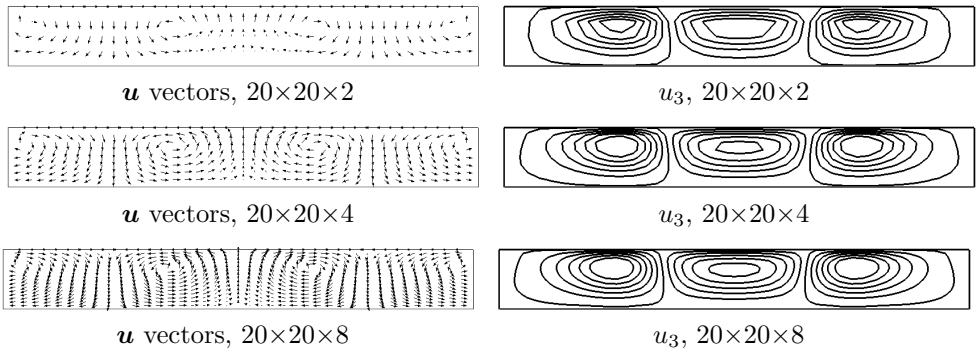


Figure 3.7: Velocity vectors (left) and contours of the vertical velocity component u_3 (right) contours for a sequence of meshes uniformly refined in the vertical direction. The same contour levels ($u_3 \in [-11.109, 8.930]$) are used in each figure.

way as

$$KE := \frac{1}{2} \int_{\Omega} |\boldsymbol{u}|^2 \, dx \tag{3.1}$$

We then analyzed the effects that uniformly refining the grid had on these quantities. Since we frequently categorize different solutions according to Nu, its accurate calculation is of interest. However, the absolute value of Nu is not as important as the trend Nu follows when we vary the nondimensional parameters. A sequence of solutions which captures the correct Nu may be considered "acceptable" even though the true Nu value may not be known to a high degree of accuracy.

The time histories in Fig. 3.8 are presented in nondimensional time units. For this particular case of interest, and assuming parameter values similar to those used by Koschmieder and Prahl, one nondimensional time unit corresponds to about 640 real-time seconds. In this study, we employed a fixed

nondimensional timestep of 0.025, or about 16 real-time seconds. Under these conditions, the initial convection pattern sets up fairly quickly and allows for rapid comparison of a non-trivial solution on different grids. The timestepping scheme is unconditionally-stable backward Euler.

Examination of Fig. 3.8 immediately reveals that the two quantities of interest are highly positively correlated. In other words, a solution with a large Nu value implies one with a large KE value and vice-versa. Second, we note that while all three meshes appear to capture the general non-decreasing nature of the quantities of interest, the final-time coarse grid solution differs from the finest grid final-time solution by approximately 17% for Nu and 15% for KE. The intermediate grid solution obviously agrees much more closely with the fine-grid solution than the coarse-grid solution.

When deciding which uniform grid to use in the following studies, in addition to the accuracy of the resulting solutions, we must also take into account the computational expense incurred by each of the different grids. For the RBM convection problem with a uniform grid of $N_x \times N_y \times N_z$ elements, the total number of degrees of freedom (N. DOFs) is given by the formula

$$\begin{aligned} \text{N. DOFs} &:= 4(2N_x+1)(2N_y+1)(2N_z+1) \\ &+ (N_x+1)(N_y+1)(N_z+1) \end{aligned} \tag{3.2}$$

since there are four $(u_1, u_2, u_3, \tilde{T})$ tri-quadratic variables and one (p) tri-linear variable.

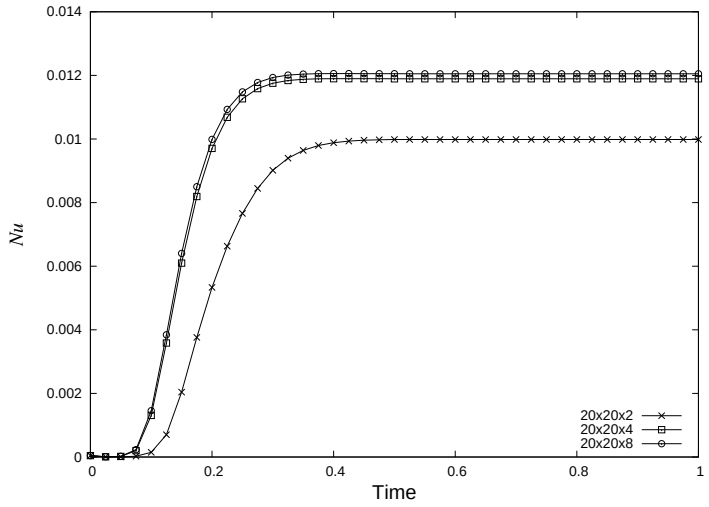
The number of degrees of freedom in each of the meshes considered here

Table 3.1: Number of degrees of freedom (N. DOFs) in each grid used in the vertical uniform grid refinement study. The number of degrees of freedom is given by the formula of Eqn. (3.2).

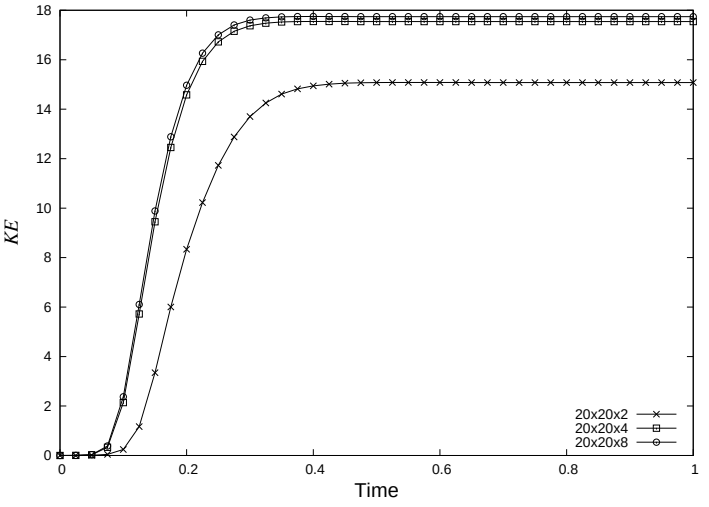
Mesh	N. DOFs
20×20×2	34,943
20×20×4	67,721
20×20×8	118,277

is given in Table 3.1. The problem size is approximately doubled with each uniform refinement in the vertical direction. The intermediate (20×20×4) grid appears to provide a reasonable compromise in terms of accuracy for the given cost: its solution differs from the fine grid solution by only about 1% (1.3% in Nu, 1.1% in KE) for about twice the computational cost of the coarse mesh.

In this work, however, we are primarily interested in mapping out large regions of parameter space, and therefore, as mentioned previously, low cost in the computations is of relatively greater importance than their absolute accuracy, so long as the general trends are captured. For this reason, in the following calculations we have employed the coarsest of the three meshes considered here. We intend, through further calculations employing adaptive mesh refinement, to investigate this issue of solution accuracy in greater detail.



Nu vs. nondimensional time for the vertical uniform refinement study



KE vs. nondimensional time for the vertical uniform refinement study

Figure 3.8: Comparison of Nu and KE values for grids with different vertical resolutions. One obvious trend we notice from these two figures is that the Nu and KE are well-correlated: one does not obtain appreciably more information from computing and comparing both measures of the solution.

3.2 Time-Accurate Calculation, $\Gamma = 6$

Before moving on to describe the time accurate calculation for the $\Gamma = 6$ aspect ratio, we shall make a few introductory comments about the details of the numerical method and the notation used. As mentioned in the introductory remarks of this chapter, a non-trivial initial condition is required to start the computation. While it is possible to initialize the solution with a random initial condition, this technique has the drawback of being difficult to reproduce when e.g. the initial mesh changes. (On the other hand, random thermal noise is almost certainly present in any numerical experiment and is probably actually responsible for the onset of convection in most cases.)

Instead of the random initial condition, in the following time-accurate calculations we have employed the condition

$$\tilde{T}(t=0) := Ax_3 \sin(3\pi x_1) \sin(3\pi x_2) \tag{3.3}$$

where $A = 2.5 \times 10^{-4}$ is a constant, relatively small, amplitude. The other system variables are initialized to zero. This particular form of the initial thermal perturbation satisfies the Dirichlet boundary condition $\tilde{T}(x_3 = 0) = 0$ and hence does not "activate" too strongly the penalty boundary condition terms and therefore does not cause a large initial residual. The choice of the third sinusoidal mode is not special; indeed using another wave-number would possibly lead to a different steady-state solution.

We next discuss our technique for "detecting" a steady-state solution. One must be slightly careful with this procedure, to avoid erroneously detecting

"almost" steady states as being steady. We use the norm of the difference between successive values of the full solution vector $\boldsymbol{U}$ (see Eqn. (2.95) for a definition of the solution vector) to estimate the distance of the most recent two solutions from being steady-state. In other words, we detect steady state when

$$\left\| \frac{\partial \boldsymbol{U}}{\partial t} \right\| \approx \frac{\|\boldsymbol{U}^{n+1} - \boldsymbol{U}^{n}\|}{\Delta t} \tag{3.4}$$

is driven to around 10^{-13} to 10^{-15} (approximately machine precision.) We denote this approximation to the time derivative as $\frac{\|\Delta \boldsymbol{U}\|}{\Delta t}$ in the following figures, and where the discrete l_2 norm is employed. Again, we stress the importance of driving the approximation of the time derivative given in Eqn. (3.4) to as small a value as possible, and also checking to be sure that it remains below this value for several steps, to avoid the detection of incorrect steady states.

Let us now describe in some detail the temporal evolution of the time-accurate solution of the unsteady equations for aspect ratio $\Gamma = 6$. This particular aspect ratio provides an interesting view into how convection states transition from one to another in time. Also, as we shall see after the discussion of some additional aspect ratios, its evolution behavior is typical of these types of problems. In Fig. 3.9 we have plotted values of the (logarithms of) both the adaptively-selected timestep and the approximated time derivative as described previously. As would be expected, the behavior of the two curves is roughly opposite: the timestep increases to its maximum value (signified by the flat segments of the timestep curve) when the time derivative is relatively

small, and vice-versa. In addition, near the point marked 1 on the plot, we note that the solution is apparently pseudo-steady.

Point 1 on Fig. 3.9 corresponds to the single convection cell shown in Fig. 3.10. It becomes unstable around timestep 50, and gradually increases in size vertically until it splits into two convection cells around timestep 190, as shown in Fig. 3.11. This transition process takes roughly half the entire simulation time. These two convection cells then begin to grow horizontally until they approximately fill the domain at timestep 195, as shown in Fig. 3.12.

These long horizontal "roll" cells then split into four coherent convection cells, which is just captured at timestep 196, shown in Fig. 3.13. The dynamics of this splitting occur relatively rapidly. The four convection cells increase in strength but do not do so completely symmetrically. The beginning of this asymmetry is visible by timestep 197, which is shown in Fig. 3.14, and becomes even more obvious by timestep 200 (see Fig. 3.15). The larger two of the four asymmetric convection cells eventually grow into the roughly circular-shaped cells seen in the final state at timestep 283, and shown in Fig. 3.16. As we can see in Fig. 3.9 around point 3, the time derivative is nearly machine zero and has remained there for several timesteps.

The time evolution of this case is representative of other aspect ratios as well. In the following sections, we will see that the plot of the time derivative encounters several local minima (such as the local minimum at point 1) before it achieves machine precision. These local minima can in turn be related to points on the continuation diagrams which will come in a later section, and

demonstrate how the unstable steady states are closely related to the time evolution properties of the simulation.

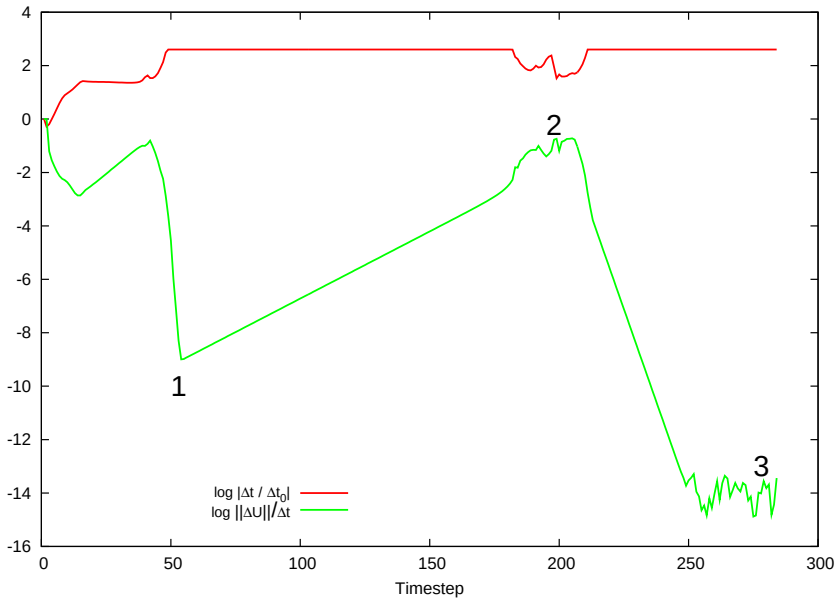
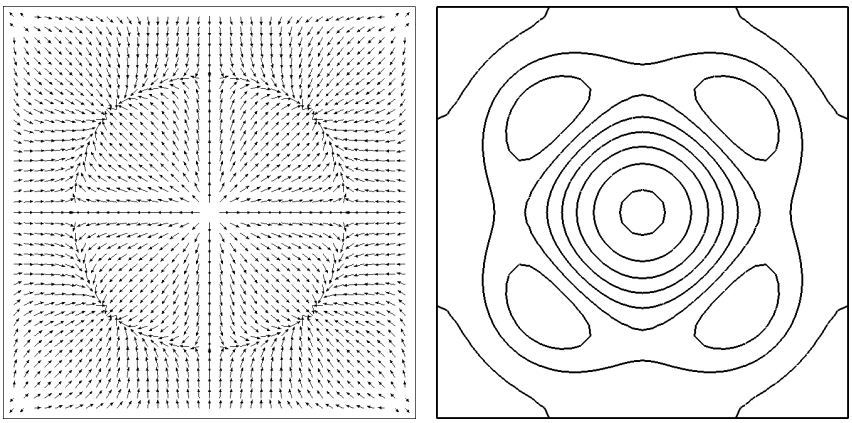
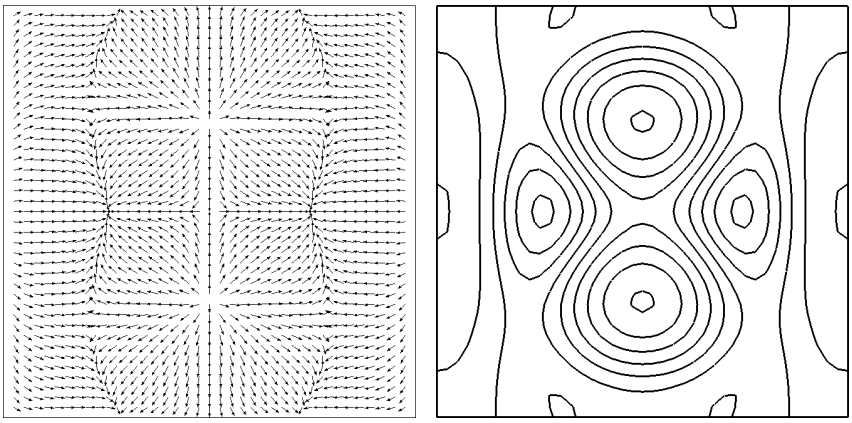


Figure 3.9: The logarithm of the adaptively selected Δt (scaled by the initial timestep) and the approximate measure of the time derivative (described in the text) for the $\Gamma = 6$ case. We observe that around timestep 50, the estimate of $\partial U/\partial t$ reaches a minimum value and slowly begins climbing again. The solution at timestep 50 is a single central convection cell. During the climbing period, from about timestep 50 to timestep 175, Δt remains fixed at the largest-allowable value, and the single convection cell remains the solution. Some time around timestep 175, the single convection cell splits into two, then briefly four convection cells before assuming a stable diagonal two-cell structure. The various intermediate solutions are detailed in the images below.



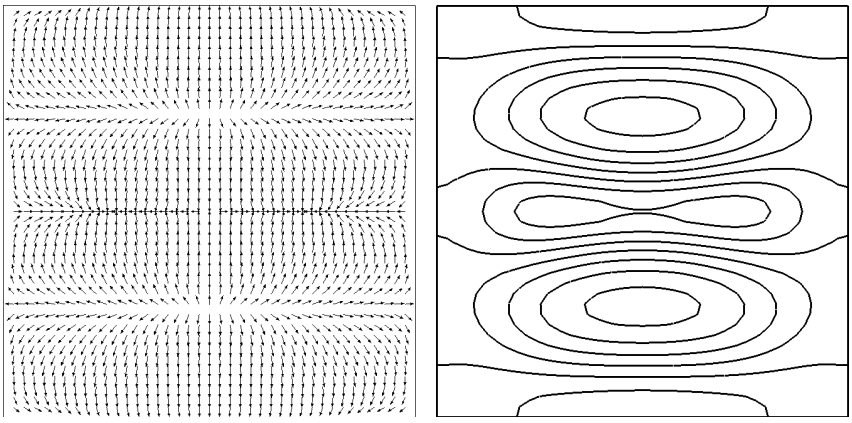
Velocity Vectors Temperature Contours

Figure 3.10: Surface velocity and temperature fields on the 20×20×2 grid at timestep 50 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.0978, 0.122]$. This state corresponds to the state marked with the number 1 in Fig. 3.9.



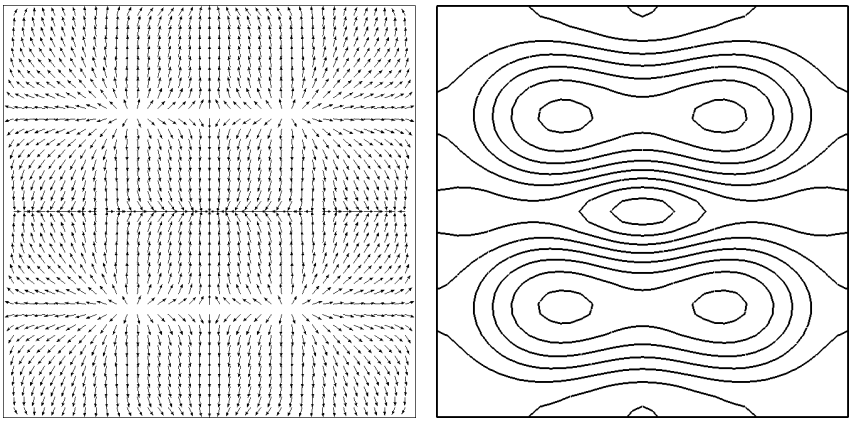
Velocity Vectors Temperature Contours

Figure 3.11: Surface velocity and temperature fields on the 20×20×2 grid at timestep 190 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.0978, 0.122]$. This state occurs near point 2 in Fig. 3.9.



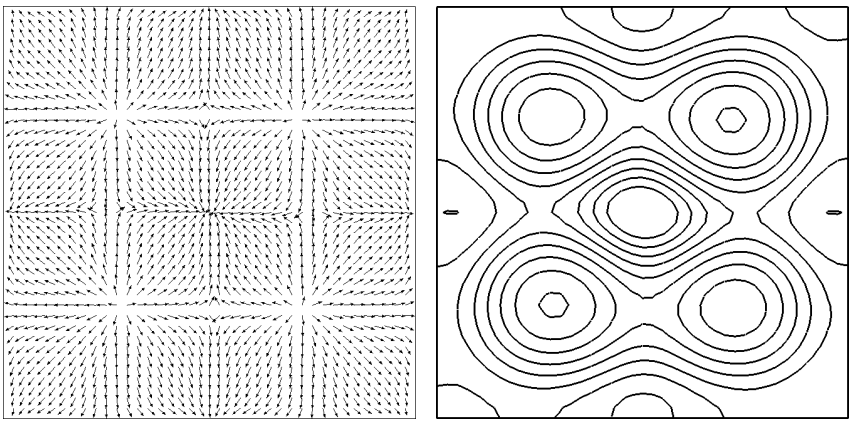
Velocity Vectors Temperature Contours

Figure 3.12: Surface velocity and temperature fields on the 20×20×2 grid at timestep 195 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.0978, 0.122]$. This state occurs near point 2 in Fig. 3.9.



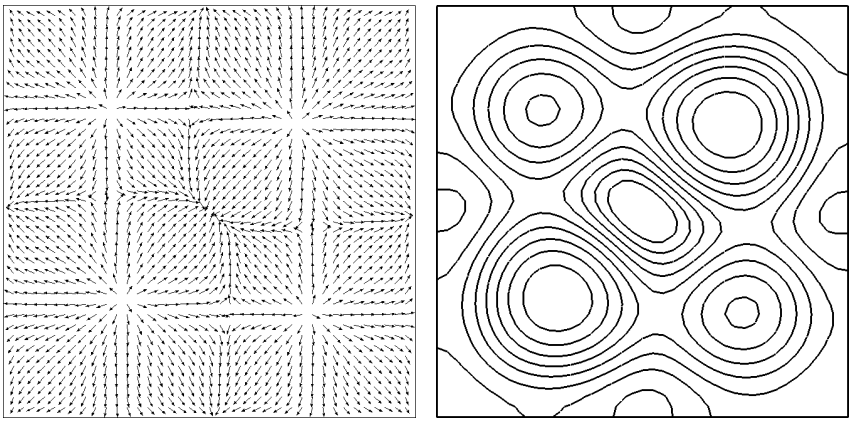
Velocity Vectors Temperature Contours

Figure 3.13: Surface velocity and temperature fields on the 20×20×2 grid at timestep 196 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.0978, 0.122]$. This state occurs near point 2 in Fig. 3.9.



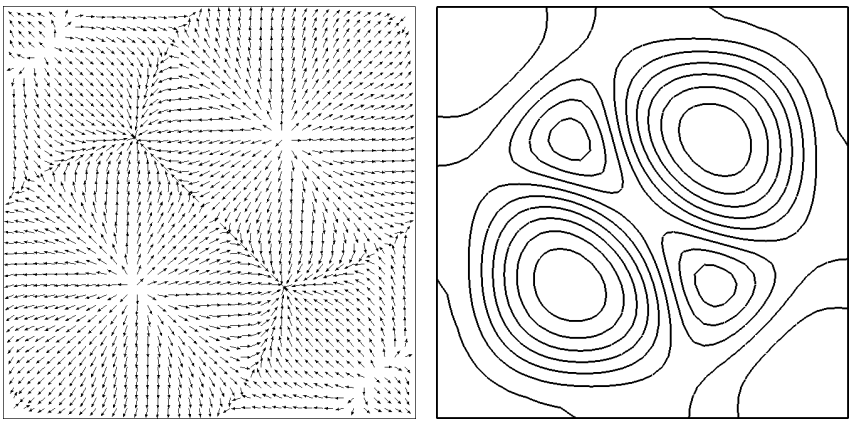
Velocity Vectors Temperature Contours

Figure 3.14: Surface velocity and temperature fields on the 20×20×2 grid at timestep 197 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.0978, 0.122]$. This state occurs near point 2 in Fig. 3.9.



Velocity Vectors Temperature Contours

Figure 3.15: Surface velocity and temperature fields on the 20×20×2 grid at timestep 200 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.0978, 0.122]$. This state occurs near point 2 in Fig. 3.9.



Velocity Vectors Temperature Contours

Figure 3.16: Surface velocity and temperature fields on the 20×20×2 grid at timestep 283 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.0978, 0.122]$. This state corresponds to the last state (marked with the number 3 in Fig. 3.9).

3.3 Time-Accurate Calculation, $\Gamma = 6.75$

Moving on to the slightly larger aspect ratio of $\Gamma = 6.75$, we observe some similarities and also some differences between this and the $\Gamma = 6$ case. Fig. 3.17, which shows the adaptively-selected timesteps and numerically-estimated time derivative, now appears to have four "intermediate" steady states (1–4 in Fig. 3.17) leading up to the final state (5) which does reach machine precision.

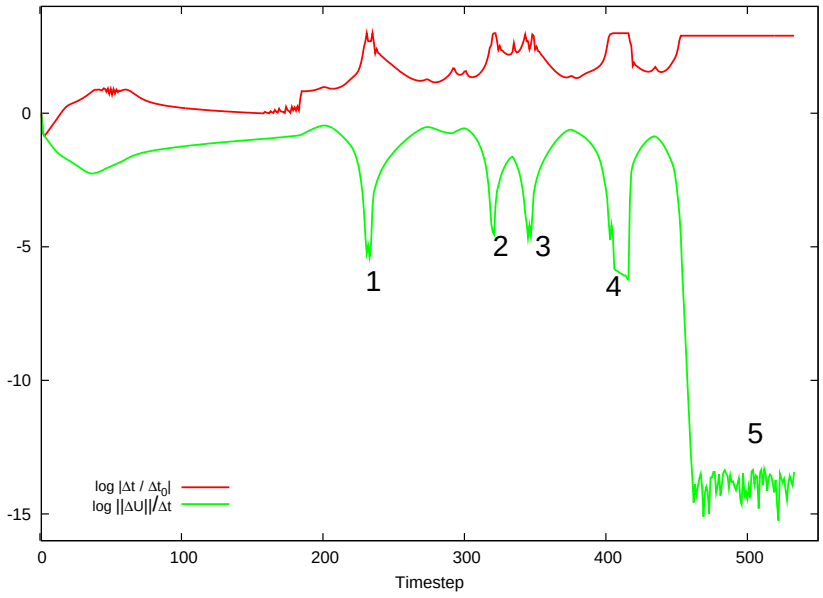
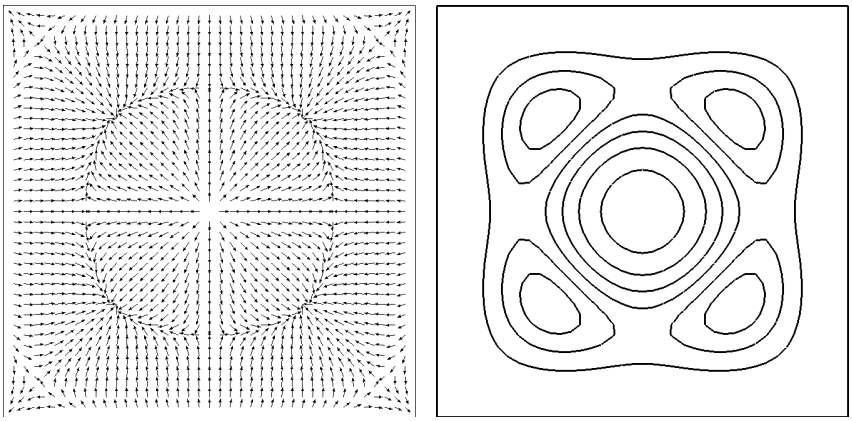


Figure 3.17: The logarithm of the adaptively selected Δt (scaled by the initial timestep) and the approximate measure of the time derivative (described in the text) for the $\Gamma = 6.75$ case.

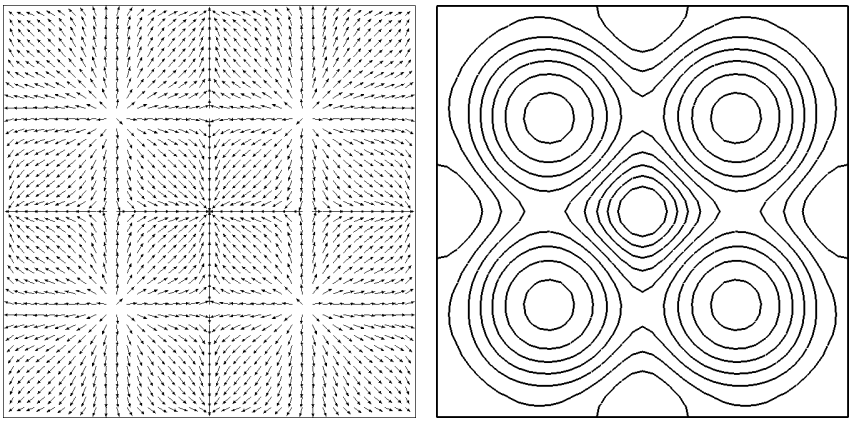
The first state (marked 1 in Fig. 3.17) occurs around timestep 230 and consists of the single central convection cell shown in Fig. 3.18. This single central cell is in fact quite similar to the initial state observed in the $\Gamma = 6$ case. Unlike the $\Gamma = 6$, case the next pseudo-steady state observed is the symmetric four-cell solution (marked 2 in Fig. 3.17) which occurs around timestep 315 and is shown in Fig. 3.19. We note also that this particular configuration has been observed experimentally, but it does not appear to be a truly stable steady-state, at least for this aspect ratio. The symmetric 4-cell solution loses symmetry and eventually transitions to (point 3 in Fig. 3.17) an asymmetric 4-cell solution around timestep 344 which is shown in Fig. 3.20.

From this configuration, one of the smaller corner cells decreases in size and the solution transitions to a three-cell pseudo-steady solution at point 4 on Fig. 3.17 around timestep 398. This pseudo-steady solution is shown in Fig. 3.21. The three-cell configuration then rotates slightly to achieve a different three-cell configuration (point 5 in Fig. 3.17) around timestep 533. This configuration finally drives the time derivative to machine zero. We note again that several of these intermediate solutions correspond to solutions on several different branches in parameter space which will be discussed later.



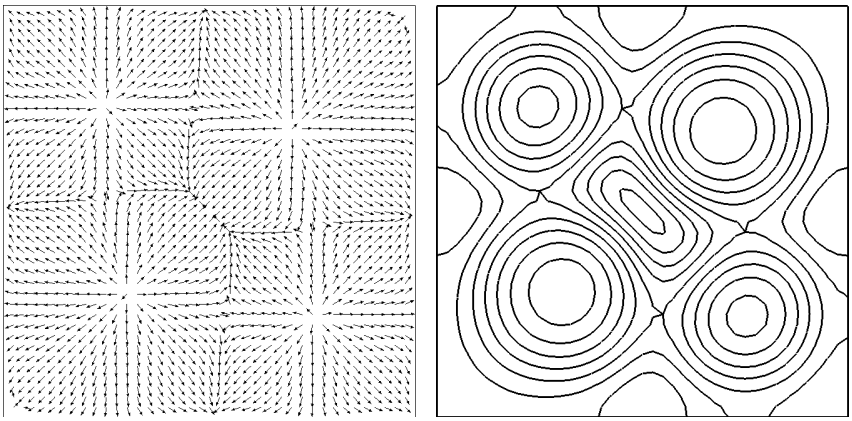
Velocity Vectors Temperature Contours

Figure 3.18: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 6.75$ case near point 1 on Fig. 3.17. The temperature contours lie in the range $\tilde{T} \in [-0.132, 0.135]$.



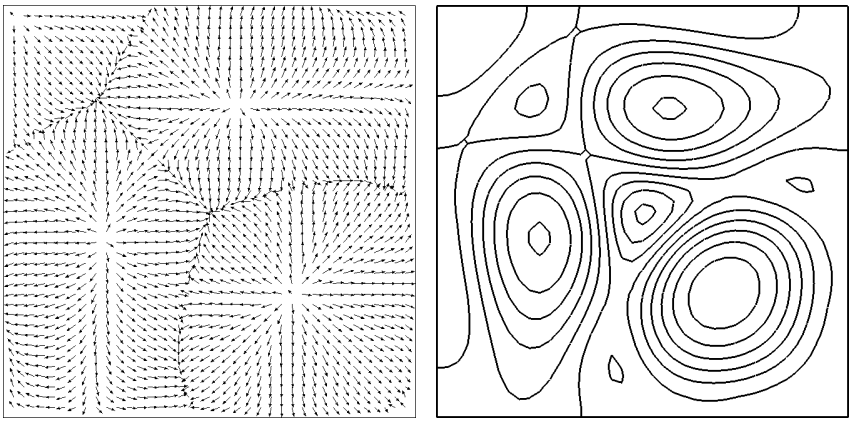
Velocity Vectors Temperature Contours

Figure 3.19: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 6.75$ case near point 2 on Fig. 3.17. The temperature contours lie in the range $\tilde{T} \in [-0.132, 0.135]$.



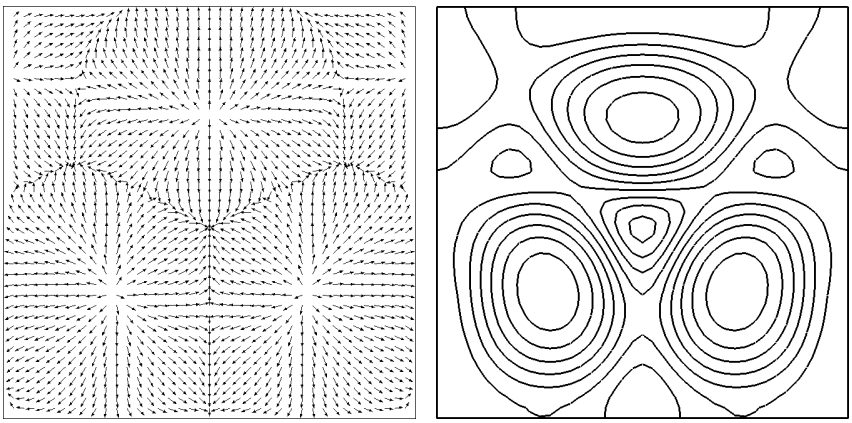
Velocity Vectors Temperature Contours

Figure 3.20: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 6.75$ case near point 3 on Fig. 3.17. The temperature contours lie in the range $\tilde{T} \in [-0.132, 0.135]$.



Velocity Vectors Temperature Contours

Figure 3.21: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 6.75$ case near point 4 on Fig. 3.17. The temperature contours lie in the range $\tilde{T} \in [-0.132, 0.135]$.



Velocity Vectors Temperature Contours

Figure 3.22: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 6.75$ case near point 5 on Fig. 3.17. The temperature contours lie in the range $\tilde{T} \in [-0.132, 0.135]$.

3.4 Time-Accurate Calculation, $\Gamma = 7.75$

After increasing the aspect ratio once again to $\Gamma = 7.75$, we observe a temporal evolution with two intermediate states, as shown in Fig. 3.23. State 1, which occurs around timestep 212, is the symmetric 4-cell pattern pictured in Fig. 3.24.

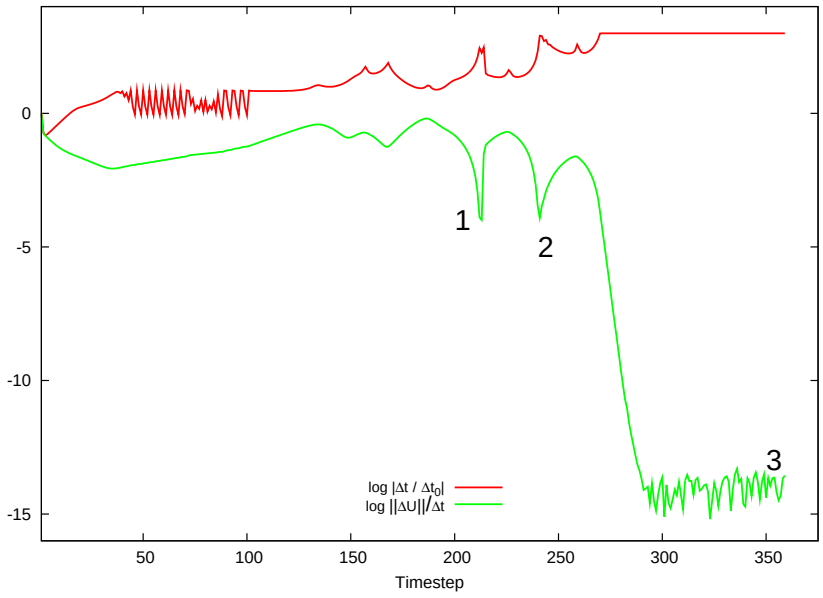
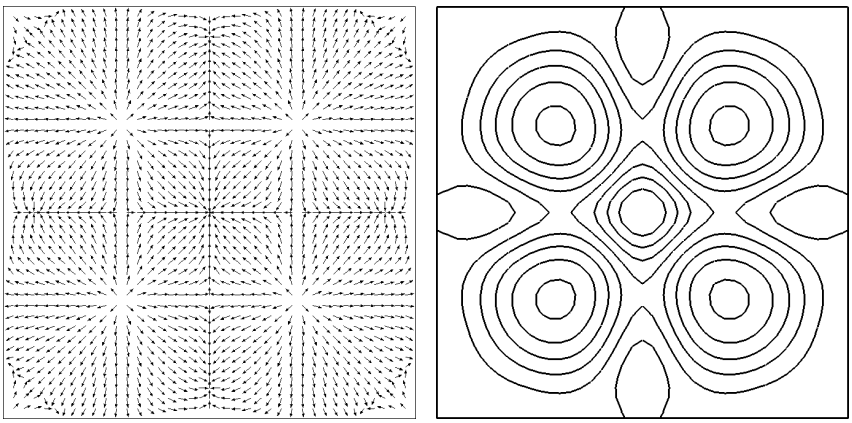


Figure 3.23: The logarithm of the adaptively selected Δt (scaled by the initial timestep) and the approximate measure of the time derivative (described in the text) for the $\Gamma = 7.75$ case.

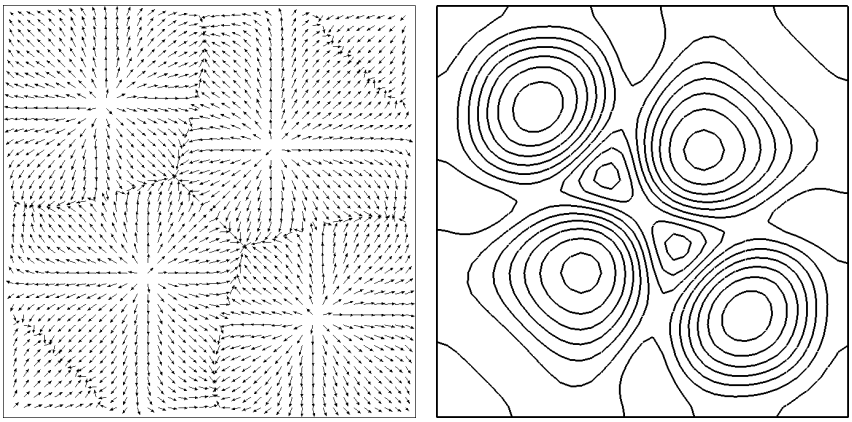
At state 1, we can observe equal fluid upwellings in the four corners of the container, which eventually evolves to the asymmetric 4-cell state marked

as 2 in Fig. 3.23. State 2 occurs around timestep 239, and is shown in detail in Fig. 3.25. This state bears some similarities (but is not identical to) the asymmetric 4-cell state observed in §3.3 on the temporal evolution of the $\Gamma = 6.75$ case. State 2 eventually evolves (while breaking another symmetry of the configuration) to the "true" steady state shown in Fig. 3.26 at timestep 369 (point 3 in Fig. 3.23). This state arises as the top right convection cell grows slightly stronger than the bottom left cell. In the later discussion of continuation along solution branches in parameter space, this evolution from symmetric to asymmetric solutions is shown to be related to jumping from (unstable) symmetric solution branch to a (stable) asymmetric one.



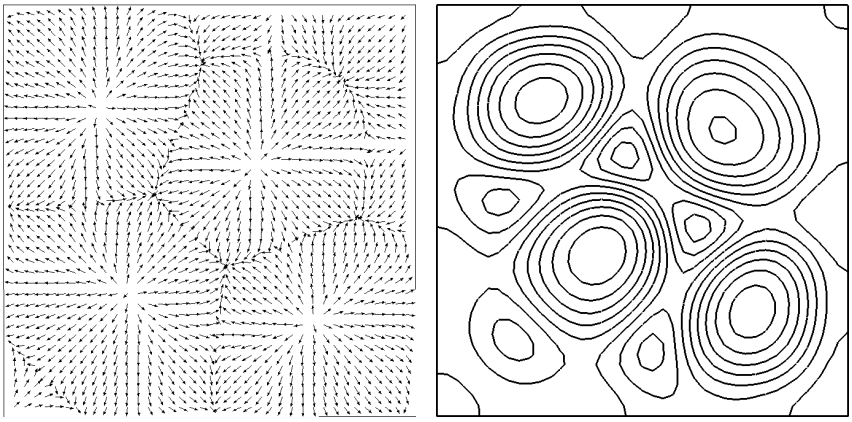
Velocity Vectors Temperature Contours

Figure 3.24: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 7.75$ case near point 1 on Fig. 3.23.



Velocity Vectors Temperature Contours

Figure 3.25: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 7.75$ case near point 2 on Fig. 3.23. The temperature contours lie in the range $\tilde{T} \in [-0.113, 0.141]$.



Velocity Vectors Temperature Contours

Figure 3.26: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 7.75$ case near point 3 on Fig. 3.23.

3.5 Time-Accurate Calculation, $\Gamma = 9$

The $\Gamma = 9$ case first evolved to the 5-cell configuration shown in Fig. 3.28. This state, which occurred around timestep 74, corresponds to solution 1 on Fig. 3.27. This 5-cell pattern appears to be unstable, however, as the time derivatives began growing rapidly upon reaching this point. The central convection cell expands vertically and becomes more hexagonal in shape, as shown in Fig. 3.29, which occurs at timestep 84, which lies between solutions 1 and 2 on Fig. 3.27. This process continues until the location marked as 2 on Fig. 3.27 is reached around timestep 90, corresponding to the state shown in Fig. 3.30.

At point 2, the central cell has become a well-defined, highly-elongated central hexagon. After this stage, new convection cells develop near the center of the left and right boundaries of the domain. These new cells then pinch off the central hexagon into an upper and lower part, as shown in Fig. 3.31, which occurs around timestep 104, and is marked as point point 3 on Fig. 3.27. After this state, the configuration begins a long, slow vertical translation of the convection cells. The slight top-to-bottom asymmetry of the configuration is clearly visible in Fig. 3.32, which occurs around timestep 110 near point 4 on Fig. 3.27, and continues as the time derivative slowly and monotonically decreases.

The configuration finally reaches a steady state around timestep 186 at point 5 on Fig. 3.27, which is shown in Fig. 3.33. This appears to be relatively stable judging from the fact that the time derivative is essentially at

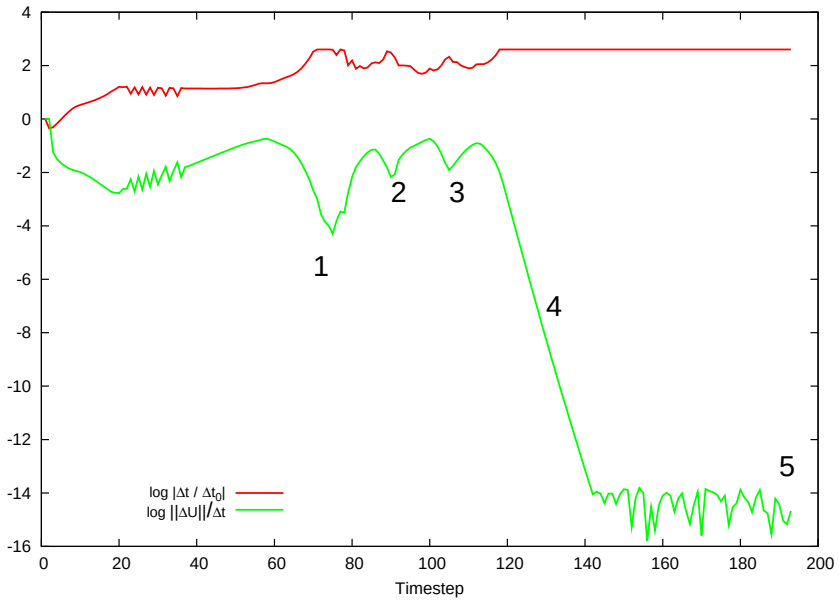
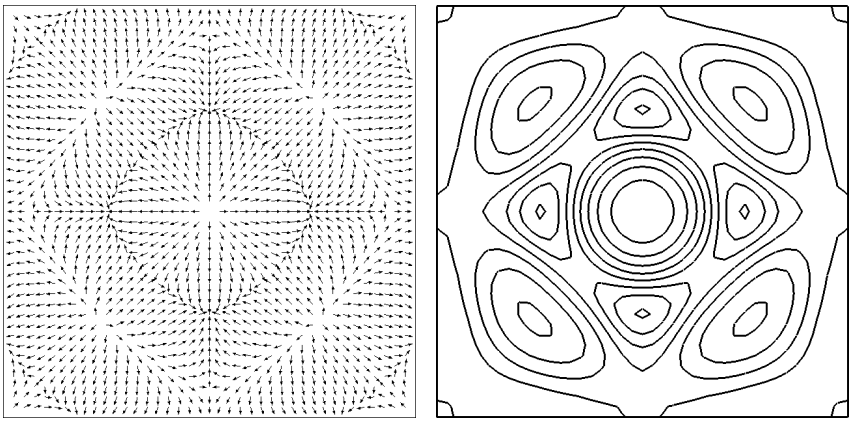


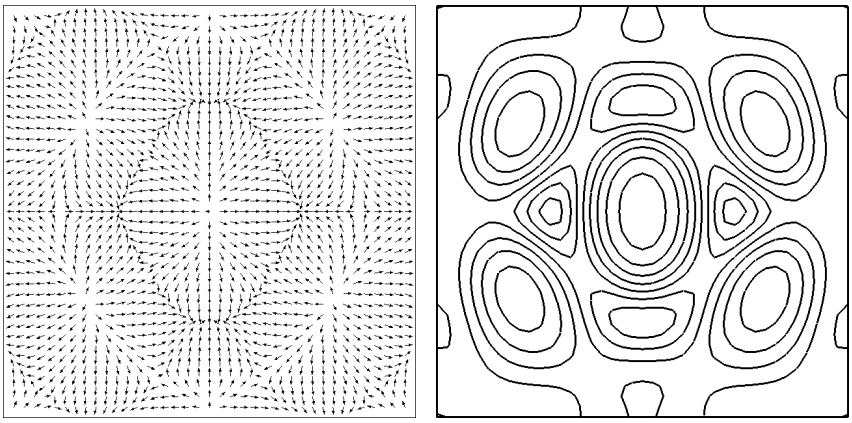
Figure 3.27: The logarithm of the adaptively selected Δt (scaled by the initial timestep) and the approximate measure of the time derivative (described in the text) for the $\Gamma = 9$ case.

machine zero at this point. The final state does appear to exhibit x-symmetry about the domain midpoint, but no additional symmetry axes are present. The temperature contours for the configurations shown in Figs. 3.28–3.33 lie in the range $\tilde{T} \in [-0.122, 0.164]$.



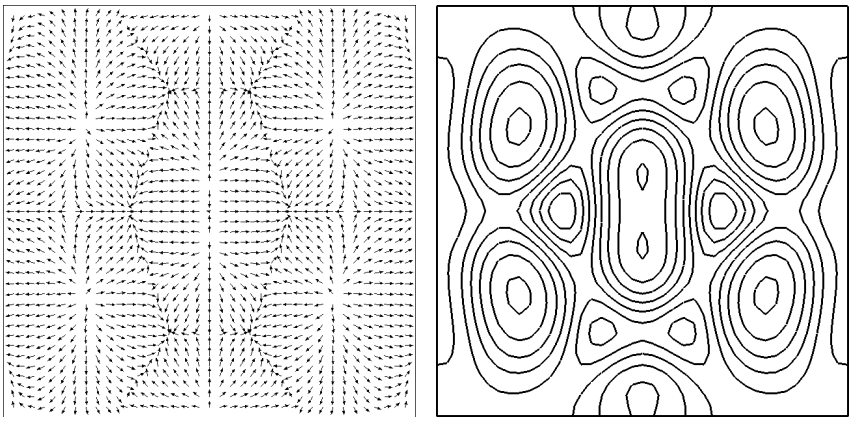
Velocity Vectors Temperature Contours

Figure 3.28: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9$ case near point 1 on Fig. 3.27



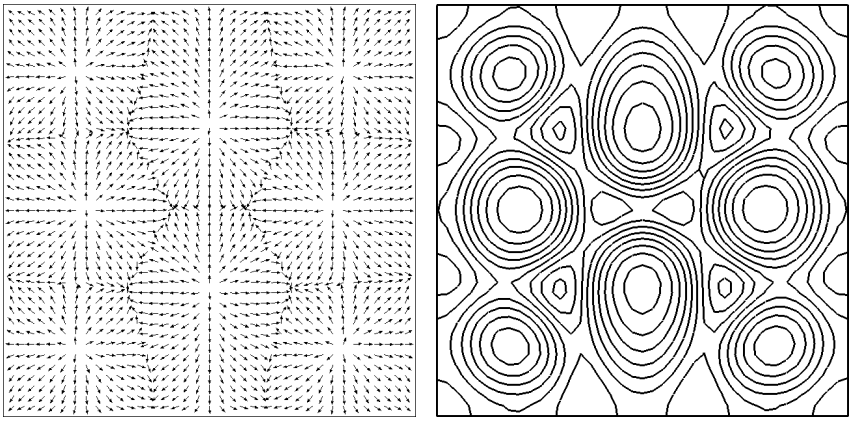
Velocity Vectors Temperature Contours

Figure 3.29: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9$ case between points 1 and 2 on Fig. 3.27



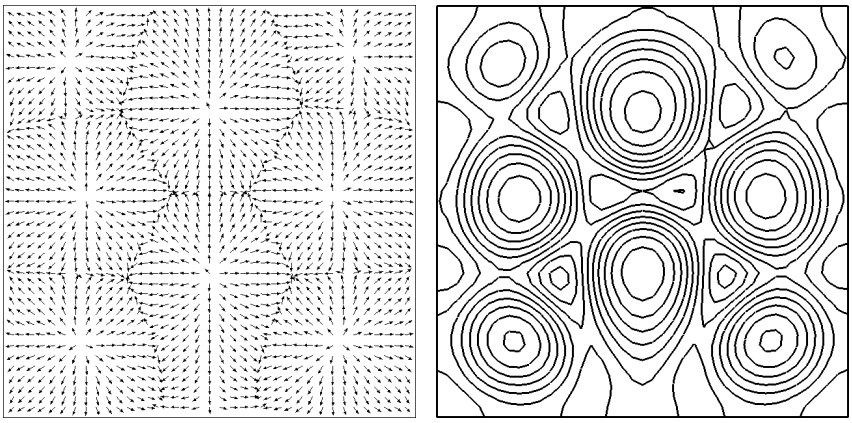
Velocity Vectors Temperature Contours

Figure 3.30: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9$ case near point 2 on Fig. 3.27



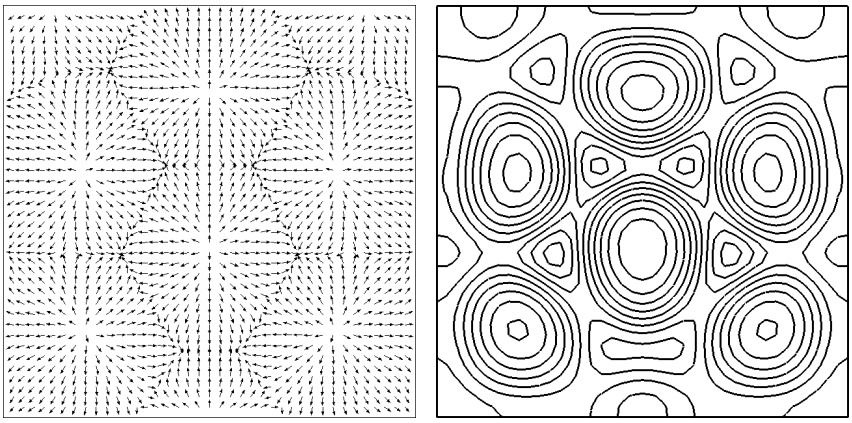
Velocity Vectors Temperature Contours

Figure 3.31: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9$ case near point 3 on Fig. 3.27. We also note that this state particularly resembles an experimental image of a solution having eight convection cells reported by Koschmeider.



Velocity Vectors Temperature Contours

Figure 3.32: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9$ case near point 4 on Fig. 3.27



Velocity Vectors Temperature Contours

Figure 3.33: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9$ case at the final calculated time (state 5 on Fig. 3.27).

3.6 Time-Accurate Calculation, $\Gamma = 9.5$

The $\Gamma = 9.5$ aspect ratio container is the first time we see a 7-convective cell pattern appear in the unsteady equations. In Fig. 3.34, we have marked a few of the interesting intermediate steady states which occurred during the simulation. In all of the temperature contour pictures, $\tilde{T}$ lies in the range $[-0.125, 0.171]$.

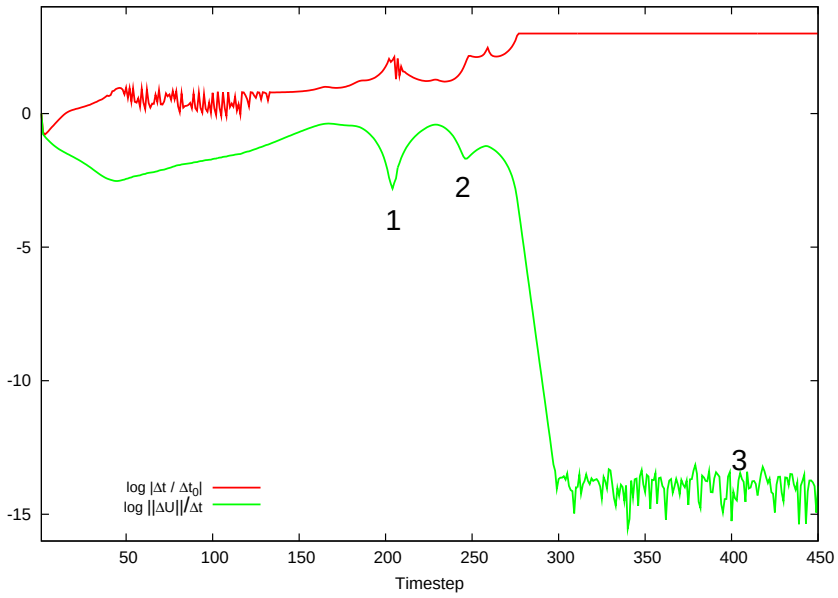
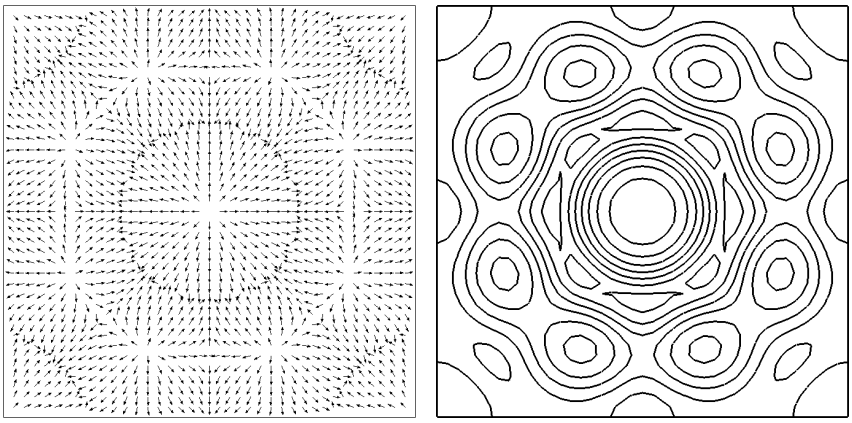


Figure 3.34: The logarithm of the adaptively selected Δt (scaled by the initial timestep) and the approximate measure of the time derivative (described in the text) for the $\Gamma = 9.5$ case.

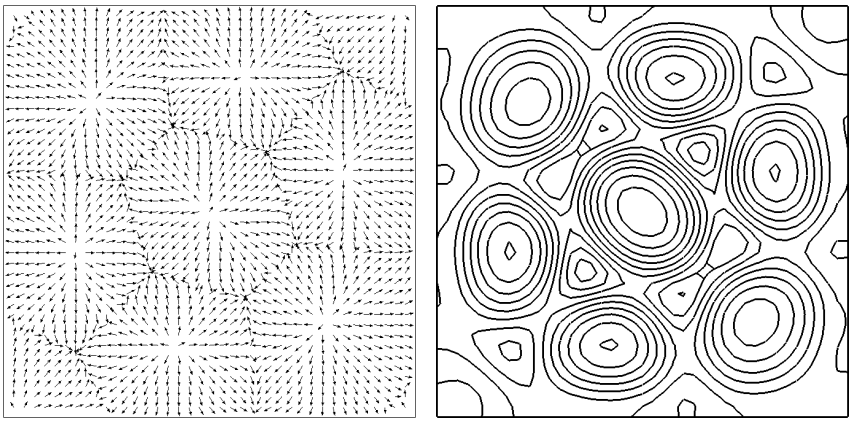
At the state marked 1 in Fig. 3.34, which occurs around timestep 206

and is shown in detail in Fig. 3.35, the solution is essentially a concentric ring of convection cells in which it is possible to count approximately 9 partially-formed cells. The central cell eventually elongates and rotates to the left, reaching state 2 at timestep 243 in Fig. 3.34 which is shown in detail in Fig. 3.36. The central cell in this configuration is hexagonal-shaped. This state is, however, also unstable and the upper-left and bottom-right cells eventually expand along the sides of the container to arrive at the final 7-cell state marked as solution 3 in Fig. 3.34 and pictured in Fig. 3.37 at timestep 457.



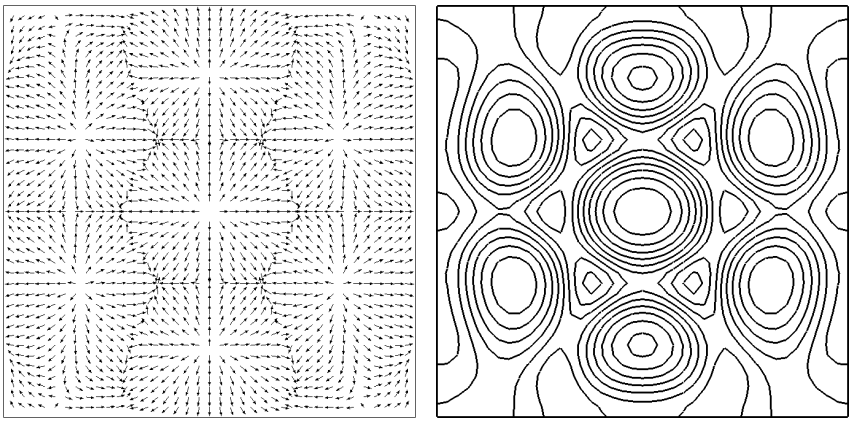
Velocity Vectors Temperature Contours

Figure 3.35: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9.5$ case near point 1 on Fig. 3.34



Velocity Vectors Temperature Contours

Figure 3.36: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9.5$ case near point 1 on Fig. 3.34



Velocity Vectors Temperature Contours

Figure 3.37: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 9.5$ case near point 3 on Fig. 3.34

3.7 Time-Accurate Calculation, $\Gamma = 10$

The time-dependent evolution characteristics of the $\Gamma = 10$ case are fairly simple: only a single intermediate (unstable) "steady" was encountered at timestep 220. This state, which is marked with a 1 in Fig. 3.38 and shown in more detail in Fig. 3.39, consists of 8 cells, the central four of which meet at a single point in the center of the domain.

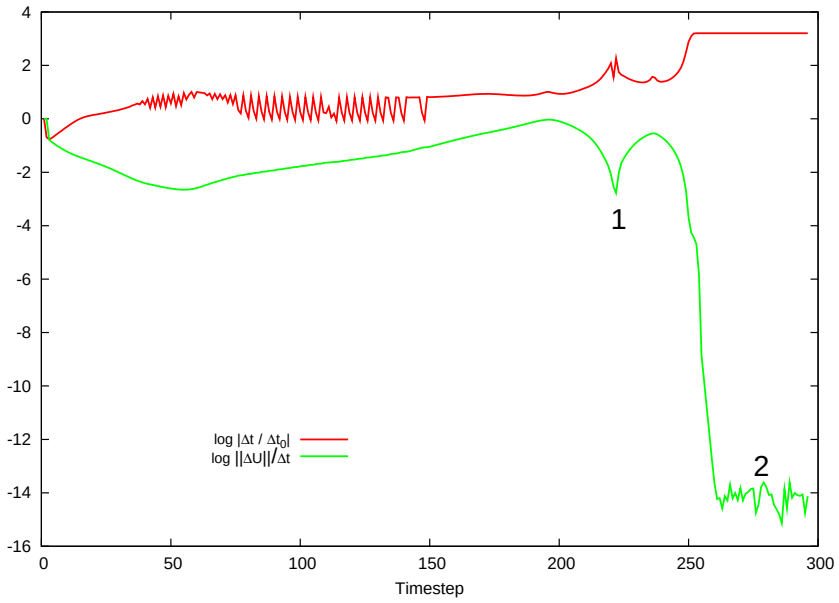
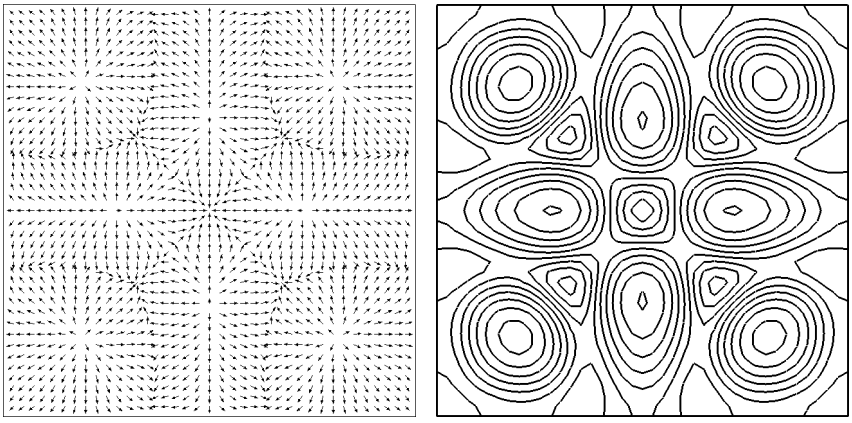


Figure 3.38: The logarithm of the adaptively selected Δt (scaled by the initial timestep) and the approximate measure of the time derivative (described in the text) for the $\Gamma = 10$ case.

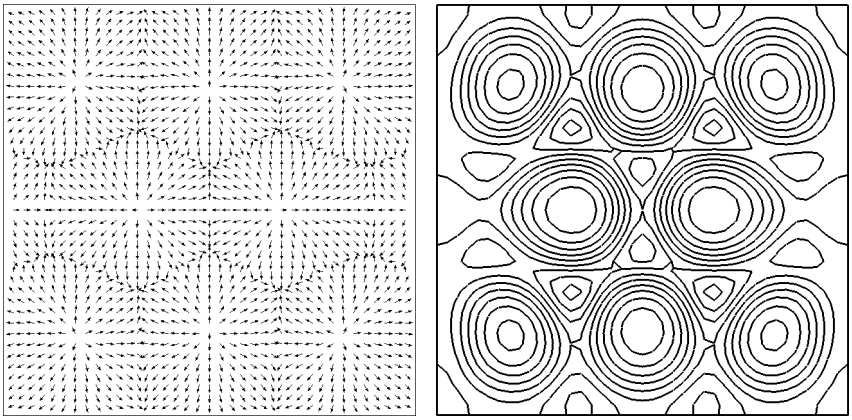
The final state, marked with a 2 in Fig. 3.38, occurs around timestep

265, and is shown in more detail in Fig. 3.40. It is fairly easy to see how state 1 transitions to state 2, it occurs essentially when the meeting place of the central four cells (which appears to be an unstable configuration in other aspect ratio cases as well) elongates and allows the two central cells to take on a hexagonal shape.



Velocity Vectors Temperature Contours

Figure 3.39: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 10$ case near point 1 on Fig. 3.38



Velocity Vectors Temperature Contours

Figure 3.40: Surface velocity and temperature fields on the 20×20×2 grid for the $\Gamma = 10$ case near point 2 on Fig. 3.38

3.8 Time-Accurate Calculation, $\Gamma = 11$

The $\Gamma = 11$ case turns out to have fairly interesting temporal dynamics. The first state, which occurs around timestep 221, is marked with a 1 in Fig. 3.41 and shown in more detail in Fig. 3.42. This state is characterized by having 8 cells, the the central four of which meet at a common central point. By the time we reach the state marked with a 2 in Fig. 3.41, around timestep 250, this centralized meeting point has spread into a line, and the central cells have taken on a hexagonal shape, which can be observed in Fig. 3.43.

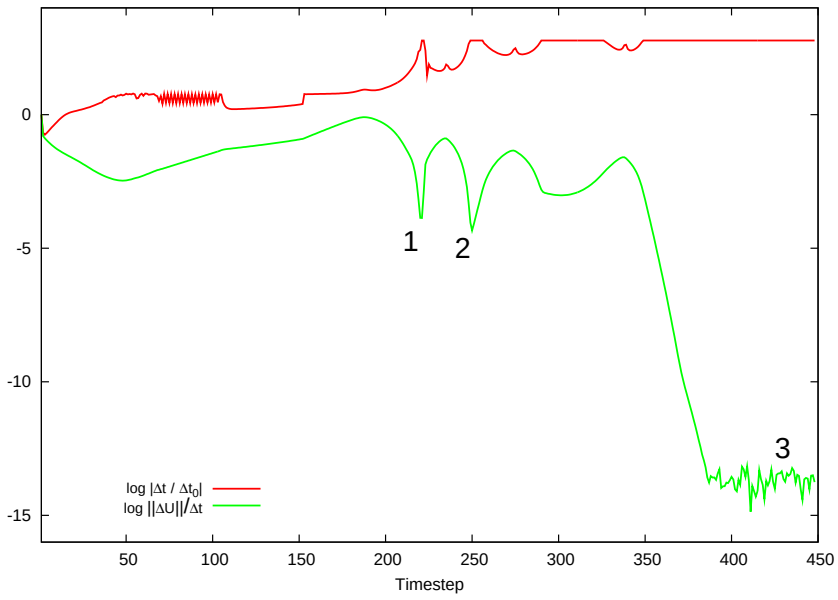
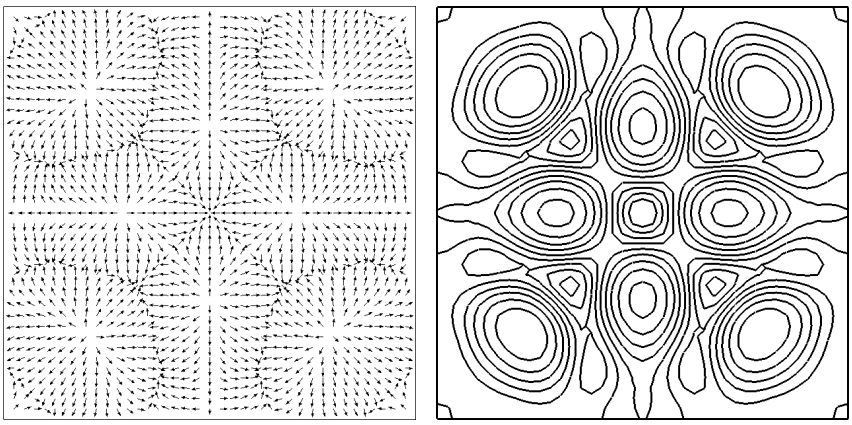


Figure 3.41: The logarithm of the adaptively selected Δt (scaled by the initial timestep) for the $\Gamma = 11$ case.

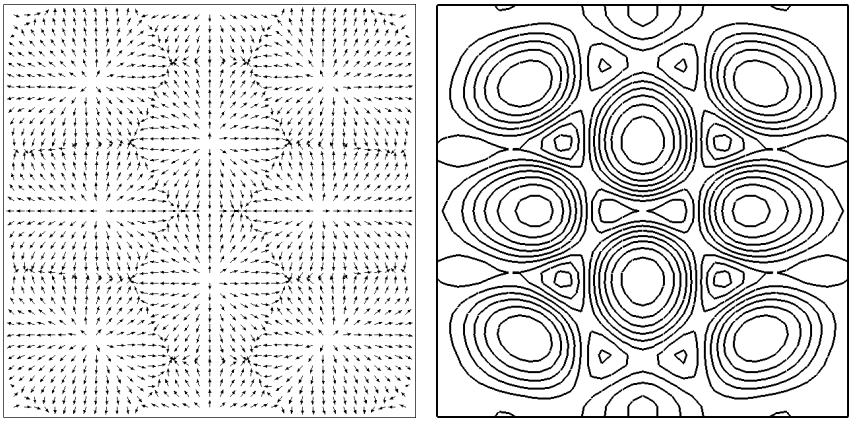
However, the second state is apparently also not a true stable state for $\Gamma = 11$ since the upper left convection cell begins growing asymmetrically in strength around timestep 280, as shown in Fig. 3.44. The entire configuration appears to rotate counter-clockwise, with the central cells aligning themselves somewhat along the main diagonal of the container.

After this point, the *lower* left convection cell begins growing in strength, this time *without* a corresponding increase in the upper right convection cell strength. This intermediate state, which occurs around timestep 338, is shown in Fig. 3.45. Eventually the lower left convection cell becomes roughly symmetric with the top left cell, but the x-symmetry of the configuration is now lost. The left half of the domain resembles the $\Gamma = 12$ (see e.g. Fig. 3.49) solution while the right half resembles the steady solution obtained for $\Gamma = 10$ (see Fig. 3.40). These solutions are stitched together by the two hexagonal cells in the center of the domain. The final state, which occurs at timestep 414, (marked with a 3 in Fig. 3.41) is shown in Fig. 3.46.



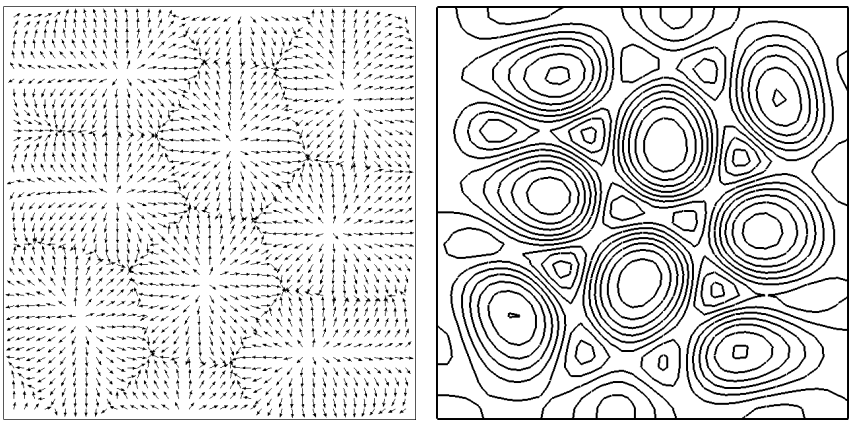
Velocity Vectors Temperature Contours

Figure 3.42: Surface velocity and temperature fields on the 20×20×2 grid at timestep 221 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.127, 0.154]$. This state corresponds to the time marked with the number 1 in Fig. 3.41.



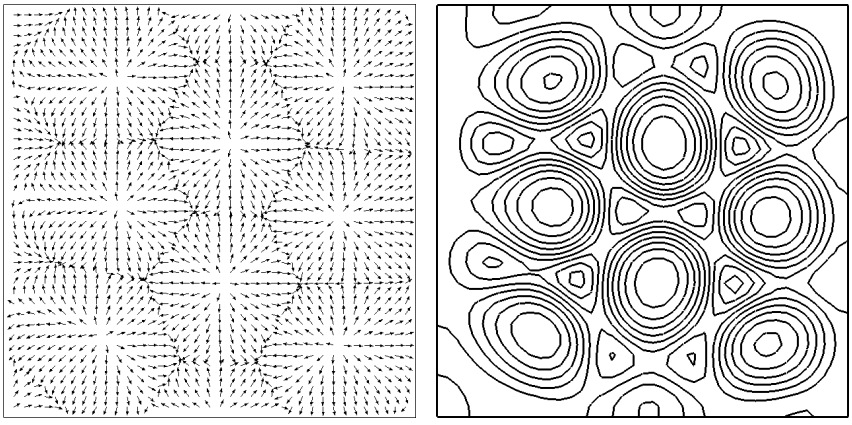
Velocity Vectors Temperature Contours

Figure 3.43: Surface velocity and temperature fields on the 20×20×2 grid at timestep 250 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.127, 0.154]$. This state corresponds to the time marked with the number 2 in Fig. 3.41.



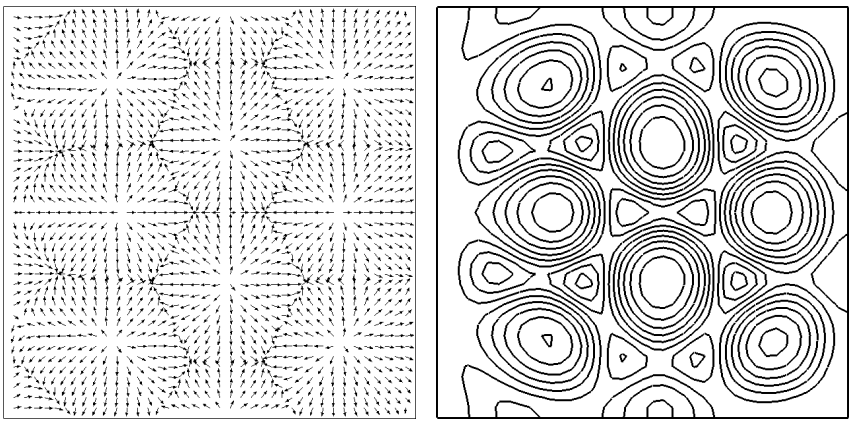
Velocity Vectors Temperature Contours

Figure 3.44: Surface velocity and temperature fields on the 20×20×2 grid at timestep 280 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.127, 0.154]$. This state occurs after state 2 in Fig. 3.41.



Velocity Vectors Temperature Contours

Figure 3.45: Surface velocity and temperature fields on the 20×20×2 grid at timestep 338 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.127, 0.154]$. This state occurs after state 2 in Fig. 3.41.



Velocity Vectors Temperature Contours

Figure 3.46: Surface velocity and temperature fields on the 20×20×2 grid at timestep 414 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.127, 0.154]$. This state corresponds to the time marked with the number 3 in Fig. 3.41.

3.9 Time-Accurate Calculation, $\Gamma = 12$

We located only a single intermediate steady state in the $\Gamma = 12$ time-accurate calculation. This state, which occurs around timestep 237, is marked as number 1 in Fig. 3.47 and shown in more detail in Fig. 3.48. We note (as has been true for the $\Gamma = 11$ and $\Gamma = 12$ cases as well) that the configuration at 1 is composed of eight convection cells, with the central four meeting in an (apparently) unstable central point.

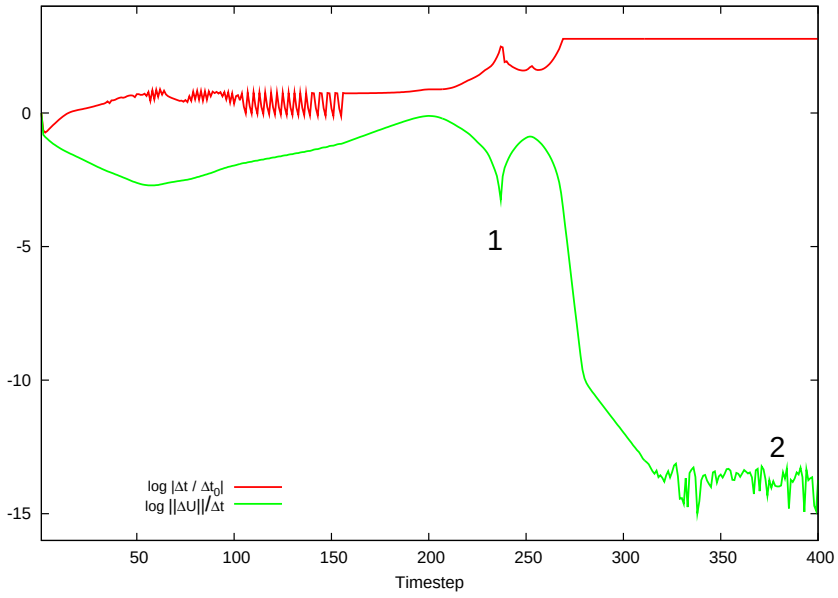
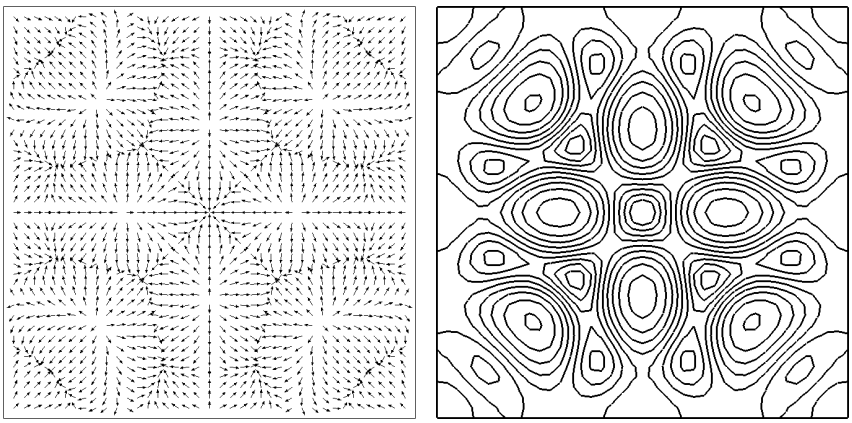


Figure 3.47: The logarithm of the adaptively selected Δt (scaled by the initial timestep) for the $\Gamma = 12$ case.

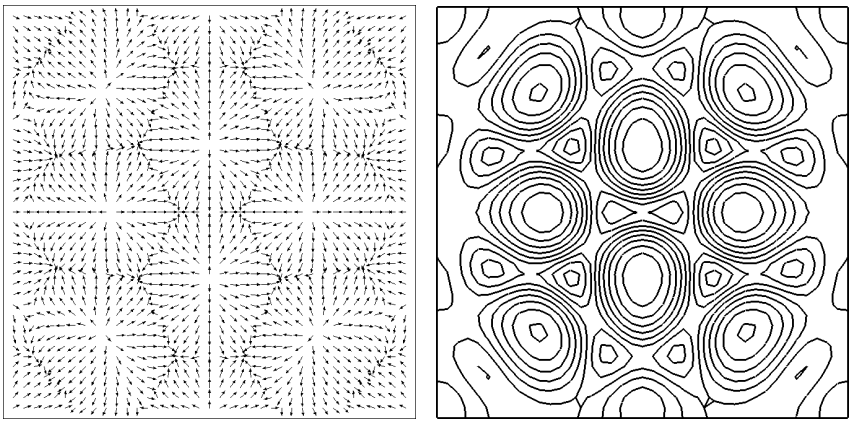
The state marked 2 in Fig. 3.47, which occurs at timestep 401, once

again depicts eight central cells, but now the central meeting point has elongated into a line, allowing the central cells to take on hexagonal shapes. This second solution is shown in more detail in Fig. 3.49.



Velocity Vectors Temperature Contours

Figure 3.48: Surface velocity and temperature fields on the 20×20×2 grid at timestep 237 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.124, 0.145]$. This state corresponds to the time marked with the number 1 in Fig. 3.47.



Velocity Vectors Temperature Contours

Figure 3.49: Surface velocity and temperature fields on the 20×20×2 grid at timestep 401 for the adaptive timestepping method. The temperature contours lie in the range $\tilde{T} \in [-0.124, 0.145]$. This state corresponds to the time marked with the number 2 in Fig. 3.47.

Chapter 4

Arclength Continuation

In this chapter, we discuss the arclength continuation algorithms employed in the present work to compute branches of steady-state solutions. Readers already familiar with pseudo-arclength continuation methods may want to skip this chapter and proceed directly to Ch. 5 where the results of the computations are discussed.

This chapter is arranged in the following way: in §4.1, we introduce the notation and several of the basic ideas associated with continuation schemes, as well as giving several references to relevant literature. In §4.2 we discuss the solution of an auxiliary "tangent" system for the generation of initial guesses for the Newton iterations used to solve the augmented arclength continuation system. In §4.3, additional details regarding the pseudo-arclength constraint used in this and other works are given.

Practical considerations of the method, including the use of a scaled form of the arclength constraint, the technique for starting the method, some adaptive arclength stepsize selection techniques, and finally a simple algorithm for solving the sparse, bordered linear systems arising from the arclength continuation method are given in §4.4. Finally, a discussion of the very important

topic of linear stability analysis and of issues related to the numerical solution of the associated generalized eigenvalue problem are given in §4.5.

4.1 Introduction and Notation

Suppose that we have a PDE which depends on a parameter λ. We write this in an abstract notation as

$$G(u, \lambda) = 0 \tag{4.1}$$

where (in the continuous setting) $G : \mathbb{B} \times \mathbb{R} \to \mathbb{B}$ for some Banach space $\mathbb{B}$, or in the finite-dimensional case which we will be considering here, $G : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, and $u \in \mathbb{R}^n$ can be thought of as e.g. a vector of finite element solution coefficients. Certain smoothness properties of G are assumed to exist. Since u depends on the parameter λ, we may think of a "branch" of solutions $u(\lambda)$ which satisfy Eqn. (4.1).

In many physically meaningful situations, it happens that for a particular value of λ, the Jacobian G_u is singular. This occurs for non-isolated solutions (points where two different solution branches cross) and for turning (also known as limit) points, in which solutions do not exist for particular ranges of the parameter or where the character of the solution changes abruptly with small changes in λ. A popular method for dealing with this situation is the arclength continuation method described by Keller [106]. We have in fact adopted the same notation as Keller in the preceding paragraph. In the arclength continuation method, Eqn. (4.1) is augmented with a constraint

equation as

$$G(u, \lambda) = 0 \tag{4.2}$$

$$N(u, \lambda, s) = 0 \tag{4.3}$$

where s is the arclength parameter, defined as the distance along the solution branch (i.e. s is non-decreasing even if λ is decreasing along the branch). The nonlinear constraint N may depend explicitly on the arclength parameter s, and of course the solution $u = u(s)$ and also the parameter $\lambda = \lambda(s)$ depend implicitly on the arclength as well.

The main difference in the solution algorithm is that now instead of solving (for $k = 0, 1, \ldots$, via the inexact Newton method) the system

$$G_u^k \delta u^k = -G^k \tag{4.4}$$

associated with Eqn. (4.1), we now must consider the *augmented* Jacobian system

$$\begin{bmatrix} G_u & G_\lambda \\ N_u^t & N_\lambda \end{bmatrix}^k \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix}^{k+1} = \begin{bmatrix} -G \\ -N \end{bmatrix}^k \tag{4.5}$$

associated with Eqs. (4.2) and (4.3). There are of course many different numerical schemes [25, 51] for solving Eqn. (4.5) efficiently. A large number of practical implementations of the arclength continuation [49, 86] and bifurcation detection [26] methods are to be found in the literature, often in combination with other techniques for determining e.g. the stability of the steady states computed using the method [173, 186].

4.2 Generation of Initial Guesses

An important aspect in these methods (which is sometimes left unmentioned, or not well-explained) is the technique by which initial guesses for the Newton iterations of the system given in Eqn. (4.5) are generated. Clearly, this is an important issue because a bad initial guess may not converge or may converge to the wrong solution (i.e. a solution on a different branch) if the initial guess is not sufficiently "good".

The main idea in generating the initial guess at each step is to introduce the variable $x := [u, \lambda]$, and think of the pair of Eqs. (4.2) and (4.3) as

$$P(x(s), s) := \begin{bmatrix} G(x(s)) \\ N(x(s), s) \end{bmatrix} \tag{4.6}$$

On a solution arc $x(s) = [u(s), \lambda(s)]$ of Eqs. (4.2) and (4.3) we therefore have $P(x(s), s) = 0$. Another way of thinking of this (and the key point in this discussion) is that if we were to "plot" P given above versus s the result would simply be the constant value of zero for each s. For such a constant function, it must also be true that the "derivative" of P with respect to s is the zero vector, i.e.

$$\frac{dP}{ds} = 0 \tag{4.7}$$

By chain-differentiation of P we obtain

$$\begin{aligned} \frac{dP}{ds} &= \begin{bmatrix} \frac{dG}{ds} \\ \frac{dN}{ds} \end{bmatrix} \\ &= \begin{bmatrix} G_u & G_\lambda \\ N_u^t & N_\lambda \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial s} \\ \frac{\partial \lambda}{\partial s} \end{bmatrix} + \begin{bmatrix} 0 \\ N_s \end{bmatrix} \end{aligned} \tag{4.8}$$

Combining Eqn. (4.8) with Eqn. (4.7) one arrives at

$$\begin{bmatrix} G_u & G_\lambda \\ N_u^t & N_\lambda \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial s} \\ \frac{\partial \lambda}{\partial s} \end{bmatrix} = \begin{bmatrix} 0 \\ -N_s \end{bmatrix} \tag{4.9}$$

which is an equation for determining the "tangent" vector $\dot{x}(s) := \partial x/\partial s$ to the solution arc $x(s)$. Eqn. (4.9) conveniently involves the same matrix as the augmented Jacobian system of Eqn. (4.5). Of course, the purpose of computing this tangent vector is that it leads us to a predictor (i.e. initial guess) for Newton's method, given by

$$x^{n+1} = x^n + \Delta s_n \dot{x}^n \tag{4.10}$$

for a fixed arclength step size Δs_n. There are various methods for solving the system associated with Eqn. (4.9), the "best" way depends on the underlying numerical method which is used to solve the original PDE. The goal is for the additional tangent calculation to fit simply and naturally into the framework of the existing code.

Aside: we note that the same concept of a solution arc can also be applied to the simpler case (sometimes called first-order continuation) where an auxiliary arclength equation is not present. In this case, the solution arc is $G(u(\lambda), \lambda) = 0$ for each value of λ on which the arc is defined, and thus

$$\begin{aligned} \frac{dG}{d\lambda} &= G_u \frac{\partial u}{\partial \lambda} + G_\lambda \\ &= 0 \end{aligned} \tag{4.11}$$

This defines the linear system

$$G_u \frac{\partial u}{\partial \lambda} = -G_\lambda \tag{4.12}$$

to be solved for $\frac{\partial u}{\partial \lambda}$. Eqn. (4.12) involves the same Jacobian matrix as for the original problem, but a different right-hand side. The benefit of solving this extra system of equations is an improved initial guess for u at the new value of the control parameter, given by

$$u^{n+1} = u^n + \left(\frac{\partial u}{\partial \lambda}\right)^n \Delta\lambda \tag{4.13}$$

for a given fixed step size $\Delta\lambda$. While it may indeed be possible to improve the convergence of the Newton iterations using this scheme, we should point out that it does not in general allow one to navigate a turning point in the solution path, and in practice this first-order continuation technique has not converged for us in any case where the even simpler "zeroth-order" continuation did not converge as well.

4.3 The Pseudo-Arclength Constraint

We still need to specify more precisely the form of the arclength constraint in Eqn. (4.3) which is to be used. Consider two "nearby" points (u, λ_u) and (v, λ_v) on a solution path P in $\mathbb{R}^{n+1}$, where n is the dimension of the finite element solution vector. This is depicted for $n = 2$ in Fig. 4.1. Note that the same concepts hold in general normed linear spaces, though they are easier to describe geometrically. Now consider the projection of the vector $u - v$ onto $\mathbb{R}^n$ defined by $\Delta u := (u_1 - v_1, \ldots, u_n - v_n, 0)$ and the projection onto $\mathbb{R}$ defined by $\Delta\lambda := (0, \ldots, 0, \lambda_u - \lambda_v)$. Then clearly the vectors Δu and $\Delta\lambda$ are

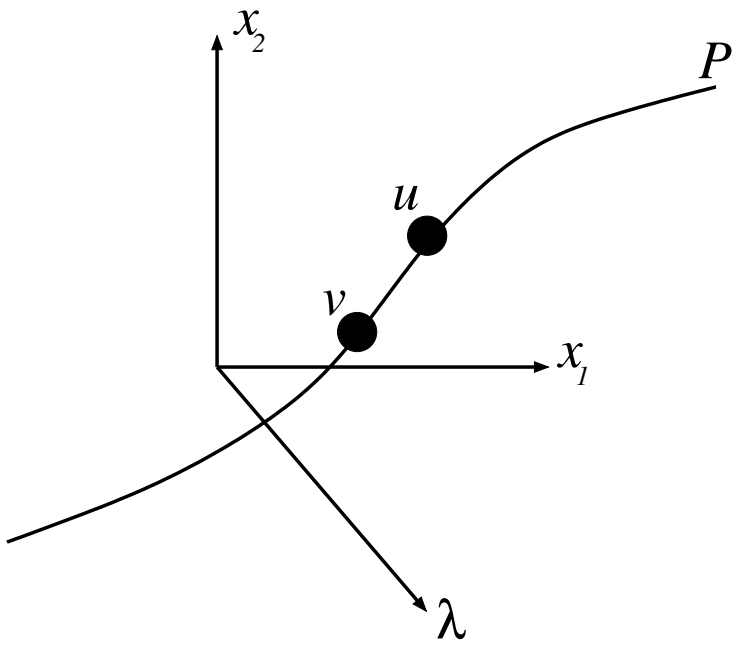


Figure 4.1: Solution path P with representative solutions u and v.

orthogonal and so their respective lengths are related by

$$\|\Delta u\|^2 + \|\Delta\lambda\|^2 = \|\Delta s\|^2 \tag{4.14}$$

in the limit as $u \to v$, for some vector Δs. In the limit as $\Delta s \to 0$, we obtain

$$\left\|\frac{\partial u}{\partial s}\right\|^2 + \left(\frac{\partial \lambda}{\partial s}\right)^2 = 1 \tag{4.15}$$

where the norm $\|\cdot\|$ is the standard Euclidian norm on $\mathbb{R}^{n+1}$. Eqn. (4.15) is in the form of Eqn. (4.3), and can used as the supplemental arclength constraint equation. However, most authors [106] agree that the use of a *nonlinear* constraint such as Eqn. (4.15) is cumbersome in practice, and instead a linearized constraint of some form is preferred. Such a linearized constraint follows from

Eqn. (4.14) by assuming $v = u(s_i)$ for some arbitrary point s_i is known, and setting $\|\Delta s\| = s - s_i$ for arbitrary s. We obtain the (discrete in s) nonlinear arclength constraint

$$N^{nl}(u, \lambda, s) := \|u - u(s_i)\|^2 + \|\lambda - \lambda(s_i)\|^2 - (s - s_i)^2 = 0 \tag{4.16}$$

which is locally accurate as $s \to s_i$. Recall that for the augmented Newton's method described in Eqn. (4.5), we have

$$(N_u^{nl})^t \frac{\partial u}{\partial s} + N_\lambda^{nl} \frac{\partial \lambda}{\partial s} + N_s^{nl} = 0 \tag{4.17}$$

Now we suppose that, instead of just being the equation to satisfy for Newton's method, Eqn. (4.17) with $\frac{\partial u}{\partial s}$ and $\frac{\partial \lambda}{\partial s}$ evaluated at the prior solution $u(s_i)$, *is the arclength constraint itself.* In other words, our linearized arclength constraint is:

$$N(u, \lambda, s) := (N_u^{nl})^t \left.\frac{\partial u}{\partial s}\right|_{s_i} + N_\lambda^{nl} \left.\frac{\partial \lambda}{\partial s}\right|_{s_i} + N_s^{nl} = 0 \tag{4.18}$$

Differentiating N^{nl} with respect to u, λ, and s yields

$$(N_u^{nl})^t \phi = 2\,(u - u(s_i))^t\,\phi \tag{4.19}$$

$$N_\lambda^{nl} = 2(\lambda - \lambda(s_i)) \tag{4.20}$$

$$N_s^{nl} = -2(s - s_i) \tag{4.21}$$

for arbitrary vector ϕ. Hence, our linearized arclength constraint becomes, in its final form,

$$N(u, \lambda, s) := (u - u(s_i))^t \left.\frac{\partial u}{\partial s}\right|_{s_i} + (\lambda - \lambda(s_i)) \left.\frac{\partial \lambda}{\partial s}\right|_{s_i} - (s - s_i) \tag{4.22}$$

4.4 Practical Considerations

In this section, we discuss several of the practical aspects involved with solving the augmented system numerically, including the details of the numerical solution of the augmented (or bordered) system in §4.4.4, and some of the details of starting up the method in §4.4.2.

4.4.1 Scaling

It is not in general feasible to use Eqn. (4.22) (as given) as a pseudo-arclength constraint, due primarily to the vastly different scales the first two terms in Eqn. (4.22) may assume. Therefore, in practice we will use a penalized or regularized version of pseudo-arclength, given by

$$N_\alpha(u, \lambda, s) := \alpha_i^2 \left(u - u(s_i)\right)^t \left.\frac{\partial u}{\partial s}\right|_{s_i} + \left(\lambda - \lambda(s_i)\right) \left.\frac{\partial \lambda}{\partial s}\right|_{s_i} - (s - s_i) \qquad (4.23)$$

where $\alpha_i \in \mathbb{R}$ are scaling parameters chosen in order to balance the relative magnitudes of the first two terms of Eqn. (4.23). To motivate the definition of α, consider the normalized "solution triangle" shown in Fig. 4.2, which has hypotenuse 1 and legs of length $\left|\frac{\partial \lambda}{\partial s}\right|$ and $\left\|\frac{\partial u}{\partial s}\right\|$ (in the continuous setting, or $\left|\frac{\Delta \lambda}{\Delta s}\right|$ and $\left\|\frac{\Delta u}{\Delta s}\right\|$ in the discrete-in-s setting). Clearly, if either of the legs is much longer than the other, then one of the terms on the left-hand side of the full arclength equation (originally Eqn. (4.15), repeated here for convenience)

$$\left\|\frac{\partial u}{\partial s}\right\|^2 + \left(\frac{\partial \lambda}{\partial s}\right)^2 = 1 \qquad (4.24)$$

will be nearly 1, while the other is nearly zero. In such cases, the included angle γ (shown in Fig. 4.2) will tend to either 0 or $\pi/2$, and the benefits of performing

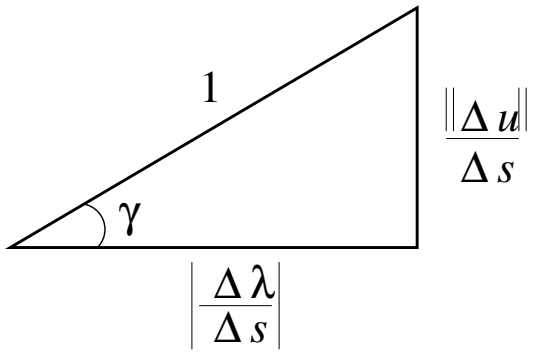


Figure 4.2: Normalized solution triangle with angle γ shown.

arclength continuation will be effectively lost. For arc-step $i = 1, 2, \ldots$, we define the regularization parameter α_i such that

$$\alpha_i \tan \gamma_i = \alpha_0 \tag{4.25}$$

for some constant α_0, whose determination we will discuss in §4.4.2 on initialization. There is apparently nothing unique in this definition of α_i, it is defined similarly to the method implemented in the Library of Continuation Algorithms (LOCA) from Sandia National Labs [172]. Note that the "equivalent" scaled full arclength equation is now

$$\alpha^2 \left\| \frac{\partial u}{\partial s} \right\|^2 + \left(\frac{\partial \lambda}{\partial s} \right)^2 = 1 \tag{4.26}$$

and the included angle may be computed at step i via

$$\tan \gamma_i := \frac{\sqrt{1 - \left(\frac{\partial \lambda}{\partial s}\big|_{s_i} \right)^2}}{\left| \frac{\partial \lambda}{\partial s}\big|_{s_i} \right|} \tag{4.27}$$

4.4.2 Initialization

Initializing the continuation algorithm can be done in a few different ways, we will only discuss one possible way here. The main difficulty is that, when one is using the linearized arclength constraint form given in Eqn. (4.22), or more precisely the suitably-normalized constraint given by Eqn. (4.23), one requires the solution and parameter derivatives with respect to s from the previous step.

At the first arc-step, of course, there is no previous step, and so we assume that two previous solutions: $\{u_0, \lambda_0\}$ and $\{u_1, \lambda_1\}$ have been computed *without* requiring arclength continuation. For example, in some problems $\{u_0, \lambda_0\} = \{0, 0\}$, and we can obtain u_1 with a small increment of the control parameter, making use of e.g. zeroth-order continuation. Then, on the first step, we impose the requirement

$$\left.\frac{\partial \lambda}{\partial s}\right|_{s_0} = \pm \frac{1}{\sqrt{2}} \tag{4.28}$$

where the sign is the same as the sign of $\lambda_1 - \lambda_0$. In other words, we use Eqn. (4.28) to ensure that the initial solution triangle is right-isosceles. Then we seek α_0 such that the scaled arclength constraint of Eqn. (4.26),

$$\alpha_0^2 \left\| \left.\frac{\partial u}{\partial s}\right|_{s_0} \right\|^2 + \left(\left.\frac{\partial \lambda}{\partial s}\right|_{s_0} \right)^2 = 1 \tag{4.29}$$

is satisfied subject to the requirement of Eqn. (4.28). We can factor out $\partial\lambda/\partial s$ via chain differentiation to obtain

$$\left(\left.\frac{\partial \lambda}{\partial s}\right|_{s_0} \right)^2 \left[1 + \alpha_0^2 \left\| \left.\frac{\partial u}{\partial \lambda}\right|_{s_0} \right\|^2 \right] = 1 \tag{4.30}$$

Then, we can solve for α_0 to obtain

$$\alpha_0 = \left\| \left. \frac{\partial u}{\partial \lambda} \right|_{s_0} \right\|^{-1} \tag{4.31}$$

where $\left. \frac{\partial u}{\partial \lambda} \right|_{s_0}$ is approximated from the initial two solutions via finite differences

$$\left. \frac{\partial u}{\partial \lambda} \right|_{s_0} \approx \frac{u_1 - u_0}{\lambda_1 - \lambda_0} \tag{4.32}$$

The initial solution tangent with respect to s is then

$$\left. \frac{\partial u}{\partial s} \right|_{s_0} = \left. \frac{\partial \lambda}{\partial s} \right|_{s_0} \frac{u_1 - u_0}{\lambda_1 - \lambda_0} \tag{4.33}$$

and the arc-distance traveled in the first step is

$$\Delta s = \frac{\lambda_1 - \lambda_0}{\left. \frac{\partial \lambda}{\partial s} \right|_{s_0}} \tag{4.34}$$

Now that we have the initial tangents $\left. \frac{\partial \lambda}{\partial s} \right|_{s_0}$ and $\left. \frac{\partial u}{\partial s} \right|_{s_0}$, we can assemble the Jacobian and residual at $\{u_1, \lambda_1\}$ and solve the tangency system (Eqns. (4.9)) to obtain $\left. \frac{\partial \lambda}{\partial s} \right|_1$ and $\left. \frac{\partial u}{\partial s} \right|_1$. Finally, α_1 can be computed according to the formula given in Eqn. (4.25).

The initial guess (represented by the dashed line in Fig. 4.3) for $\{u_2, \lambda_2\}$ is then obtained using these tangents as

$$\tilde{u}_2 = u_1 + \left. \frac{\partial u}{\partial s} \right|_{s_1} \Delta s \tag{4.35}$$

$$\tilde{\lambda}_2 = \lambda_1 + \left. \frac{\partial \lambda}{\partial s} \right|_{s_1} \Delta s \tag{4.36}$$

The tangents $\left. \frac{\partial u}{\partial s} \right|_{s_1}$ and $\left. \frac{\partial \lambda}{\partial s} \right|_{s_1}$ are then also used in the Newton iterations for obtaining $\{u_2, \lambda_2\}$, as well as updating the tangents at point 2. The continuation algorithm then proceeds as described previously for steps 3,4,...

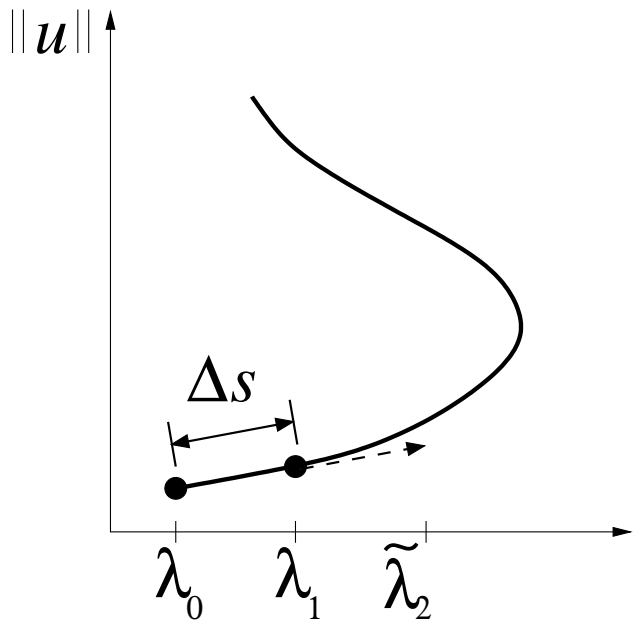


Figure 4.3: Procedure for starting the arclength continuation procedure.

One particularly attractive feature of this form of pseudo-arclength constraint is that enough information is present (if one is willing to save a single additional vector) to compute a higher-order predictor. This predictor, which is based on the second-order explicit Adams-Bashforth formula with variable-size steps is given by

$$\tilde{u}_{n+1} = u_n + \frac{\Delta s_{n+1}}{2}\left[\left(2 + \frac{\Delta s_{n+1}}{\Delta s_n}\right)\left.\frac{\partial u}{\partial s}\right|_n - \left(\frac{\Delta s_{n+1}}{\Delta s_n}\right)\left.\frac{\partial u}{\partial s}\right|_{n-1}\right] \qquad (4.37)$$

We have observed a moderate reduction (on the order of 5-10%) in the number of linear solver iterations required when solving the bordered systems using this predictor as the starting guess, with no additional calculation required and little additional storage overhead.

4.4.3 Adaptive Arclength Stepsize Selection

It is generally unwise to maintain a fixed arclength stepsize throughout the tracing of a solution branch for the following reasons:

1. The initial arclength stepsize may be small if the initial solutions u_0 and u_1 were relatively close together. It rarely makes sense, from an efficiency standpoint, to continue using such a small arclength stepsize in regions where the solution is not changing much with respect to the parameter λ.

2. Unless we can reduce the arclength stepsize, failure of the Newton iterations to converge must signal a failure for the method. This frequently occurs near bifurcations and turning points. The ability to reduce the arclength stepsize and try the Newton iterations again is thus key to the robustness of the branch tracing algorithm.

3. More detail (additional arc-steps) are often desirable near turning and bifurcation points, since the character of the solution typically changes rapidly in these regions, and the transition from one solution regime to another is a topic of special interest when computing the branches.

4. Without the ability to shrink the arclength stepsize, important features of the solution branch (such as e.g. turning points) can actually be "stepped over" and missed entirely. An example in which a particular tracing of a branch misses two turning points is shown in Fig. 4.4.

There are several possible techniques for adaptively choosing the arclength stepsize. One, which is used in the LOCA [172] library, is to compute the parameter

$$\tau_i := \frac{\frac{\partial u}{\partial \lambda}\Big|_{s_i} \cdot \frac{\partial u}{\partial \lambda}\Big|_{s_{i-1}}}{\left\| \frac{\partial u}{\partial \lambda}\Big|_{s_i} \right\| \left\| \frac{\partial u}{\partial \lambda}\Big|_{s_{i-1}} \right\|} \tag{4.38}$$

which is the cosine of the angle between the two most recent solution tangent vectors (with respect to λ). If the two most recent tangent vectors were in essentially the same direction, $\tau \to 1$, while if they were nearly orthogonal (such as near a turning point) $\tau \to 0$. The next arclength stepsize is then given by

$$\Delta s_{i+1} = \tau_i \Delta s_i \tag{4.39}$$

This method does require the storage of an additional vector (the old solution tangent) however this is a common characteristic of most steplength selection methods. In our experience, we found that when the solution vector u has entries varying by several orders of magnitude, then the contributions of the smaller components to the dot product in the numerator of τ can be essentially lost, rendering $\tau \approx 1$ even near turning points.

Another possibility is to simply scale the stepsize by the ratio of the norms of the two most recent solution tangents with respect to λ, i.e.

$$\Delta s_{i+1} = \frac{\left\| \frac{\partial u}{\partial \lambda}\Big|_{s_{i-1}} \right\|}{\left\| \frac{\partial u}{\partial \lambda}\Big|_{s_i} \right\|} \Delta s_i \tag{4.40}$$

This avoids the need to store the entire old tangent vector (we need store only its norm) and seems to work reasonably well at shrinking the stepsize in

practice. Unfortunately, it frequently appears to grow the stepsize more slowly than one might otherwise prefer in regions of slow solution change.

Another commonly-used technique for growing the arclength stepsize is to look at the number of Inexact Newton iterations required to converge the previous step. As in Seydel's excellent book [184], we may assume an "optimal" number of Newton iterations N_{opt} exists based on the other tolerances in a given problem, and then if the most recent continuation step i required N_i inexact Newton iterations, we scale the arclength stepsize according to

$$\Delta s_{i+1} = \frac{N_{\text{opt}}}{N_i} \Delta s_i \tag{4.41}$$

In a variation on this theme, and in acknowledgment of the fact that N_{opt} can be difficult to define in general, the LOCA authors adopt

$$\Delta s_{i+1} = \left[1 + a \left(\frac{N_{\text{max}} - N_i}{N_{\text{max}} - 1} \right)^2 \right] \Delta s_i \tag{4.42}$$

where N_{max} is the maximum-allowable number of Newton iterations, and a is an "aggressiveness" factor determining how quickly to grow the step. In practice, we have found that combining both Eqns. (4.40) and (4.42) with an aggressiveness factor $a \approx 1$ works reasonably well for arclength stepsize selection at negligible extra cost.

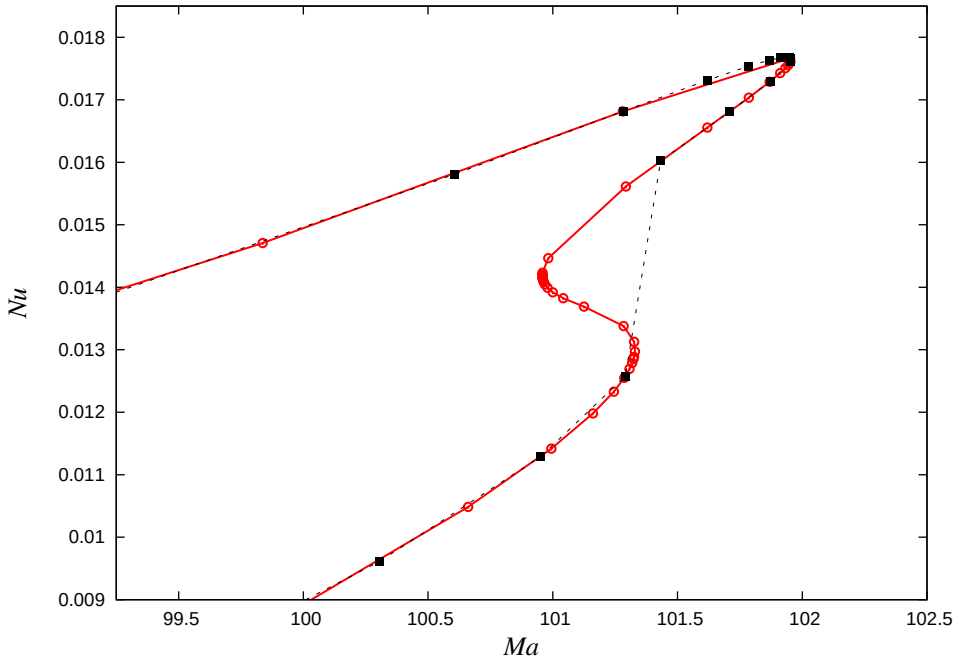


Figure 4.4: Example of two tracings (computed in opposite directions) showing the effect of too large an arclength stepsize. In this case, the continuation scheme was conducted in the Ma variable, first on the red branch (circular markers) going from top to bottom, and a second time with the black dashed-line branch (square markers) which goes from bottom to top. The second branch tracing misses two turning points because it fails to reduce the arclength stepsize quickly enough.

4.4.4 Numerical Solution of Bordered System

By inspection of Eqns. (4.5) and (4.9), it is obvious that one needs to frequently solve linear systems of the form

$$\begin{bmatrix} G_u & G_\lambda \\ N_u^t & N_\lambda \end{bmatrix} \begin{bmatrix} \delta u \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} r \\ \rho \end{bmatrix} \tag{4.43}$$

where the Jacobian matrix is reused, but the right-hand side and unknown vectors are interchangeable depending on whether we are solving the augmented system (Eqn. (4.5)) or the tangency condition (Eqn. (4.9)). While it would technically be possible to solve (4.43) directly or iteratively as a sparse, bordered system, it turns out to be more convenient (in terms of reuse of existing computational tools) to use the following two-solve procedure for each solve of Eqn. (4.43). In the case of Eqn. (4.5), Algorithm 1 must be employed at

Algorithm 1 Two-step solve procedure for systems in the form of Eqn. (4.43).

Solve $G_u y = G_\lambda$ for y (first solve).
Solve $G_u z = r$ for z (second solve).
Set $\delta\lambda = \frac{\rho - N_u^t z}{N_\lambda - N_u^t y}$
Set $\delta u = z - (\delta\lambda) y$

each Newton step, making this scheme roughly twice as expensive as a normal Newton solve, and therefore only attractive to use near turning points or regions of rapid change in the solution relative to the control parameter λ. To see how Algorithm 1 arises, simply write out the first "row" of Eqn. (4.43)

$$G_u \delta u + (\delta\lambda) G_\lambda = r \tag{4.44}$$

(Recall that $\delta\lambda$ is a scalar, G_λ is a vector of the same length as the residual vector G.) Multiplying Eqn. (4.44) by a notional G_u^{-1} (note: we of course do not explicitly form the inverse Jacobian, just compute its action) we obtain

$$\delta u + (\delta\lambda) G_u^{-1} G_\lambda = G_u^{-1} r \tag{4.45}$$

Now, by our definitions of y and z given in Algorithm 1 we have

$$\delta u + (\delta\lambda) y = z \tag{4.46}$$

Or simply,

$$\delta u = z - (\delta\lambda) y \tag{4.47}$$

From the second "row" of Eqn. (4.43) we obtain

$$N_u^t \delta u + (\delta\lambda) N_\lambda = \rho \tag{4.48}$$

Or, plugging in Eqn. (4.47) for δu

$$N_u^t \left(z - (\delta\lambda)\, y\right) + (\delta\lambda) N_\lambda = \rho \tag{4.49}$$

Finally, Eqn. (4.49) can be rearranged to obtain a scalar equation for $\delta\lambda$

$$\delta\lambda = \frac{\rho - N_u^t z}{N_\lambda - N_u^t y} \tag{4.50}$$

and plugging $\delta\lambda$ back into Eqn. (4.47) we obtain the final result for δu.

We observe that this two-step solve procedure never involves the "full" augmented matrix, and still requires linear system solves with the Jacobian G_u which may be singular. Therefore, while the scheme is attractive to implement in an existing numerical code because it is non-intrusive (does not

require changing the existing Jacobian assembly routine) it may not succeed if the solution ends up precisely at a turning point. However, Keller gives a proof [106] of how successive approximation iterations (and hints for Newton iterations as well) will always be able to "jump" over such singular points provided the initial guess lies in a particular domain of attraction around the singular point. In practice, we have never seen the scheme fail to pass a turning point provided it is allowed enough arclength stepsize reductions.

Another important issue is choosing the correct tolerance for the iterative solutions of the $G_u y = G_\lambda$ and $G_u z = -G$ systems at each Newton step. We typically employ the method of Eisenstat and Walker [76] discussed in §2.3 for the solution of the z system, since solving this system is the prototypical application of the inexact Newton method.

Some additional care must be taken when selecting the tolerance for the y system, since its right-hand side $G_\lambda \nrightarrow 0$ as the Newton iterations proceed. Our linear algebra package (PETSc [19]) requires the selection of a "relative" tolerance where relative in this instance means relative to $\|G_\lambda\|$. Therefore, we modify the standard tolerance selection of Eqn. (B-2.27) in a rather straightforward way by taking

$$\eta_k \leftarrow \eta_k \frac{\|G\|}{\|G_\lambda\|} \tag{4.51}$$

where $\|G\|$ is the current nonlinear system residual.

4.4.5 Boundary Conditions

There is also the question of how the boundary conditions, which are posed on the original problem $G(u, \lambda) = 0$ should enter into the augmented system. For example, how should boundary conditions be applied in the tangency system $G_u \frac{\partial u}{\partial \lambda} = -G_\lambda$? It seems logical that any non-homogeneous Dirichlet boundary conditions imposed on the original problem should be imposed as homogeneous Dirichlet conditions on the tangency condition, effectively enforcing that the change in that finite element coefficient with respect to changes in λ is zero.

4.5 Linear Stability Analysis

In this section, we consider the linear stability of the solutions u obtained on solution arcs using the pseudo-arclength continuation methods described previously. The general time-dependent form of Eqn. (4.1) may be written as

$$B\frac{\partial u}{\partial t} = G(u, \lambda) \tag{4.52}$$

where B is a linear operator (an $n \times n$ matrix in the discrete setting) which we will refer to from here on as the "mass matrix." A wide variety of problems can be handled by taking B as the $n \times n$ identity matrix, but for our purposes we will consider a general linear operator. Suppose further that a steady solution u_0 satisfying

$$0 = G(u_0, \lambda) \tag{4.53}$$

is known, and the linear stability of u_0 is of interest. We define

$$\eta := u - u_0 \tag{4.54}$$

as a "small" perturbation away from the known solution u_0. Differentiating η with respect to time, we observe that

$$\begin{aligned} B\dot{\eta} &= B\dot{u} - B\dot{u_0} \\ &= G(u,\lambda) - G(u_0,\lambda) \\ &= G(u,\lambda) \end{aligned} \tag{4.55}$$

Holding λ fixed, we expand about u_0 to obtain

$$\begin{aligned} B\dot{\eta} &= G(u_0+\eta,\lambda) \\ &= G(u_0,\lambda) + G_u(u_0,\lambda)\eta + \mathcal{O}(\|\eta\|^2) \\ &\approx G_u(u_0,\lambda)\eta \end{aligned} \tag{4.56}$$

since the first term on the right-hand side of Eqn. (4.56) vanishes by Eqn. (4.53). We now have a linear evolution problem for η. Inserting the *ansatz*

$$\eta = \exp(\sigma t)x \tag{4.57}$$

for σ a (possibly complex) scalar and $x \in \mathbb{R}^n$ (or $\mathbb{C}^n$) into Eqn. (4.56), and letting $A := G_u(u_0,\lambda)$ stand for the Jacobian matrix evaluated at u_0, one obtains

$$\sigma Bx = Ax \tag{4.58}$$

Eqn. (4.58) is in the standard form of a generalized eigenvalue problem, with eigenvalue σ and eigenvector x.

4.5.1 Numerical Solution of the Generalized Eigenvalue Problem

Direct calculation of the eigenvalues of an $n \times n$ matrix is an expensive $\mathcal{O}(n^3)$ operation, and is not feasible for the three-dimensional coupled heat transfer and fluid-flow systems considered here. Fortunately, linear stability analysis requires only knowledge of the sign of the "rightmost" (on the real axis) eigenvalue. If at least one eigenvalue has positive real part, the solution is linearly unstable, whereas, if all the eigenvalues have negative real part, the solution is said to be linearly stable.

A classical method for computing the largest-in-magnitude eigenvalue of the spectrum is the power iteration. Unfortunately, the power iteration is unsuitable for our purposes because (1) the largest-in-magnitude eigenvalue is not necessarily the right-most eigenvalue, and (2) it is known to have very poor convergence properties when σ_1 and σ_2 (the largest and second-largest in magnitude eigenvalues, respectively) are close together. (The convergence rate is geometric with ratio $|\sigma_1/\sigma_2|$.)

The Arnoldi iteration [11] somewhat overcomes difficulty (2) listed above by computing the eigenvalues of the orthogonal projection of the matrix A onto the Krylov subspace. When A is symmetric, the Arnoldi iteration is equivalent to the Lanczos iteration [171], another method for determining eigenvalues. In this work, we employ the Arnoldi iteration as implemented by the SLEPc [94] eigensolver library, which is a natural extension of the PETSc library, and makes extensive use of its Krylov subspace solvers.

The Arnoldi iteration, like the power iteration, also converges to the largest-in-magnitude eigenvalues of A, and hence does not overcome difficulty (1) listed above. For example, in our particular application, the time-independent constraint (continuity equation) rows of A effectively lead to $-\infty$ eigenvalues, and as such will be the eigenvalues obtained by the power or Arnoldi iterations. To overcome this issue, we typically employ the shift-and-invert spectral transformation (as implemented by the SLEPc library). Beginning with Eqn. (4.58), we choose a shift $s \neq \sigma$ and subtract sBx from both sides to obtain

$$(A - sB)\, x = (\sigma - s)\, Bx \tag{4.59}$$

In general it is possible to use a complex shift s, although in this work we employ only real-valued shifts. Next, multiplying both sides of Eqn. (4.59) by $(\sigma - s)^{-1} (A - sB)^{-1}$ we obtain

$$(\sigma - s)^{-1}\, x = (A - sB)^{-1}\, Bx \tag{4.60}$$

Now, Eqn. (4.60) is a standard eigenvalue problem

$$Tx = \gamma x \tag{4.61}$$

where $T := (A - sB)^{-1} B$ is the transformed matrix and $\gamma := (\sigma - s)^{-1}$ is the transformed eigenvalue. The inverse is of course never computed explicitly; its action is determined with linear system solves using PETSc's iterative methods.

The shift-and-invert transform maps eigenvalues $|\sigma| \gg |s|$ to $\gamma \approx 0$, ensuring that they are no longer the dominant eigenvalues of the spectrum and

that they are not converged by the Arnoldi iteration. This is ideal behavior for e.g. the $-\infty$ eigenvalues in our systems of interest. The Arnoldi iteration can also be made to converge very quickly to a suspected eigenvalue σ by choosing a shift s very close to said eigenvalue. This shift essentially maps σ to the dominant eigenvalue of the spectrum, and can be useful in cases where a positive eigenvalue is perhaps partially (but not to a sufficient tolerance, say) converged in a preceding Arnoldi iteration.

The shift-and-invert spectral transformation just described still suffers from the drawback that the eigenvalues which it converges are not guaranteed to be rightmost eigenvalues, just those closest to the shift s. A different, but related spectral transformation which attempts to map rightmost eigenvalues to the dominant eigenvalues of the spectrum is the Cayley transform [127]. In addition to the shift s, we also choose "anti-shift" $t \neq \sigma$, and beginning again with Eqn. (4.58) we now add tBx to both sides to obtain

$$(A + tB)\, x = (\sigma + t)\, Bx \tag{4.62}$$

Multiplying both sides of Eqn. (4.62) by $(\sigma - s)$ we obtain

$$\begin{aligned} (\sigma - s)\,(A + tB)\, x &= (\sigma - s)\,(\sigma + t)\, Bx \\ &= (\sigma + t)\,(\sigma Bx - sBx) \\ &= (\sigma + t)\,(Ax - sBx) \\ &= (\sigma + t)\,(A - sB)\, x \end{aligned} \tag{4.63}$$

Finally, we multiply both sides of Eqn. (4.63) by $(\sigma - s)^{-1}\,(A - sB)^{-1}$ to ob-

tain

$$(A - sB)^{-1}(A + tB)\,x = \left(\frac{\sigma + t}{\sigma - s}\right)x \tag{4.64}$$

Eqn. (4.64) is thus a standard eigenvalue problem in the same form as Eqn. (4.61), where now $T := (A - sB)^{-1}(A + tB)$ and $\gamma := \left(\frac{\sigma+t}{\sigma-s}\right)$. The Cayley transform has the property that it maps the line $Re\{\sigma\} = \frac{1}{2}(s - t)$ in the σ-plane to the unit circle in the γ-plane [127]. To see this, we can evaluate γ (as defined above) at an arbitrary point $\sigma^* = \frac{1}{2}(s - t) + bi$ ($b \in \mathbb{R}$ arbitrary) on the line $Re\{\sigma\} = \frac{1}{2}(s - t)$ defined previously to obtain

$$\gamma|_{\sigma^*} = \frac{\frac{1}{2}(s + t) + bi}{-\frac{1}{2}(s + t) + bi} \tag{4.65}$$

which, in polar coordinates, becomes

$$\gamma|_{\sigma^*} = \exp(2i\theta) \tag{4.66}$$

with $\theta := \tan^{-1}\left(\frac{2b}{s+t}\right)$, i.e., the unit circle in the γ-plane. Since, for linear stability analysis, we are mostly interested in eigenvalues σ with positive real part, it has been suggested [41, 118] that one should select the shift and anti-shift according to the following guidelines

1. Select $s > 0 \in \mathbb{R}$ such that $s > Re\{\sigma_1\}$ and, if possible, $|s| \approx Im\{\sigma_1\}$, where σ_1 is the rightmost eigenvalue of interest. (Note that this guideline is "implicit," since of course σ_1 is the eigenvalue we are trying to find. However, if an approximation to σ_1 is available, one may use it to select the shift by the preceding guideline.)

2. $t = s$

The second guideline is designed to ensure that the imaginary axis ($Re\{\sigma\} = 0$) is mapped to the unit circle in the γ-plane, and that eigenvalues to the right of the imaginary axis are mapped to dominant eigenvalues, and hence converged by Arnoldi iterations. Conversely, eigenvalues with negative real parts are mapped by this transformation to small magnitudes and hence not converged by the eigensolver.

In practice, the Cayley transform will converge to eigenvalues at $+\infty$ if any exist. This is in contrast to the shift-invert transform which maps eigenvalues at $\pm\infty$ to zero, and (though we are not completely clear why) appears to lead to problems when the penalty formulation is used to enforce Dirichlet boundary conditions. We believe the penalty terms may lead to spurious positive eigenvalues which are $\mathcal{O}$(penalty) in size, and are currently working to understand exactly what is happening.

Finally, even when using the Cayley transform, Lehoucq [118] states that "There is no available theory to verify that the rightmost eigenvalue has been computed." The Cayley transform is therefore no panacea, and the practitioner must rely on additional information which is available about the solution when determining if an eigenvalue with positive real part truly exists.

This additional information includes nearby converged time-accurate solutions (time-dependent solvers should only converge to linearly-stable steady states) as well as trusted eigenvalue information from other nearby steady so-

lutions. It is quite often possible to track the movement of a single negative eigenvalue as it approaches, and finally crosses over, the imaginary axis to become unstable. Finally, features of the solution branch, including turning points and symmetry-breaking bifurcations, generally signal the birth of a positive eigenvalue. For example, we know this is the case near turning points since the Jacobian is singular (has a zero eigenvalue) there, implying a change in sign of at least one eigenvalue near such points.

Chapter 5

RBM Results for Continuation in Aspect Ratio

We now comment briefly on the question of the connection between experimentally-reproducible results for the R-B-M problem and the calculations considered here. Parameters which can be most easily and continuously controlled by the experimentalist typically include the temperature difference applied across the enclosure, the depth of the liquid layer, and possibly the height of the cooling surface above the layer (i.e. the thickness d_{air} of the air layer above the fluid).

In general, it is difficult to experimentally vary the aspect ratio of the layer in a continuous way *while simultaneously holding all other problem parameters constant*. Experimental results which vary the aspect ratio (such as the classical results of Koschmieder and Prahl [113] and work by Ramon [162] in cylindrical containers) typically involve first increasing the depth of the liquid layer (which also increases the Ra and Ma numbers, though at different rates) and then subsequently increasing the temperature difference (and hence further increasing the Ra and Ma numbers) across the layer until a convection pattern appears. We generally regard the fluid properties such as density, vis-

cosity, and thermal diffusivity as fixed for the fluids considered here (though they do vary slightly with temperature) and not continuously variable.

Under these experimental restrictions, the variation of the non-dimensional parameters in the governing equations are not perfectly independent. For example, if the researcher increases the temperature gradient across the layer, both the Ra and Ma numbers vary simultaneously, making it difficult to determine experimentally the independent effects of increasing either parameter alone (a test which is possible in numerical simulation). Likewise, if he performs the experiment in two different containers with different depths, the two experiments will have different values for *all* of the non-dimensional parameters mentioned in the beginning of this section, since they all depend on the depth of the layer, d. Other parameters, such as the Bi number, which are easy to vary in numerical simulations can be much more difficult (though potentially possible) to vary in experimental apparatus.

For these reasons, it may be difficult to discover which single parameter (if any) has the greatest effect on pattern formation in the enclosure. Furthermore, it suggests that one need not (and indeed should not) limit numerical simulations to only what is possible in the laboratory, but to still interpret the results in light of what is physically realizable. In the following results, where the variation of the total heat flux (per unit area) through the fluid layer is plotted vs. the container aspect ratio, we do not suggest that the solution branches shown should be reproducible by experiment, but rather that single points on the solution branches should be observable experimentally (for

particular values of Ra, Ma, and Bi). We have included experimental images (though we do not currently have detailed experimental data for further comparison) where appropriate in order to show that the numerical solutions obtained are in good qualitative and semi-quantitative agreement even though high-resolution validation of the solutions is not possible.

In §5.1, we summarize the unsteady results already discussed in some detail in Chapter 3. Then in the remaining sections of this Chapter we discuss the various convection patterns obtained by solving the steady form of the governing equations with pseudo-arclength continuation. The format of the rest of this Chapter is given in more detail in §5.1.

5.1 Summary of Unsteady Solutions

We now plot all of the time-accurate steady states obtained by the unsteady equations on a single plot. The scalar measure used to plot each solution is the Nusselt number, in this case defined by

$$Nu := \int_{\partial\Omega_s} \tilde{T}\, ds \tag{5.1}$$

The Nusselt number is invariant under $\pi/2$–radian rotations of the surface temperature field, hence we cannot differentiate in the following plots between such solutions. Since the domain is already non-dimensionalized to be of unit area, we do not need to further scale Nu by the surface area, as is the usual practice. Finally, in all the solutions presented here, $Nu \geq 0$, and $Nu = 0$ implies the quiescent (trivial) solution. We observe two general trends from Fig. 5.1. The first is that increasing aspect ratios correlate with increasing numbers of convection cells. The second is that higher Nu generally results from larger aspect ratio containers. We also note that the change in the number of convection cells is not continuous with respect changes in the aspect ratio Γ. Near certain aspect ratios, the stable steady state obtained by the time-accurate calculation jumps rapidly to a different number of convection cells. This jump in the number of cells is typically (but not always, as in the change from 4–5 cells) accompanied by a large jump in Nu.

These results from the unsteady solves lead to the following questions: Do there exist multiple stable steady states with different numbers of convection cells for a single aspect ratio? What is the (nonlinear-dynamics) mech-

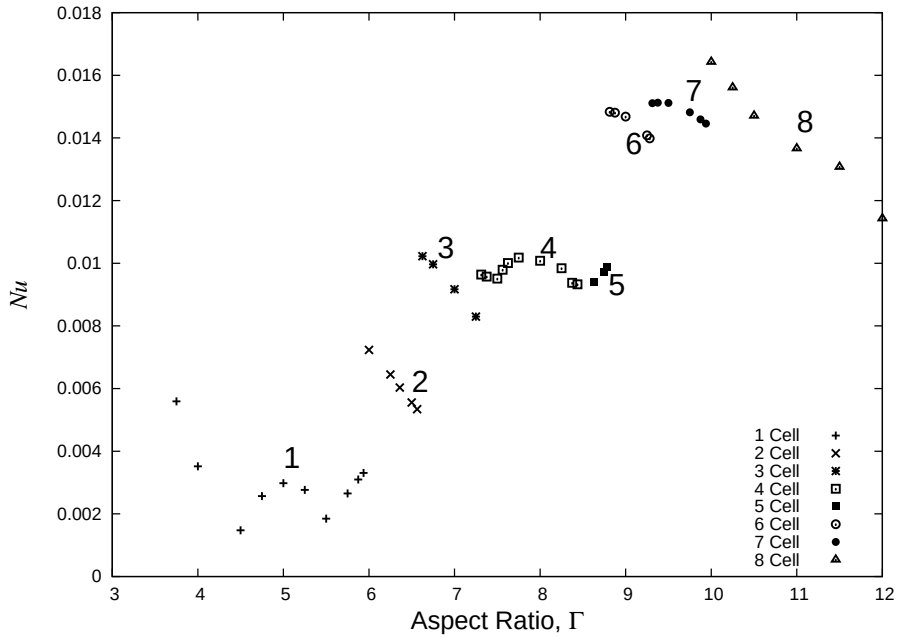


Figure 5.1: Nusselt numbers at various aspect ratios computed with the unsteady equations. The different styles of the data points correspond to the different numbers of convection "cells" observed at each aspect ratio. We have also placed these numerical values nearby on the plot to assist in readability.

anism for transition from one cellular state to another? Is there a unifying principle which leads one cellular configuration to be selected over others at a given aspect ratio?

In order to answer these questions, we have used standard arclength continuation techniques in conjunction with linear stability analysis to extend the data shown in Fig. 5.1. The results of these studies are presented in the

following way: in §5.2 we investigate more closely the 1, 2, and 3-cell solution branches, in §5.3 we investigate the 4 and 5-cell configurations, and in §5.4 we tackle the 6, 7, and 8-cell configurations. Along the way we also discuss some relevant experimental results, and how they relate to the numerical solutions obtained in this study.

5.2 One, Two, and Three-Cell Solution Branches

Here we show the stability information for several of the smallest aspect-ratio solution branches, which comprise 1, 2, and 3 convection cells. The vertical arrows in Fig. 5.2 denote the (unsteady) transition from branch to branch which would occur if the aspect ratio was slowly increased. Hysteresis near these transition points (evidenced by overlapping regions of stable steady solutions) is present near $\Gamma \approx 6$ and $\Gamma \approx 6.5$. The maximum aspect ratio increment of the curves in Fig. 5.2 and subsequent stability plots is $\Delta\Gamma \approx 0.025$, although the actual increments in Γ are of course selected by the continuation algorithm itself, and are often substantially smaller near turning points. The circular markers in Fig. 5.2 and those like it which follow, correspond to points in (Γ,Nu) space obtained from the solution of the unsteady governing equations, and shown originally in Fig. 5.1.

We shall describe briefly the hysteresis near $\Gamma \approx 6$. Starting on the one-cell branch (the lowest branch, near $\Gamma = 6$) and slightly increasing the aspect ratio causes the solution to transition (via unsteady time evolution processes) from the one-cell configuration (shown in Fig. 5.4) to a two-cell solution (shown in Fig. 5.17). We note that the instabilities may be of oscillatory (Hopf) type or exponential growth, depending on whether the unstable eigenvalue is real or complex. A good general reference for instability is Seydel's book [184].

A subsequent slight decrease in the aspect ratio would not send this two-cell solution back to the previous one-cell configuration, but rather to a two-cell configuration with a slightly higher Nu value. This two-cell solution eventually

becomes unstable somewhere between $\Gamma = 5.5$ and $\Gamma = 5.4$, at which point the one-cell solution (on the lower branch, shown in Fig. 5.4) is the stable one. A similar process occurs when the two-cell branch becomes unstable under increasing aspect ratio (around $\Gamma = 6.5$ in Fig. 5.2) and transitions to the three-cell branch (shown in Fig. 5.27).

The algorithm for computing the solution branches shown in Fig. 5.2 and similar figures which follow is given below as Algorithm 2. The method is initialized with u_0, a finite element solution obtained by timestepping the unsteady governing equations to steady state with corresponding parameter value λ_0. Here, λ_0 is a generic control parameter, e.g. the aspect ratio Γ or the Marangoni number Ma. In the first solve step, to obtain u_1, we employ so-called "zeroth-order" incremental continuation in λ, which typically converges rapidly if $\lambda_1 \approx \lambda_0$ and we are far from any turning points. Here we have covered the highlights of the scheme; for more details, see the description of the pseudo-arclength continuation method in Ch. §4.

Algorithm 2 Algorithm for tracing non-trivial solution branches pseudo-arclength continuation. See Ch. §4 for additional details.

(Initialization) Obtain u_0, λ_0 from unsteady solve.
Choose next control parameter value: $\lambda_1 = \lambda_0 + \Delta\lambda$ ($\Delta\lambda$ small).
Solve the steady equations for u_1, using u_0 as initial guess.
Compute tangents $\partial u/\partial s|_1$, $\partial\lambda/\partial s|_1$ via finite differences.
Compute initial guess $\tilde{u}_2$, $\tilde{\lambda}_2$ using the tangent.
for $n = 2$ to n_{steps} **do**
  Solve the augmented Jacobian system for new values u_n, λ_n.
  Update tangents $\partial u/\partial s|_n$, $\partial\lambda/\partial s|_n$ with one additional linear solve.
  Compute initial guess $\tilde{u}_{n+1}$, $\tilde{\lambda}_{n+1}$
end for

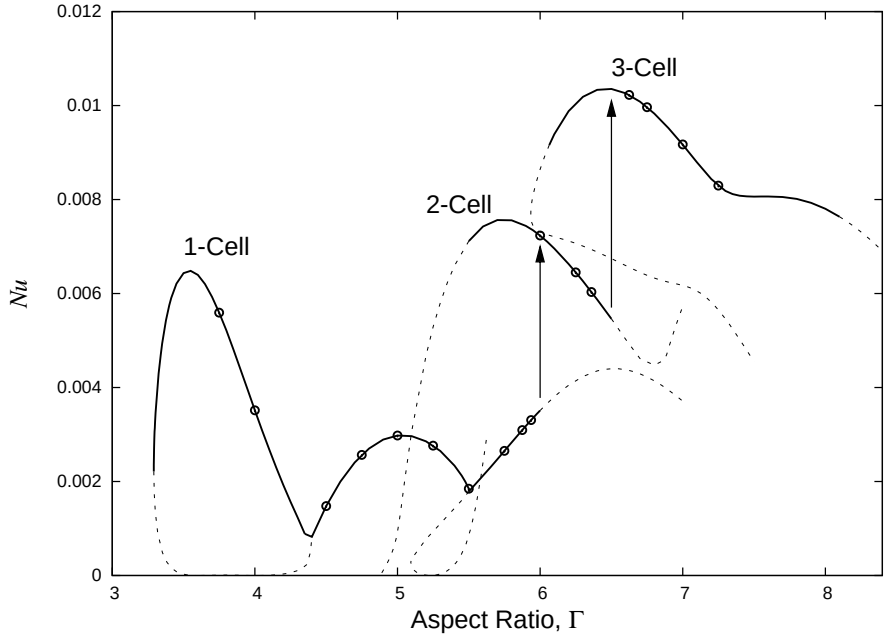


Figure 5.2: Solution branches computed via arclength continuation for 1, 2, and 3-cell solutions. Branch segments composed of linearly-stable solutions are shown with solid lines while branch segments composed of linearly-unstable (unstable to infinitesimal perturbations) solutions are depicted with dashed lines. The circular markers represent the one, two, and three-cell data points shown originally in Fig. 5.1.

5.2.1 One-Cell Solution Branch

We now describe in more detail the stable one-cell solution branch depicted in Fig. 5.2. We note that, at least for the values of the other control parameters (Ma, Ra, Bi, Pr) used, the smallest aspect ratio for which a non-trivial solution was found was $\Gamma \approx 3.3$. Experimental [113] and numerical [126] solutions have been found for smaller aspect ratios at higher Ma than considered at present. Solution 1, which is shown in detail subfigure 1 of Fig. 5.4, occurs around $\Gamma = 3.3$ and consists of a single upwelling "cell" at the center of the domain. At these small aspect ratios (relatively "deep" fluid layers) the buoyant instability is more important relative to the surface-tension driven instability, and the convection cell on the surface does not have the characteristic well-defined boundaries it does at higher aspect ratios.

This single central upwelling continues for the solutions marked 2 and 3 in Fig. 5.3, though after solution 2 with a markedly lower Nusselt number. By solution 3, we just begin to see warm, upwelling fluid in the container's corners (subfigure 3 of Fig. 5.4), and by solution 4, we observe a single, circular convection cell whose borders are just "tangent" to the sides of the container. Solution 4 also corresponds to a local minimum in the Nu value. Representative dominant (rightmost) eigenvalues for this branch of solutions are given in Table 5.1.

Also at solution 4, the symmetric one-cell solution loses stability to the asymmetric branch of solutions labeled 5–9 in Fig. 5.3. Here and elsewhere, we shall refer to branches composed of asymmetric solutions simply as "asym-

Table 5.1: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment consisting of stable solutions 1–4 shown in Fig. 5.3. Near $\Gamma = 3.3$, the rightmost eigenvalue has just crossed over zero to become stable. While on this branch, the rightmost eigenvalue takes on a value close to -4. The stable solution branch is the solid line connecting solutions 1–4 in Fig. 5.3.

Γ	σ_i		
3.300	-0.536	-10.927	-15.459
3.550	-4.041	-14.418	
3.750	-4.824		
4.000	-4.261		

metric branches." In solution 5 of Fig. 5.3, we observe warm fluid from one of the four corners (none of the four is preferred, the particular form of the symmetry-breaking being most likely selected by finite precision arithmetic affects) forcing the central cell towards the opposite corner. Representative dominant eigenvalues from this branch are given in Table 5.2. As parameter continuation proceeds, the central cell is forced further and further towards the corner in solutions 6 and 7 of Fig. 5.3, and it finally begins to move back toward the center in solutions 8 and 9. By solution 9 (which occurs around $\Gamma = 5.5$) this asymmetric branch ends by intersecting with the symmetric branch of one-cell solutions labeled 10 and 11.

We note that the symmetric one-cell solutions 10 and 11 are different from the symmetric one-cell solutions 1–4 encountered previously. They have a well-defined cell boundary, and do not cover as much of the free surface.

Table 5.2: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment consisting of stable solutions 4–9 shown in Fig. 5.3. Complex-conjugate eigenpairs are presented together. Near $\Gamma = 4.4$, the rightmost eigenvalue has just turned negative, and it continues to become more negative with increasing Γ until it reaches the new branch which connects solutions 9–11.

Γ	σ_i		
4.400	-0.013	$-0.538 \pm 0.643i$	
4.500	-0.066	$-2.070 \pm 2.332i$	
4.600	-0.114	$-3.567 \pm 3.754i$	
4.700	-0.188	$-4.926 \pm 4.959i$	
4.800	-0.311	$-6.178 \pm 5.918i$	
4.900	-0.500	$-7.401 \pm 6.688i$	-8.809
5.000	-0.755	$-8.630 \pm 7.416i$	
5.100	-1.057	$-9.004 \pm 0.335i$	
5.400	-1.375	-3.123	-5.447
5.500	-0.167	-0.257	-3.835

Table 5.3 gives the rightmost eigenvalues along this branch, including unstable regions of the branch (as marked by dashed lines).

Finally, Figures 5.5–5.15 give additional details on the surface temperature and velocity fields at solutions 1–11 just described.

Table 5.3: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment consisting of solutions 9–11 shown in Fig. 5.3. Complex-conjugate eigenpairs are presented together. A positive eigenvalue is first detected for increasing Γ around $\Gamma = 6$, and the magnitude of this positive eigenvalue increases with increasing Γ along the remainder of the branch.

Γ	σ_i		
5.400	1.673		
5.450	0.934		
5.500	-0.170612	-0.170618	-3.886
6.010	0.475	-9.248	
6.120	1.165	-9.911	
7.497	8.570	-9.012	

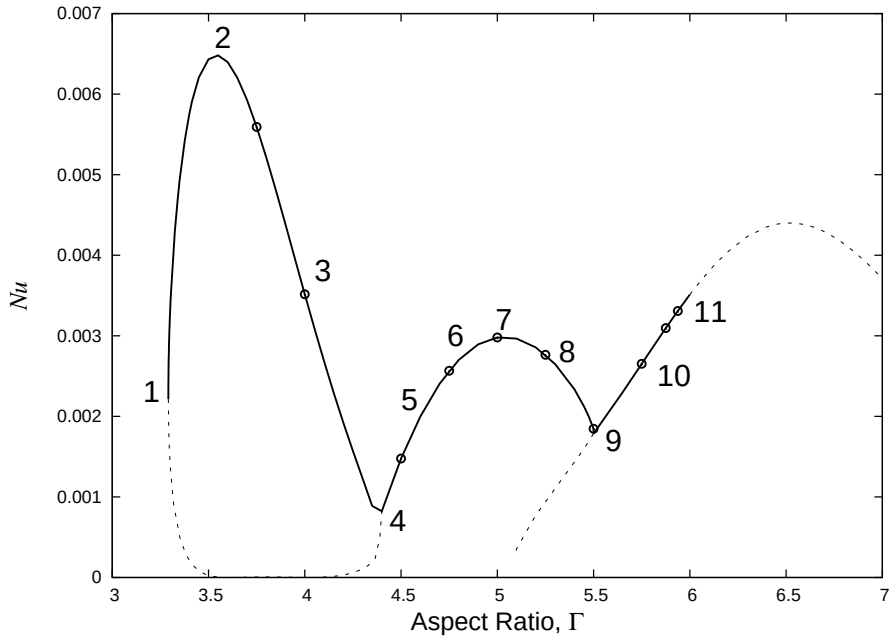


Figure 5.3: Computed solution branch for one-cell solutions. The numbers correspond to solutions discussed in greater detail in the text, and in Fig. 5.4. Also shown, as circular markers, are solutions from "unsteady" branch number 1 described previously and used as a starting point for computing this branch. The solutions marked here are shown as surface velocity vector plots in subfigures 1–11 of Fig. 5.4.

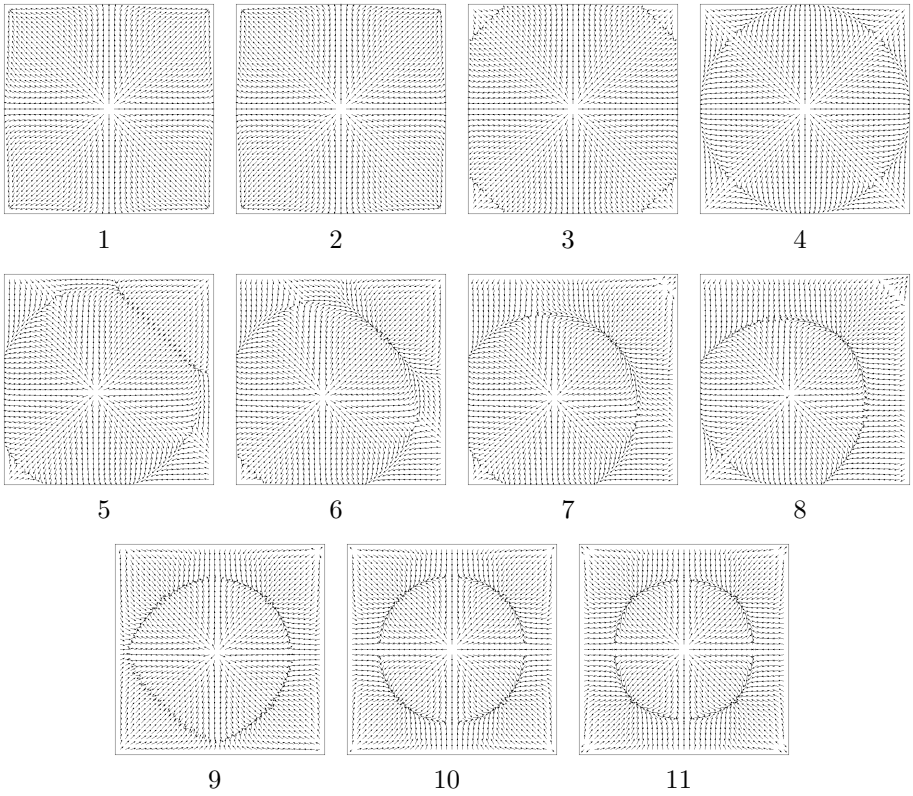
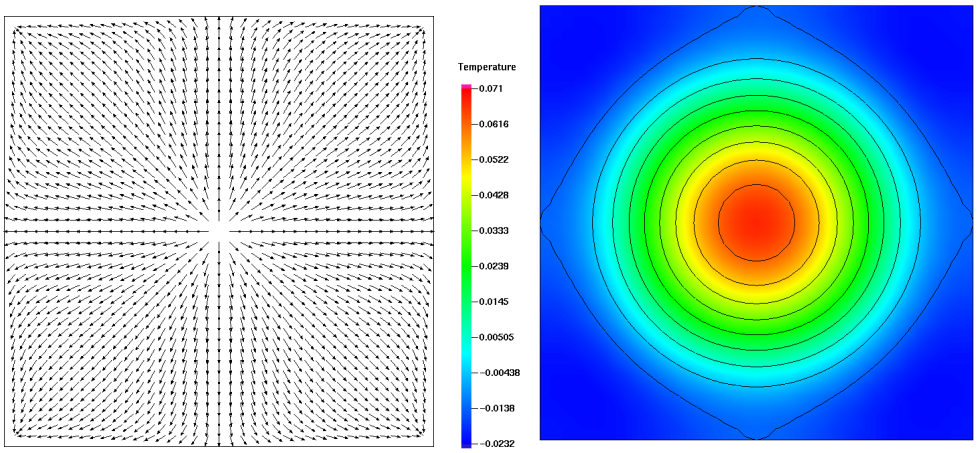
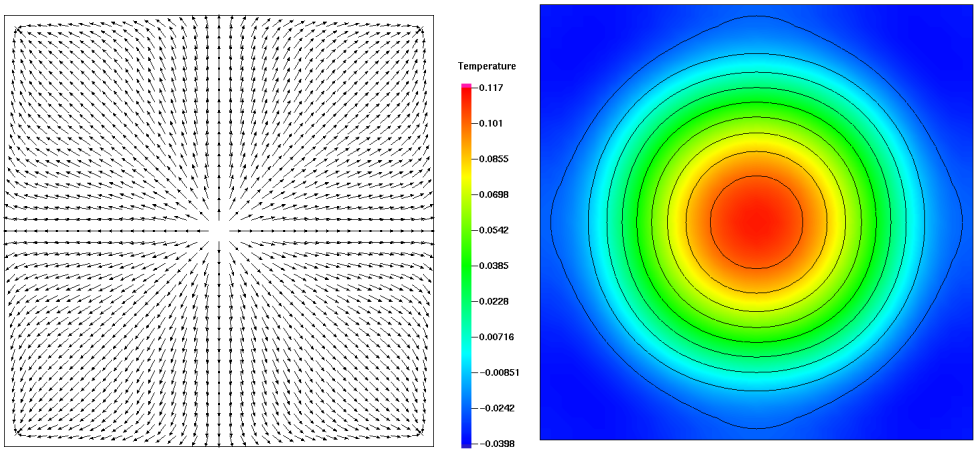


Figure 5.4: Velocity vectors for solutions on the branch depicted in Fig. 5.3.



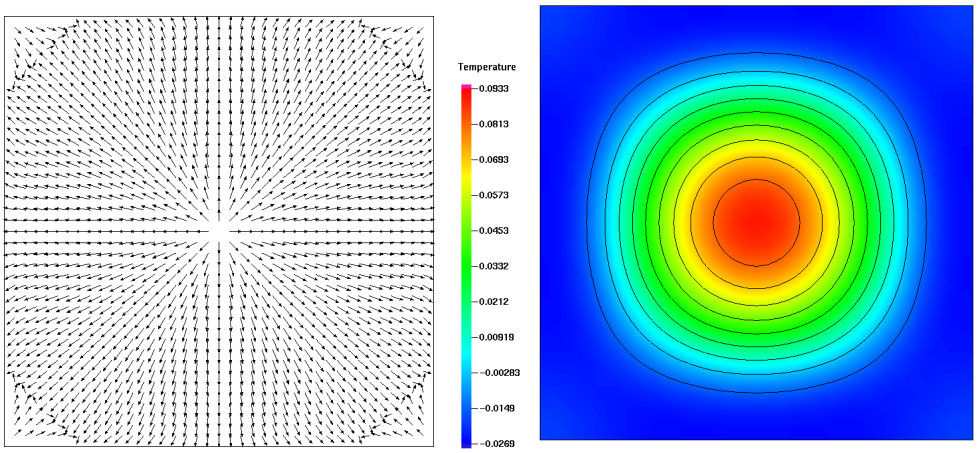
Velocity Vectors Temperature Contours

Figure 5.5: Solution 1 on the branch depicted in Fig. 5.3.



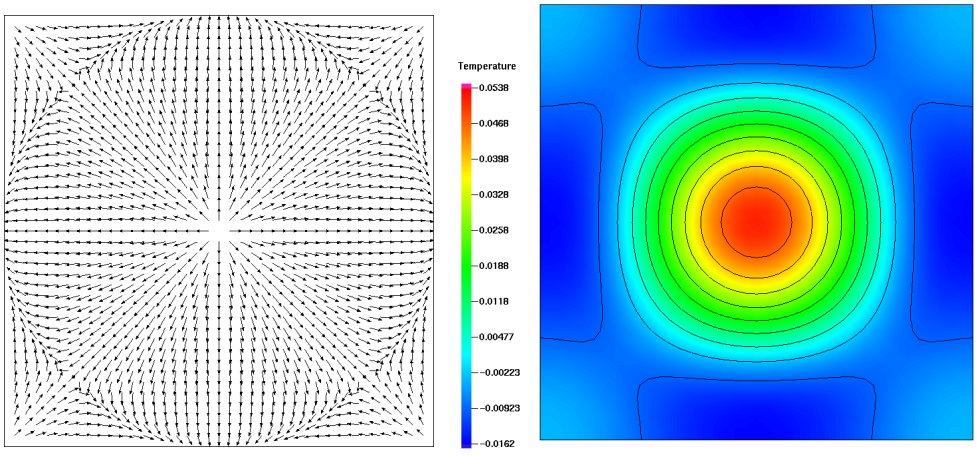
Velocity Vectors Temperature Contours

Figure 5.6: Solution 2 on the branch depicted in Fig. 5.3.



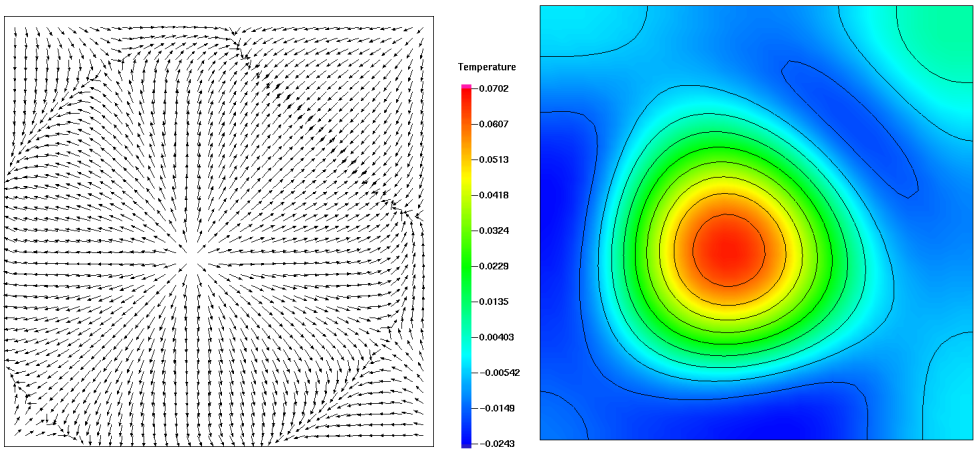
Velocity Vectors Temperature Contours

Figure 5.7: Solution 3 on the branch depicted in Fig. 5.3.



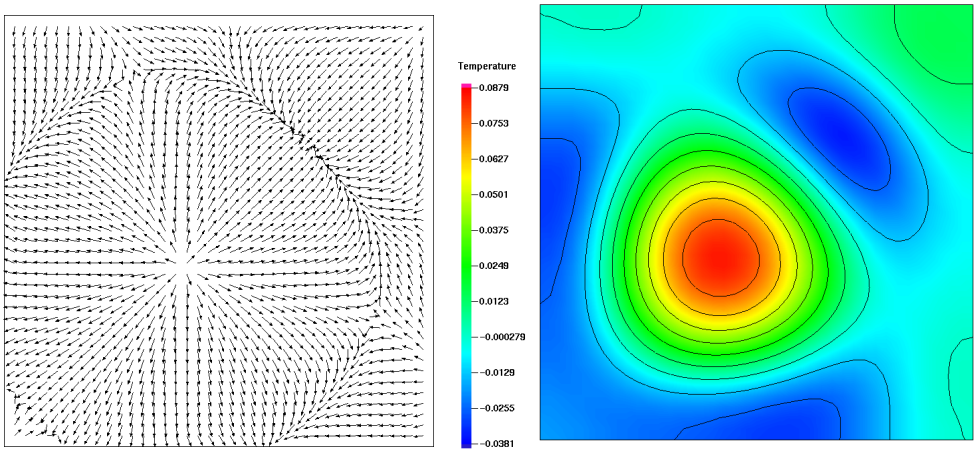
Velocity Vectors Temperature Contours

Figure 5.8: Solution 4 on the branch depicted in Fig. 5.3.



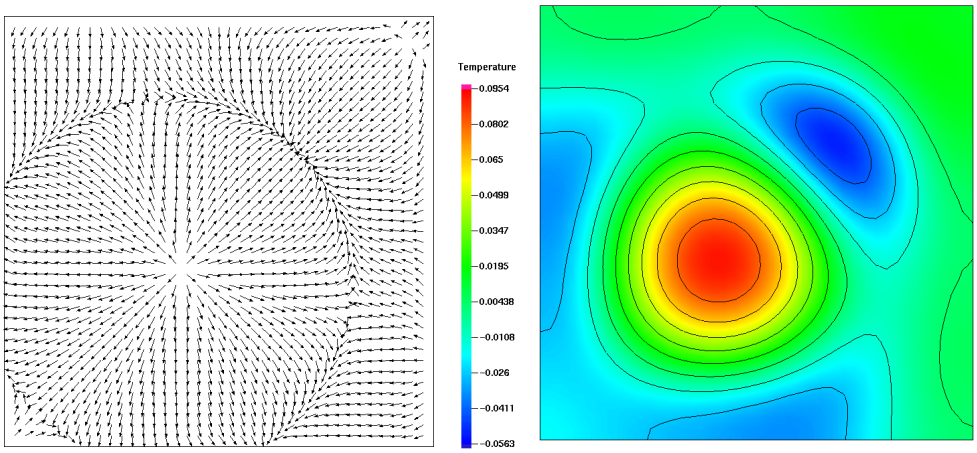
Velocity Vectors Temperature Contours

Figure 5.9: Solution 5 on the branch depicted in Fig. 5.3.



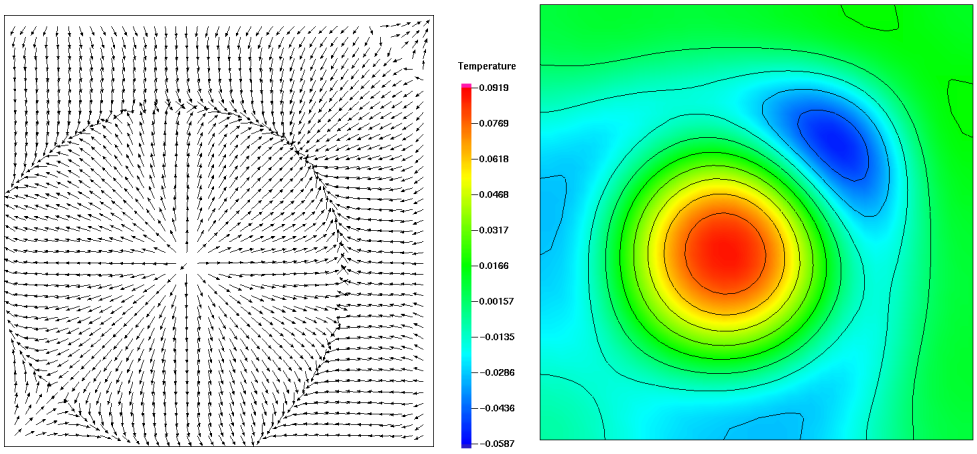
Velocity Vectors Temperature Contours

Figure 5.10: Solution 6 on the branch depicted in Fig. 5.3.



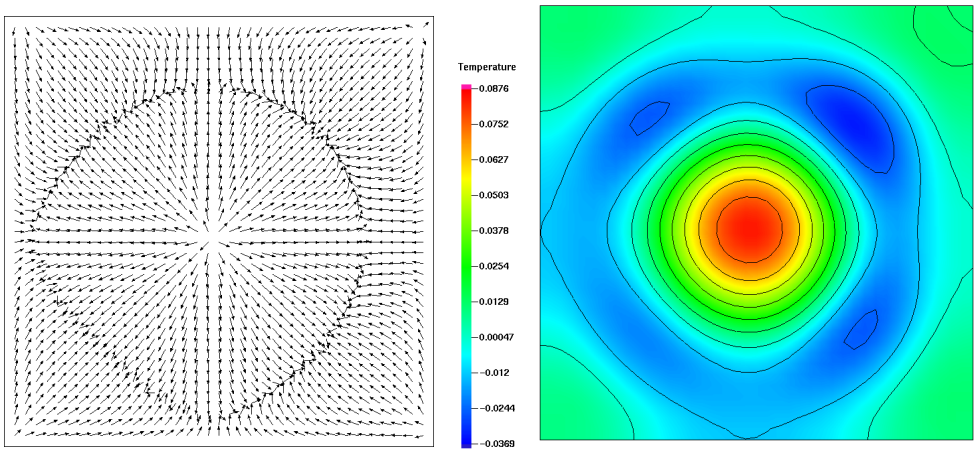
Velocity Vectors Temperature Contours

Figure 5.11: Solution 7 on the branch depicted in Fig. 5.3.



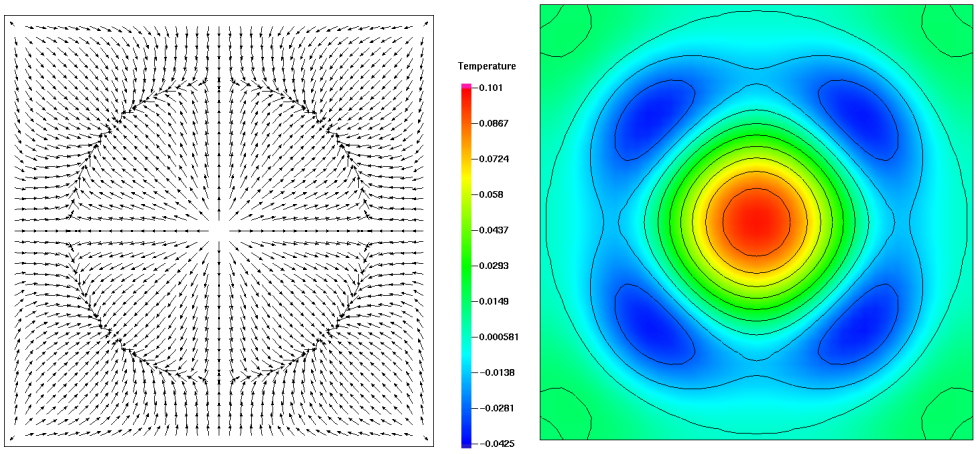
Velocity Vectors Temperature Contours

Figure 5.12: Solution 8 on the branch depicted in Fig. 5.3.



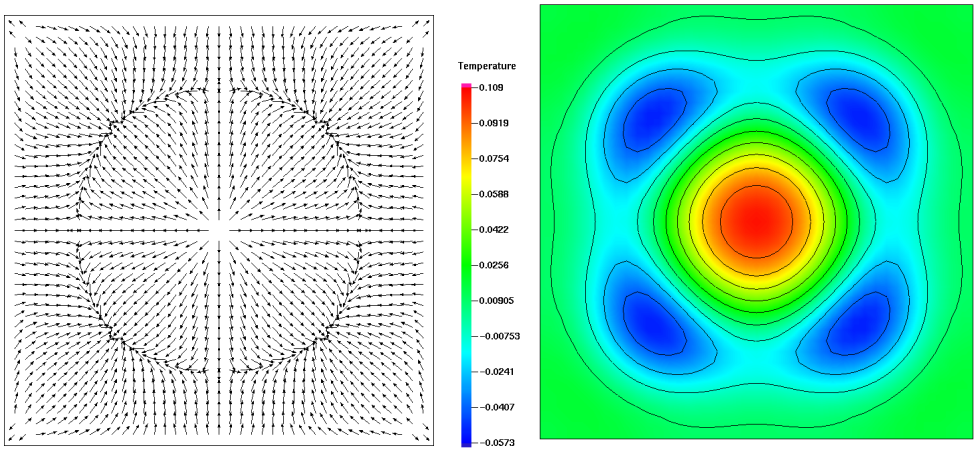
Velocity Vectors Temperature Contours

Figure 5.13: Solution 9 on the branch depicted in Fig. 5.3.



Velocity Vectors Temperature Contours

Figure 5.14: Solution 10 on the branch depicted in Fig. 5.3.



Velocity Vectors Temperature Contours

Figure 5.15: Solution 11 on the branch depicted in Fig. 5.3.

5.2.2 Two-Cell Solution Branch

We have already briefly described the transition from the one-cell to the two-cell branch, the solutions of which (including hysteresis effects) are depicted in Fig. 5.17. The two-cell solutions are characterized by a "stretching" of the convection cells along the diagonal axis as the aspect ratio is increased. These two-cell solutions are similar to experimentally and numerically-observed two-cell solutions at similar aspect ratios. This two-cell branch loses stability to three-cell solutions around $\Gamma = 6.5$, as discussed physically below.

The numerically-computed eigenvalues for this case are given in Table 5.4. Here and in other tables, we have used a box to denote branch segments which are composed of linearly-stable solution, having only eigenvalues with negative real parts. As the aspect ratio increases, the solutions become stable somewhere between $\Gamma = 5.4$ and $\Gamma = 5.5$, and remain stable until around $\Gamma = 6.4$ where (as mentioned) they lose stability to three-cell patterns. None of the two-cell patterns observed here correspond particularly well with experimentally-observed two-cell patterns, one of which (by Medale and Cerisier [126]) is shown in Fig. 5.18. As we will see in a later section, however, by performing continuation in Ma it is indeed possible to achieve two-cell solutions more similar to the experimental image, suggesting that the experiments may have been conducted at higher Ma.

The surface thermal and velocity fields are shown in more detail in Figs. 5.19–5.26.

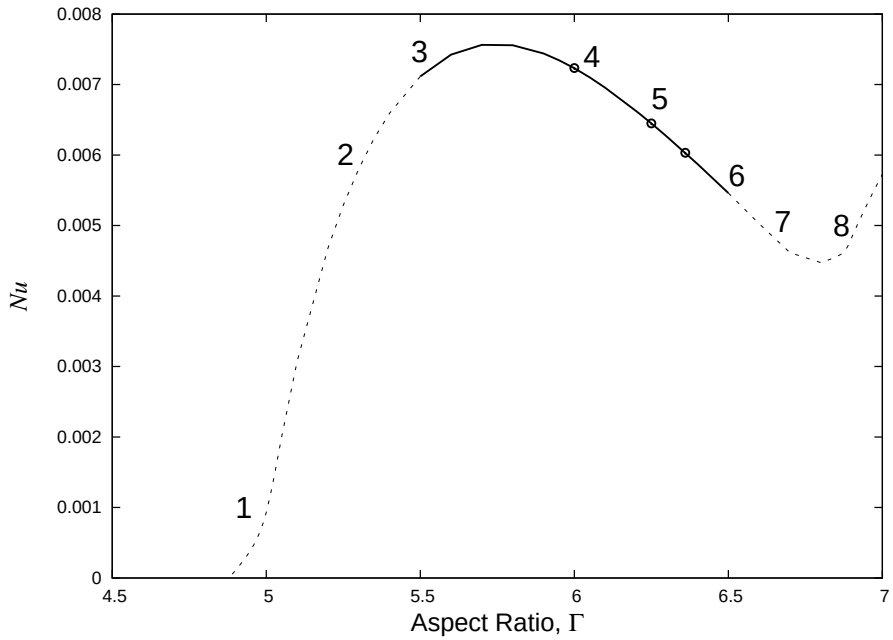


Figure 5.16: Computed solution branch for two-cell solutions. The numbers correspond to solutions discussed in greater detail in the text, and in Fig. 5.17. Also shown, as circular markers, are solutions from "unsteady" branch number 2 described previously and used as a starting point for computing this branch.

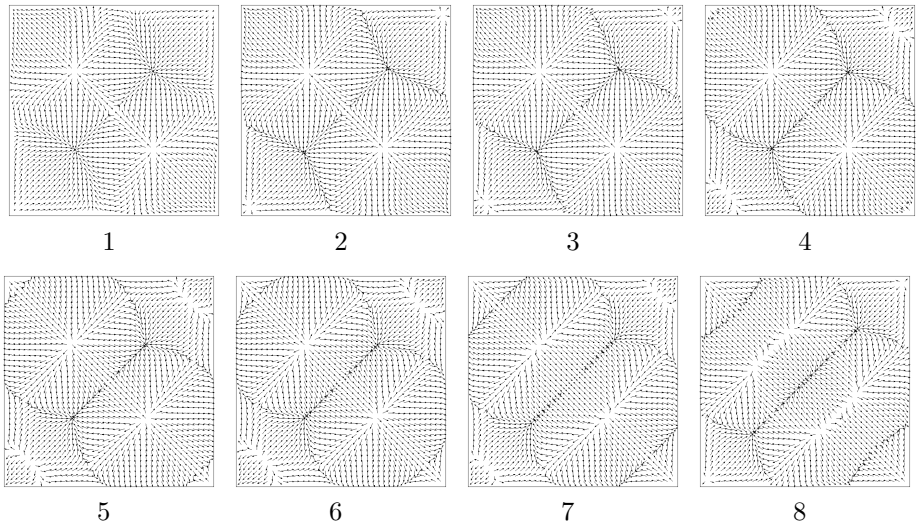


Figure 5.17: Velocity vectors for solutions on the branch depicted in Fig. 5.16.

Table 5.4: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment shown in Fig. 5.16. Complex-conjugate eigenpairs are presented together. The stable solution branch is the solid line in Fig. 5.16.

Γ	σ_i		
6.500	0.639		
6.400	-0.071	-9.701	-12.413
6.000	-2.703		
5.700	-5.393	$-10.067 \pm 6.277i$	
5.600	-6.412	-6.877	-12.987
5.500	-1.811	-7.471	
5.400	1.484	-8.546	-12.138
5.350	2.913	-9.075	-10.719
5.250	5.480	-7.795	-10.070
5.200	-6.283	6.625	-9.970
4.900	$-0.815 \pm 0.387i$	2.514	
4.880	-0.030	-1.840	3.848
4.879	-3.566×10^{-4}	-1.887	4.244

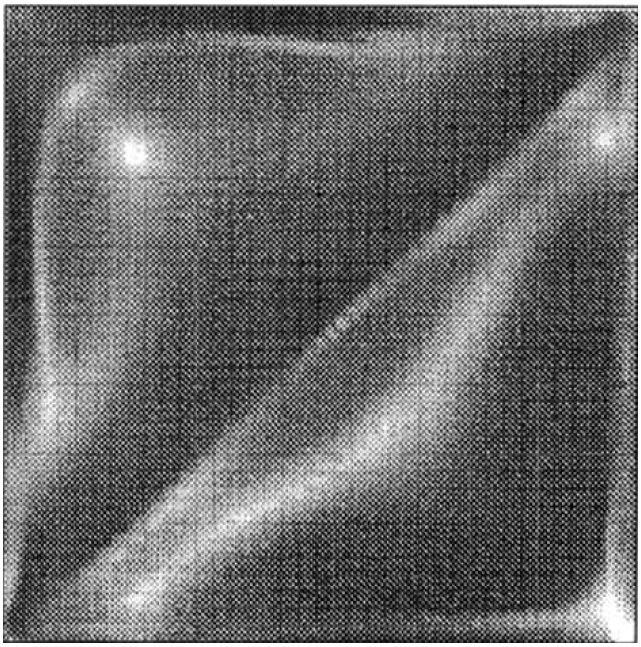
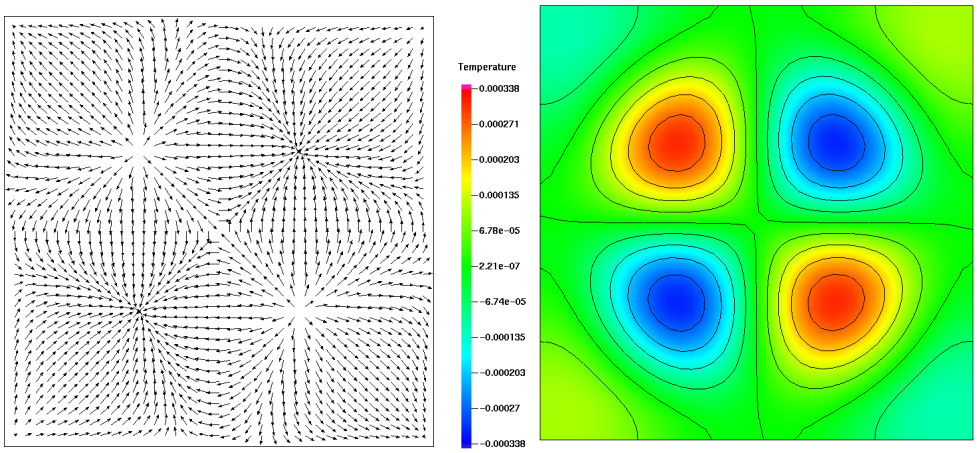
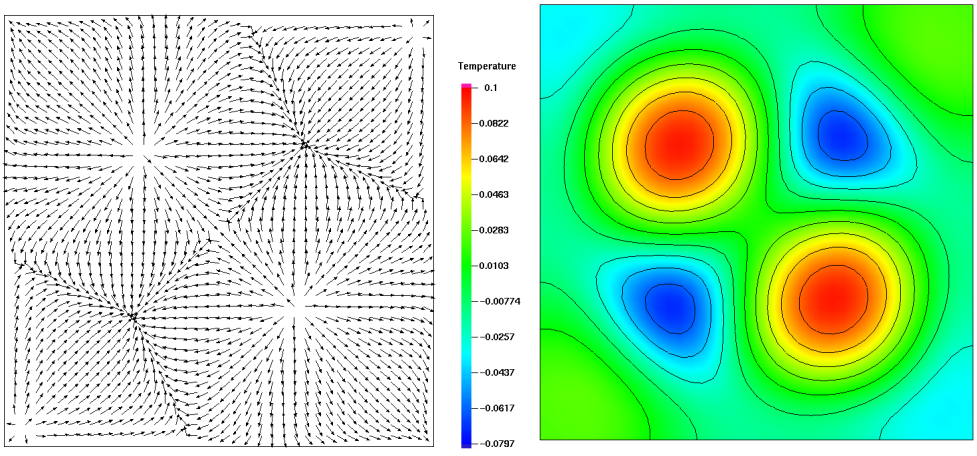


Figure 5.18: Experimentally-observed 2-cell configuration by Medale and Cerisier [126], who report observing it for parameter values $Ma = 64$, $Ra = 33$, and $\Gamma = 5.68$. While no stable two-cell solution precisely matching the experimental image was computed for these parameter values, we did find, by increasing Ma, a solution which appears to match the experimental image more closely (see e.g. solution 9 in Fig. 6.2 of §6.1.1 which occurs for $Ma \approx 140.4$ and $\Gamma = 5.7$).



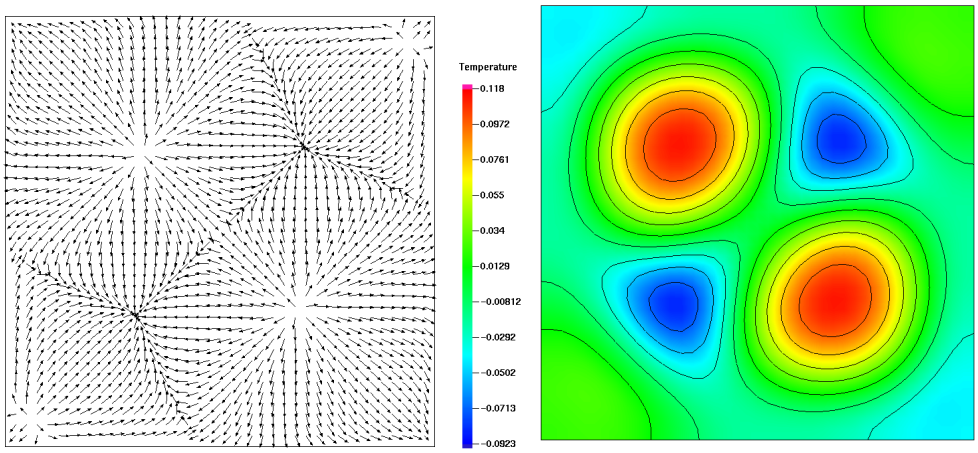
Velocity Vectors Temperature Contours

Figure 5.19: Solution 1 on the branch depicted in Fig. 5.16.



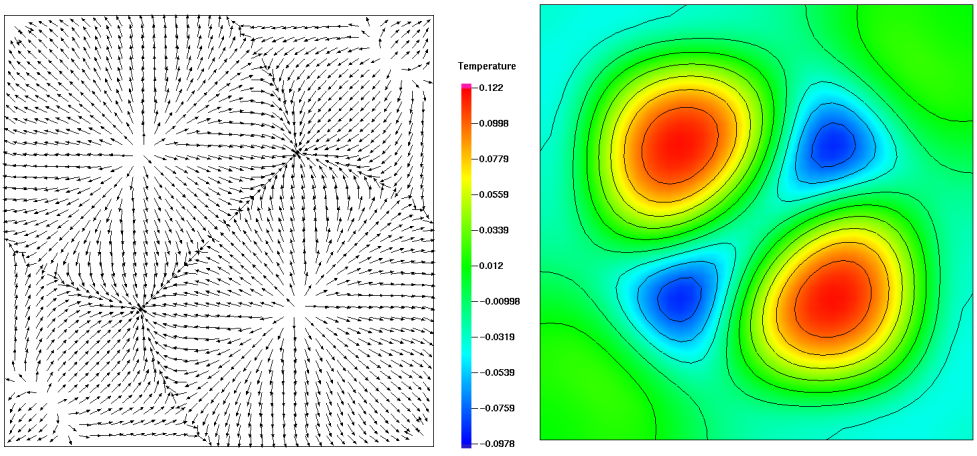
Velocity Vectors Temperature Contours

Figure 5.20: Solution 2 on the branch depicted in Fig. 5.16.



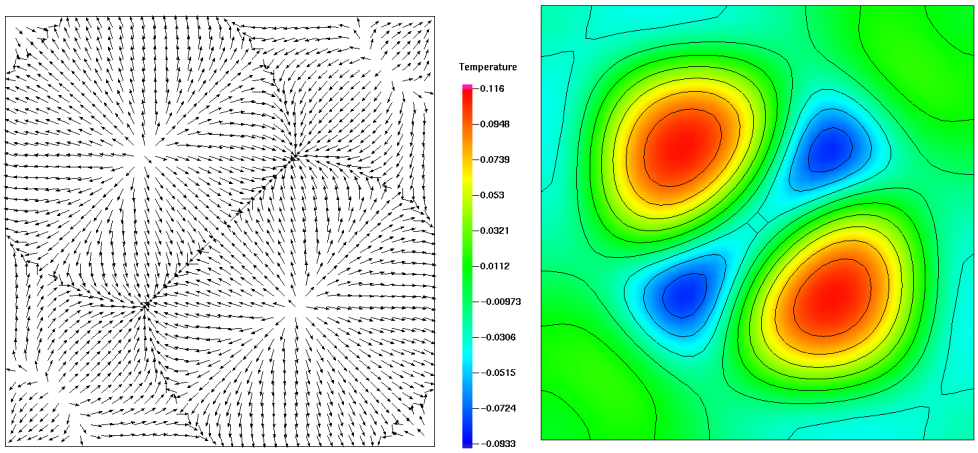
Velocity Vectors Temperature Contours

Figure 5.21: Solution 3 on the branch depicted in Fig. 5.16.



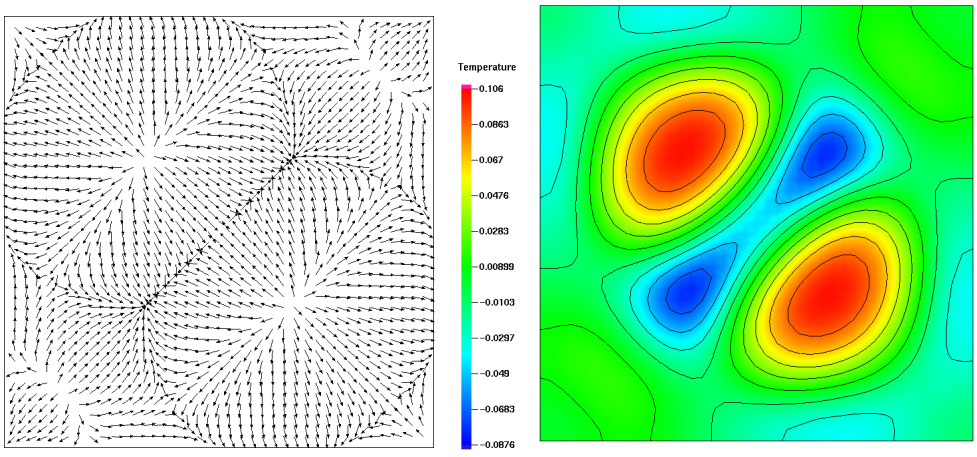
Velocity Vectors Temperature Contours

Figure 5.22: Solution 4 on the branch depicted in Fig. 5.16.



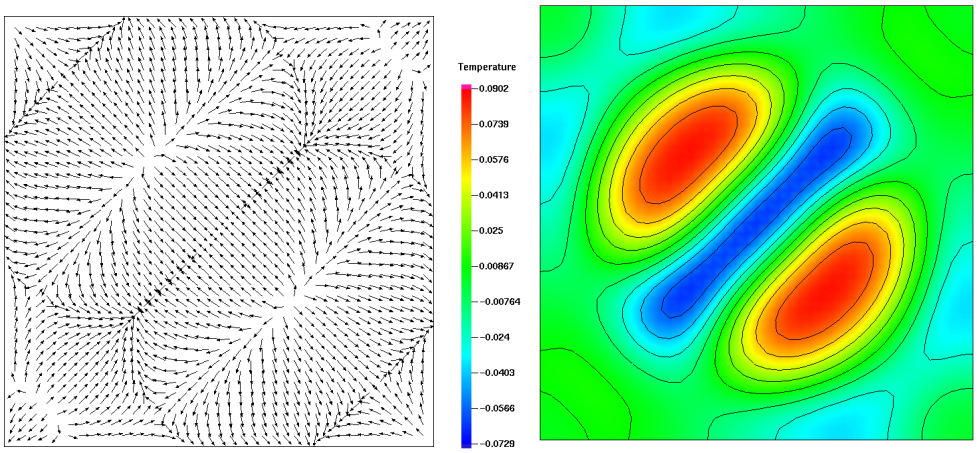
Velocity Vectors Temperature Contours

Figure 5.23: Solution 5 on the branch depicted in Fig. 5.16.



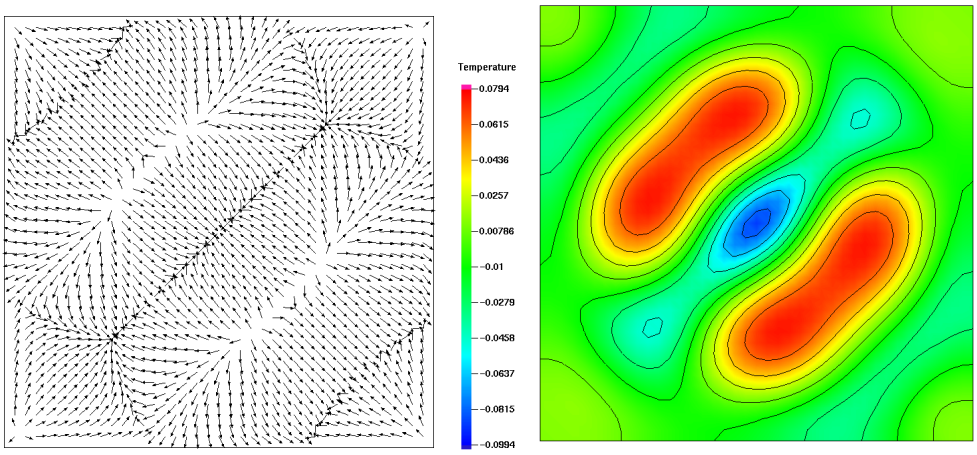
Velocity Vectors Temperature Contours

Figure 5.24: Solution 6 on the branch depicted in Fig. 5.16.



Velocity Vectors Temperature Contours

Figure 5.25: Solution 7 on the branch depicted in Fig. 5.16.



Velocity Vectors Temperature Contours

Figure 5.26: Solution 8 on the branch depicted in Fig. 5.16.

5.2.3 Three-Cell Solution Branch

The branch shown in Fig. 5.27, depicts a transition from stable three-cell configurations (with symmetry about $y = 0$) to an unstable four-cell configuration (with both $x = 0$ and $y = 0$ symmetry). A stable, symmetric three-cell pattern of this type has not been observed experimentally as far as we know. We conjecture that the asymmetric experimental three cell configuration, shown in Fig. 5.29 would likely exist as a symmetry-breaking bifurcation somewhere along this branch.

The numerically-computed eigenvalues for the branch in Fig. 5.27 are given in Table 5.5. We found positive eigenvalues (and hence unstable steady states) along the top branch for $\Gamma \geq 8$ and $\Gamma \leq 6$. The entire lower branch (after the turning point near $\Gamma = 6$) was found to be unstable, with solutions having at least two positive eigenvalues. While for most aspect ratios we find configurations with more convection cells to have higher Nu values, we found the opposite to be the case here. The lower (unstable) branch, although it has more convection cells, has substantially smaller Nu values, though we cannot draw too strong a conclusion since the solutions on the lower branch are unstable.

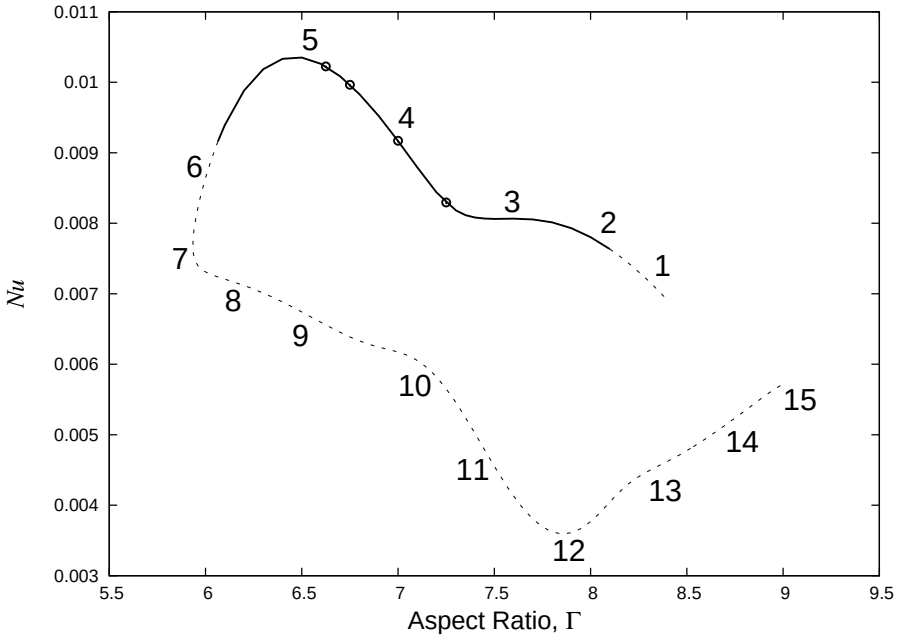


Figure 5.27: This branch depicts the transition from a 3-cell solution with one symmetry to a 4-cell solution with 2 symmetries. Note that these configurations have not been observed experimentally. Also shown, as circular markers, are solutions obtained from the "unsteady" equations described previously and used as a starting point for computing this branch.

Beginning with point 1 on Fig. 5.27, for which the velocity vector field is shown in detail in subfigure 1 of Fig. 5.28, we observe a configuration with two relatively strong cells and one slightly weaker. The thermal flux of this configuration grows as we proceed along the branch to solutions 2 and 3, and at solution 3, the strength and size of the three cells is nearly identical as seen in Fig. 5.28 subfigure 4. The alignment of the cell boundaries at the center of the container are roughly separated at 60° from one another.

Beyond point 3 on the continuation path, the thermal flux increases rapidly until it reaches a maximum around point 5. The paired cells on the left side of the container have both grown to fill the left-hand side of the domain by this point, and the unpaired cell on the right-hand boundary has expanded as well. As we move to solution 6 and finally to the turning point at solution 7, the third (unpaired) cell is observed to shrink in size relative to the other two, as they grow to cover more of the container.

Recall that turning points are points where the system Jacobian G_u is singular, i.e. has a zero eigenvalue. A new eigenmode (the mode associated with the singular eigenvalue) changes stability as we pass through the turning point. The physical significance of this event is that small disturbances to the system which were once damped out in time are now amplified and drive the solution away from the steady state of interest. This change typically occurs through rapid, transient processes, e.g. the buckling of a beam or the combustion of reacting species.

The trend of decreasing heat transfer as well as the growth of the paired

cells continues through solutions 8, 9, and 10, and at solution 11 we first observe the beginnings of a fourth and fifth cell near the left-hand boundary of the domain. These new cells appear to pinch the (now) centralized top and bottom cell pair, and are most clearly visible by solution 13, by which point the heat transfer rate has once again begun to increase. Although there appear to be two new cells at solution 13, they have already begun to coalesce by solution 14, and by solution 15, a four-cell pattern with both x and y symmetry is at last observed.

The solution branch "continues" from this point in the sense that the sequence of solutions 14, 13, 12, etc. are repeated in reverse as now the left-hand cell grows in strength and the right-hand cell is diminished. These symmetric solutions have identical heat transfer rates (the quantity of interest is rotation-invariant) and thus do not appear as a separate branch, but rather an overlapping branch. By selecting a different quantity of interest, for example, the velocity at an arbitrary point in the upper right quadrant of the surface, one could differentiate between these sorts of solutions in the continuation plots.

Table 5.5: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for selected aspect ratios along the solution branch shown in Fig. 5.27. Complex-conjugate eigenpairs are presented together. The boxed region delimits the stable and unstable regions of the solution branch (marked as solid and dashed lines, respectively, in Fig. 5.27.)

Γ	σ_i		
8.388	0.091	-0.989	7.914
8.240	0.778	$-0.761 \pm 6.843i$	
8.170	0.733	$-1.262 \pm 8.450i$	
8.135	0.609	$-1.587 \pm 9.046i$	
8.1	0.447	$-1.955 \pm 9.540i$	
8.0	-0.160	$-3.234 \pm 10.495i$	
7.7	-2.218	$-7.967 \pm 9.093i$	
7.6	-2.639	$-9.585 \pm 6.775i$	
7.5	-2.436	-9.692	$-11.715 \pm 2.242i$
7.4	-1.604	-7.692	-9.008
7.3	-1.657	-6.421	-11.879
7.2	-3.760	-7.026	$-11.393 \pm 7.063i$
6.6	$-6.987 \pm 10.295i$	$-11.422 \pm 9.067i$	
6.5	$-6.451 \pm 9.285i$	$-10.380 \pm 9.116i$	
6.2	$-4.073 \pm 3.866i$	$-6.599 \pm 7.787i$	
6.1	-0.469	-5.811	$-5.233 \pm 6.787i$
6.064	1.018	-6.618	$-4.725 \pm 6.302i$
6.0	3.008	$-3.767 \pm 5.110i$	
6.127	1.326	-5.301	7.925
7.293	-0.329	5.124	-8.879
8.746	$0.508 \pm 2.872i$	-5.243	
9.006	-0.371	4.591	6.037

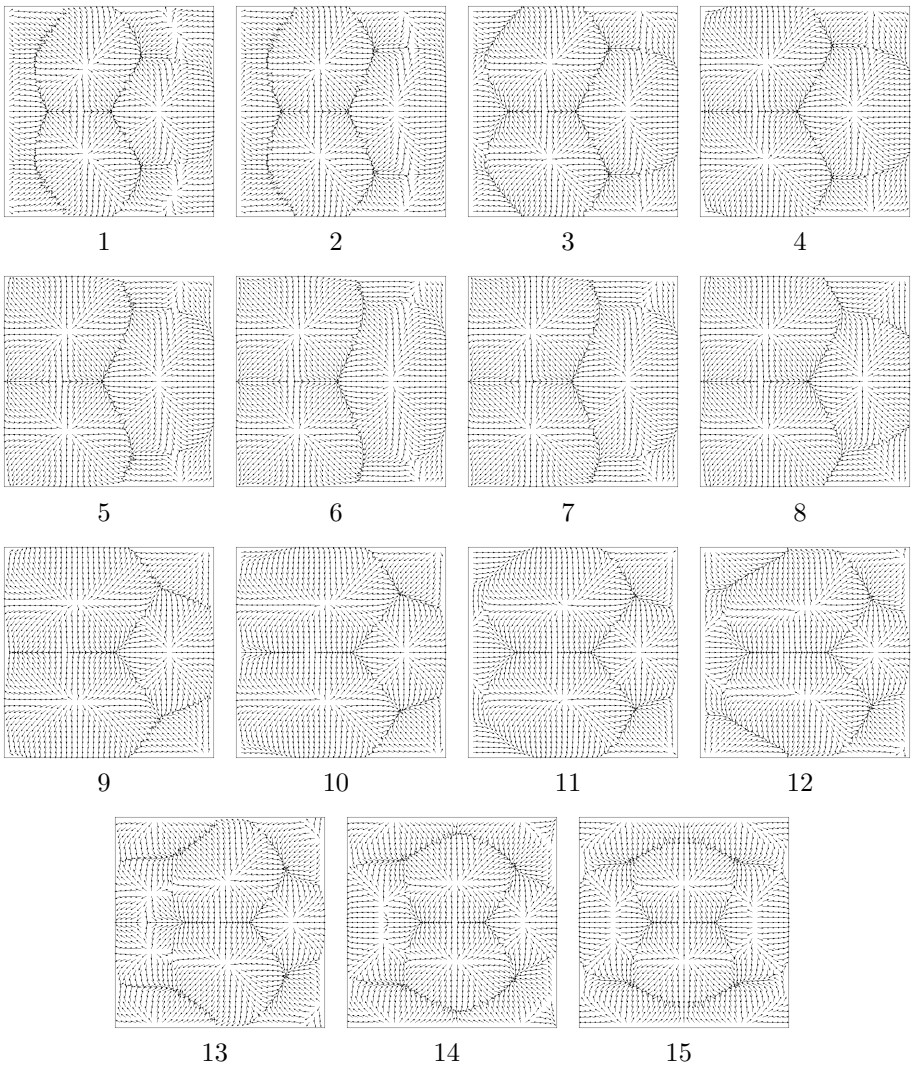


Figure 5.28: Velocity vectors for solutions on the branch depicted in Fig. 5.27.

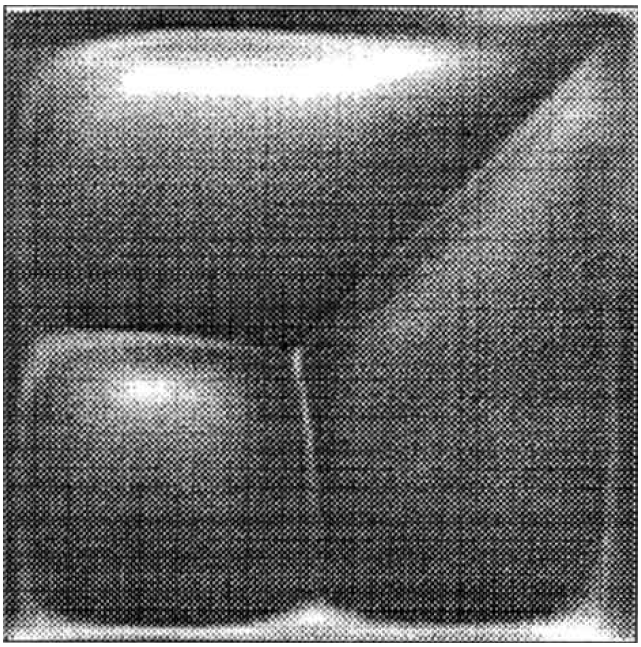
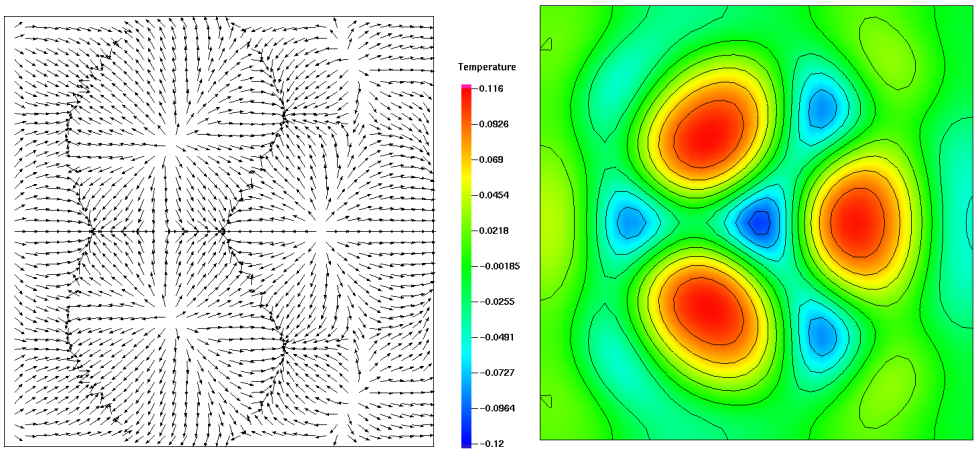
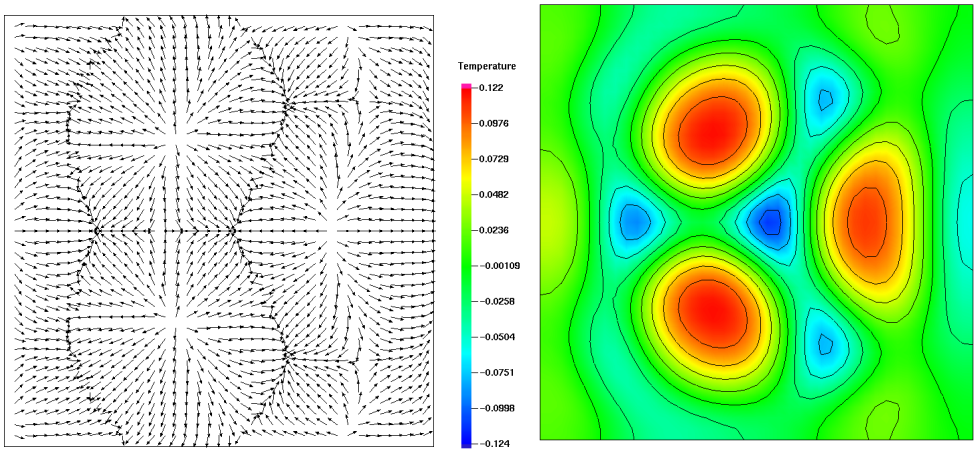


Figure 5.29: Experimentally-observed 3-cell configuration by Medale and Cerisier [126] at aspect ratio $\Gamma = 6.18$. This configuration was not observed computationally by us, but would likely exist as a symmetry-breaking bifurcation from the branch in Fig. 5.27.



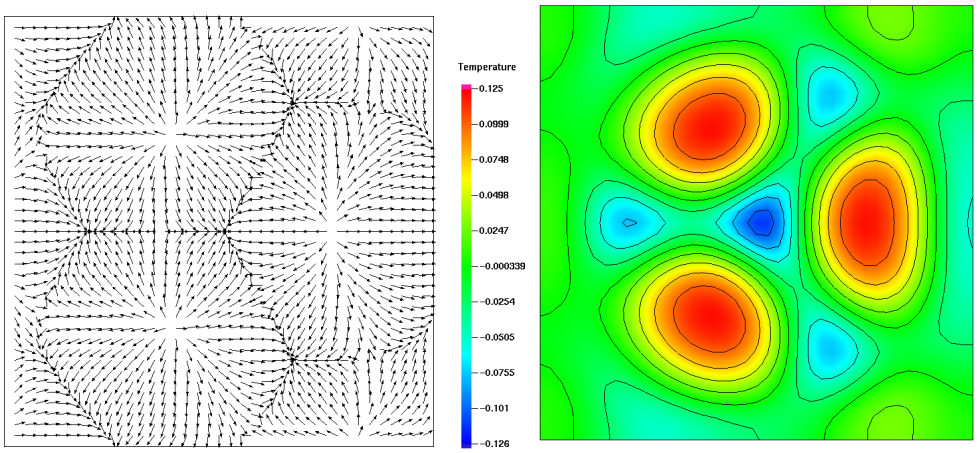
Velocity Vectors Temperature Contours

Figure 5.30: Solution 1 on the branch depicted in Fig. 5.27.



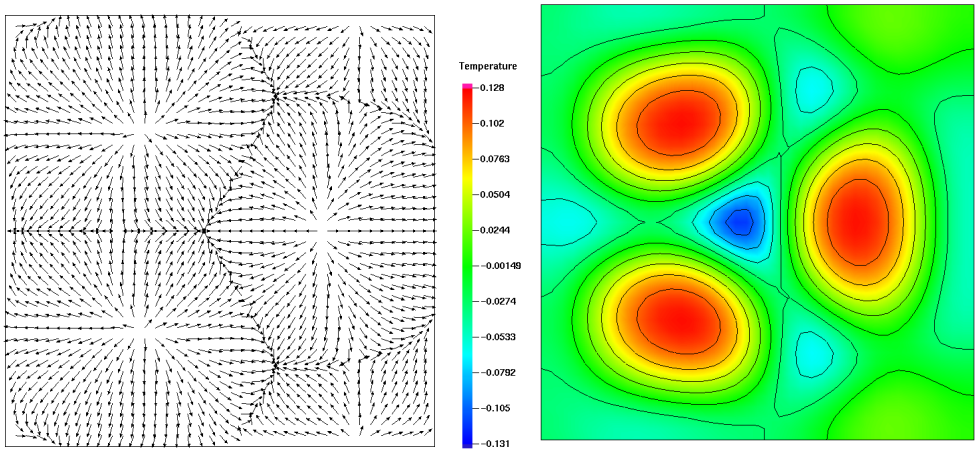
Velocity Vectors Temperature Contours

Figure 5.31: Solution 2 on the branch depicted in Fig. 5.27.



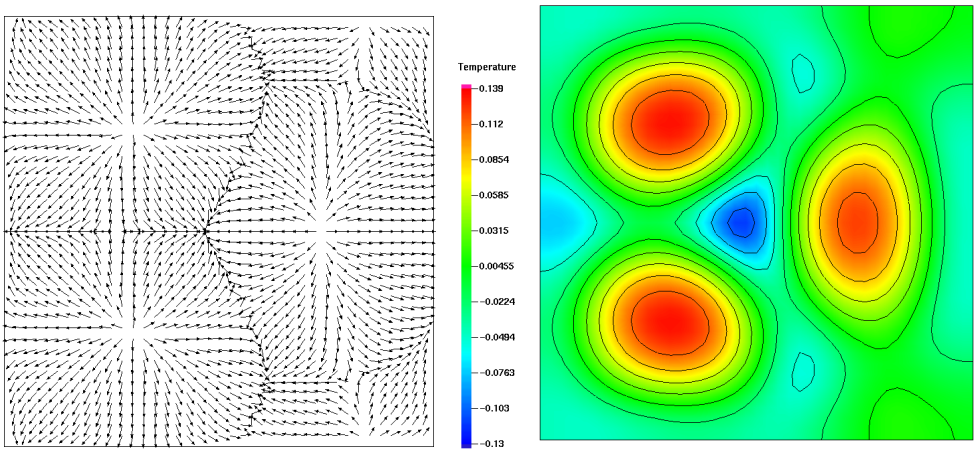
Velocity Vectors Temperature Contours

Figure 5.32: Solution 3 on the branch depicted in Fig. 5.27.



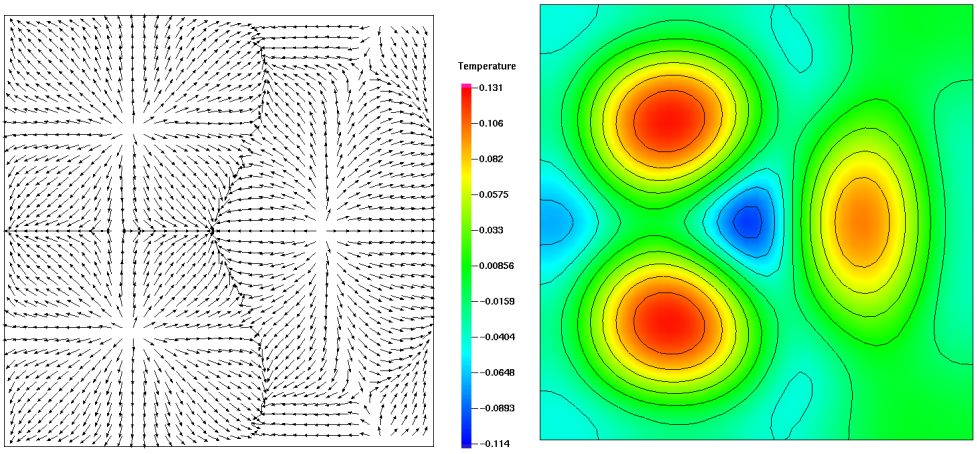
Velocity Vectors Temperature Contours

Figure 5.33: Solution 4 on the branch depicted in Fig. 5.27.



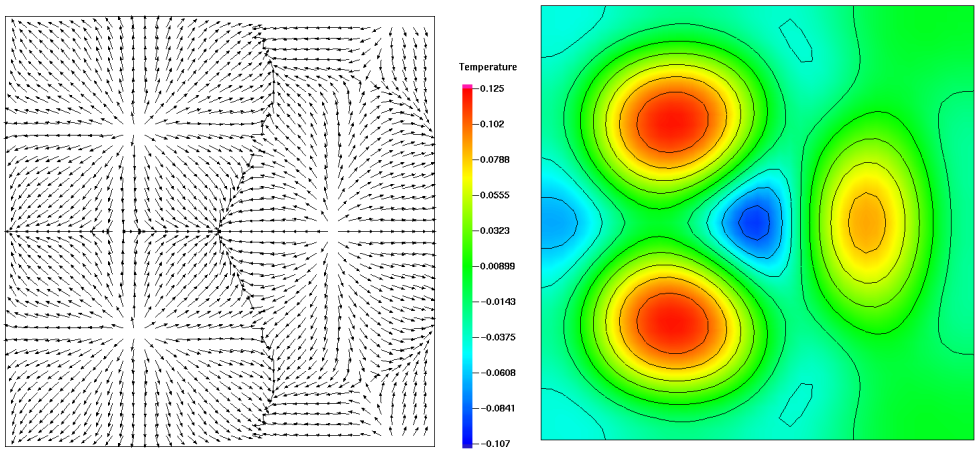
Velocity Vectors Temperature Contours

Figure 5.34: Solution 5 on the branch depicted in Fig. 5.27.



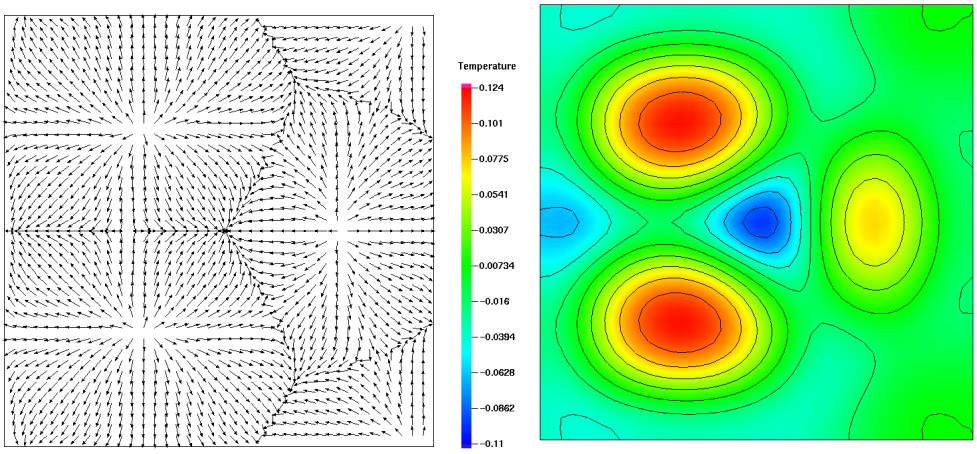
Velocity Vectors Temperature Contours

Figure 5.35: Solution 6 on the branch depicted in Fig. 5.27.



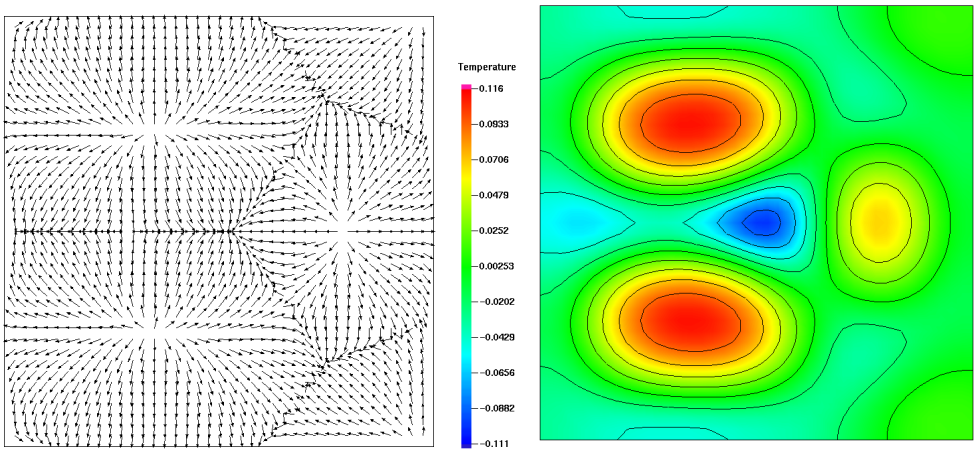
Velocity Vectors Temperature Contours

Figure 5.36: Solution 7 on the branch depicted in Fig. 5.27.



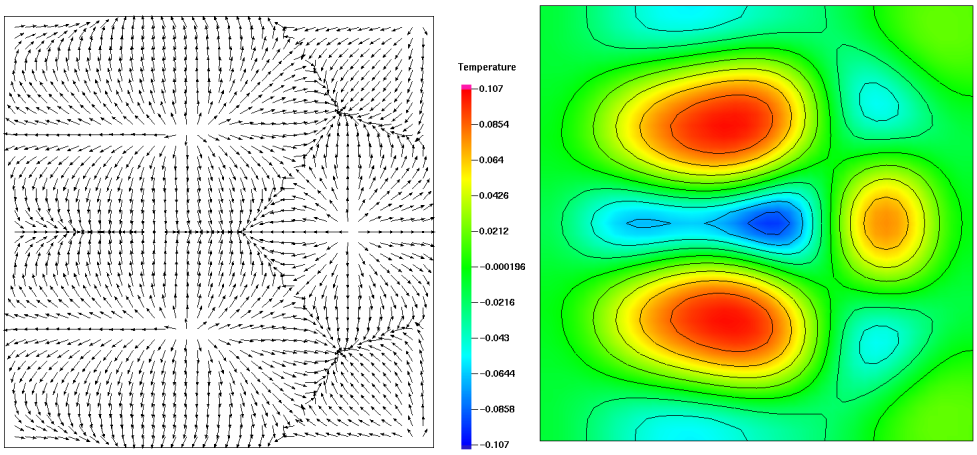
Velocity Vectors Temperature Contours

Figure 5.37: Solution 8 on the branch depicted in Fig. 5.27.



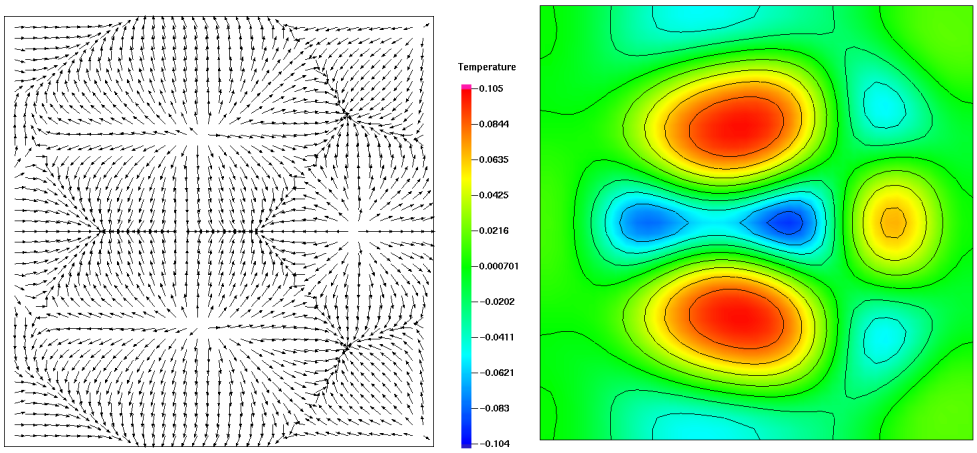
Velocity Vectors Temperature Contours

Figure 5.38: Solution 9 on the branch depicted in Fig. 5.27.



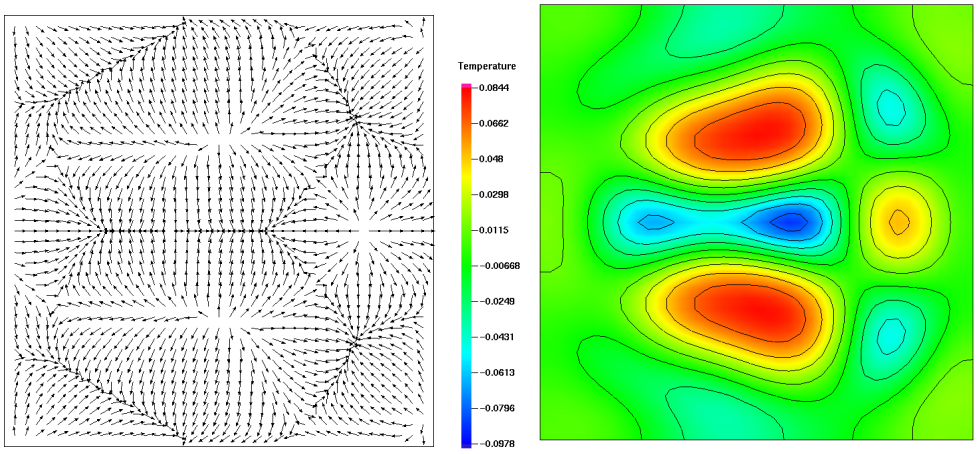
Velocity Vectors Temperature Contours

Figure 5.39: Solution 10 on the branch depicted in Fig. 5.27.



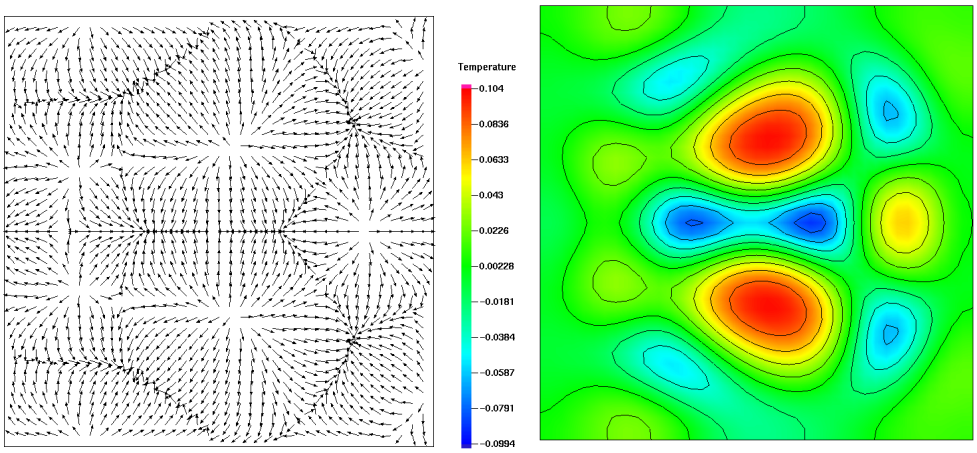
Velocity Vectors Temperature Contours

Figure 5.40: Solution 11 on the branch depicted in Fig. 5.27.



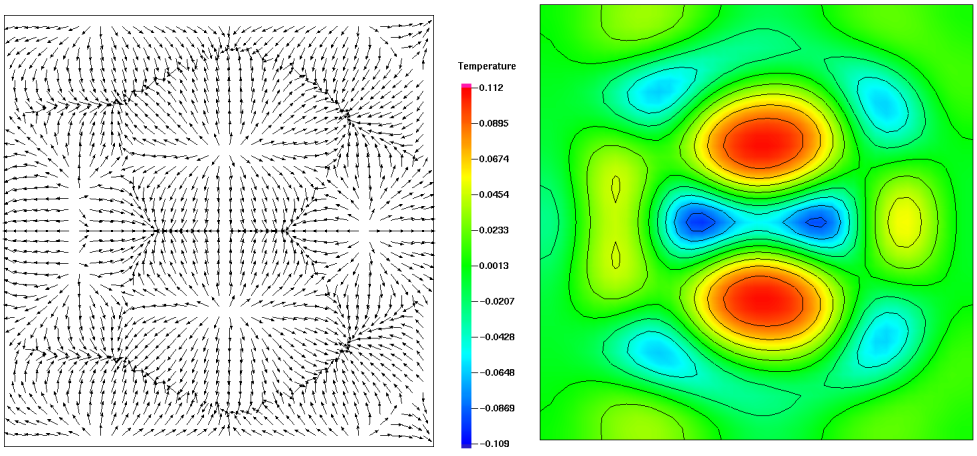
Velocity Vectors Temperature Contours

Figure 5.41: Solution 12 on the branch depicted in Fig. 5.27.



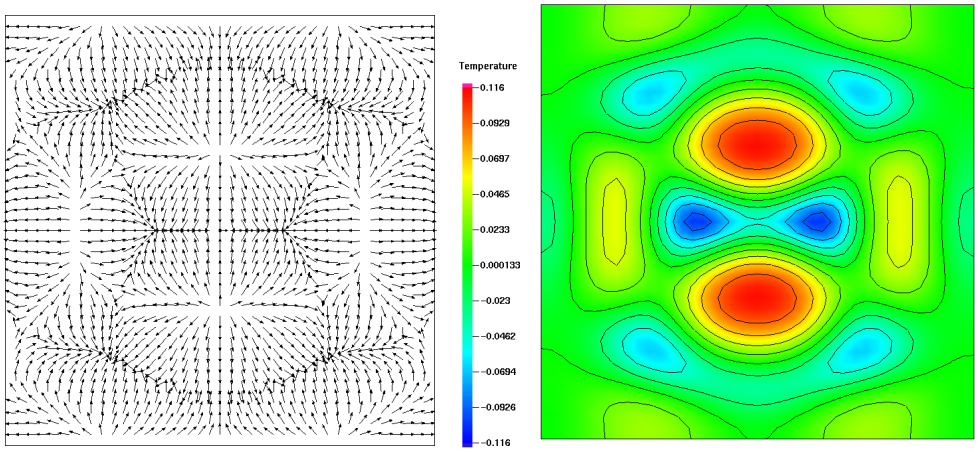
Velocity Vectors Temperature Contours

Figure 5.42: Solution 13 on the branch depicted in Fig. 5.27.



Velocity Vectors Temperature Contours

Figure 5.43: Solution 14 on the branch depicted in Fig. 5.27.



Velocity Vectors Temperature Contours

Figure 5.44: Solution 15 on the branch depicted in Fig. 5.27.

5.3 Four and Five-Cell Solution Branches

In Figure 5.45, the three, four, and five convection cell branches are shown. (The three-cell branch is repeated from the previous section for reference.) In §5.3.1 we discuss in more detail the stable four-cell solution branch, as well as some interesting nearby unstable solutions. In §5.3.2, we discuss in more detail a double symmetry-breaking transition which occurs along the four-cell branch between aspect ratios $\Gamma = 7.5$ and $\Gamma = 8.4$. Finally, in §5.3.3, we discuss the computed five-cell solutions observed, which correspond to a stable, symmetric branch which bifurcates into a stable asymmetric branch between $\Gamma = 8.7$ and $\Gamma = 8.75$.

We once again observe hysteresis effects in Fig. 5.45, with multiple stable steady states (e.g. both three and four-cell or both four and five-cell states) existing for certain aspect ratios. It is difficult to predict why the unsteady solver would "select" a particular pattern over another for a given aspect ratio, but we conjecture that it must depend on the initial conditions used in the unsteady solve. (Near $\Gamma = 8.75$ in Fig. 5.45 for example, had we used initial data slightly perturbed from the four-cell branch we would expect the unsteady solver to converge to the four-cell solution, whereas if our initial data was slightly perturbed from the five-cell solution we would expect the unsteady solver to converge to it instead.) None of the stable states observed here correspond particularly well with experimental results, but as we point out in §5.3.1, one of the unstable steady states bears an interesting resemblance to one of Koschmieder's experimental results, a relationship which is further

examined later.

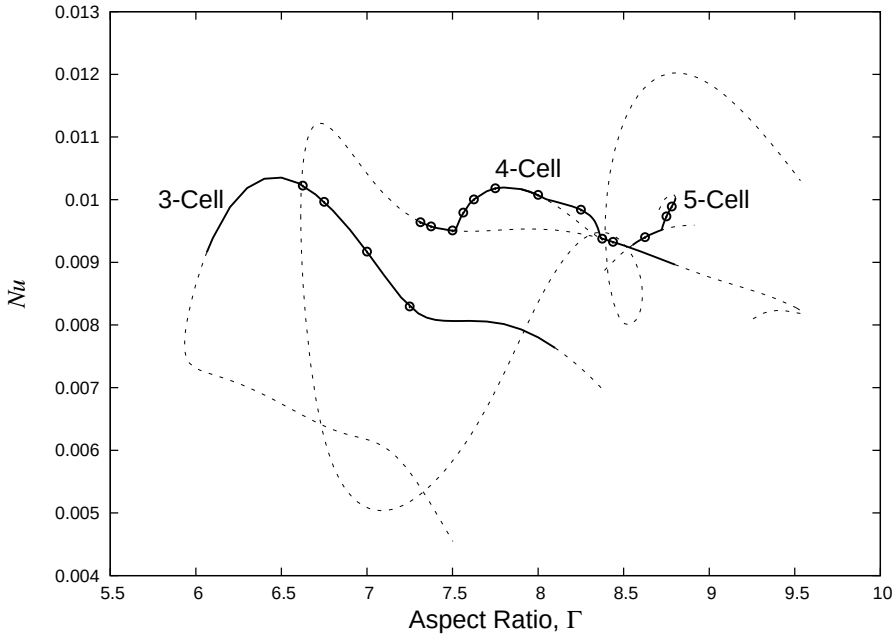


Figure 5.45: Solution branches for 3, 4, and 5-cell solutions.

5.3.1 Four-Cell and Surrounding Solutions

On this branch, the numerical solution transitions from a four-cell solution to a unstable six-cell solution which slightly resembles an experimentally-observed six-cell solution.

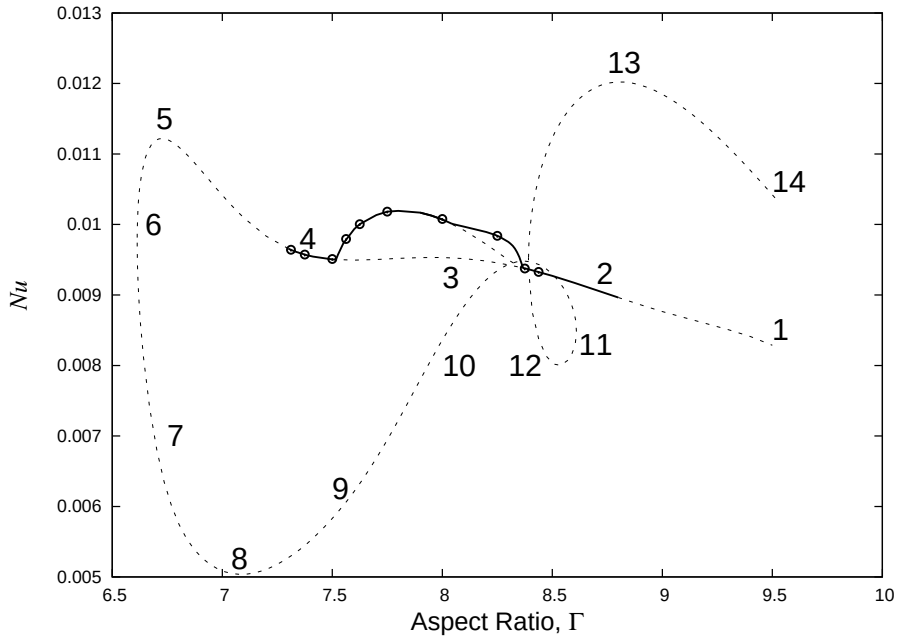


Figure 5.46: Computed solution branch for symmetry-breaking bifurcations. The numbers correspond to solutions discussed in greater detail in the text, and in Fig. 5.47. Also shown, as circular markers, are solutions from "unsteady" branch number 4 described previously and used as a starting point for computing this branch.

Solutions 1–5 on the branch shown in Fig. 5.46 are the same solutions

with $y = \pm x$ symmetry described in §5.3.2 and depicted in Fig. 5.66. Numerically computed eigenvalues for the part of the branch comprising solutions 1–5 are given in Table 5.6. The two regions of stable solutions found for $8.4 \leq \Gamma \leq 8.8$ and $7.3 \leq \Gamma \leq 7.5$ are shown inside boxes in the table.

This same $y = \pm x$ symmetry will be maintained throughout the branch discussed in this section, see e.g. the surface velocity field plots in Fig. 5.47. (We discuss the nearby symmetry-breaking branches in later sections.) As we move from solution 5→6, the first turning point on the branch is encountered. Around this point, the upper-left and lower-right convection cells start to become more rounded in shape, and also begin increasing in size relative to the lower-left and upper-right cells as we go from 6→7. Going from 7→8 we approach a local minimum heat transfer rate for the surface on the branch. Also at this point, we observe the top left and lower right convection cells beginning to stretch along the symmetry direction.

This stretching continues until solution 9, at which point a "pinching" effect begins to take over for these two cells. (It has been observed several times in these simulations that this pinching is a common symmetry-preserving cell transition mechanism.) Indeed, moving from solution 9→10, this pinching has proceeded to the point where the pinched cells have now split into 2, generating a total of six convection cells. A configuration remarkably similar to this six-cell configuration has in fact been observed by Koschmieder in his experiments with containers having small aspect ratio.

Although it is somewhat difficult to make out in this reproduced im-

age, we have included Koschmieder's 6-cell experimental result here as well in Fig. 5.48. While Koschmieder's result certainly confirms that there is a physically-realizable six-cell solution in the vicinity of these parameter values, we remain skeptical as to whether solution 11 presented in Fig. 5.47 can truly be the same solution, since our calculations showed it to be linearly unstable. It is possible that the mathematical model has neglected an important physical effect (e.g. surface deformation) which would tend to stabilize the solution. On the other hand, it might also be possible to "stabilize" solution 11 by using a finer grid or by varying other problem parameters such as the Ma number.

After passing this turning point, a new characteristic has arisen in the solution: a central, seventh convection cell has emerged by solution 12. The apparent effect of the existence of the new cell is to dramatically increase the heat transfer rate, to levels above those on the original symmetric branch. The central cell continues to grow in strength until solution 13, at which point it has achieved a distinctly rhomboid-shaped appearance. Solution 13 corresponds approximately with a local maximum in the heat transfer rate. Beyond solution 13, the central cell appears to be shrinking again, possibly due to the increasing size of the top right and bottom left convection cells.

More detailed images of solutions 1–13 including the surface temperature contours are given in Figs. 5.49– 5.62 below.

Table 5.6: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for selected aspect ratios along the main (symmetric) solution branch (solutions 1–5) shown in Fig. 5.46. Complex-conjugate eigenpairs are presented together. The boxed-in rows of the table denote segments of the branch composed of stable solutions (marked as solid lines in Fig. 5.46.) The symmetric branch appears to become unstable somewhere between $\Gamma = 8.8$ and $\Gamma = 8.9$.

Γ	σ_i		
9.2	-4.478	-6.520	6.912
9.1	5.438	-5.685	-6.306
9.0	3.623	-5.125	-9.938
8.9	1.357	-4.537	-9.378
8.8	-1.770	-3.830	-7.312
8.7	-3.004	$-5.178 \pm 3.155i$	
8.6	-2.099	$-5.930 \pm 4.775i$	
8.5	-1.164	$-6.803 \pm 5.459i$	
8.4	-0.246	$-7.809 \pm 5.489i$	
8.3	0.610	$-8.955 \pm 4.801i$	
8.0	2.414	-6.257	$-13.738 \pm 7.727i$
7.7	2.085	-4.561	-6.148
7.6	1.177	-3.976	-4.488
7.5	-0.297	-2.302	-4.680
7.3125	-0.226	-4.934	-5.895
7.3	-0.134	-5.345	-6.020
7.2	0.395	-7.227	-9.346
7.1	0.625	-8.008	-8.786
6.618	-1.263	8.547	$-17.593 \pm 5.300i$

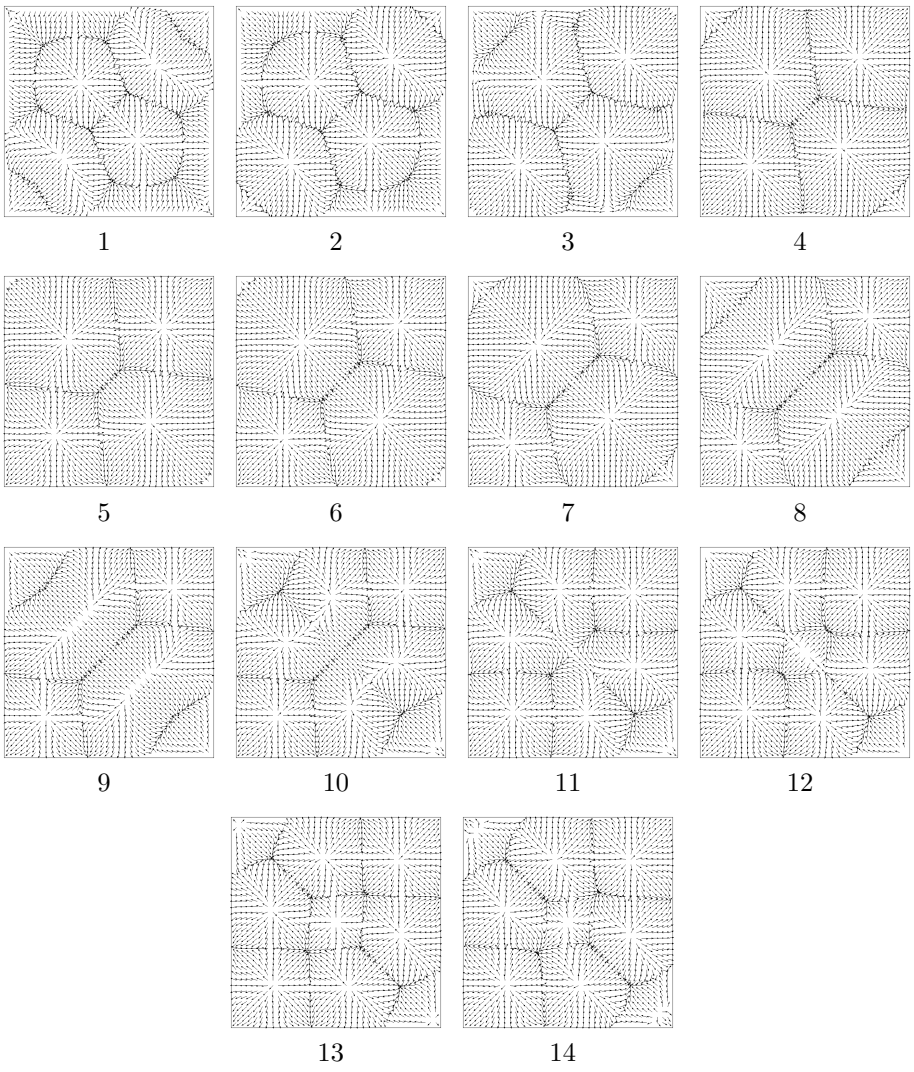


Figure 5.47: Velocity vectors for solutions on the branch depicted in Fig. 5.46.

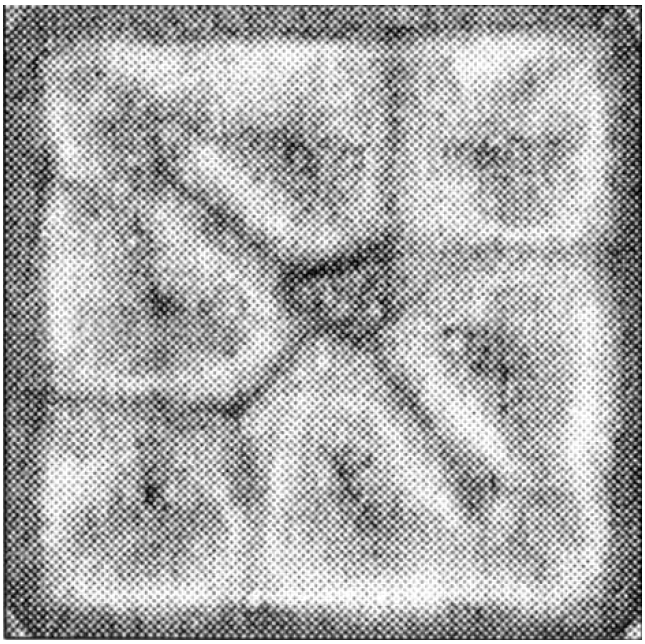
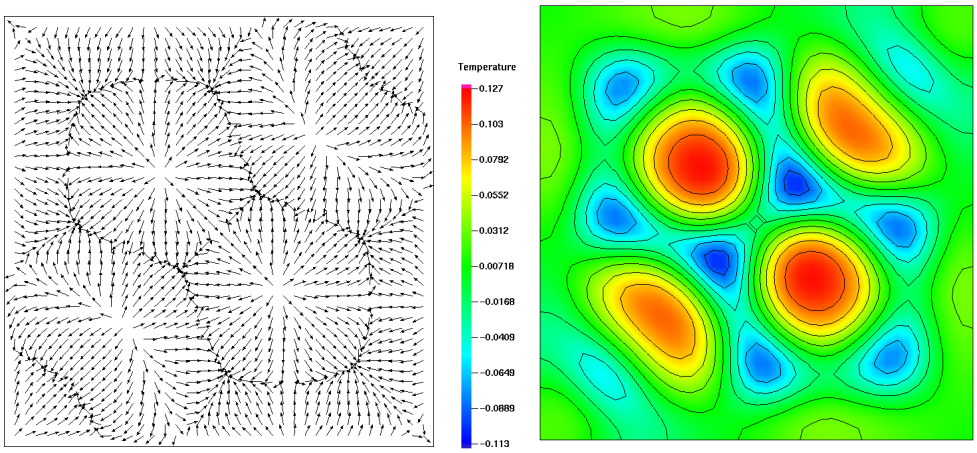
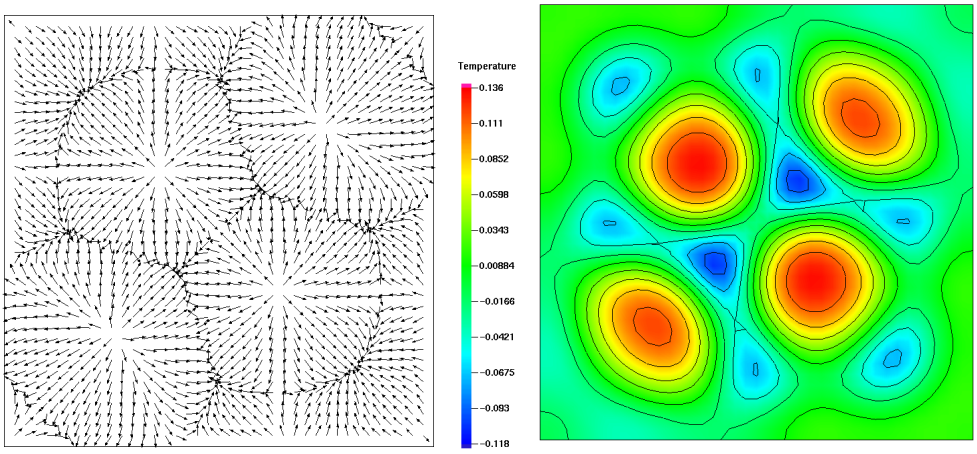


Figure 5.48: Experimentally-observed 6-cell configuration by Koschmieder at $Ma = 86$, $Ra = 22$, and $\Gamma = 8.08$. This corresponds approximately to solution 11 on the branch depicted in Fig. 5.46.



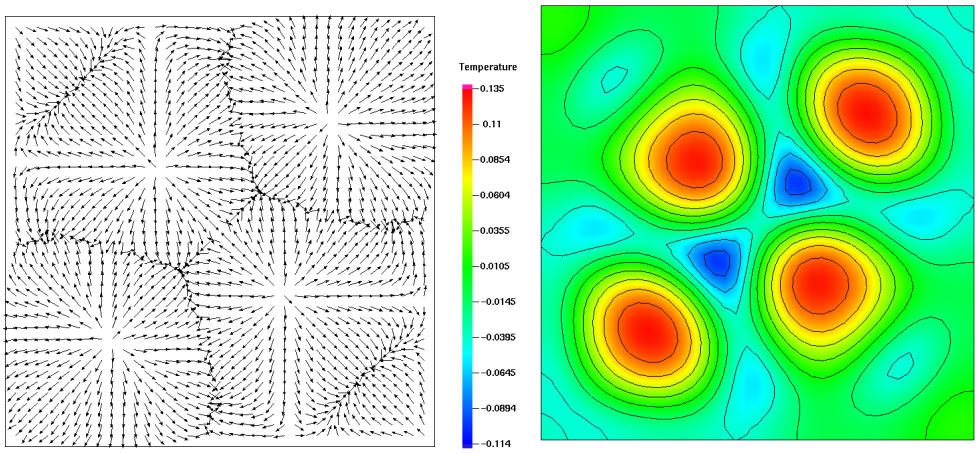
Velocity Vectors Temperature Contours

Figure 5.49: Solution 1 on the branch depicted in Fig. 5.46.



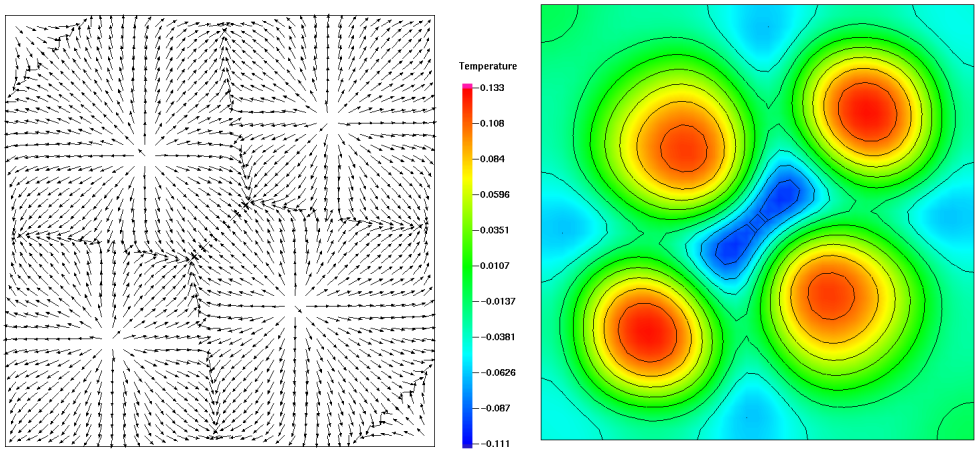
Velocity Vectors Temperature Contours

Figure 5.50: Solution 2 on the branch depicted in Fig. 5.46.



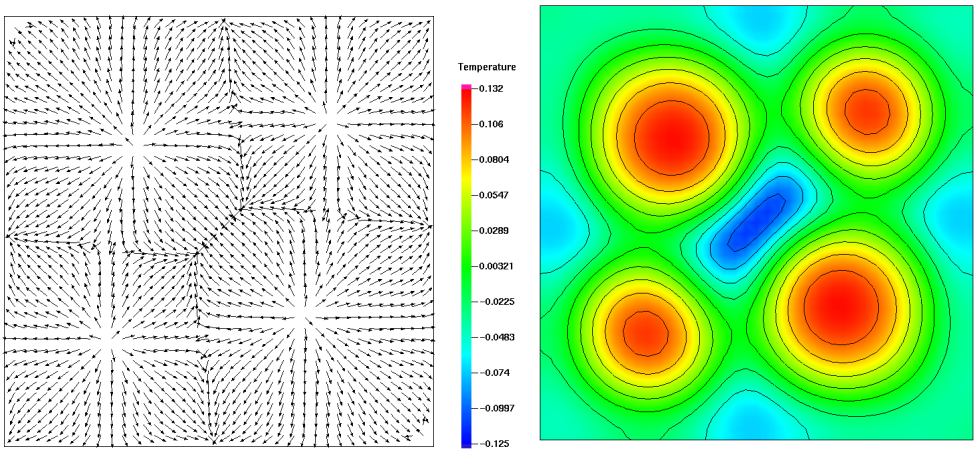
Velocity Vectors Temperature Contours

Figure 5.51: Solution 3 on the branch depicted in Fig. 5.46.



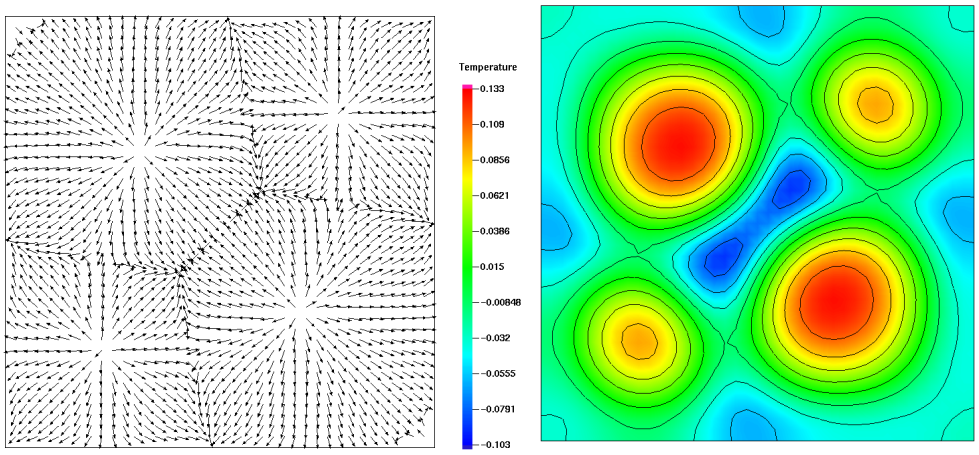
Velocity Vectors Temperature Contours

Figure 5.52: Solution 4 on the branch depicted in Fig. 5.46.



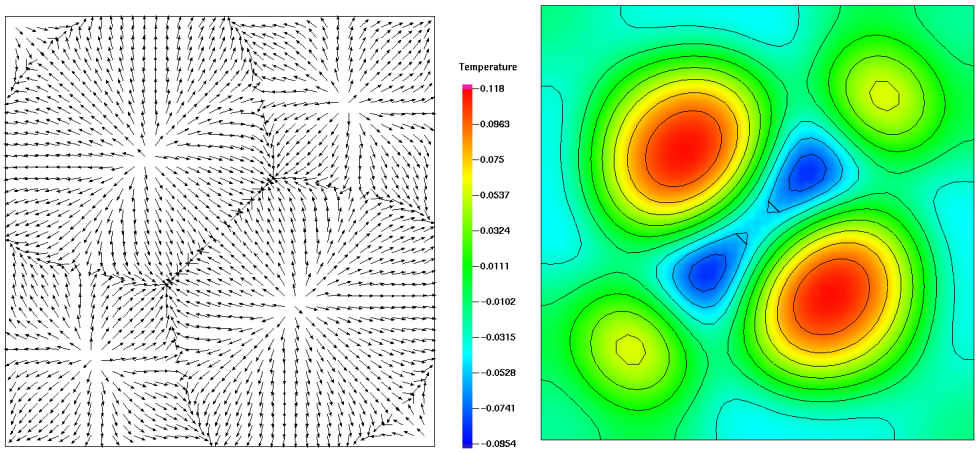
Velocity Vectors Temperature Contours

Figure 5.53: Solution 5 on the branch depicted in Fig. 5.46.



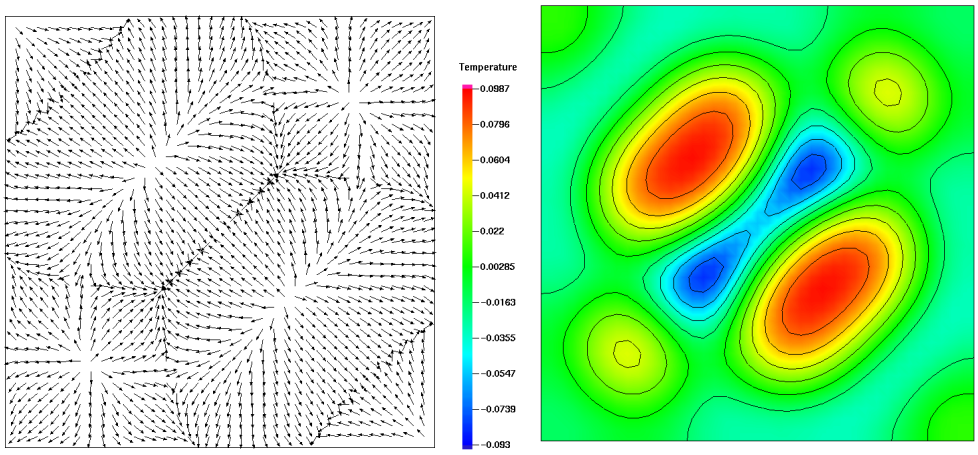
Velocity Vectors Temperature Contours

Figure 5.54: Solution 6 on the branch depicted in Fig. 5.46.



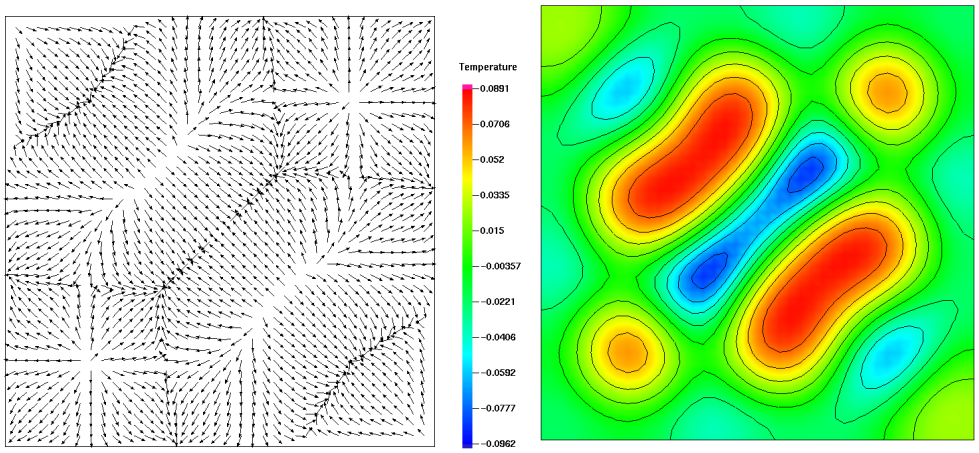
Velocity Vectors Temperature Contours

Figure 5.55: Solution 7 on the branch depicted in Fig. 5.46.



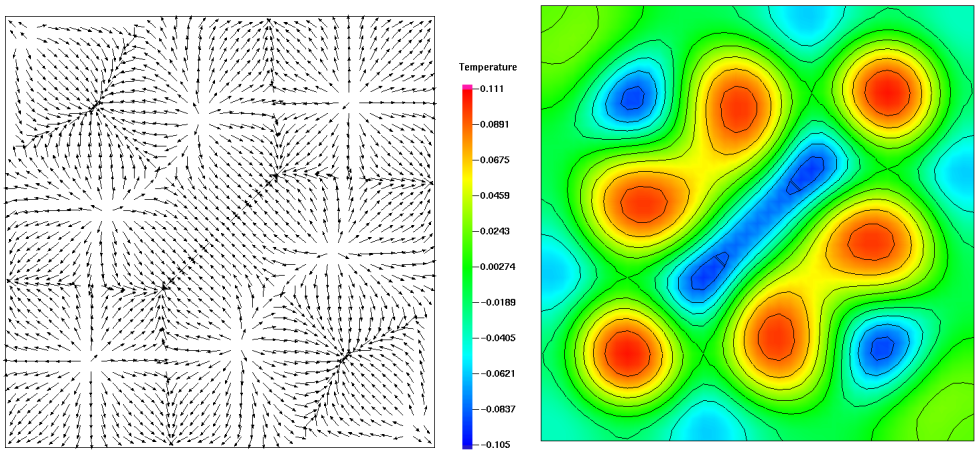
Velocity Vectors Temperature Contours

Figure 5.56: Solution 8 on the branch depicted in Fig. 5.46.



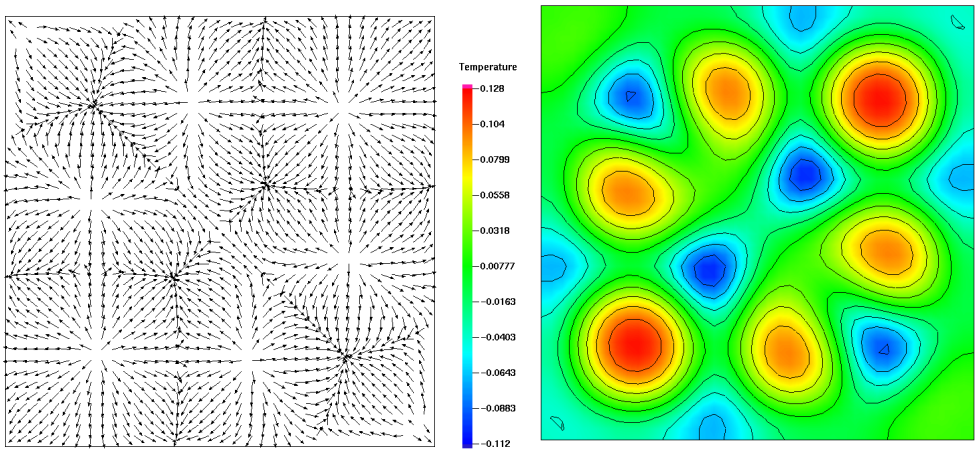
Velocity Vectors Temperature Contours

Figure 5.57: Solution 9 on the branch depicted in Fig. 5.46.



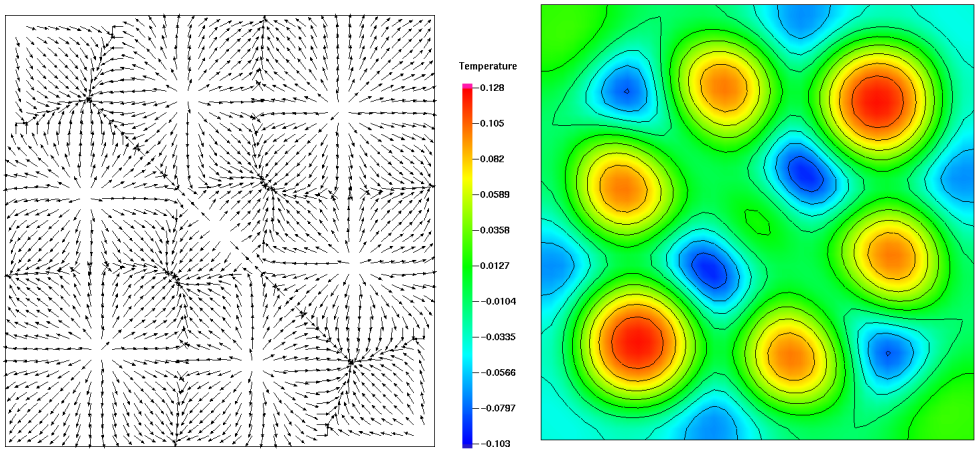
Velocity Vectors Temperature Contours

Figure 5.58: Solution 10 on the branch depicted in Fig. 5.46.



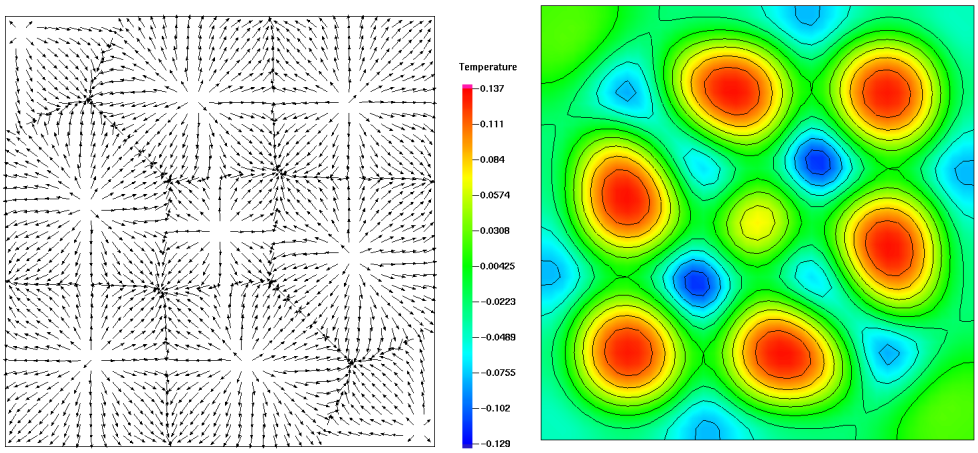
Velocity Vectors Temperature Contours

Figure 5.59: Solution 11 on the branch depicted in Fig. 5.46.



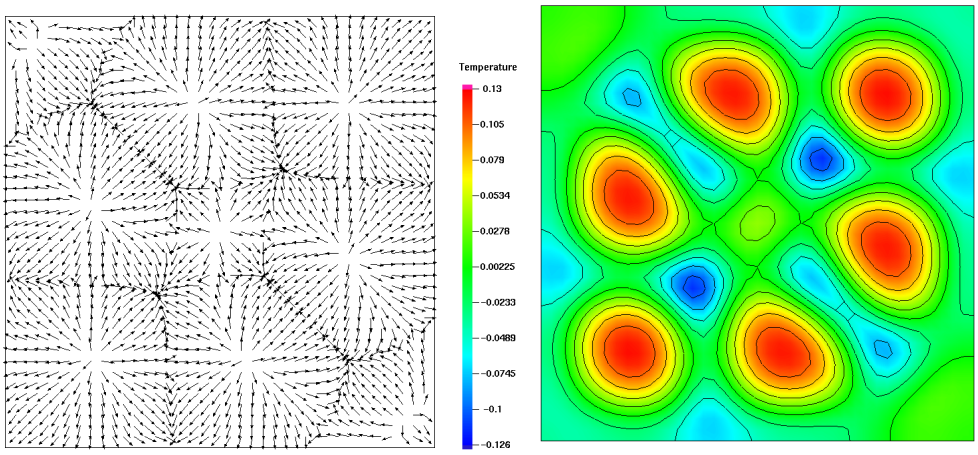
Velocity Vectors Temperature Contours

Figure 5.60: Solution 12 on the branch depicted in Fig. 5.46.



Velocity Vectors Temperature Contours

Figure 5.61: Solution 13 on the branch depicted in Fig. 5.46.



Velocity Vectors Temperature Contours

Figure 5.62: Solution 14 on the branch depicted in Fig. 5.46.

5.3.2 Four-Cell Double Symmetry-Breaking Branches

This branch shows how first one symmetry and then another is broken, resulting in three separate branches which all eventually rejoin the main branch. Numerically computed eigenvalues for just the stable branch segments are given in Table 5.7. We note that the stable solutions in this case are all asymmetric.

We shall begin by describing the symmetric (or main) branch, which is comprised of solutions 1–5. By "symmetric" we mean here that the solution displays both $y = x$ and $y = -x$ symmetries: i.e. if you "fold" the solution domain along either of these lines, it falls directly on top of itself. Beginning from solution 1 (which is itself a symmetry-breaking solution from the symmetric 4-cell pattern described previously) and moving toward 2, we observe a strengthening of the upward flow in the top left and bottom right corners of the domain, which "flattens" the top left and bottom right convection cells and squeezes them toward the center of the domain.

This process continues from 2→3, with little net change in Nu. As we move from 3→4, the Nusselt number begins decreasing significantly, and from the surface velocity vectors we observe that the top left and bottom right cells have become more rounded and continue to shrink. This trend of shrinking central cells continues from 4→5, at which point the cells which share a common boundary in the center of the domain are at their smallest observed size thus far. We note that solution 5 still displays both symmetries observed at the beginning of this branch.

Table 5.7: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for selected aspect ratios on the asymmetric four-cell branch shown in Fig. 5.63. Complex-conjugate eigenpairs are presented together. The stable solution branch is the solid line in Fig. 5.63.

Γ	σ_i		
7.400	-1.028	-2.437	-5.177
7.500	-0.296	-2.302	-4.679
7.600	-2.357	-3.023	$-12.687 \pm 6.102i$
7.700	-2.784	-4.127	$-11.339 \pm 9.814i$
7.800	-2.154	-5.060	$-10.777 \pm 10.793i$
7.900	-1.336	-5.367	$-10.551 \pm 10.979i$
8.000	-0.425	-5.168	$-10.467 \pm 10.819i$
8.030	-0.143	-5.021	$-10.449 \pm 10.732i$
8.050	-0.090	-4.797	$-10.455 \pm 10.674i$
8.100	-1.034	-3.773	$-10.681 \pm 10.425i$
8.250	-1.344	-3.452	$-11.319 \pm 9.311i$
8.260	-1.254	-3.497	$-11.329 \pm 9.222i$
8.300	-0.885	-3.486	$-11.247 \pm 8.884i$
8.350	-0.536	-1.799	$-10.553 \pm 9.182i$

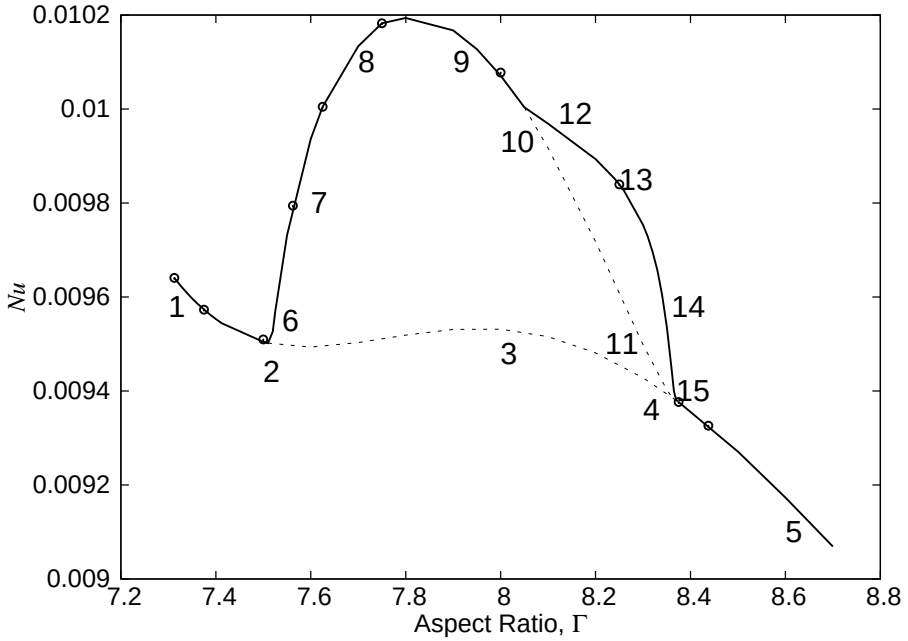


Figure 5.63: Computed solution branch for a double symmetry-breaking bifurcation. The "main" branch (numbers 1–5) has a single symmetry-breaking branch at point 6, and then a second symmetry is broken from the second branch (numbers 6–11) near point 12. This case exhibits how first one symmetry and then a second symmetry can be broken as we trace along a solution branch. Also shown, as circular markers, are solutions from "unsteady" branch number 4 described previously and used as a starting point for computing this branch.

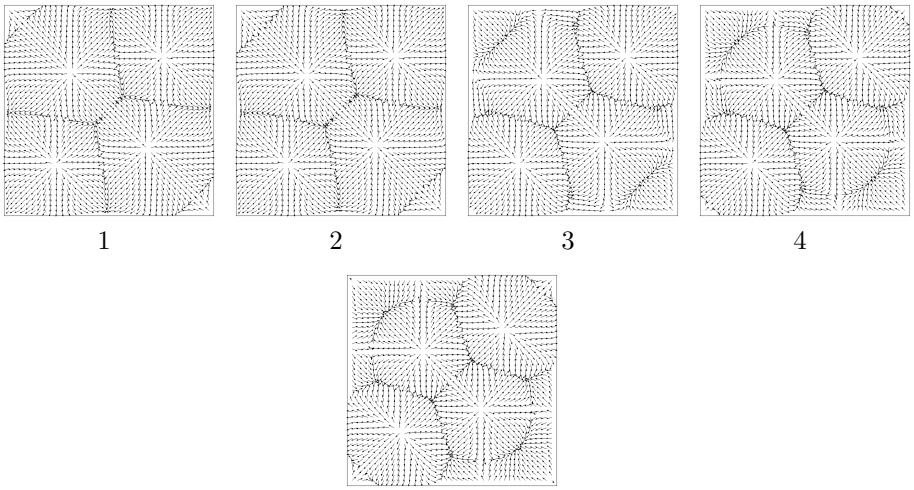


Figure 5.64: Velocity vectors for solutions on the main branch depicted in Fig. 5.63.

The first symmetry-breaking branch, which occurs near solution 6, breaks the $y = x$ symmetry, as the top left convection cell begins moving into the center of the domain, while the other three cells increase in size but remain relatively motionless. Moving from solution 7→8, we observe the characteristic "rounding" of the top cell, which was also observed on the symmetric branch. At some point between solutions 8 and 9, the configuration with the maximum-computed Nu occurs. This is interesting since, in most of the cases studied thus far, symmetric solutions at a given aspect ratio have obtained higher Nu values. As we continue from solution 9→10, the gradual rounding and shrinking of the top left cell has ceased, and the warm fluid rising in the bottom right corner of the container has begun to move the lower-right convec-

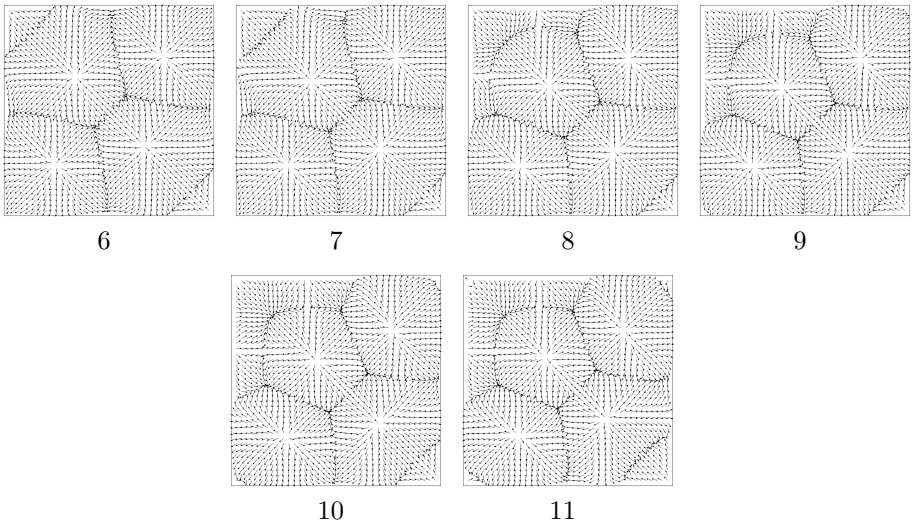


Figure 5.65: Velocity vectors for solutions on the first symmetry-breaking branch depicted in Fig. 5.63.

tion cell towards the center of the domain. That trend continues from 10→11, until eventually this branch rejoins the main (symmetric) branch where the top left and bottom right cells are of the same size.

While the first symmetry-breaking branch just described breaks the $y = x$ symmetry, the second symmetry-breaking bifurcation, which occurs around solution 12 in Fig. 5.63, breaks the $y = -x$ symmetry. On this third branch, the solution displays neither of the two original symmetries. As we move from solution 12→13, the lower left convection cell is observed to increase in size relative to the top right cell. Once again, this asymmetric solution has a higher heat transfer rate than the symmetric, or the once-broken symmetric solutions. Moving from solution 13→14, we observe the lower-right convection

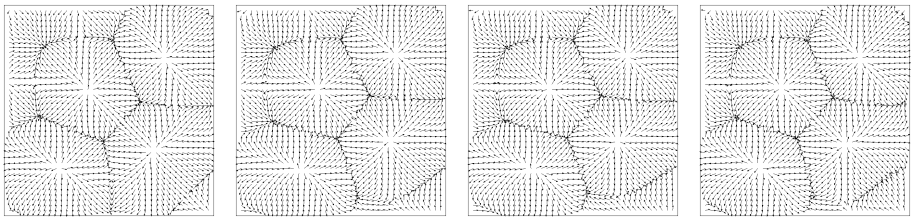
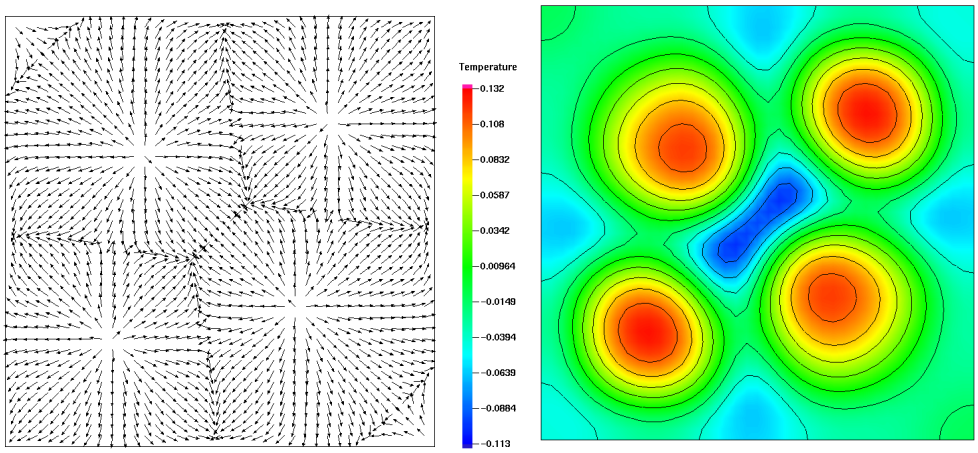


Figure 5.66: Velocity vectors for solutions on the second symmetry-breaking branch depicted in Fig. 5.63.

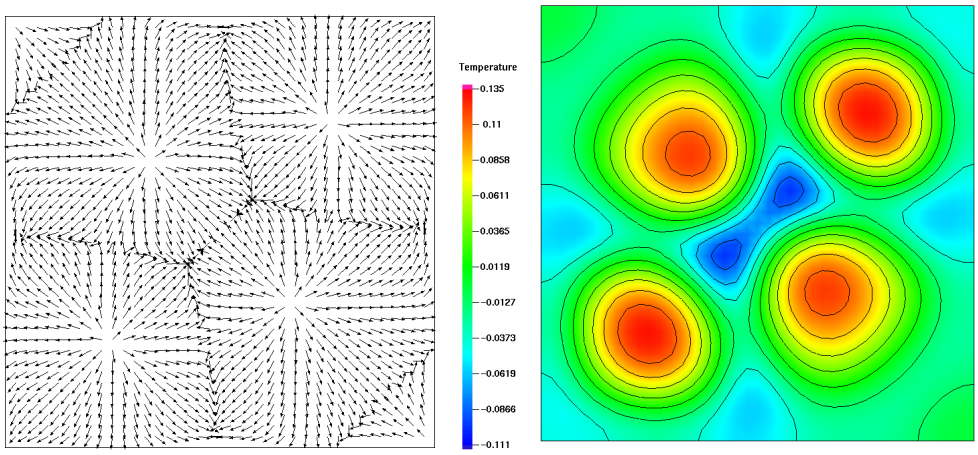
cell decreasing in size as warm fluid rising in the bottom right corner pushes it toward the center of the domain. This shrinking of the lower-right cell continues from 14→15, until eventually this branch also rejoins the original, symmetric branch around solution 4.

Detailed images of the surface temperature contours are given in Figs. 5.67–5.81.



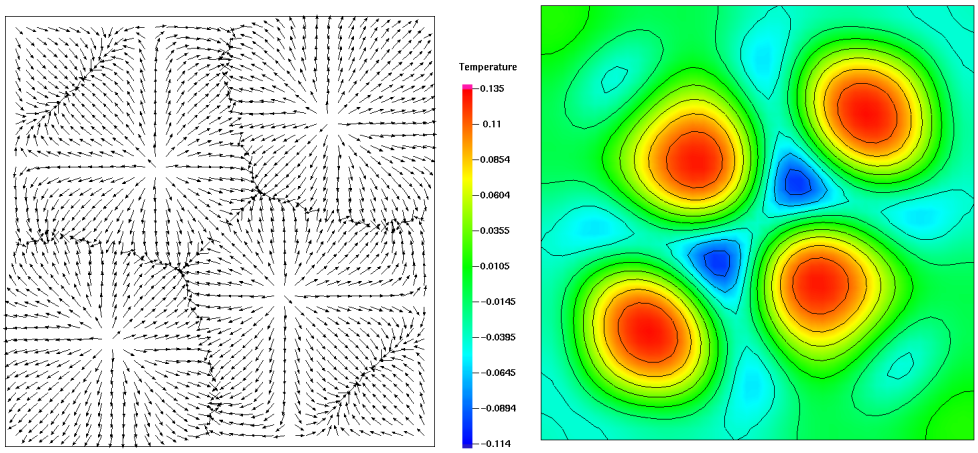
Velocity Vectors Temperature Contours

Figure 5.67: Solution 1 on the branch depicted in Fig. 5.63.



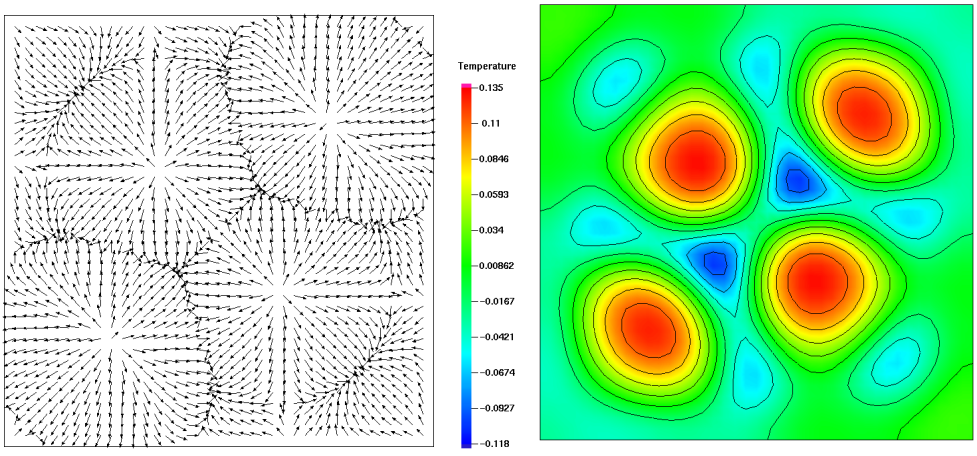
Velocity Vectors Temperature Contours

Figure 5.68: Solution 2 on the branch depicted in Fig. 5.63.



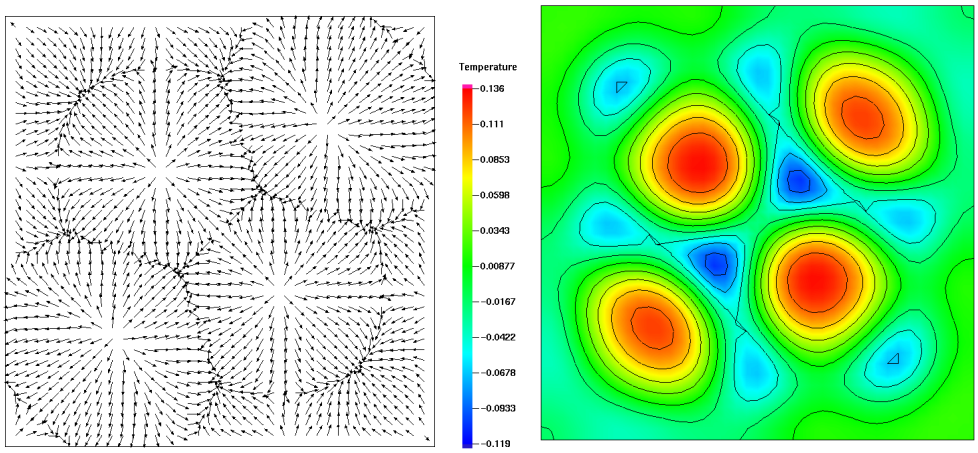
Velocity Vectors Temperature Contours

Figure 5.69: Solution 3 on the branch depicted in Fig. 5.63.



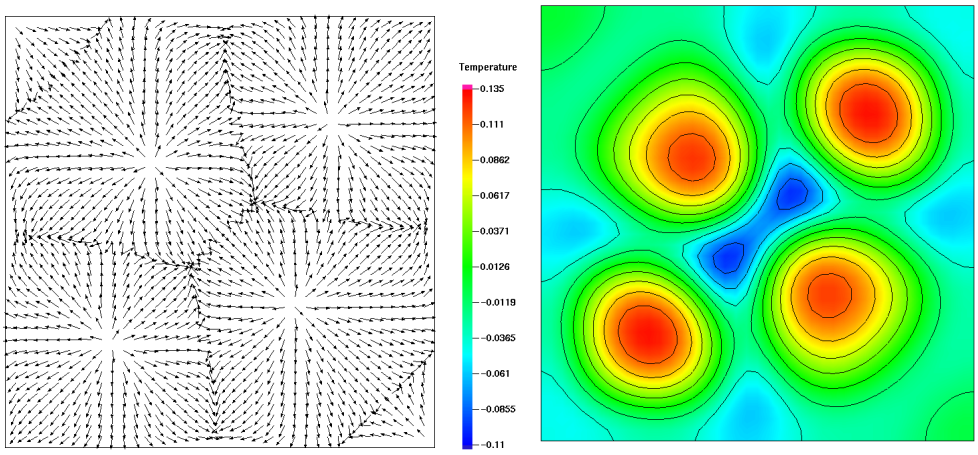
Velocity Vectors Temperature Contours

Figure 5.70: Solution 4 on the branch depicted in Fig. 5.63.



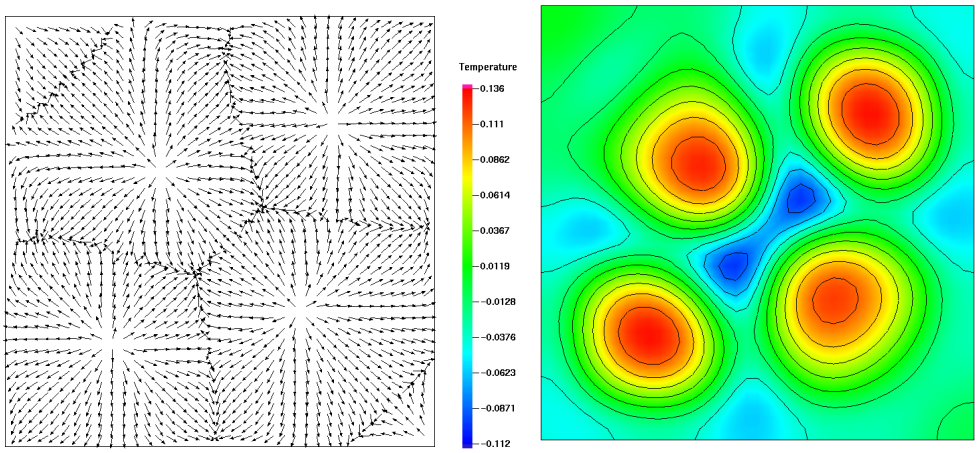
Velocity Vectors Temperature Contours

Figure 5.71: Solution 5 on the branch depicted in Fig. 5.63.



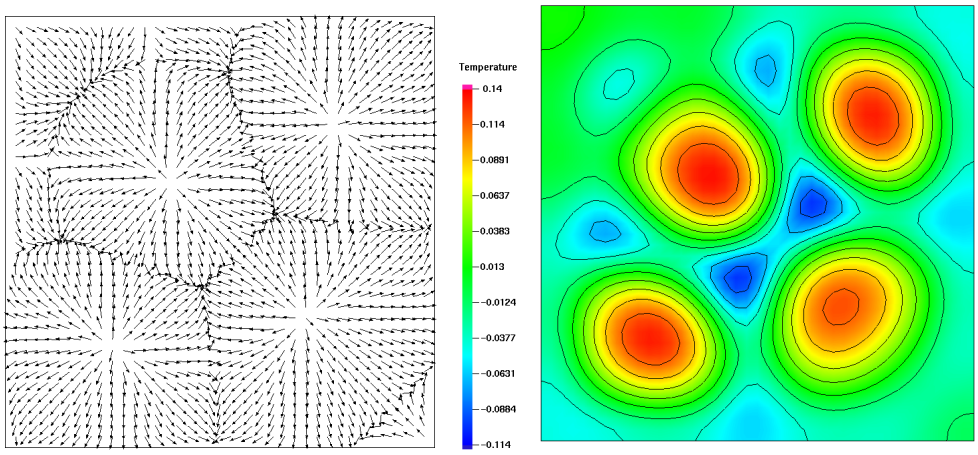
Velocity Vectors Temperature Contours

Figure 5.72: Solution 6 on the branch depicted in Fig. 5.63.



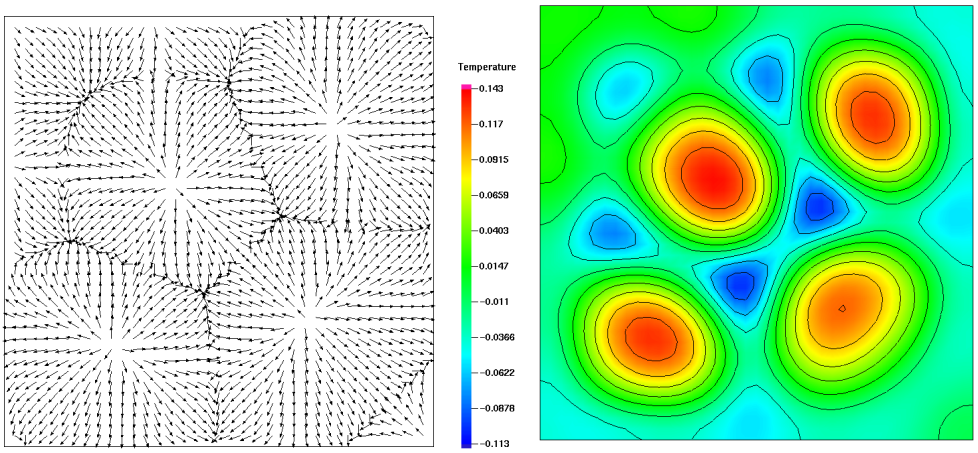
Velocity Vectors Temperature Contours

Figure 5.73: Solution 7 on the branch depicted in Fig. 5.63.



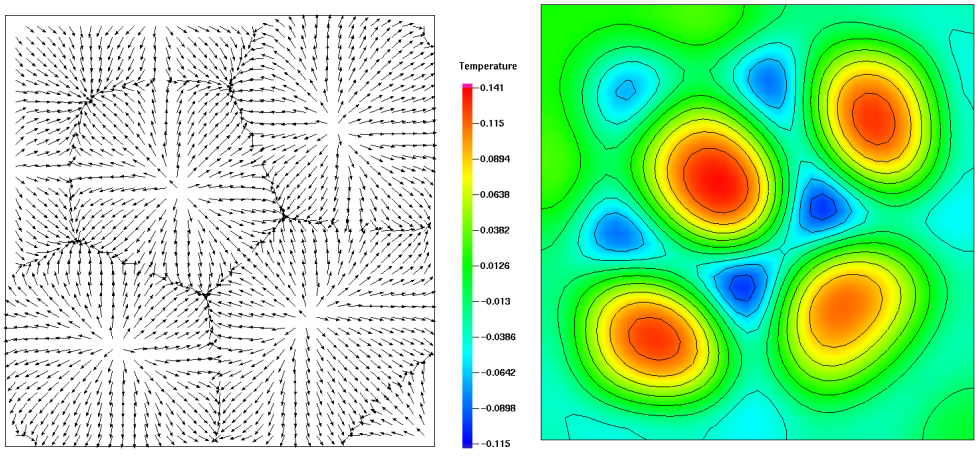
Velocity Vectors Temperature Contours

Figure 5.74: Solution 8 on the branch depicted in Fig. 5.63.



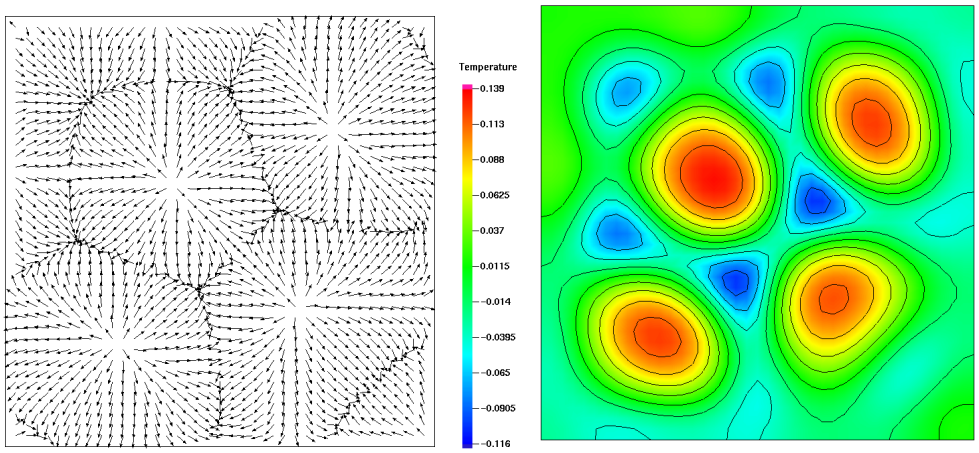
Velocity Vectors Temperature Contours

Figure 5.75: Solution 9 on the branch depicted in Fig. 5.63.



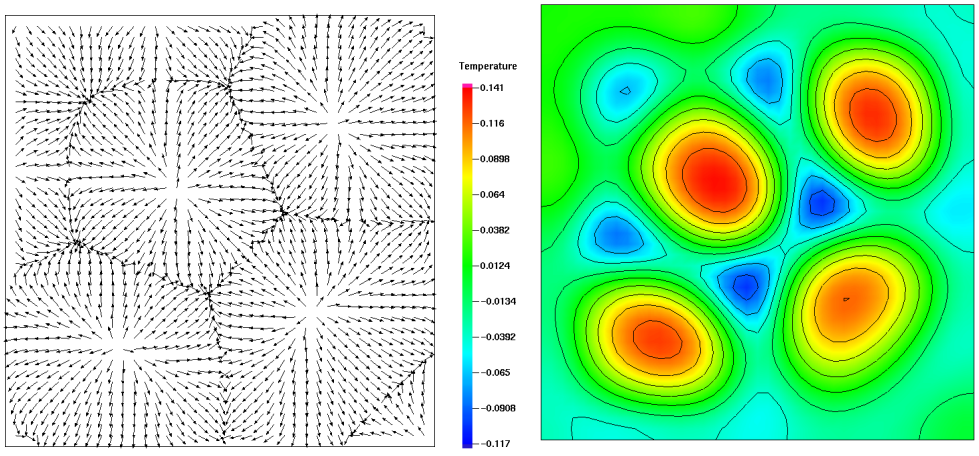
Velocity Vectors Temperature Contours

Figure 5.76: Solution 10 on the branch depicted in Fig. 5.63.



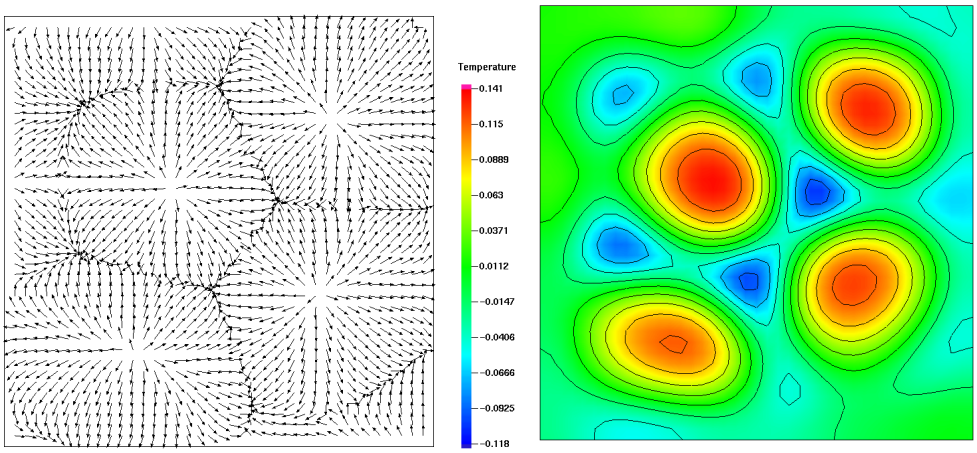
Velocity Vectors Temperature Contours

Figure 5.77: Solution 11 on the branch depicted in Fig. 5.63.



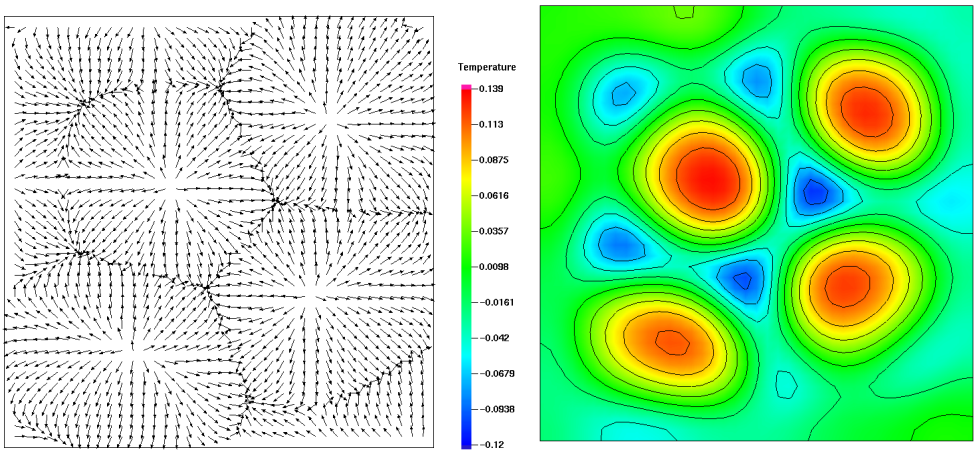
Velocity Vectors Temperature Contours

Figure 5.78: Solution 12 on the branch depicted in Fig. 5.63.



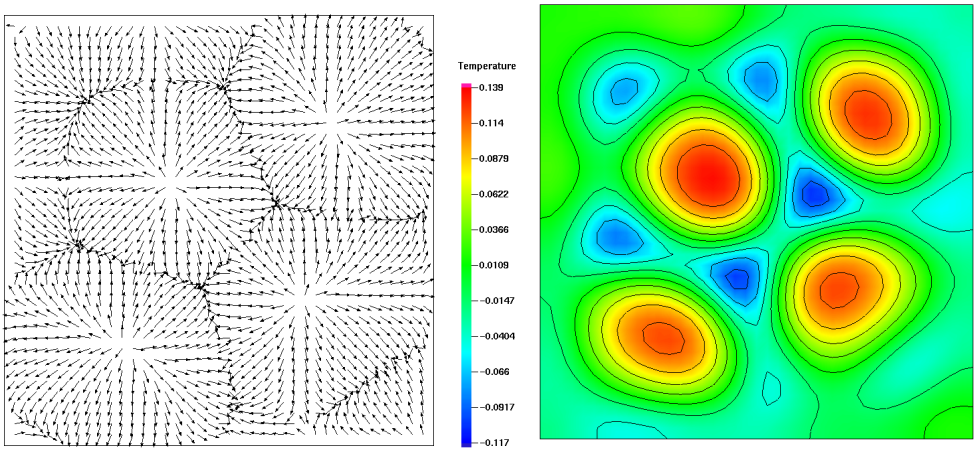
Velocity Vectors Temperature Contours

Figure 5.79: Solution 13 on the branch depicted in Fig. 5.63.



Velocity Vectors Temperature Contours

Figure 5.80: Solution 14 on the branch depicted in Fig. 5.63.



Velocity Vectors Temperature Contours

Figure 5.81: Solution 15 on the branch depicted in Fig. 5.63.

5.3.3 Five-Cell Symmetry-Breaking Bifurcation

Here we examine the transition from a symmetric five-cell branch (solutions 1, 2, and 3 in Fig. 5.82) to an asymmetric five-cell branch (solutions 4–9 in Fig. 5.82). The stable asymmetric branch pictured in Fig. 5.82 illustrates the property that asymmetric solutions often have a higher Nu value than symmetric solutions at the same aspect ratio. In the course of our calculations we have also noticed that the kinetic energy KE is also positively correlated with Nu. It may be useful to think of the symmetric solutions observed here as being similar to "minimal energy" states. Asymmetric solutions at the same aspect ratios having larger Nu values is consistent with this observation.

Solutions 1, 2, and 3, whose surface velocity vectors are shown in Fig. 5.83, are symmetric five-cell configurations with two "half" central cells filling the rest of the domain. These solutions have relatively low rates of heat transfer (compared to the asymmetric branch of solutions to be described next) and feature a central hexagonal cell. As we move along the branch from 1→3, this central hexagonal cell is observed to "flatten" and stretch horizontally while at the same time the central half-cells are growing slightly in size. Solution 2 is near the bifurcation to the asymmetric branch. Numerically computed eigenvalues for both stable and unstable symmetric solutions along this branch are given in Table 5.8.

Solution 4, which is on the asymmetric branch, is quite close to solution 2, but the $x = 0$ (i.e. vertical, through the center of the domain) symmetry has just been broken. Numerically computed eigenvalues for this branch are given

Table 5.8: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for selected aspect ratios along the main (symmetric) solution branch shown in Fig. 5.82. Complex-conjugate eigenpairs are presented together. The double horizontal lines delimit the stable and unstable regions of the solution branch (marked as solid and dashed lines, respectively, in Fig. 5.82.) The symmetric branch appears to become unstable as we pass the symmetry-breaking bifurcation around solution 2 which lies between $\Gamma = 8.7$ and $\Gamma = 8.75$.

Γ	σ_i		
8.500	2.166	-2.656	-6.448
8.540	0.543	-3.964	-4.246
8.553	-0.154	-3.127	-4.703
8.555	-0.327	-2.932	-4.792
8.560	-0.760	-2.460	-4.968
8.573	$-1.573 \pm 1.203i$	-4.691	
8.679	-1.062	$-1.766 \pm 2.630i$	
8.714	-0.202	$-1.954 \pm 2.291i$	
8.749	0.602	$-2.190 \pm 1.510i$	
8.905	1.551	-3.875	4.683

in Table 5.9. As we proceed from 4→5 along the asymmetric branch, the entire configuration of cells is observed to "rotate" slightly clockwise, while the central convection cell remains nearly hexagonal in shape. Driving this clockwise rotation appears to be the strengthening upwelling of fluid in the top left and bottom right corners. This clockwise rotation continues as we move from solution 6 to the turning point near 7, and we observe the central hexagonal cell beginning to stretch and bend into a convection cell with more than six sides by solution 8. The move from 7→8 was also accompanied by a decrease in Nu. The bending of the central cell continues to solution 9, along with the continuing decrease in the Nusselt number. The branch continues from this point (not shown) with the bottom left, central, and top right cells eventually combining into a single cell.

Table 5.9: Numerically-computed eigenvalues for selected aspect ratios along the asymmetric solution branch shown in Fig. 5.82. The double horizontal lines delimit the stable and unstable regions of the solution branch (marked as solid and dashed lines, respectively, in Fig. 5.82. The branch becomes unstable after passing through the turning point.)

Γ	σ_i		
8.725	-0.131	$-1.946 \pm 1.741i$	
8.75	-0.811	-1.301	-2.851
8.798	-1.082	-2.465	-3.289
8.792	1.725	5.092	-5.576
8.786	2.598	-6.082	6.490
8.771	4.853	-7.273	-7.593
8.738	-7.608	10.414	

Plots of surface temperature contours for these 9 states are given in Figs. 5.85– 5.93.

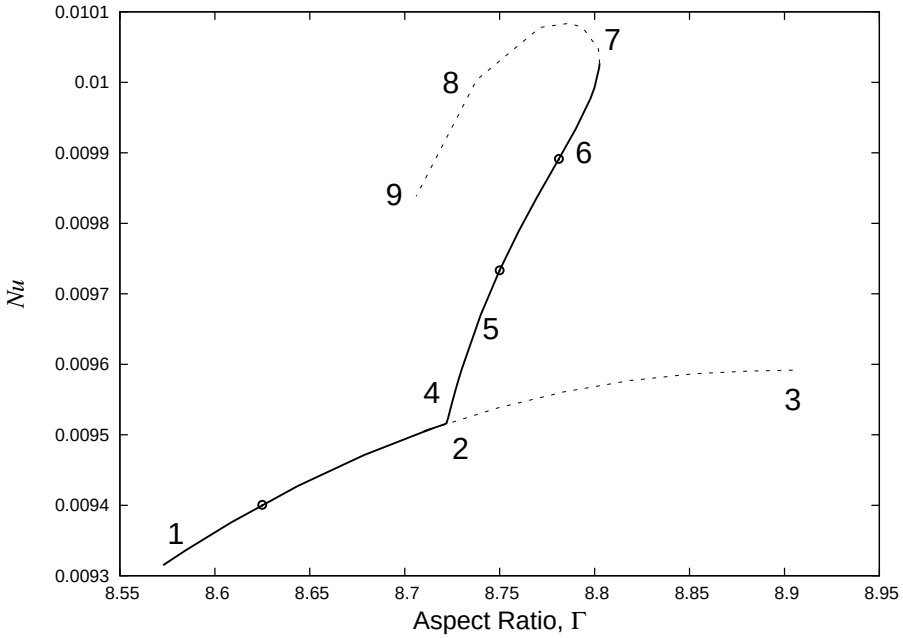


Figure 5.82: Computed solution branch displaying bifurcation from a symmetric 5-cell configuration to an asymmetric 5-cell configuration with higher Nusselt number. Also shown, as circular markers, are solutions from "unsteady" branch number 5 described previously and used as a starting point for computing this branch.

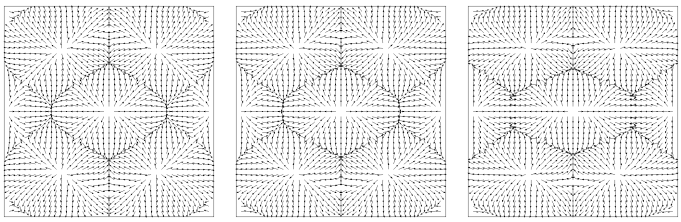


Figure 5.83: Velocity vectors for solutions on the symmetric branch depicted in Fig. 5.82.

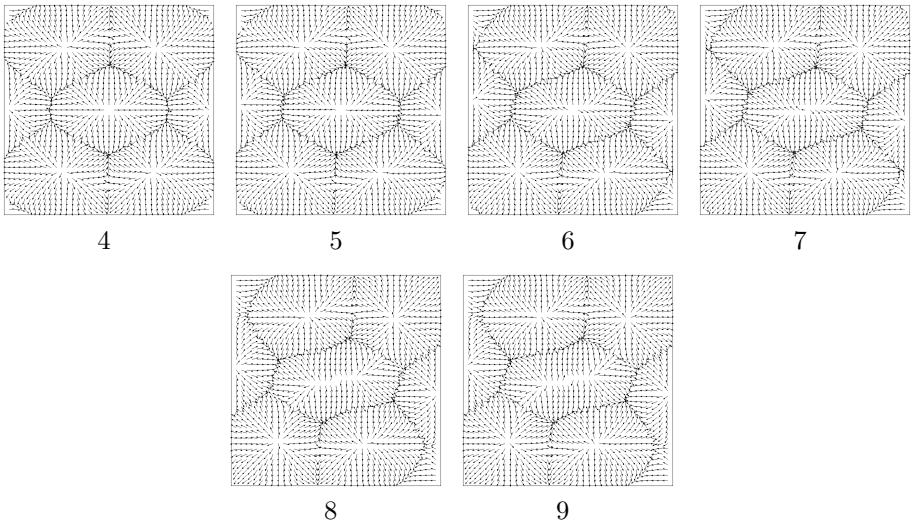
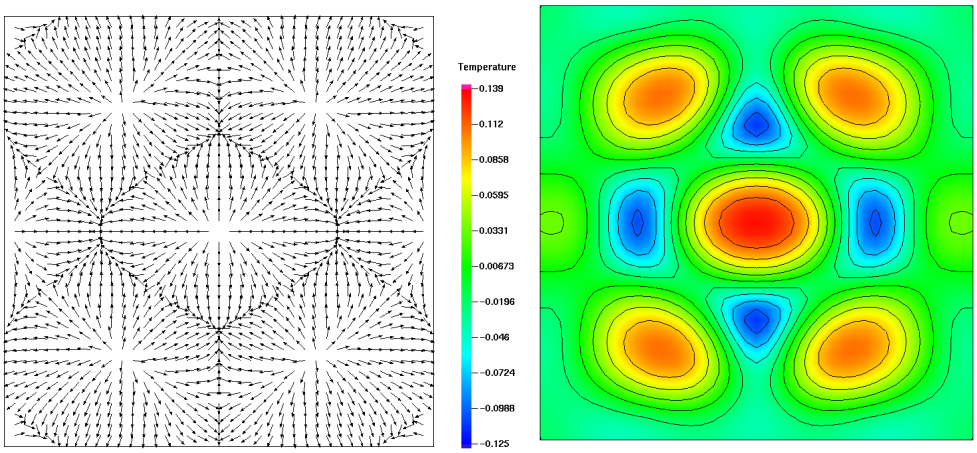
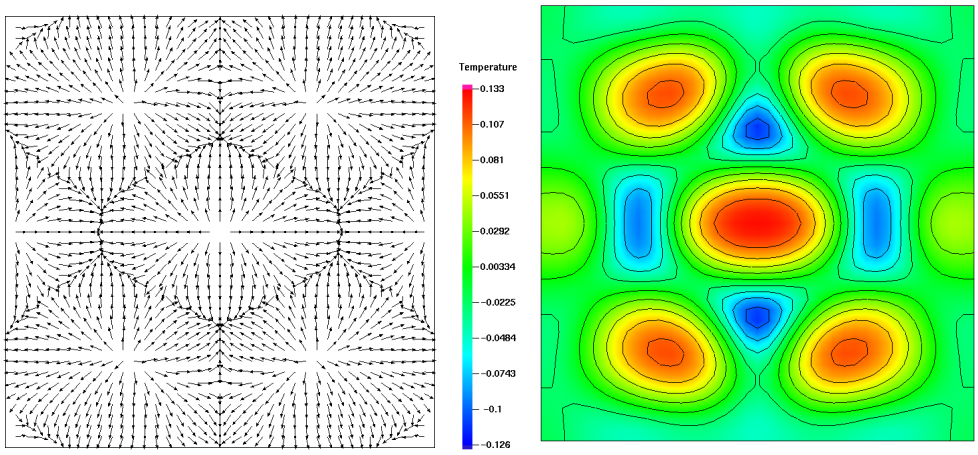


Figure 5.84: Velocity vectors for solutions on the asymmetric branch depicted in Fig. 5.82.



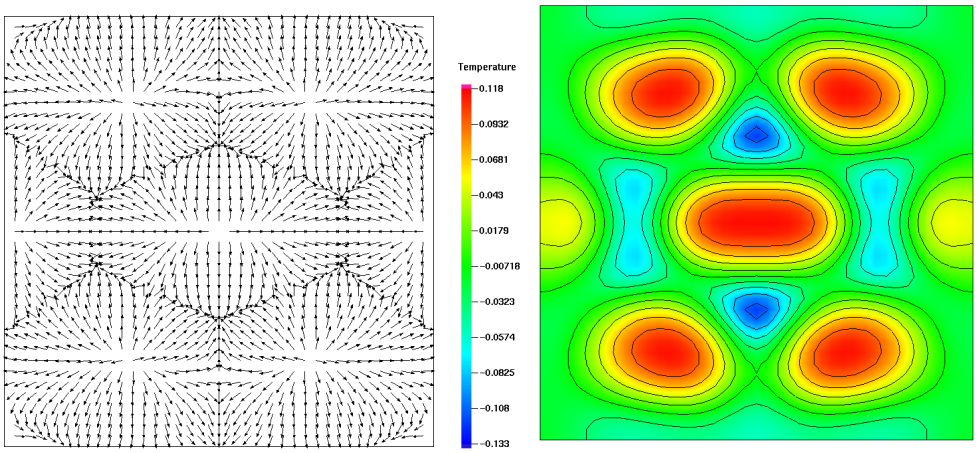
Velocity Vectors Temperature Contours

Figure 5.85: Solution 1 on the branch depicted in Fig. 5.82.



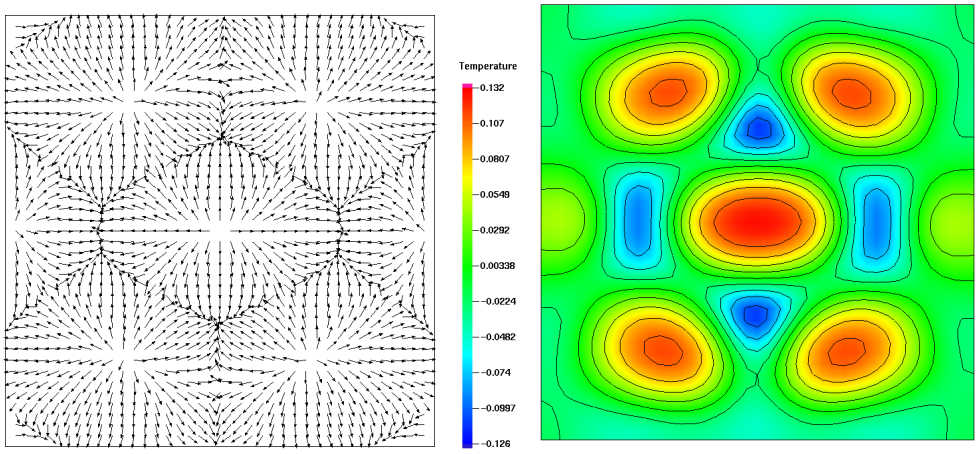
Velocity Vectors Temperature Contours

Figure 5.86: Solution 2 on the branch depicted in Fig. 5.82.



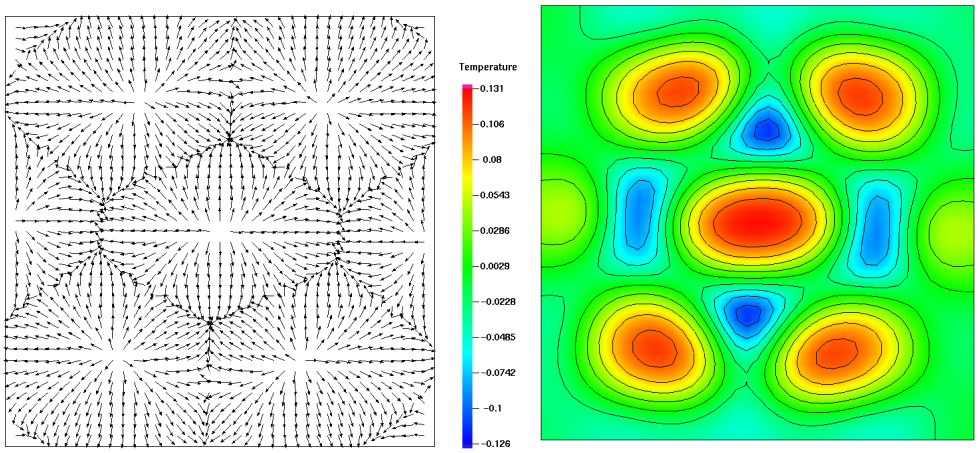
Velocity Vectors Temperature Contours

Figure 5.87: Solution 3 on the branch depicted in Fig. 5.82.



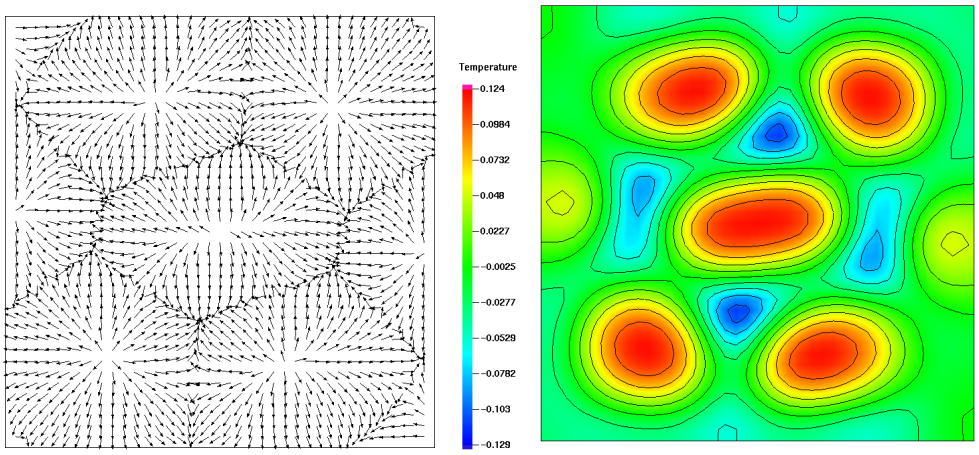
Velocity Vectors Temperature Contours

Figure 5.88: Solution 4 on the branch depicted in Fig. 5.82.



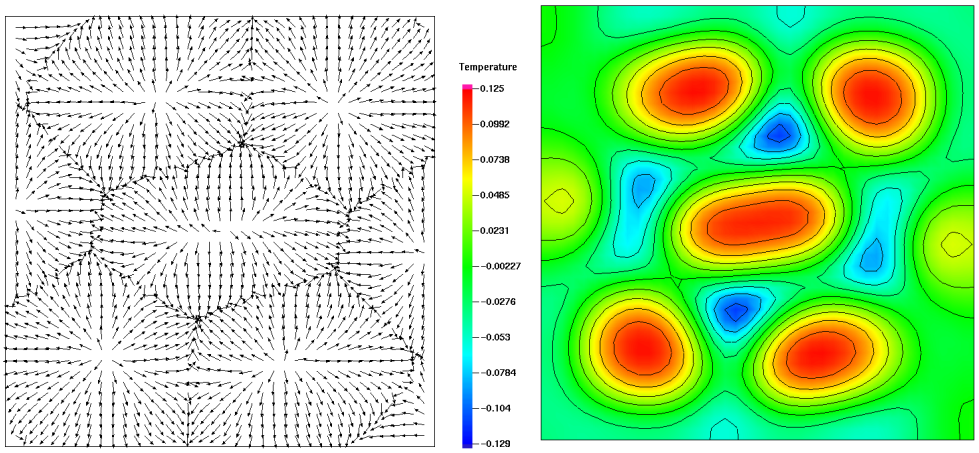
Velocity Vectors Temperature Contours

Figure 5.89: Solution 5 on the branch depicted in Fig. 5.82.



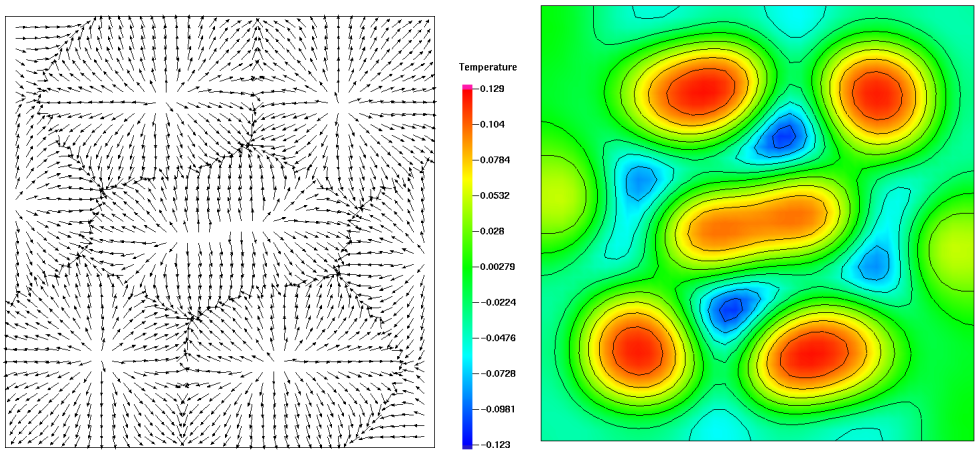
Velocity Vectors Temperature Contours

Figure 5.90: Solution 6 on the branch depicted in Fig. 5.82.



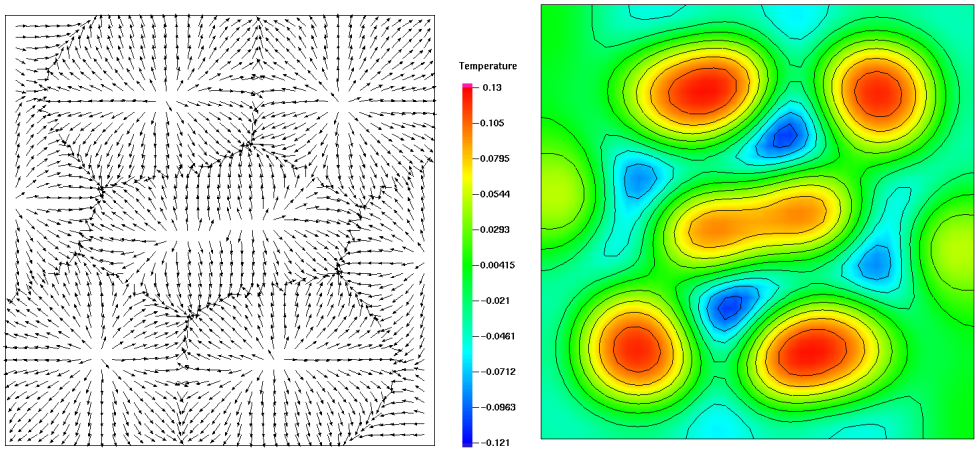
Velocity Vectors Temperature Contours

Figure 5.91: Solution 7 on the branch depicted in Fig. 5.82.



Velocity Vectors Temperature Contours

Figure 5.92: Solution 8 on the branch depicted in Fig. 5.82.



Velocity Vectors Temperature Contours

Figure 5.93: Solution 9 on the branch depicted in Fig. 5.82.

5.4 Six, Seven, and Eight-Cell Solution Branches

In Fig. 5.94, the six, seven, and eight convection cell branches are shown. The subsections describing these branches in detail are organized as follows: §5.4.1 we describe the general characteristics of the six and symmetric eight-cell branches. They are convenient to discuss together since an unstable branch exists connecting them. The numerical results are compared qualitatively with an eight-cell experimental result by Koschmieder shown later.

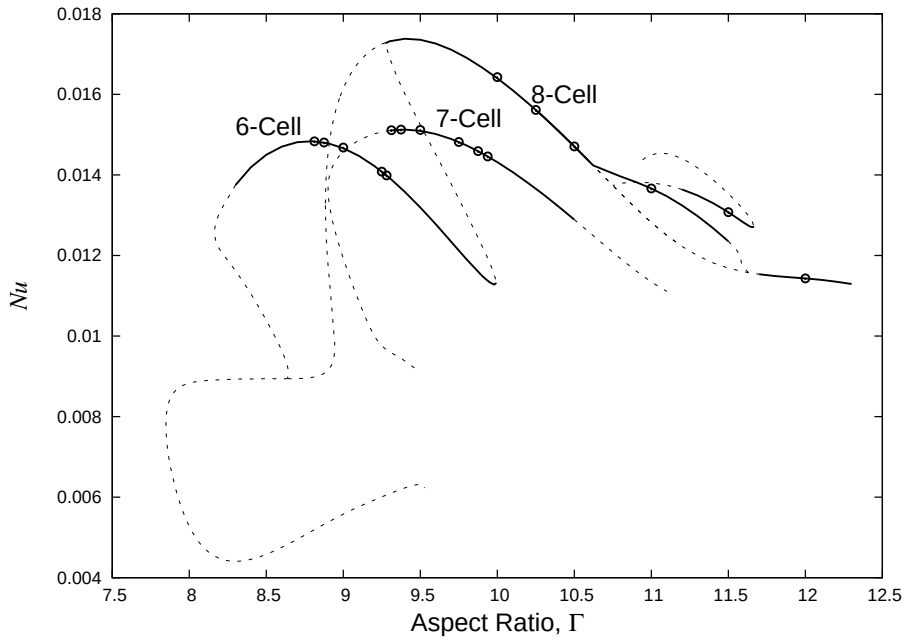


Figure 5.94: Solution branches for 6, 7, and 8-cell solutions.

Next, in §5.4.2, we discuss the seven-cell branch in detail. The seven-

cell patterns, which in some sense appear "natural" due to their symmetry, have not been observed experimentally to our knowledge. Finally, in §5.4.3 we discuss the asymmetric eight-cell solutions which occurred for the largest aspect ratios considered by this study. Like the four-cell double symmetry breaking results discussed in §5.3.2, the eight-cell configurations also exhibit multiple stable asymmetric solutions before returning to a single, symmetric solution.

5.4.1 Six and Eight-Cell Branches

In Figure 5.95, we show in more detail the six and eight-cell branches from Fig. 5.94 The surface velocity vectors of solutions 1–6 are shown in detail in Fig. 5.96, and consist of six-convection cell configurations with one ($y = 0$) axis of symmetry. Numerically computed eigenvalues for this segment of the branch are presented in Table 5.10, and suggest that the six-cell configuration becomes stable somewhere between $\Gamma = 8.2$ and $\Gamma = 8.3$.

As the aspect ratio increases, the convection cell in the middle of the left side of the container becomes smaller in size, until it is nearly the size of the neighboring hexagonal cell to its right. The convection cells in the top and bottom corners are growing in strength at the same time. The configuration with the highest Nu value is solution 2, while solution 6 has the lowest Nusselt number for this group.

In contrast, solutions 7–12 are characterized by having eight convection cells (instead of 6) and also have both $x = 0$ and $y = 0$ lines of symmetry. Nu-

Table 5.10: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment containing solutions 1–6 shown in Fig. 5.95. The first eigenvalue is computed just to the right of the turning point. Complex-conjugate eigenpairs are presented together.

Γ	σ_i		
8.170	-1.438	-5.144	5.499
8.200	-5.898	3.186	-3.002
8.300	-1.444	-5.547	-7.237
8.400	-5.179	-7.498	-8.224
8.600	-10.616	$-10.681 \pm 0.441i$	
9.000	-11.512	-13.815	-20.075
9.500	-8.320	-11.500	-18.814
9.900	-3.531	-5.586	-8.343

merically computed eigenvalues for these solutions are presented in Table 5.11. Solutions 7 and 8 look remarkably similar to the experimental eight convection cell solution observed by Koschmieder at a reported value of $\Gamma = 8.75$ (Fig. 5.98). Solutions 7–12 are characterized by higher Nu values than solutions 1–6. Beyond approximately $\Gamma = 10.6$, this branch becomes unstable.

Stable solutions for higher aspect ratios will be discussed in a later section. Several solutions (13–15) on the unstable branch connecting the stable six and eight-cell branches are shown in Fig. 5.99. Although they should not be regarded as physically-realizable (since they are unstable) they do show how the solution transitions continuously from six cells to eight, as the top and bottom cells on the left side of the container move into the domain, and

Table 5.11: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for solutions 7–12 on the six-cell branch shown in Fig. 5.95.

Γ	σ_i		
9.200	2.186	−10.164	−10.295
9.300	−0.851	−12.175	−14.581
9.500	−6.997	−13.679	−19.586
10.600	−0.433	−2.797	−13.061
11.000	4.088	4.579	−8.775

the tiny "half" cell on the right is lost. Numerically computed eigenvalues for these unstable solutions are presented in Table 5.12.

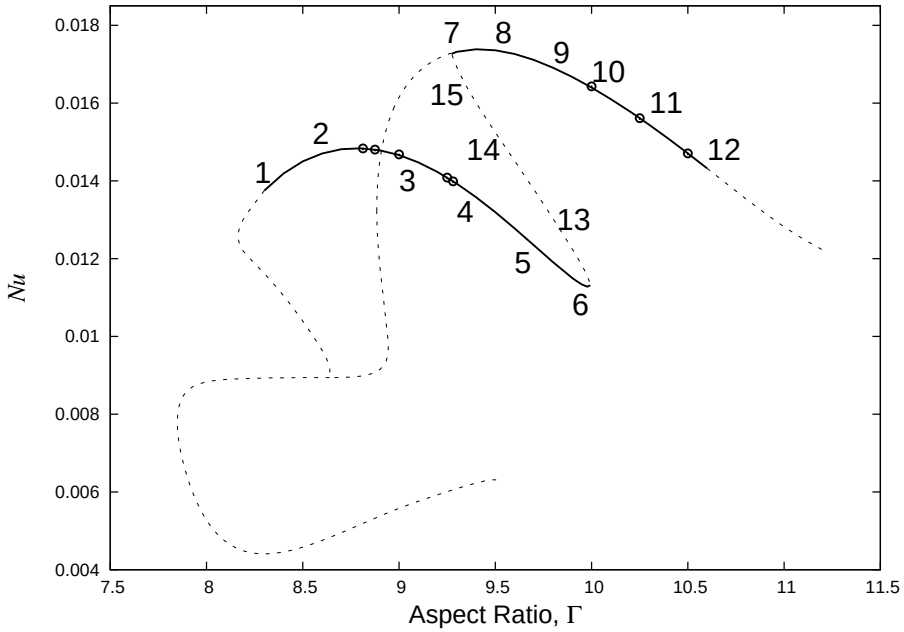


Figure 5.95: Computed solution branch showing hysteresis and multiple stable steady state solutions. The numbers correspond to solutions discussed in greater detail in the text, and in Figs. 5.96, 5.97, and 5.99. Also shown, as circular markers, are solutions from "unsteady" branch numbers 6 and 8 described previously and used as a starting point for computing this branch.

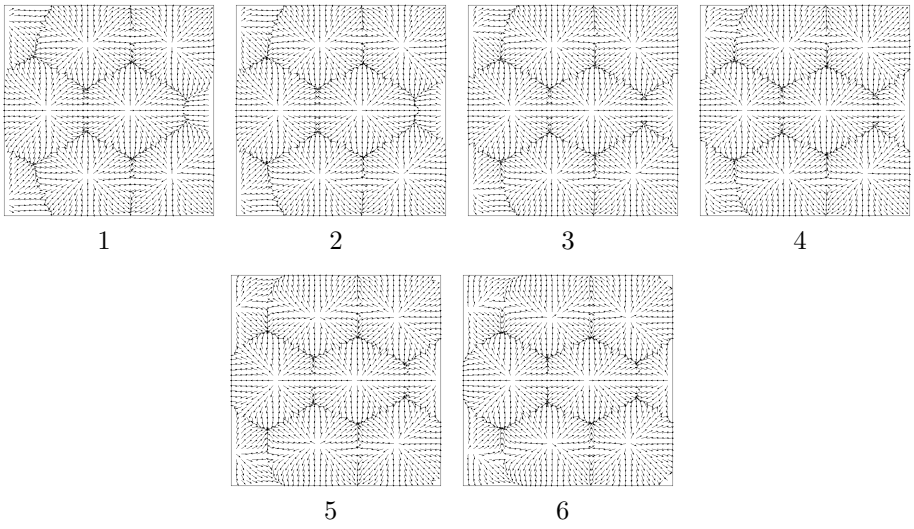


Figure 5.96: Velocity vectors for solutions on the lower stable branch depicted in Fig. 5.95.

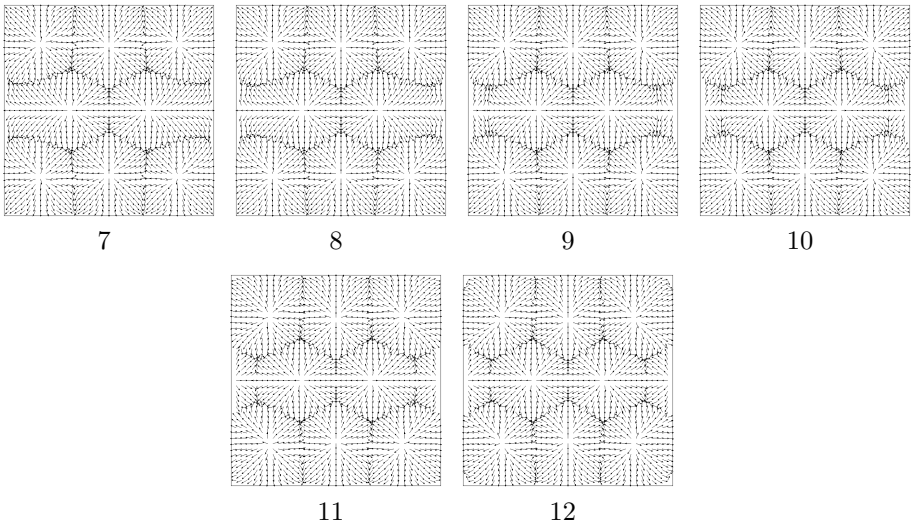


Figure 5.97: Velocity vectors for solutions on the upper stable branch depicted in Fig. 5.95.

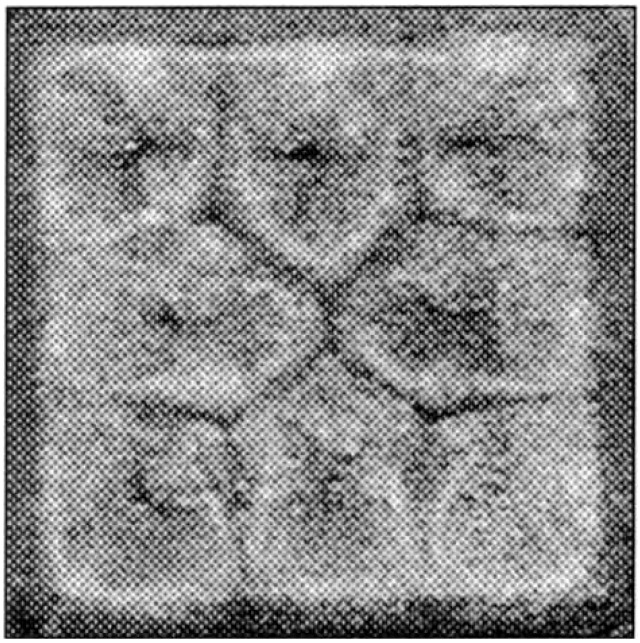
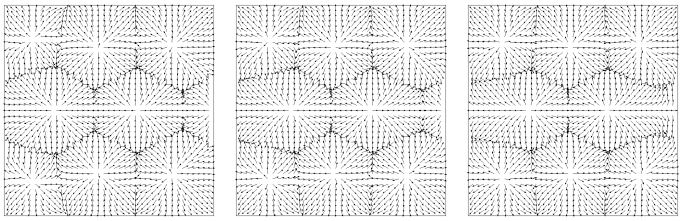


Figure 5.98: Experimentally-observed 8-cell configuration by Koschmieder at $Ra = 16$, $Ma = 63$, and aspect ratio $\Gamma = 8.75$. This experimental result corresponds closely to configurations "7" and "8" on the upper stable branch of Fig. 5.95, and which are shown in detail in Fig. 5.97. The computed solutions occur for $Ra = 30$, $Ma = 92$, and $9.3 \leq \Gamma \leq 9.5$.



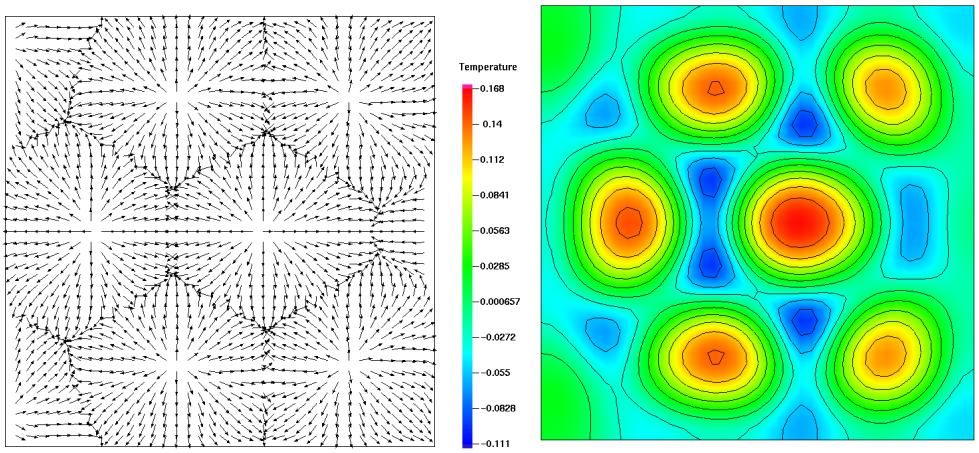
13 14 15

Figure 5.99: Velocity vectors for solutions on the unstable branch depicted in Fig. 5.95.

Table 5.12: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for solutions 13–15 shown in Fig. 5.95. Complex-conjugate eigenpairs are presented together.

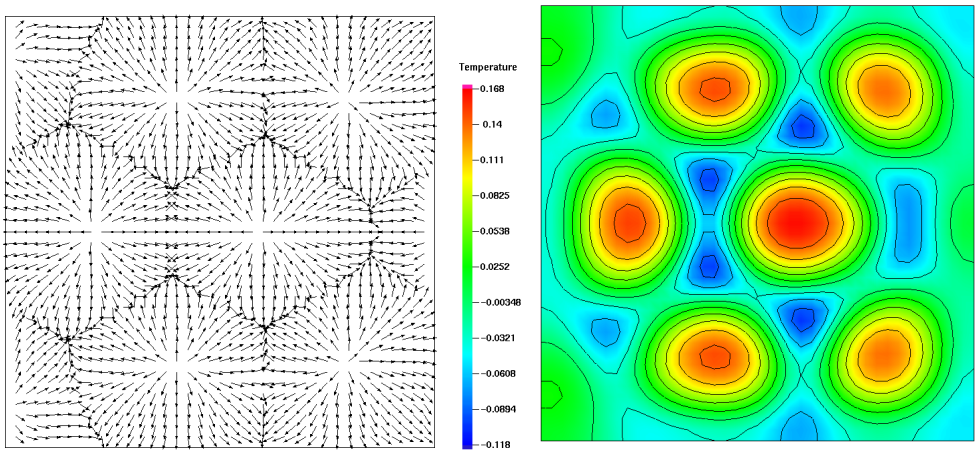
Γ	σ_i		
9.989	1.576	1.700	−5.178
9.284	0.736	−6.889	−16.494

Additional images displaying the surface temperature contours and surface velocity fields for solutions 1–15 just discussed can be found in Figs. 5.100–5.114.



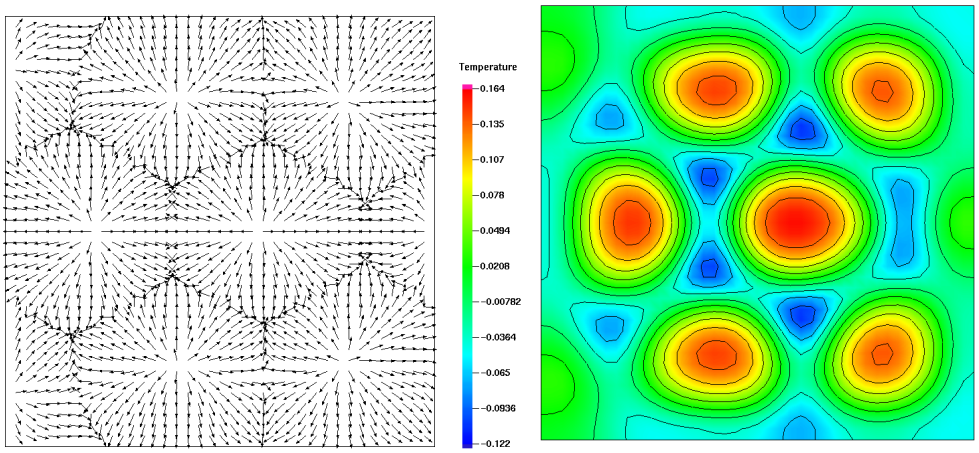
Velocity Vectors Temperature Contours

Figure 5.100: Solution 1 on the branch depicted in Fig. 5.95.



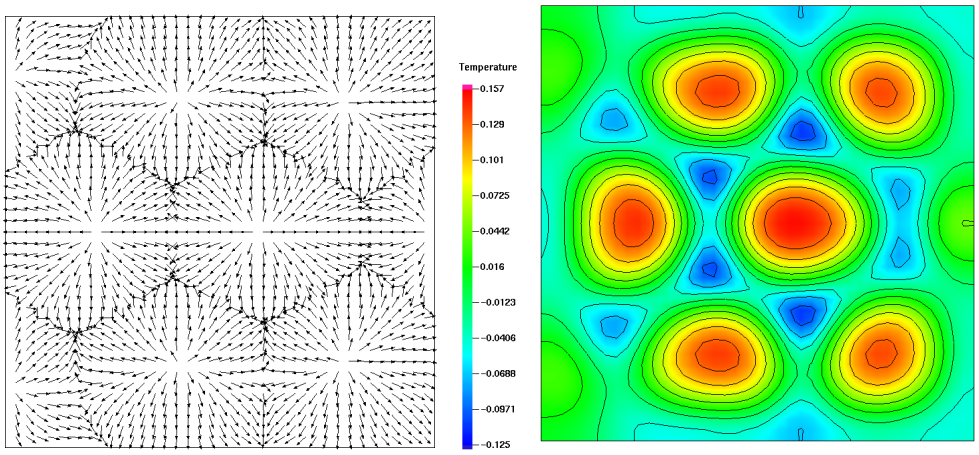
Velocity Vectors Temperature Contours

Figure 5.101: Solution 2 on the branch depicted in Fig. 5.95.



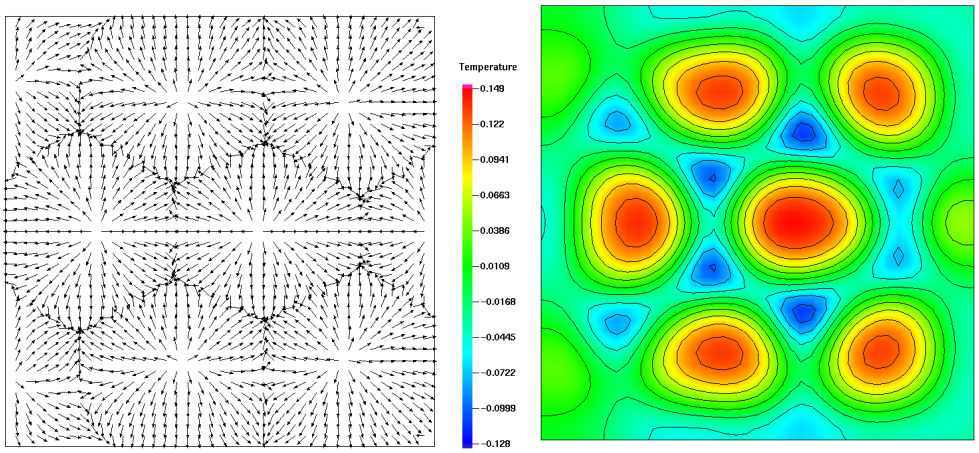
Velocity Vectors Temperature Contours

Figure 5.102: Solution 3 on the branch depicted in Fig. 5.95.



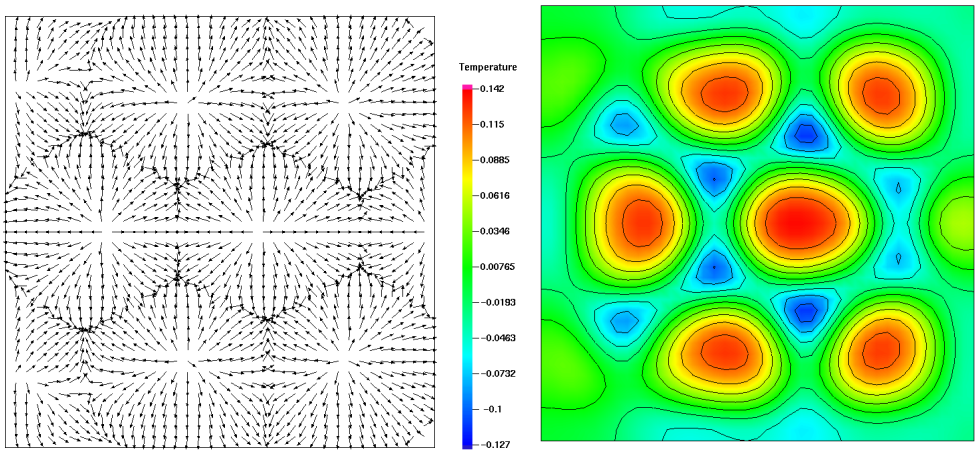
Velocity Vectors Temperature Contours

Figure 5.103: Solution 4 on the branch depicted in Fig. 5.95.



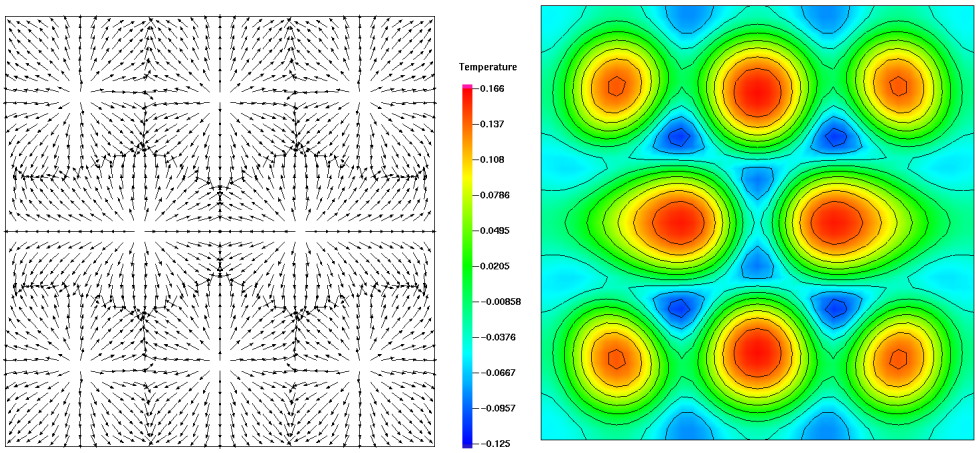
Velocity Vectors Temperature Contours

Figure 5.104: Solution 5 on the branch depicted in Fig. 5.95.



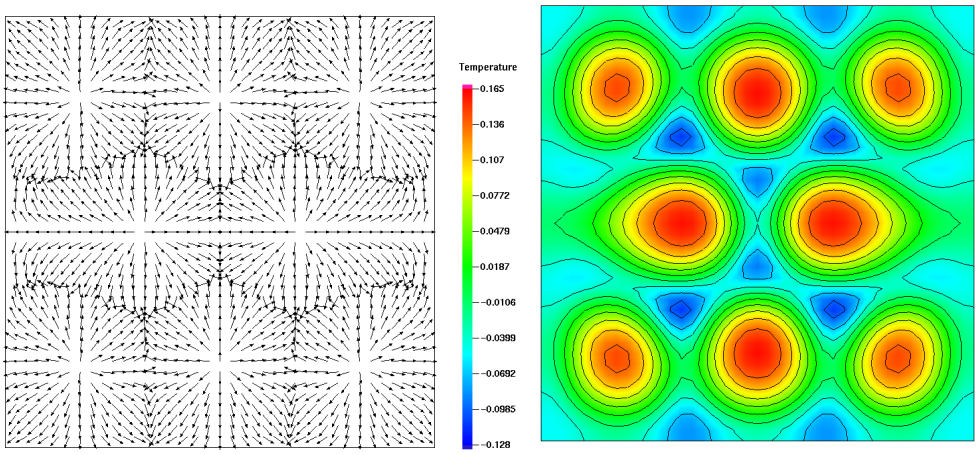
Velocity Vectors Temperature Contours

Figure 5.105: Solution 6 on the branch depicted in Fig. 5.95.



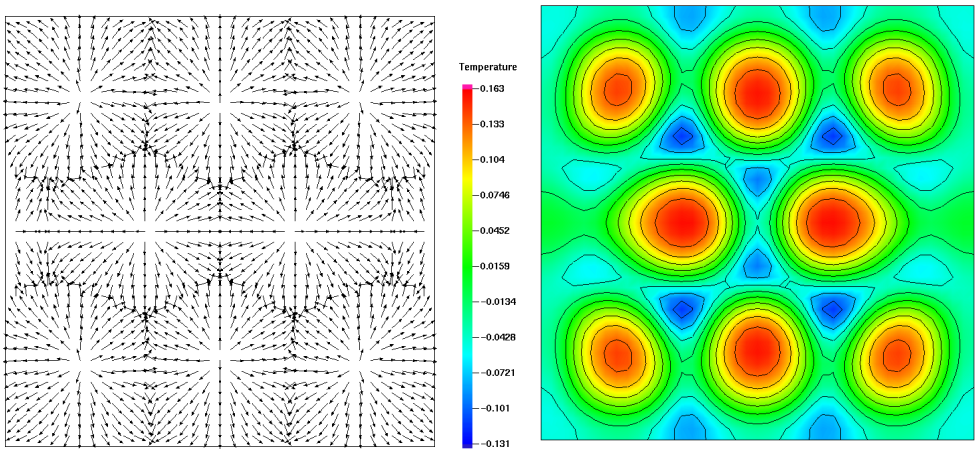
Velocity Vectors Temperature Contours

Figure 5.106: Solution 7 on the branch depicted in Fig. 5.95.



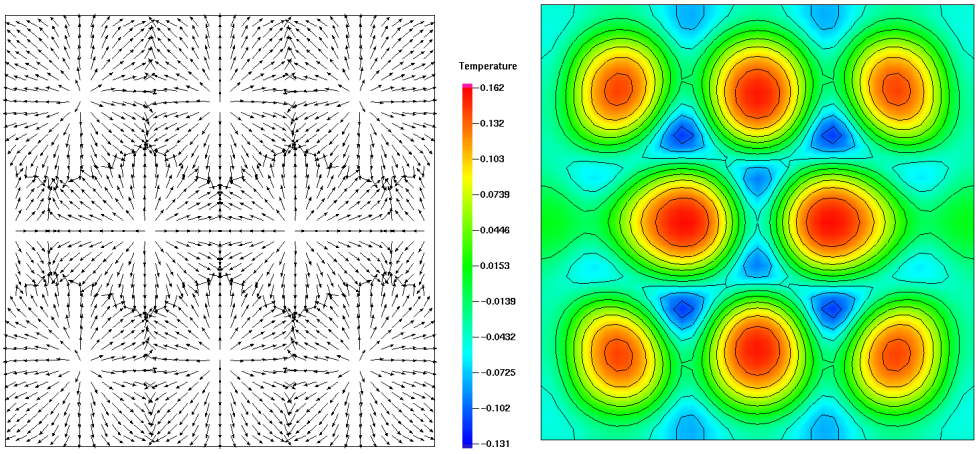
Velocity Vectors Temperature Contours

Figure 5.107: Solution 8 on the branch depicted in Fig. 5.95.



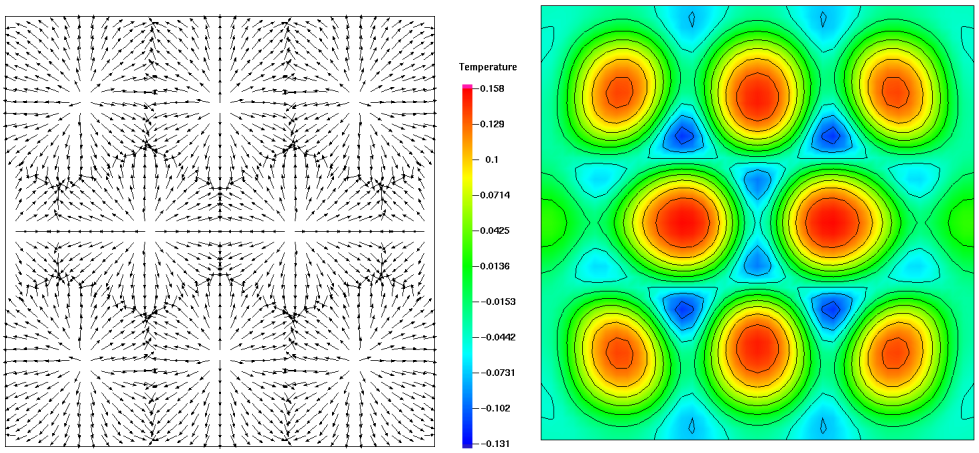
Velocity Vectors Temperature Contours

Figure 5.108: Solution 9 on the branch depicted in Fig. 5.95.



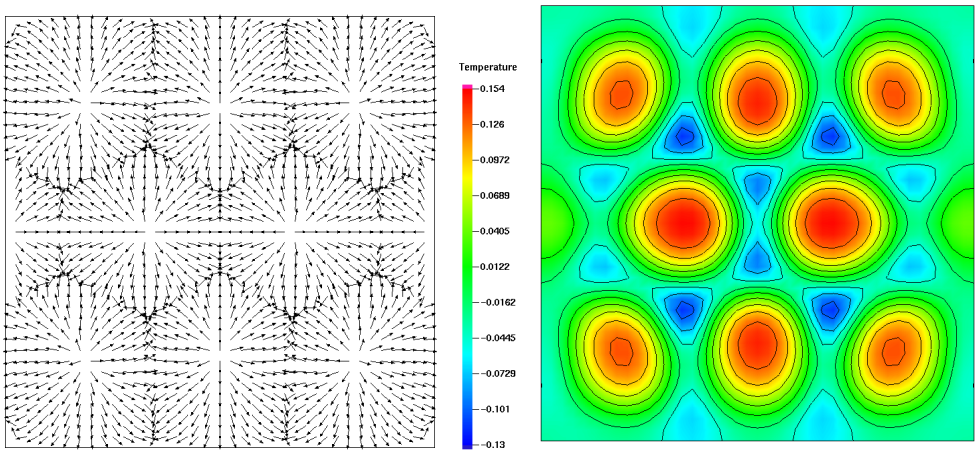
Velocity Vectors Temperature Contours

Figure 5.109: Solution 10 on the branch depicted in Fig. 5.95.



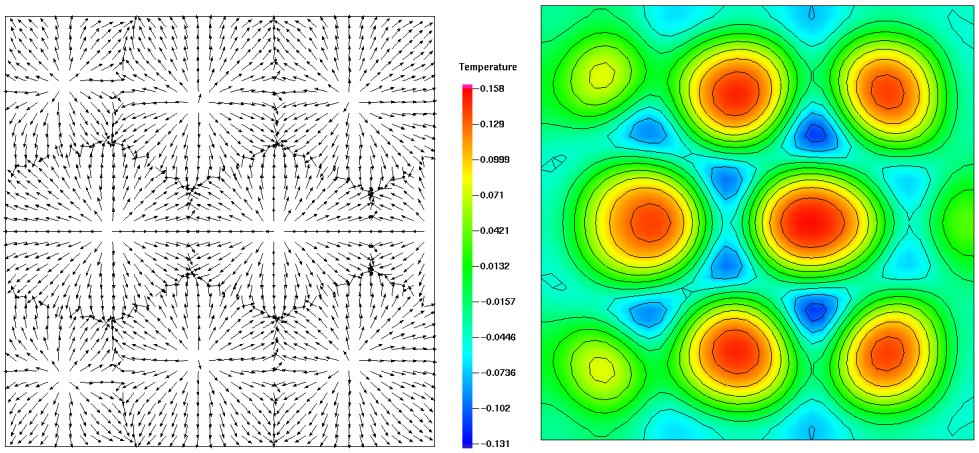
Velocity Vectors Temperature Contours

Figure 5.110: Solution 11 on the branch depicted in Fig. 5.95.



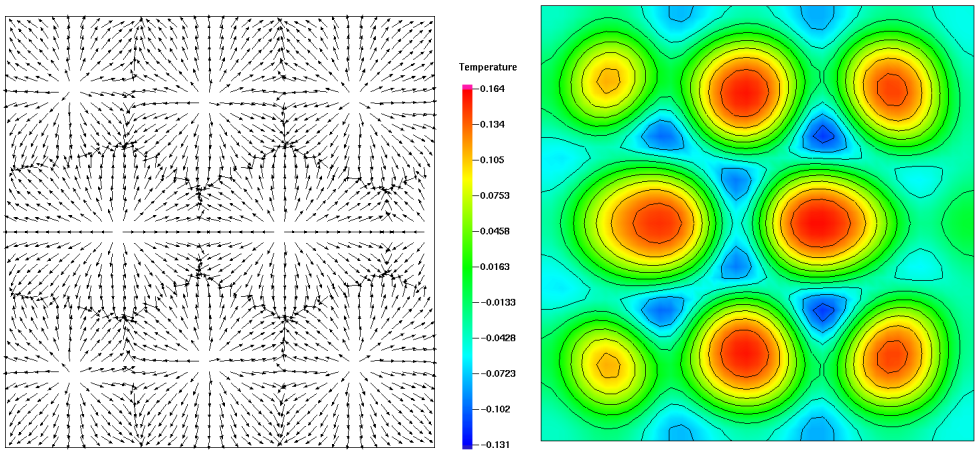
Velocity Vectors Temperature Contours

Figure 5.111: Solution 12 on the branch depicted in Fig. 5.95.



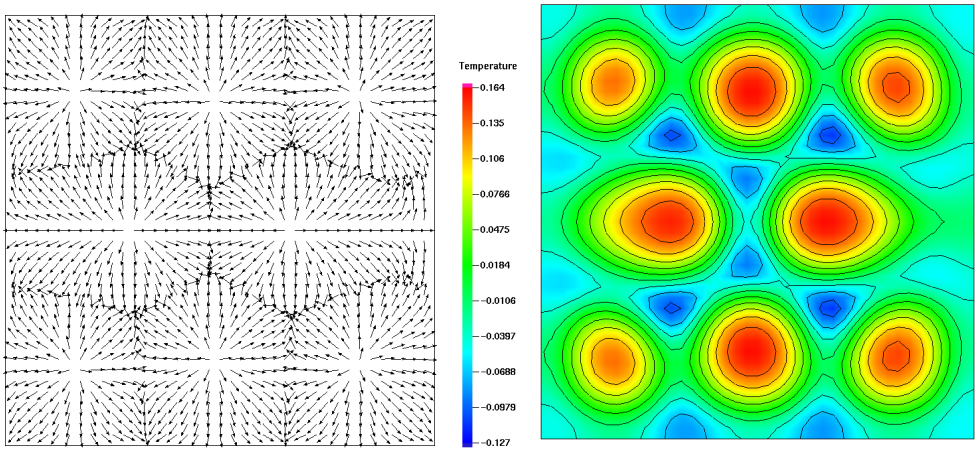
Velocity Vectors Temperature Contours

Figure 5.112: Solution 13 on the branch depicted in Fig. 5.95.



Velocity Vectors Temperature Contours

Figure 5.113: Solution 14 on the branch depicted in Fig. 5.95.



Velocity Vectors Temperature Contours

Figure 5.114: Solution 15 on the branch depicted in Fig. 5.95.

5.4.2 Seven-Cell Branch

This branch depicts several stable seven-cell solutions, as well as some eigenvalues at different points along the branch. These seven-cell solutions are characterized by intermediate Nu values between the six and eight-cell solution branches also shown in Fig. 5.94. Numerically computed eigenvalues for this branch are given in Table 5.13 and predict that the seven-cell solution is stable for approximately $9.3 \leq \Gamma \leq 10.5$.

Beginning with solution 1 in Fig. 5.115, and continuing through solution 7, we observe a central hexagonal convection cell surrounded six other cells having either 4 or five sides. For example, in solution 1 (which is shown in detail in Fig. 5.116) we observe elongated five-sided cells at the top and bottom of the domain, while the middle-left and middle-right cells are four-sided. As the aspect ratio increases, the top and bottom convection cells shrink in size and become irregular pentagons while the middle-left and middle-right cells move towards the center of the domain and become irregularly-shaped hexagons.

The solution structure near subfigure 4 of Fig. 5.116 has an almost circular ring of six cells surrounding the central hexagon. Continuing to increase the aspect ratio, we observe in solutions 5, 6, and 7 a "sharpening" of the cell boundaries of the middle-left and middle-right encircling convection cells, in contrast to their more rounded appearance at smaller aspect ratios. This pattern loses stability to the eight cell configurations discussed in §5.4.3. Several numerically-computed eigenvalues along the branch are given in Table 5.13,

Table 5.13: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment shown in Fig. 5.115. Complex-conjugate eigenpairs are presented together. The stable part of the solution branch is denoted by a solid line in Fig. 5.115.

Γ	σ_i		
9.000	2.181	8.769	-9.720
9.100	1.367	5.483	-12.029
9.200	0.049	2.653	-15.629
9.300	-0.099	-1.582	$-18.421 \pm 2.076i$
9.400	-2.853	-3.384	$-20.590 \pm 1.143i$
10.000	-9.194	-12.956	-14.353
10.500	-1.092	-4.479	-12.210
10.600	0.717	-2.572	-11.307
10.700	-0.651	2.446	-10.369
11.000	4.546	6.644	-7.925

and they show that the x and y-symmetric seven-cell solution becomes stable somewhere between $\Gamma = 9.2$ and $\Gamma = 9.3$, and becomes unstable again somewhere between $\Gamma = 10.5$ and $\Gamma = 10.6$.

The surface velocity and temperature fields are shown in more detail in Figs. 5.117–5.123.

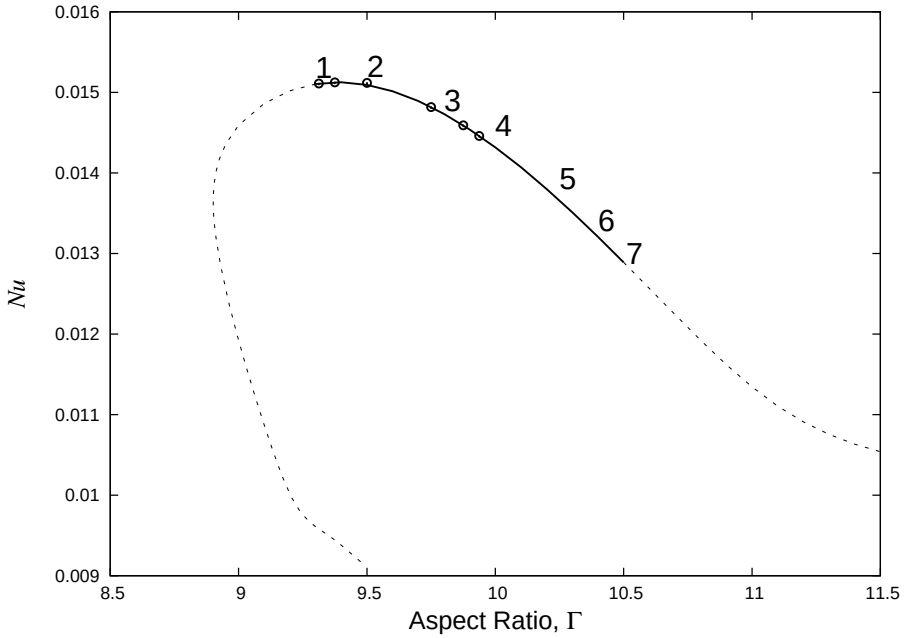


Figure 5.115: Branch with seven-cell solutions. The numbers correspond to solutions discussed in greater detail in the text, and in Fig. 5.116. Also shown, as circular markers, are six-cell solutions obtained as the solution in the steady-state limit of the unsteady governing equations. These solutions were used as a starting point for computing this branch.

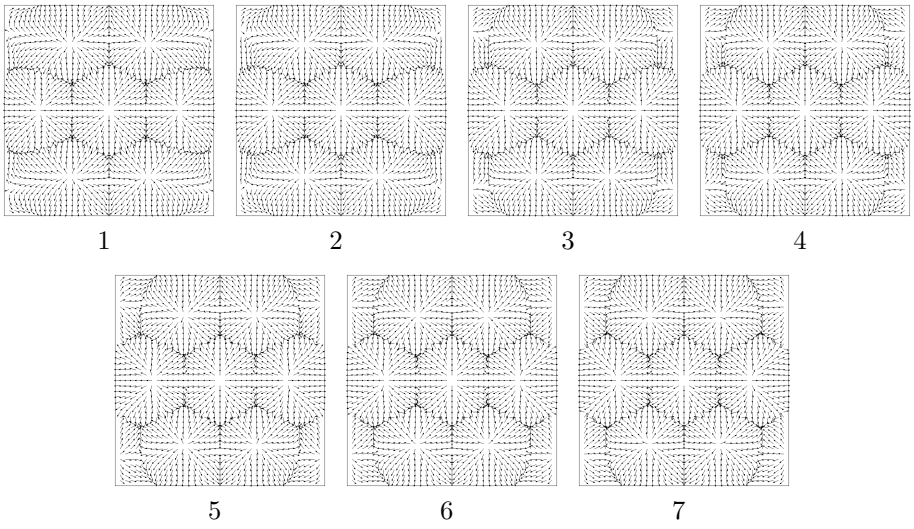
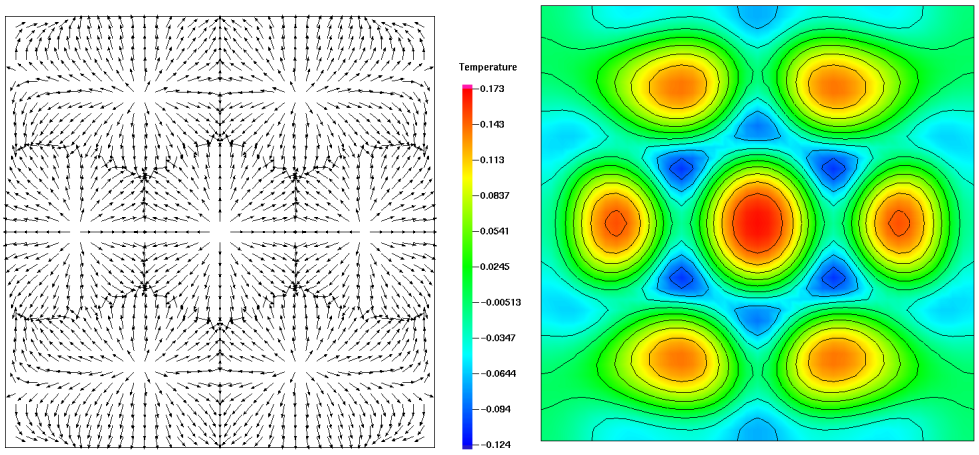
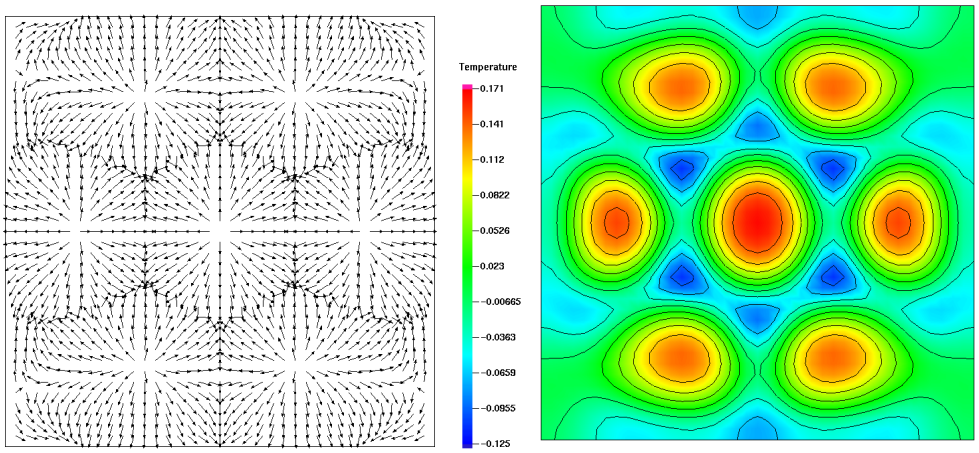


Figure 5.116: Velocity vectors for solutions on the branch depicted in Fig. 5.115.



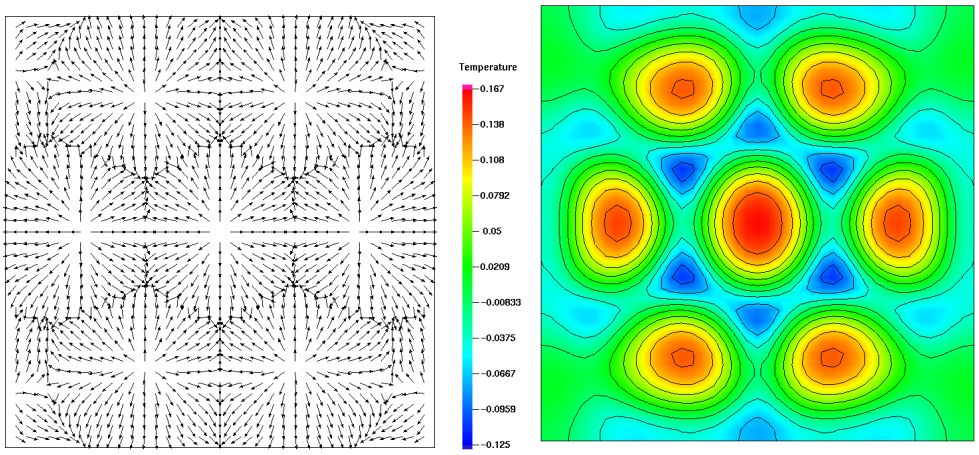
Velocity Vectors Temperature Contours

Figure 5.117: Solution 1 on the branch depicted in Fig. 5.115.



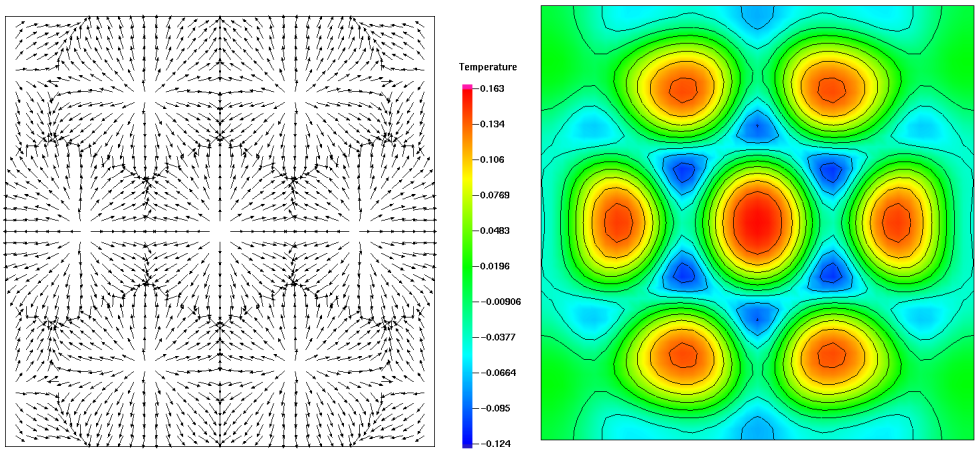
Velocity Vectors Temperature Contours

Figure 5.118: Solution 2 on the branch depicted in Fig. 5.115.



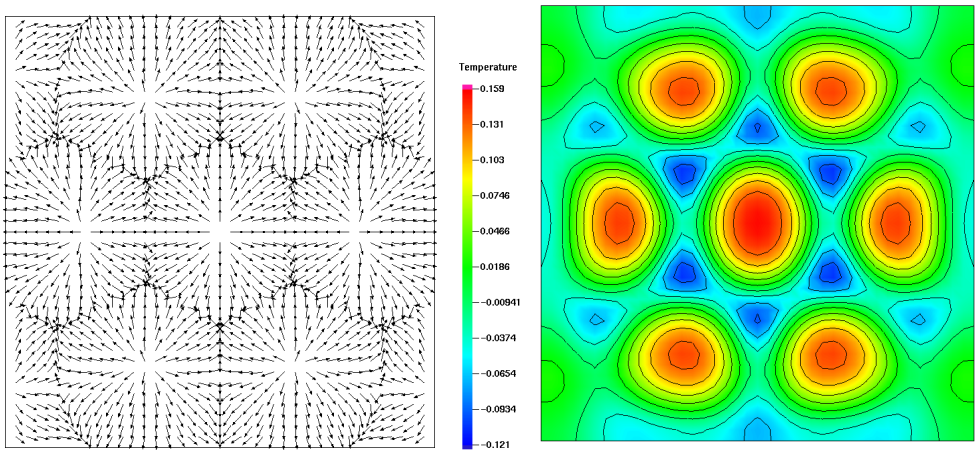
Velocity Vectors Temperature Contours

Figure 5.119: Solution 3 on the branch depicted in Fig. 5.115.



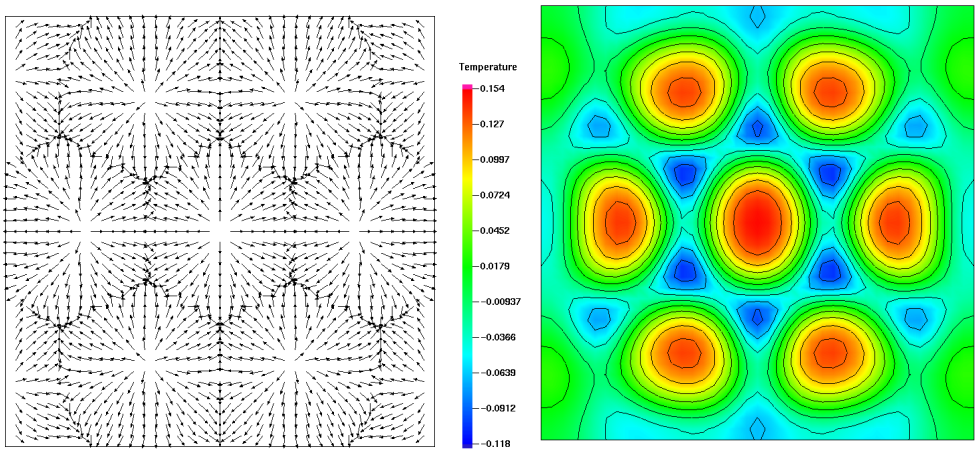
Velocity Vectors Temperature Contours

Figure 5.120: Solution 4 on the branch depicted in Fig. 5.115.



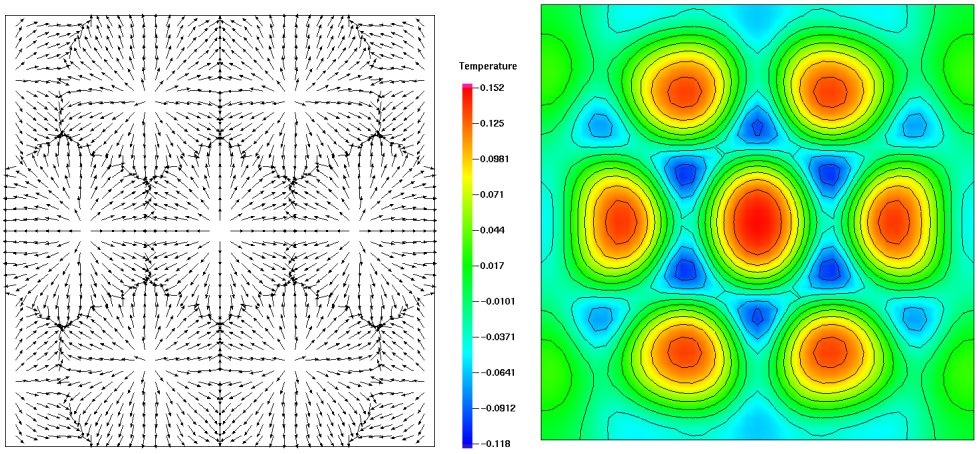
Velocity Vectors Temperature Contours

Figure 5.121: Solution 5 on the branch depicted in Fig. 5.115.



Velocity Vectors Temperature Contours

Figure 5.122: Solution 6 on the branch depicted in Fig. 5.115.



Velocity Vectors Temperature Contours

Figure 5.123: Solution 7 on the branch depicted in Fig. 5.115.

5.4.3 Eight-Cell Symmetry-Breaking Branches

This branch shown in Fig. 5.124 depicts the transition (with hysteresis) between 8-cell configurations with different symmetries and Nu values. The symmetric eight cell solutions (with representative examples marked as numbers 1 and 2 in Fig. 5.124) become linearly-unstable around solution 3, at which point a branch of asymmetric eight-cell configurations characterized by solutions 3–6 become stable. Numerically computed eigenvalues for the lower, symmetric branch are presented in Table 5.14, while those for the first symmetry-breaking branch are presented in Table 5.15. In this case, the vertical ($x = 0$) line of symmetry is broken, and the configuration "moves" to the left as the row of cells along the left boundary of the container is pushed inward by rising, warmer fluid.

These configurations are stable until aspect ratio $\Gamma \approx 11.5$ at which point a slight increase in the aspect ratio causes the branch with solutions 7 and 8 in Fig. 5.124) to become the stable one. This unsteady transition is accompanied by an increase in the Nusselt number, and is represented by the vertical arrow between solutions 6 and 7. The branch containing solutions 7 and 8 has broken both the $x = 0$ and $y = 0$ symmetries observed in configurations (e.g. solutions 1 and 2) on the original branch. Numerically computed eigenvalues for this branch are given in Table 5.16.

Solution 7 is recognizable as a counter-clockwise "twist" of solution 6. This type of configuration remains linearly stable up until the turning point at $\Gamma \approx 11.67$, at which point a slight increase in the aspect ratio will cause the

solution to jump down to the lower, symmetric branch which is characterized by solution 9 in Fig. 5.124. This jump back down to the symmetric branch is denoted by a downward vertical arrow.

Comparing solutions 1 and 9, we observe that they are both symmetric, eight cell solutions, but solution 9 occurs at a much higher aspect ratio than solution 1 ($\Gamma \approx 12$ vs. $\Gamma \approx 10.2$) and has a lower Nu value. If we consider the reverse situation, in which the aspect ratio is slowly decreased from $\Gamma = 12$, a different solution path in which the upper branch (with two broken symmetries) is stable to around $\Gamma \approx 11.2$ would be followed instead. This example shows how multiple symmetry-breaking branches can combine before returning to the stable, symmetric branch at a higher aspect ratio.

Detailed surface velocity and temperature contour plots are presented in Figs. 5.126–5.134.

Table 5.14: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment consisting of stable and unstable symmetric 8-cell solutions (the lowest branch in Fig. 5.124, which includes solutions 1, 2, and 9). Complex-conjugate eigenpairs are presented together. The stable solution branch is denoted by the solid line in Fig. 5.124.

Γ	σ_i		
10.600	-0.433	-2.797	-13.061
10.700	-0.921	1.197	-12.162
11.000	4.088	4.579	-8.775
11.500	2.103	-4.365	5.710
11.600	0.555	-4.024	4.224
11.700	-1.336	2.064	-3.696
11.800	-0.832	-3.291	-3.450
11.900	-2.782	$-5.319 \pm 1.900i$	
12.300	-1.425	-7.460	$-10.291 \pm 4.844i$

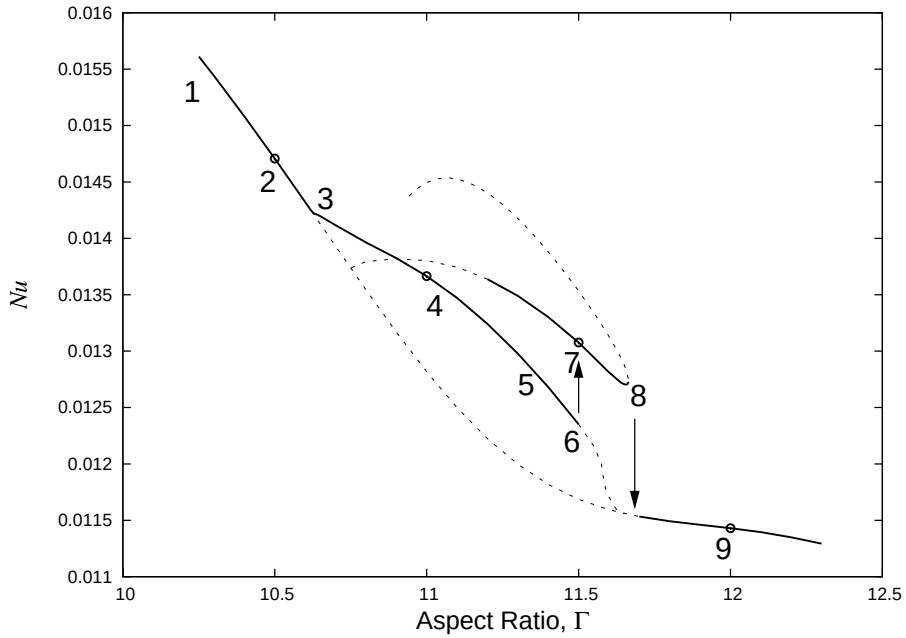


Figure 5.124: Computed solution branches 8-cell transition. The numbers correspond to solutions discussed in greater detail in the text, and in Figs. 5.125. Based on linear stability analysis of the steady solutions, branch segments consisting of stable steady states are marked with solid lines while unstable segments are marked with dashed lines. The arrows depict the (unsteady) transition between stable branches which would occur if the aspect ratio is slowly increased. Also shown, as circular markers, are time-accurate steady state solutions obtained from the unsteady equations which were used as starting points for computing this branch.

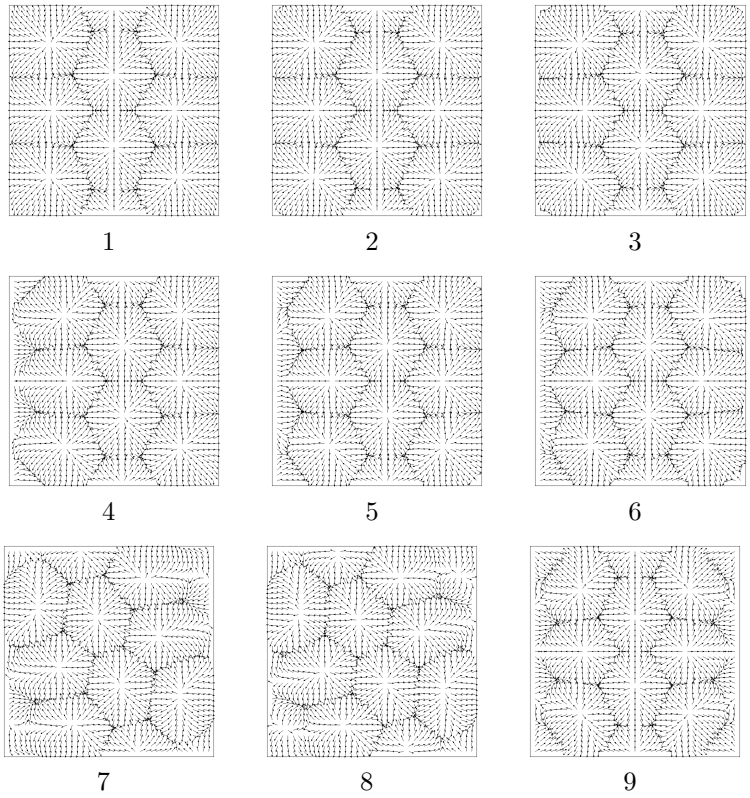


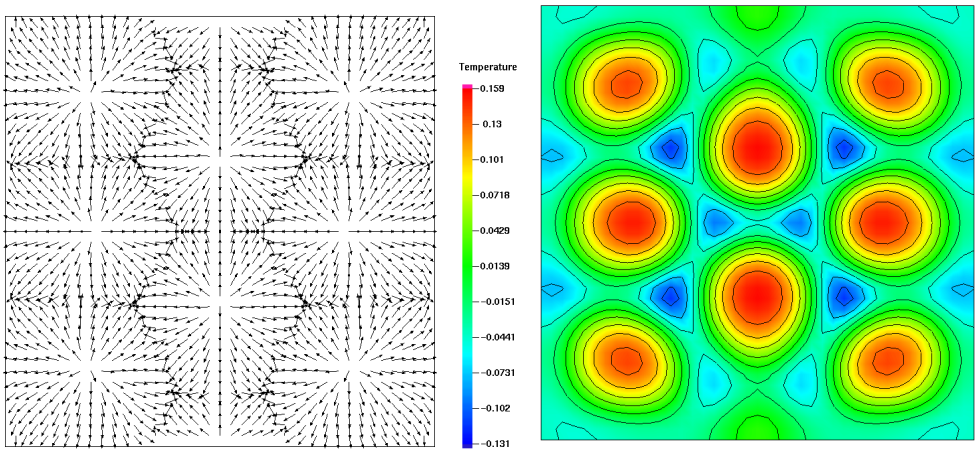
Figure 5.125: Velocity vectors for solutions on the branch depicted in Fig. 5.124.

Table 5.15: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment consisting of 8-cell solutions 3–6 shown in Fig. 5.124. Complex-conjugate eigenpairs are presented together. Stable solutions are denoted by solid lines in Fig. 5.124.

Γ	σ_i		
10.600	-0.433	-2.797	-13.062
10.630	-0.147	-2.217	-12.833
10.700	-1.730	-2.430	-12.130
10.900	-4.416	-6.613	-8.420
11.100	$-6.063 \pm 1.700i$	-8.979	
11.300	$-4.618 \pm 1.088i$	-7.746	
11.500	-0.263	-3.365	-4.178
11.550	1.200	-1.689	-3.962
11.600	-0.742	4.012	-4.128

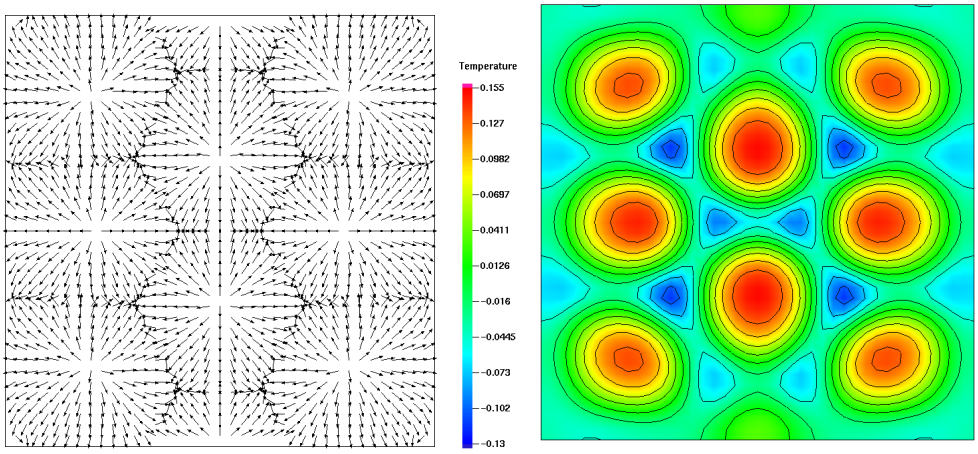
Table 5.16: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the branch segment consisting of the rotated 8-cell solutions 7 and 8 shown in Fig. 5.124.

Γ	σ_i		
11.000	1.303	-8.243	-8.461
11.100	0.802	-9.480	-10.441
11.200	-0.090	-10.025	-12.532
11.300	-1.219	-9.707	-15.125
11.600	-3.976	-8.295	$-12.384 \pm 12.293i$
11.629	-10.392	12.065	12.262



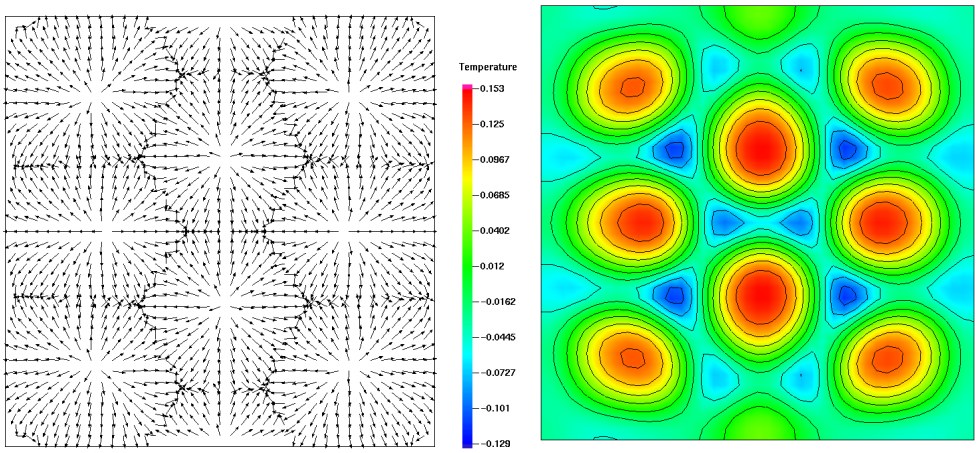
Velocity Vectors Temperature Contours

Figure 5.126: Solution 1 on the branch depicted in Fig. 5.124.



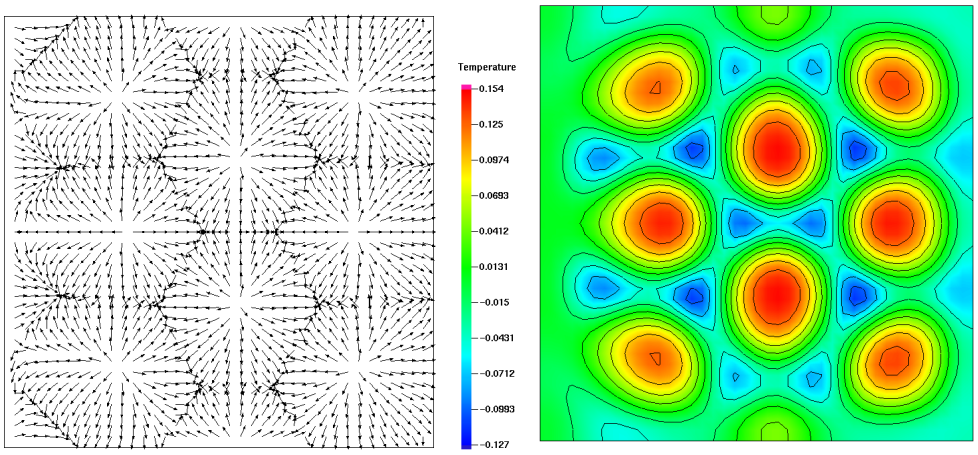
Velocity Vectors Temperature Contours

Figure 5.127: Solution 2 on the branch depicted in Fig. 5.124.



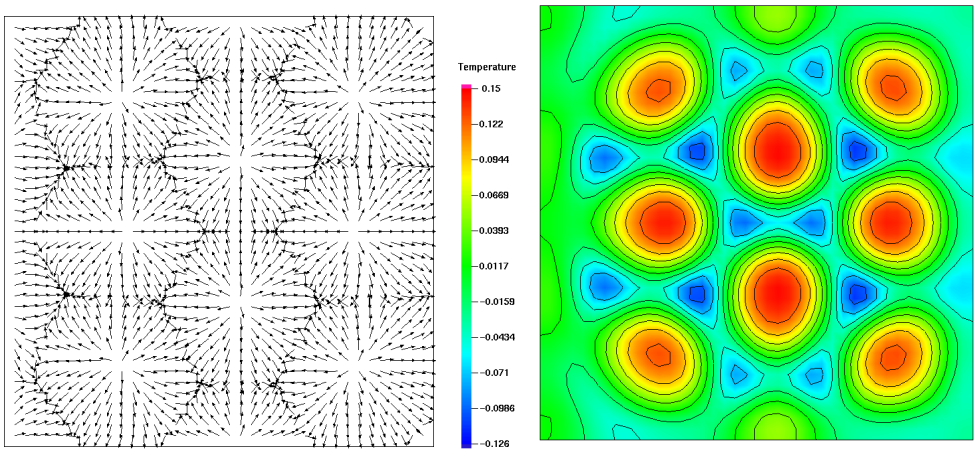
Velocity Vectors Temperature Contours

Figure 5.128: Solution 3 on the branch depicted in Fig. 5.124.



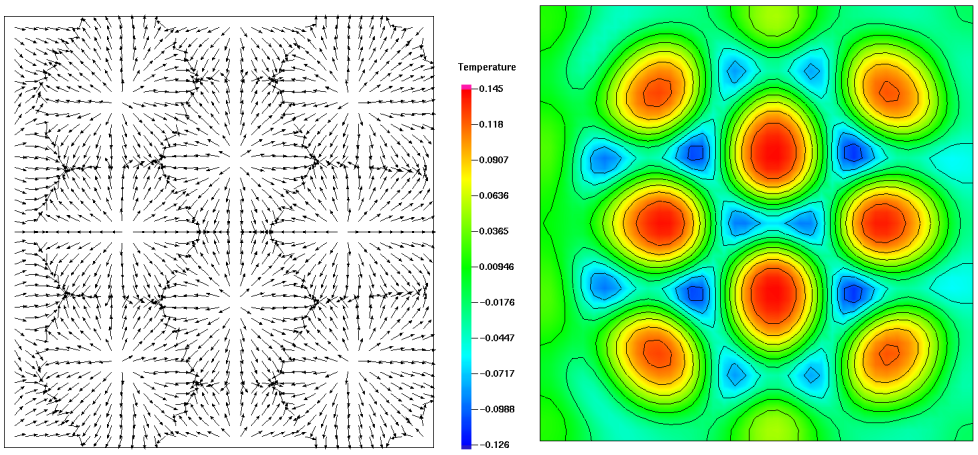
Velocity Vectors Temperature Contours

Figure 5.129: Solution 4 on the branch depicted in Fig. 5.124.



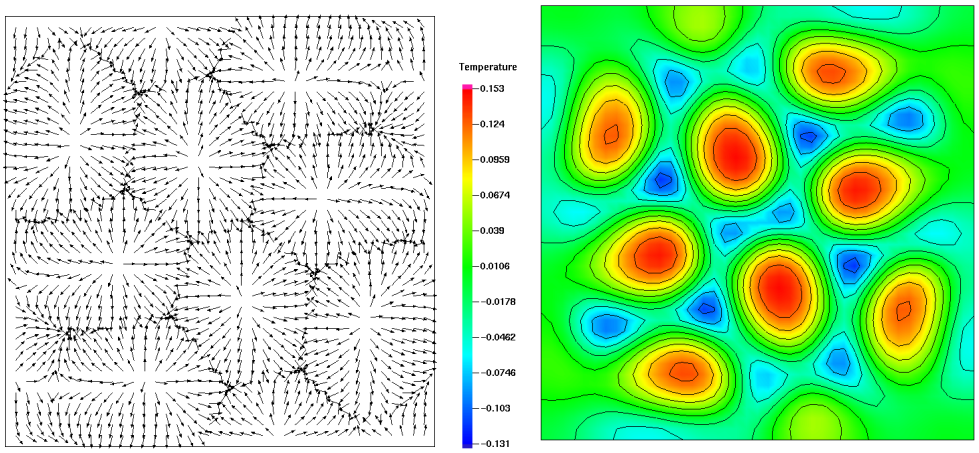
Velocity Vectors Temperature Contours

Figure 5.130: Solution 5 on the branch depicted in Fig. 5.124.



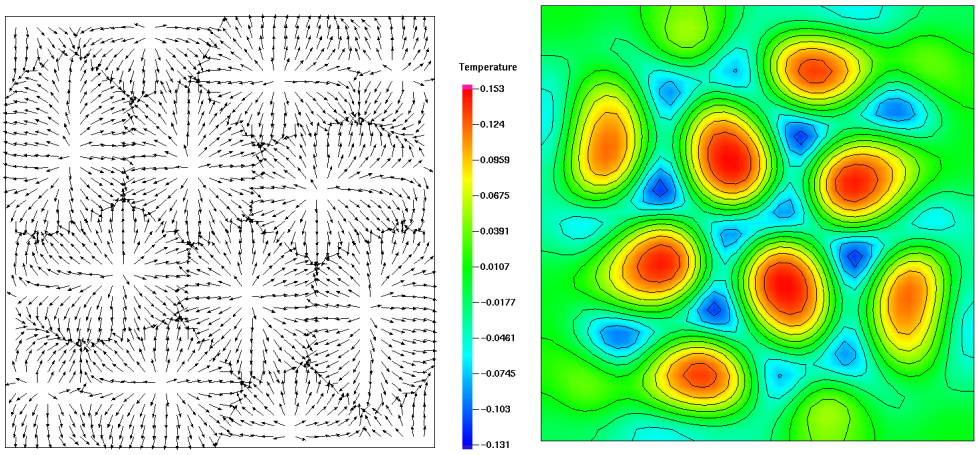
Velocity Vectors Temperature Contours

Figure 5.131: Solution 6 on the branch depicted in Fig. 5.124.



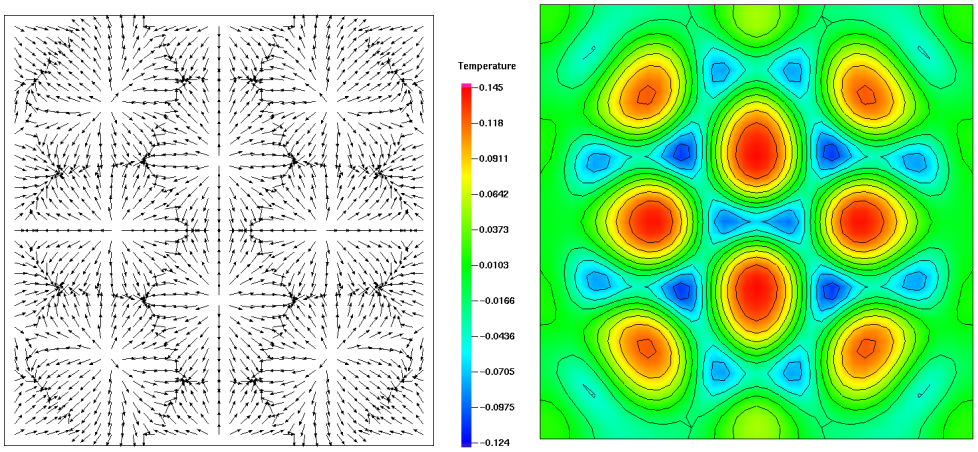
Velocity Vectors Temperature Contours

Figure 5.132: Solution 7 on the branch depicted in Fig. 5.124.



Velocity Vectors Temperature Contours

Figure 5.133: Solution 8 on the branch depicted in Fig. 5.124.



Velocity Vectors Temperature Contours

Figure 5.134: Solution 9 on the branch depicted in Fig. 5.124.

5.5 Summary of Aspect Ratio Continuation Results

In order to sum up the results of the preceding sections, we now present on a single plot, the solution branches for configurations having one through eight convection cells. In general, solutions with higher aspect ratios have more convection cells, and solutions with more convection cells have higher Nu values. The transition from configurations with n to $n+1$ cells usually occurs with hysteresis, and our simulations predict the existence of multiple, stable steady states for some values of the aspect ratio. These transitions are usually, but not always, accompanied by a large jump in Nu, even though the equations depend continuously on the aspect ratio, and stable branches are typically connected by unstable branches during the continuation procedure.

Our results also confirm that in all cases the presence of convection increases the rate of heat transfer through the fluid layer. The effects of container shape are known to be pronounced at small aspect ratios, and these results confirm that the convection cell configuration depend strongly on a single non-dimensional geometric parameter which we have called the aspect ratio. These results are not complete, since they do not adequately explain all of the experimental evidence which is available to date. In the next chapter, we expand on the present results by considering additional parametric studies in terms of the Marangoni number, Ma. As we will see, increasing the Ma can have a profound effect on the shape of the cells and the resulting Nu.

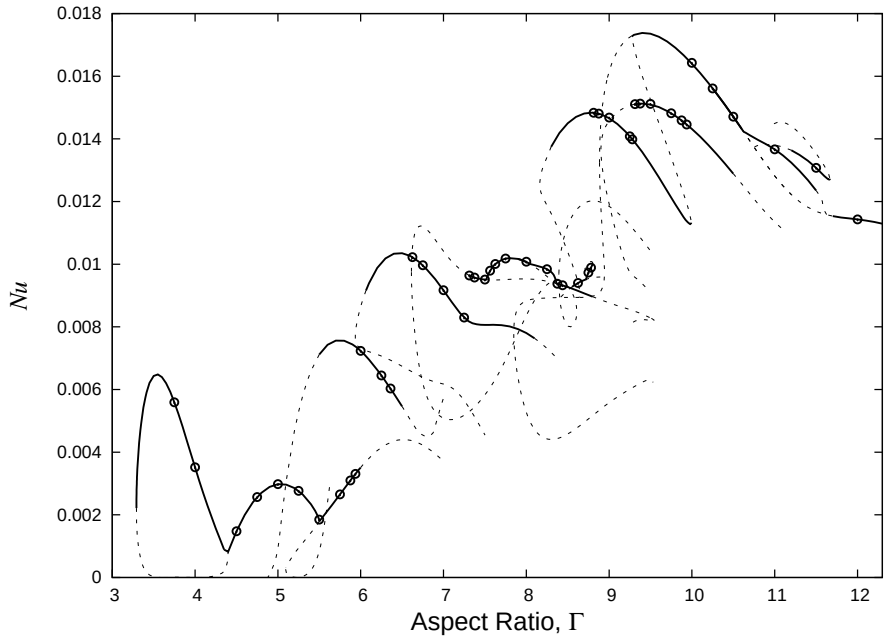


Figure 5.135: Solution branches computed via arclength continuation for $3 \leq \Gamma \leq 12$. Branch segments composed of linearly-stable solutions are shown with solid lines while branch segments composed of linearly-unstable (unstable to infinitesimal perturbations) solutions are depicted with dashed lines. The circular markers represent the time-accurate solutions shown originally in Fig. 5.1.

Chapter 6

RBM Results for Continuation in Marangoni Number

6.1 Two-cell Solutions: Ma continuation

While computing solution branches in aspect ratio space using pseudo-arclength continuation, we observed several symmetric two-cell solutions which have been discussed in detail in §5.2.2 and displayed in Figs. 5.16 and 5.17. These numerically-computed simulations slightly resemble the symmetric two-cell configuration reported by Koschmieder and Medale and Cerisier which is shown in Fig. 5.18. The experimental cells are much more triangular in shape and much less rounded than the computationally-obtained configuration. Furthermore, with increasing aspect ratio, the two-cell configuration elongates and eventually becomes unstable, rather than more closely resembling the experimental image.

6.1.1 Continuation in Ma for $\Gamma = 5.7$

To explore the behavior further, continuation in Marangoni number was applied with the aspect ratio $\Gamma = 5.7$ now fixed. The value 5.7 was selected since it corresponded to the solution closest in aspect ratio to the experimentally-reported value of 5.68. The associated solution branch is shown

in Fig. 6.1. The velocity vector plots of solutions corresponding to the points marked 1–9 in Fig. 6.1 are shown in more detail in Fig. 6.2. On this solution branch, we observe the convection cells becoming less rounded (and hence, more triangular) in shape with increasing Ma.

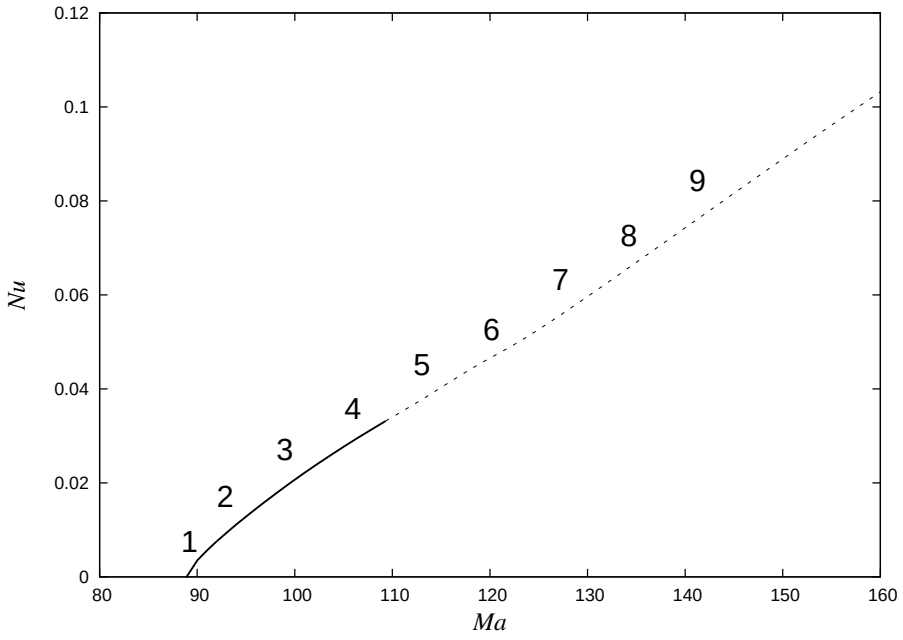


Figure 6.1: Solution branch of symmetric two-cell solutions which resemble an experimental configuration reported by Koschmieder. This configuration starts off stable and loses stability with increasing Ma.

According to the eigenvalue computations shown in Table 6.1 this two-cell solution becomes unstable somewhere after solution 4, around $Ma = 110$. We note that the unstable eigenvalues in this case have very large imaginary

part (signaling a Hopf bifurcation [184]) which makes them difficult to locate in a scheme designed to find eigenvalues with magnitude nearest to zero. Such a scheme will preferentially detect stable eigenvalues (with negative real part) whose imaginary parts are small relative to the imaginary part of the unstable eigenvalue. It may be possible to rectify this situation by choosing a spectral transformation with imaginary shift. (See §4.5.1 for details about the spectral shift employed in this work.) However such a transformation requires complex matrix calculations as well, which is beyond the scope of the current work and a feature not currently supported by our eigenvalue computation software.

Table 6.1: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the increasing-Ma branch depicted in Fig. 6.1. The eigenvalues predict that the two-cell configuration is stable up to approximately $Ma = 110$, beyond which point this configuration becomes stable.

Ma	σ_i		
93.000	-5.683	$-11.281 \pm 9.757i$	
100.912	$-9.978 \pm 24.720i$	-13.853	-18.707
102.307	$-16.622 \pm 7.812i$	-22.253	
103.704	$-15.303 \pm 13.853i$	-22.127	
106.508	$-8.975 \pm 21.993i$		
109.325	$0.257 \pm 32.530i$	$-2.094 \pm 27.348i$	-21.477
110.740	$1.194 \pm 29.716i$	-21.080	-24.673
120.783	-10.075	$14.832 \pm 53.132i$	-18.219

Solutions corresponding to points 7, 8, and 9 most strongly resemble the experimental result, though we found them to be unstable at this aspect

ratio. At this time, it is not clear whether more accurate eigenvalue computations would eventually show these solutions to be linearly stable, or if they would be stable at a slightly different aspect ratio. As an aside, we note that Lehoucq and Salinger [118] demonstrated the converse situation in a large-scale simulation of a disk reactor used for chemical vapor deposition. A coarse-grid eigenvalue computation initially predicted a stable solution (a rightmost eigenvalue with negative real part) but under mesh refinement, the rightmost eigenvalue was eventually shown to have a positive real part, thereby signaling an unstable solution.

Increasing Ma in this case (and those which follow) has the effect of "stretching" the existing dominant convection cells in the configuration to more completely cover the domain. For this particular case, we did not observe convection cell splitting or symmetry-breaking while increasing Ma, and the branch structure is much simpler here than for the aspect ratio continuation cases discussed previously.

Additional images showing the surface temperature contours and velocity fields in detail are given in Figs. 6.3–6.11.

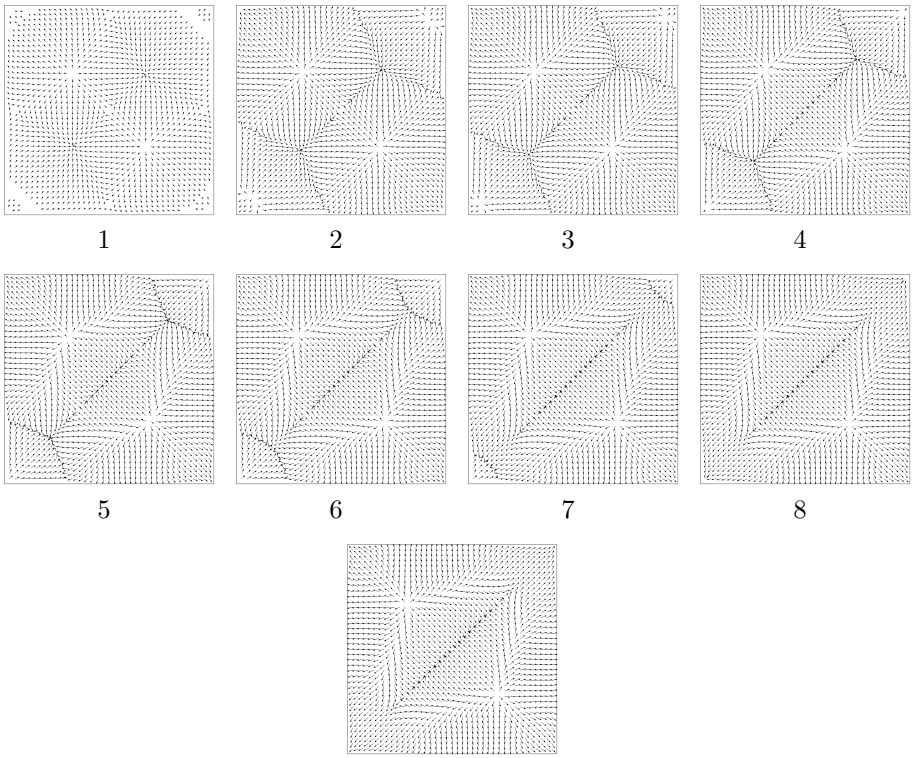
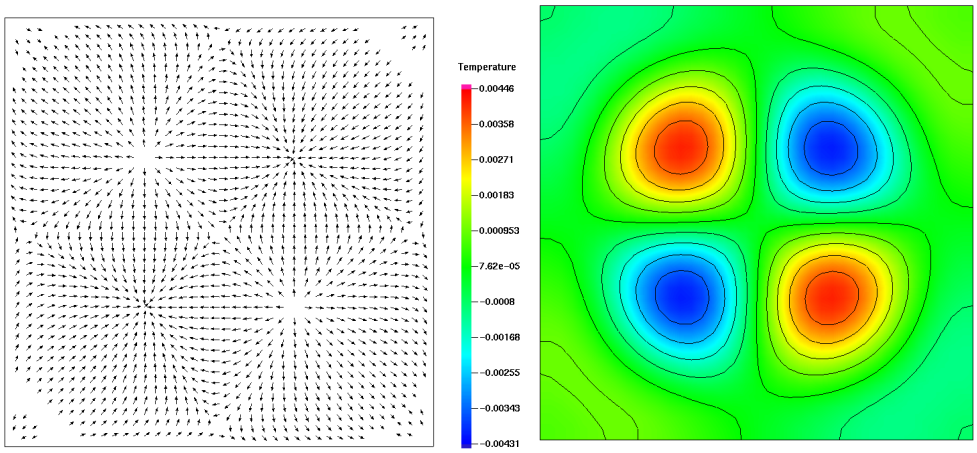
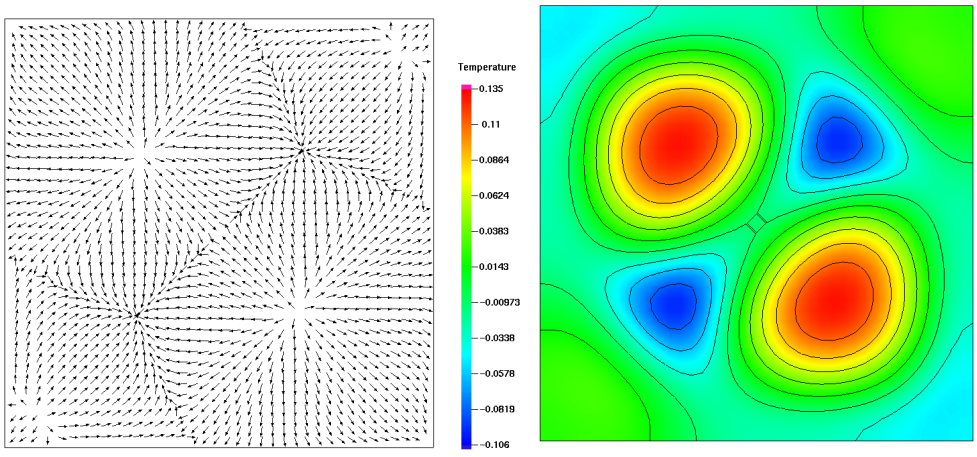


Figure 6.2: Velocity vectors for solutions on the branch depicted in Fig. 6.1.



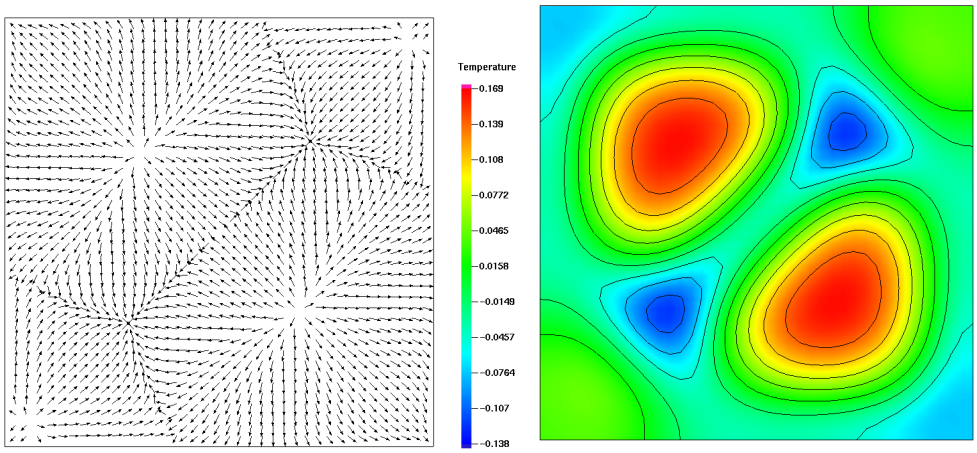
Velocity Vectors Temperature Contours

Figure 6.3: Solution 1 on the branch depicted in Fig. 6.1.



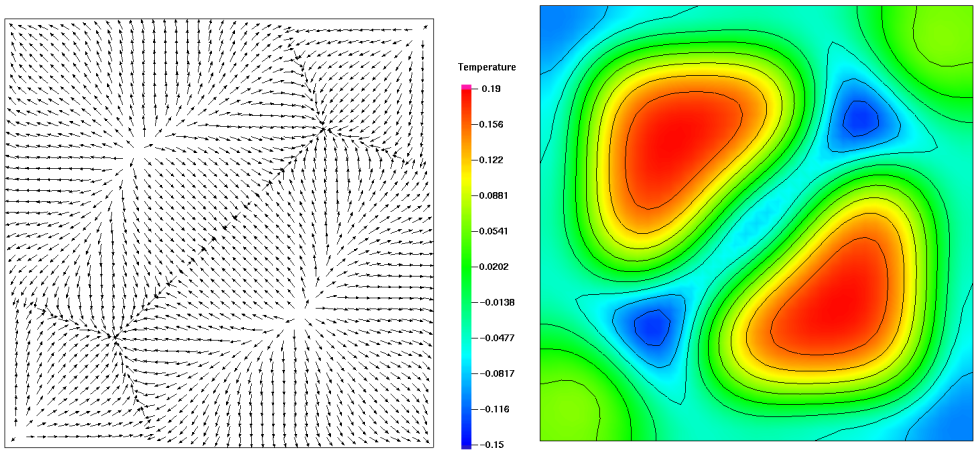
Velocity Vectors Temperature Contours

Figure 6.4: Solution 2 on the branch depicted in Fig. 6.1.



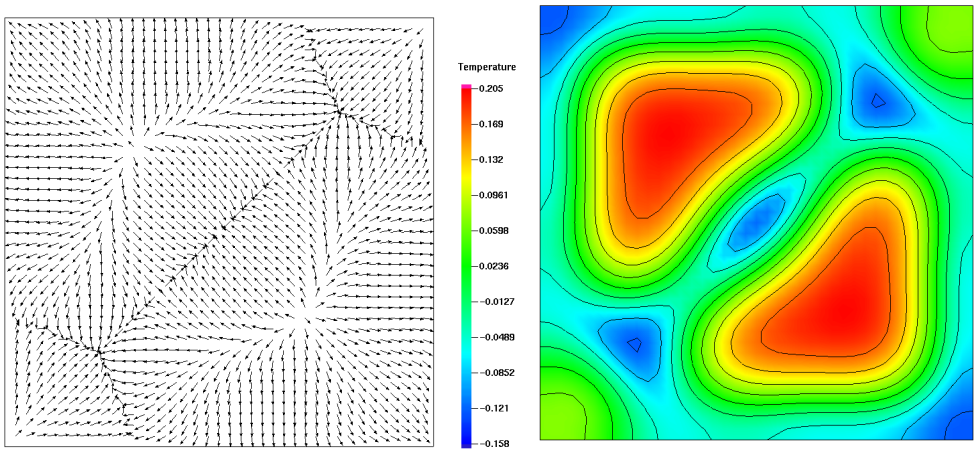
Velocity Vectors Temperature Contours

Figure 6.5: Solution 3 on the branch depicted in Fig. 6.1.



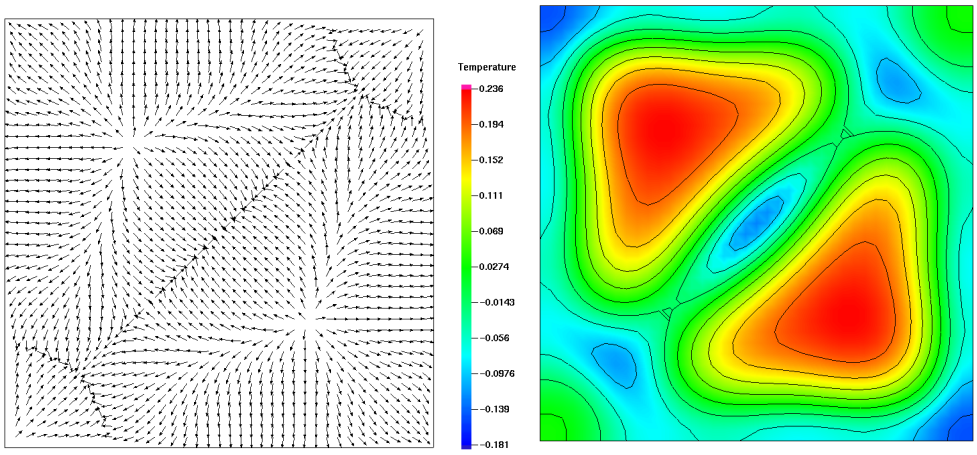
Velocity Vectors Temperature Contours

Figure 6.6: Solution 4 on the branch depicted in Fig. 6.1.



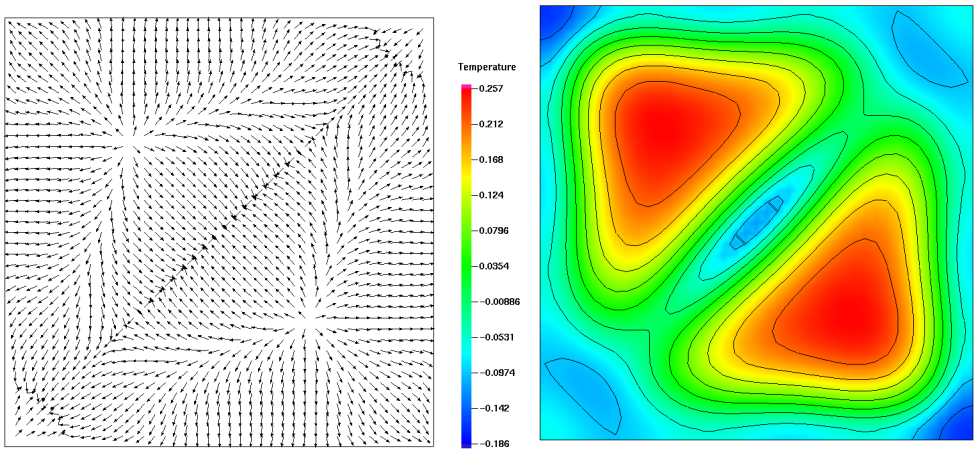
Velocity Vectors Temperature Contours

Figure 6.7: Solution 5 on the branch depicted in Fig. 6.1.



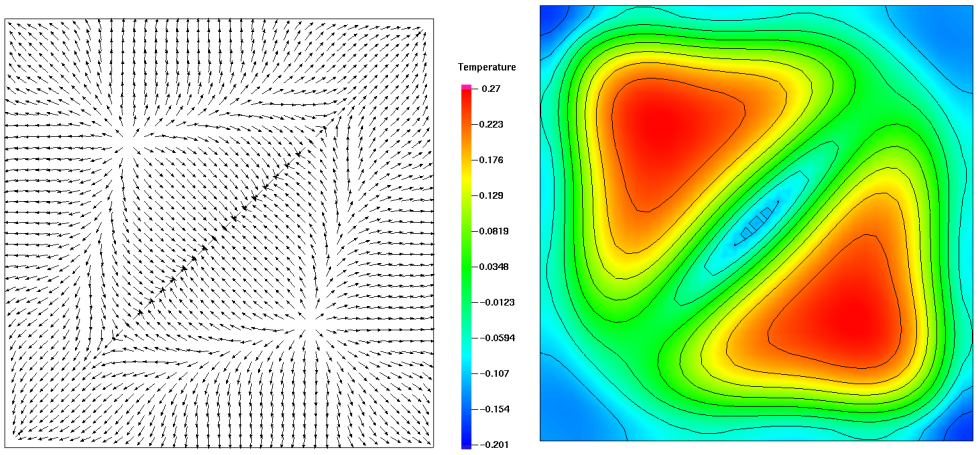
Velocity Vectors Temperature Contours

Figure 6.8: Solution 6 on the branch depicted in Fig. 6.1.



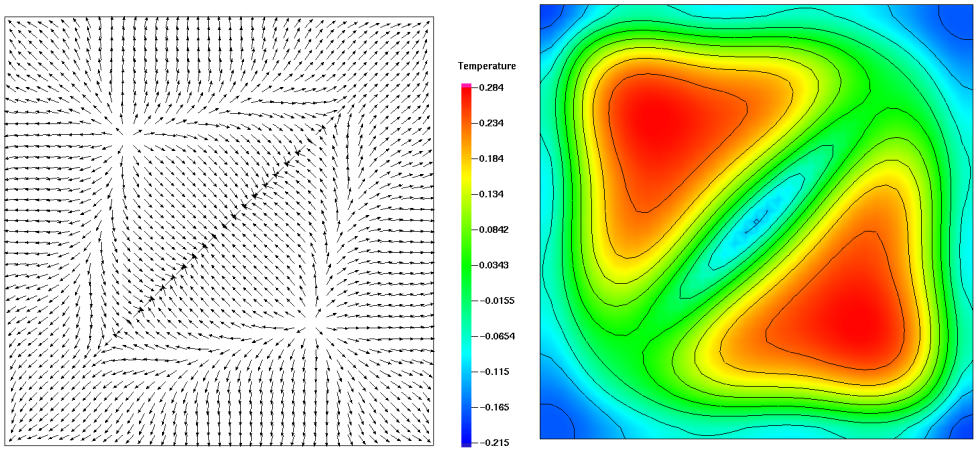
Velocity Vectors Temperature Contours

Figure 6.9: Solution 7 on the branch depicted in Fig. 6.1.



Velocity Vectors Temperature Contours

Figure 6.10: Solution 8 on the branch depicted in Fig. 6.1.



Velocity Vectors Temperature Contours

Figure 6.11: Solution 9 on the branch depicted in Fig. 6.1.

6.2 Symmetric 2×2 Cell Solution

During the course of aspect ratio continuation, we observed an unstable symmetric 2×2 square convection cell solution which persisted over a range of aspect ratios. In Fig. 6.12, we show this unstable branch, which has a local maximum of Nu around $\Gamma \approx 6.436$. We used this value as a starting point for Ma continuation to determine if any linearly-stable 2×2 solutions could be found.

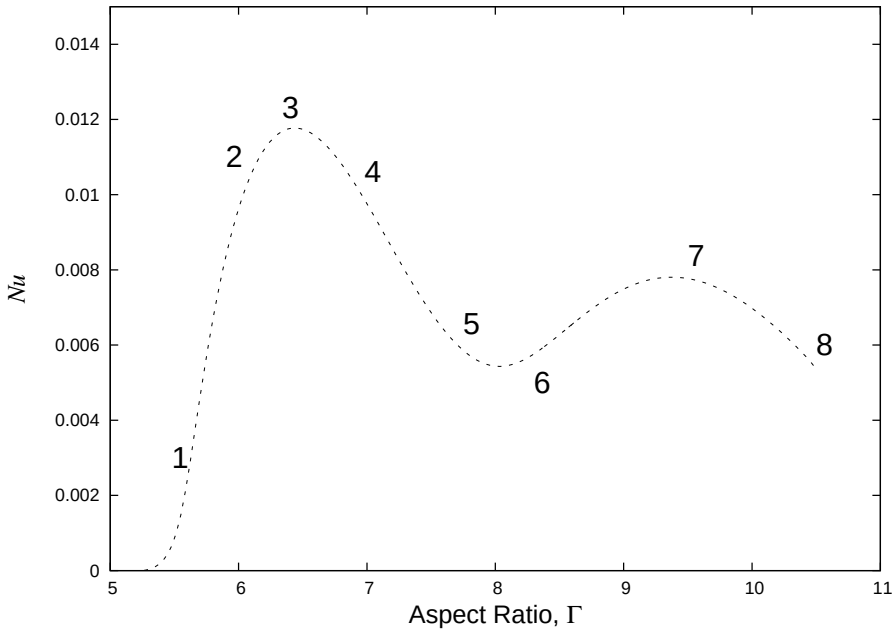


Figure 6.12: Branch of symmetric, unstable four-cell solutions computed via arclength continuation in Γ. We use a solution near point 3 ($\Gamma \approx 6.436$) as a starting point for continuation in Ma.

The symmetric 2×2 solution has been observed experimentally [113] (see Fig. 6.13) and thus we were surprised to find a similar solution that was unstable.) Detailed surface velocity fields for the unstable Γ-continuation branch of Fig. 6.12 are given in Fig. 6.14. We observe that solutions 1–4 in Fig. 6.14 are "about right" in that they resemble the experimental image, whereas solutions 5–8 do not. This observation also influenced our decision on where to begin the Ma continuation.

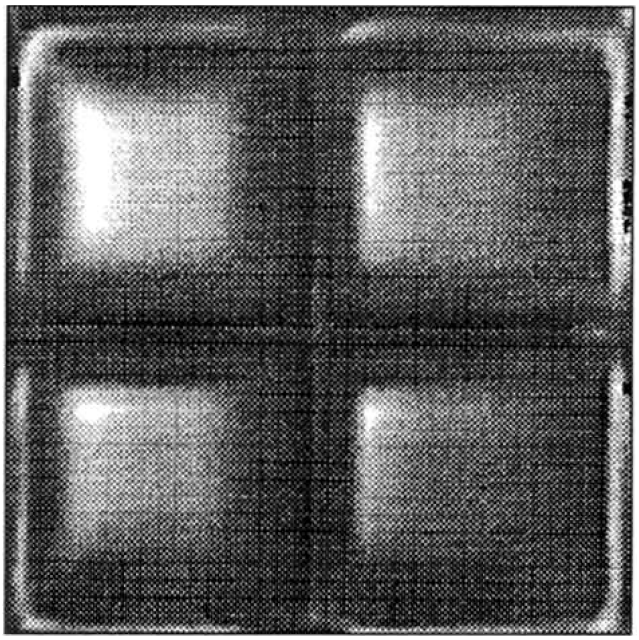


Figure 6.13: Experimentally-observed 4-cell configuration by Koschmieder [113] as well as Medale and Cerisier [126]. Koschmieder reports the values $Ra = 38$, $Ma = 93$, and $\Gamma = 6.36$ for this case.

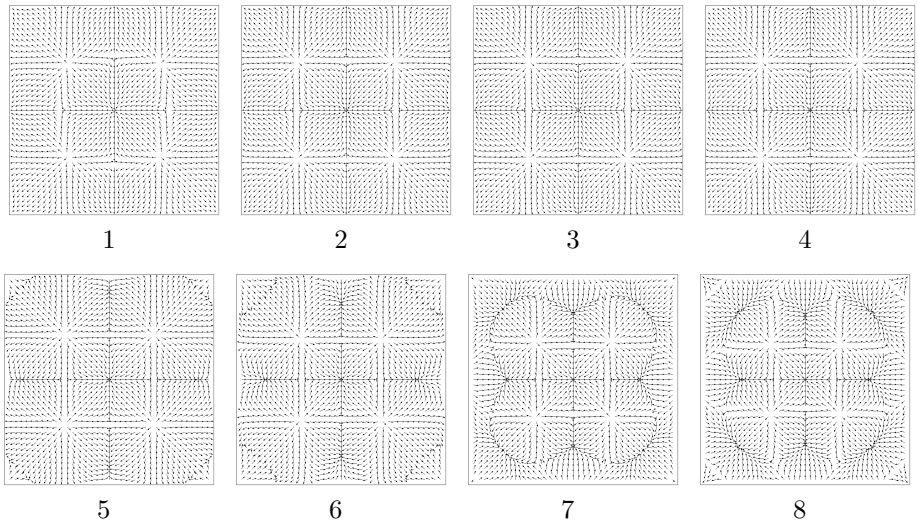


Figure 6.14: Velocity vectors for unstable, symmetric four-cell solutions on the branch depicted in Fig. 6.12.

6.2.1 Continuation in Ma for $\Gamma = 6.436$

The branch computed via pseudo-arclength continuation in Ma with a constant value of $\Gamma = 6.436$ is shown in Fig. 6.15. This branch is similar to others we have observed for increasing Ma. Nu increases monotonically with Ma, and there are no turning or hysteresis points present, unlike the situation observed in the Γ-continuation plots. The solutions for the points marked with the numbers 1–8 in Fig. 6.15 are shown in more detail in Fig. 6.16.

The numerical eigenvalue calculations, shown in Table 6.2, suggest that the symmetric 2×2-cell solution is unstable for $Ma \leq 96$, but becomes stable after this point. These findings are in reasonable agreement with the experi-

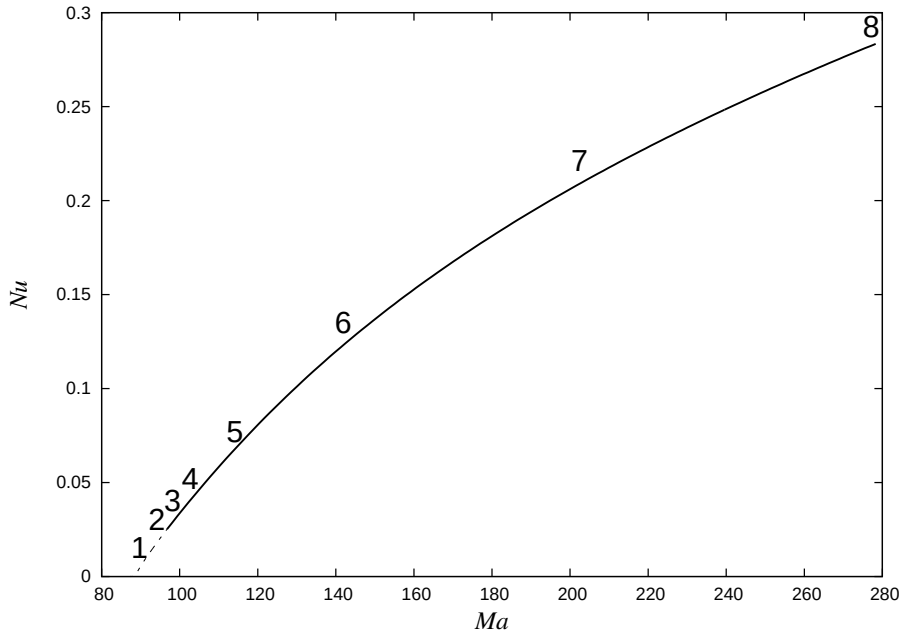


Figure 6.15: Solution branch computed via arclength continuation of 2×2 square-cell solutions. Koschmieder observed such a configuration in his experiments at parameter values $\Gamma = 6.36$, $Ma = 93$, $Ra = 38$.

mental observations of Koschmieder for this configuration. We also note that the symmetric four-cell solution remained the preferred configuration up to the highest Ma number ($Ma \approx 278.265$) computed, much higher than what was safely possible experimentally.

Additional images showing detailed surface temperature contours and the surface velocity field are given in Figs. 6.17–6.24

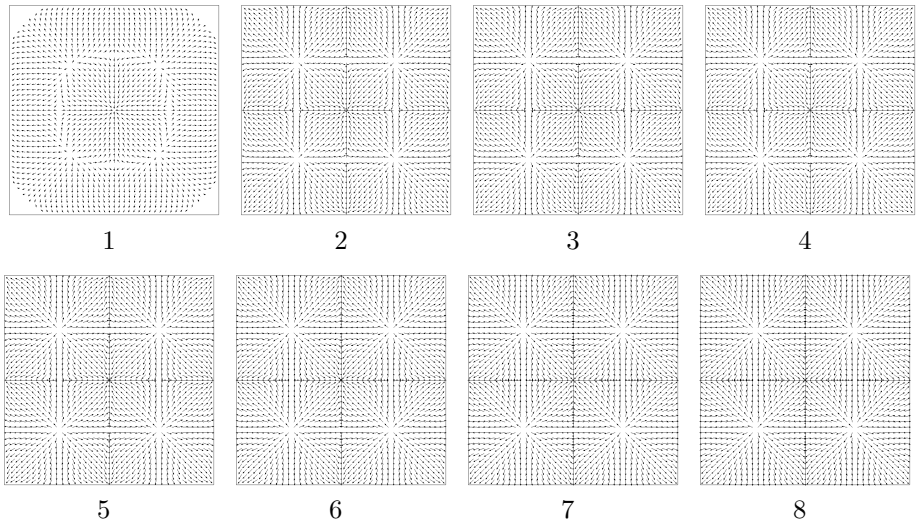
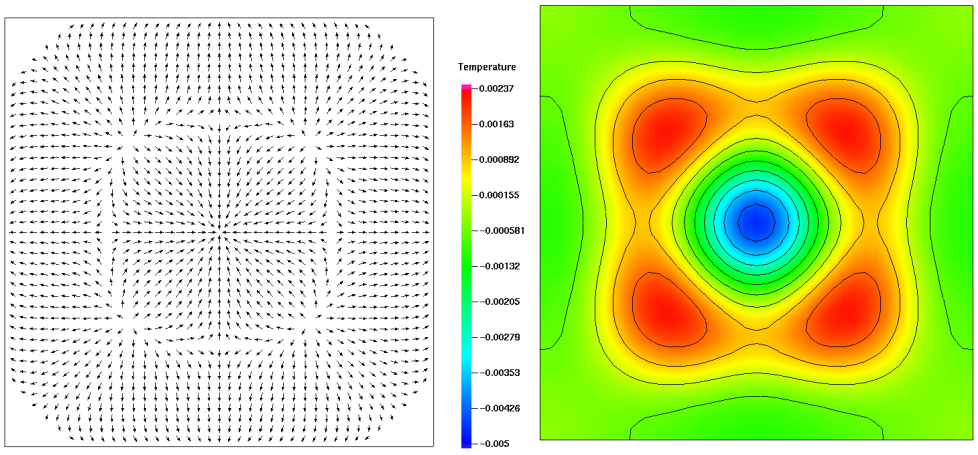


Figure 6.16: Velocity vectors for solutions on the Ma continuation branch depicted in Fig. 6.15.

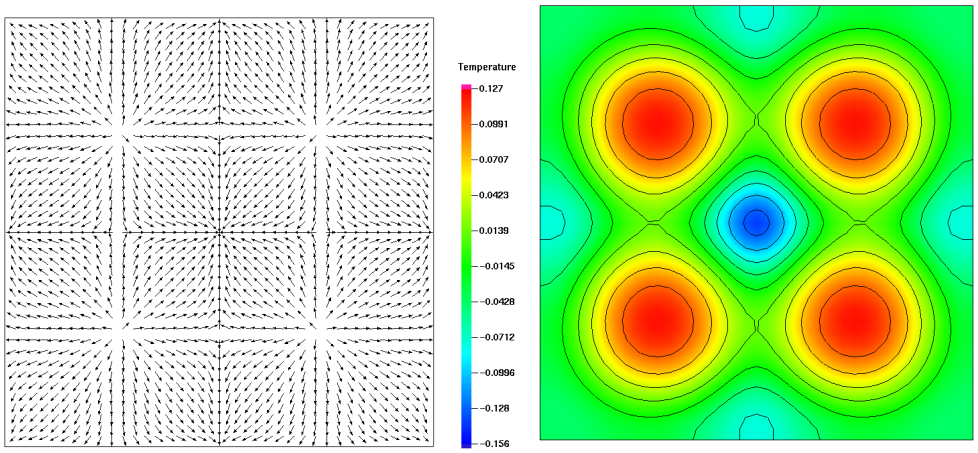


Velocity Vectors Temperature Contours

Figure 6.17: Solution 1 on the branch depicted in Fig. 6.15.

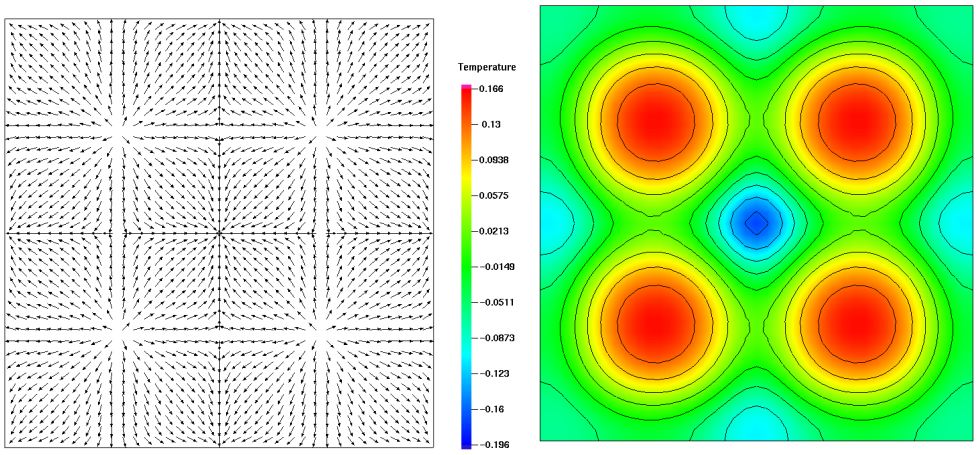
Table 6.2: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the increasing-Ma branch depicted in Fig. 6.15. The eigenvalues predict that the symmetric four-cell configuration is unstable up to approximately $Ma = 96$, beyond which point this configuration becomes stable. This case suggests that increasing the Ma may have the effect of "stabilizing" cellular configurations which would otherwise be unstable to perturbations.

Ma	σ_i		
89.614	−3.816	−6.059	7.538
91.013	−6.09193	−6.09192	6.304
93.000	3.786	−12.2319	−12.232
93.989	2.554	−14.321	
95.375	0.883		
96.748	−0.722		
105.038	−9.927	−22.093	−23.241
111.963	−17.256	−20.298	−26.171
200.314	−6.514	−39.875	−44.1454
278.265	−7.347	−32.700	−38.698



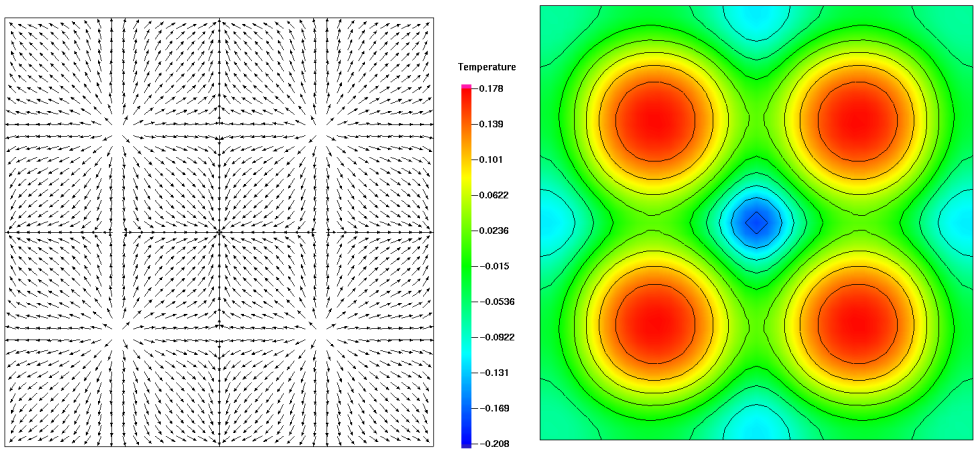
Velocity Vectors Temperature Contours

Figure 6.18: Solution 2 on the branch depicted in Fig. 6.15.



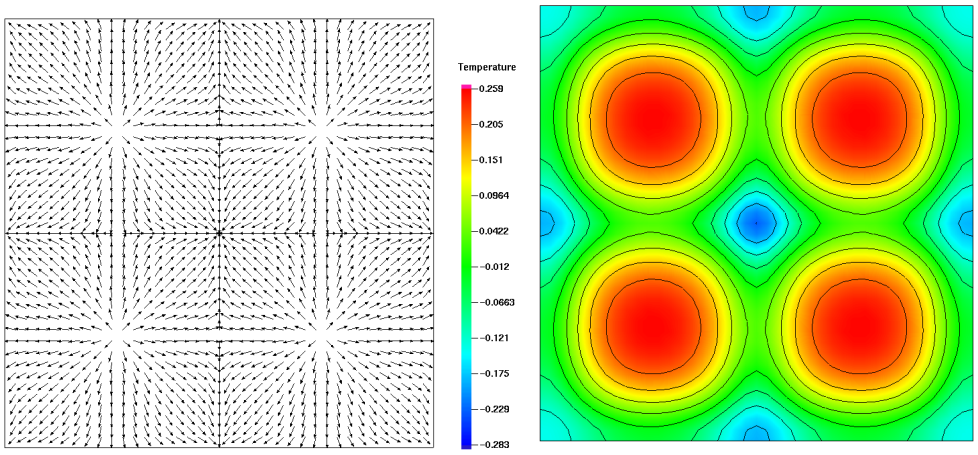
Velocity Vectors Temperature Contours

Figure 6.19: Solution 3 on the branch depicted in Fig. 6.15.



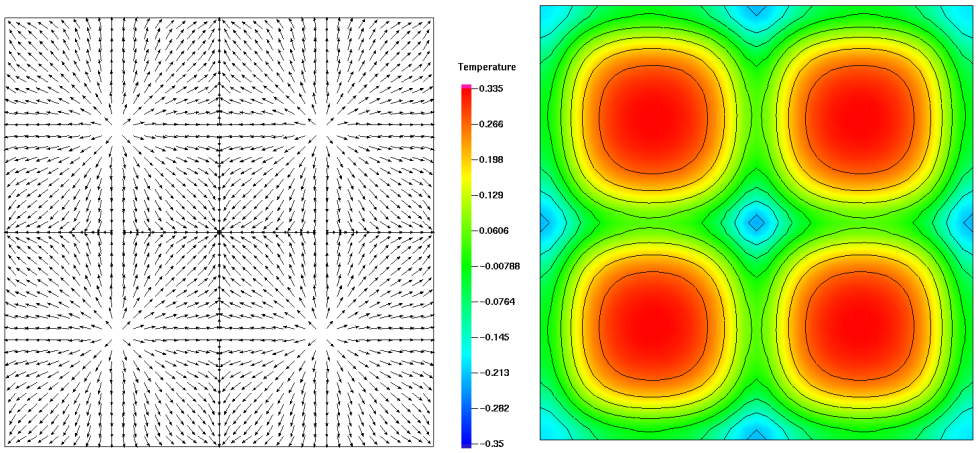
Velocity Vectors Temperature Contours

Figure 6.20: Solution 4 on the branch depicted in Fig. 6.15.



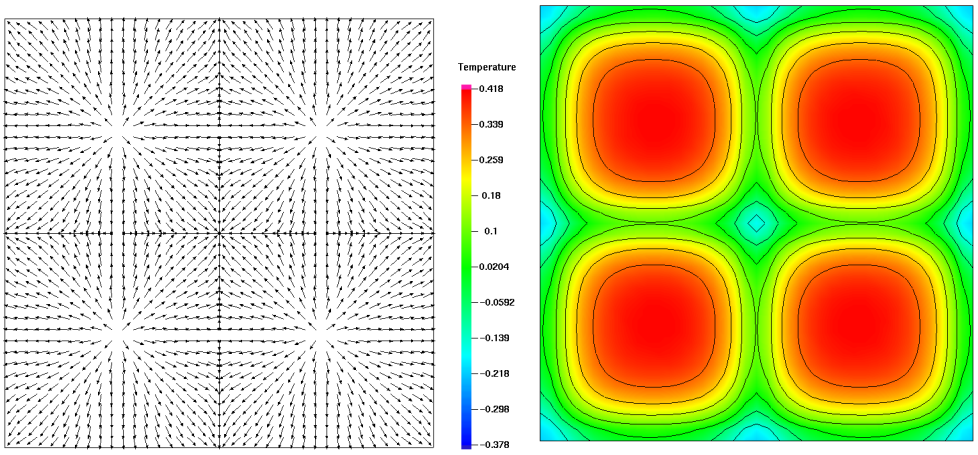
Velocity Vectors Temperature Contours

Figure 6.21: Solution 5 on the branch depicted in Fig. 6.15.



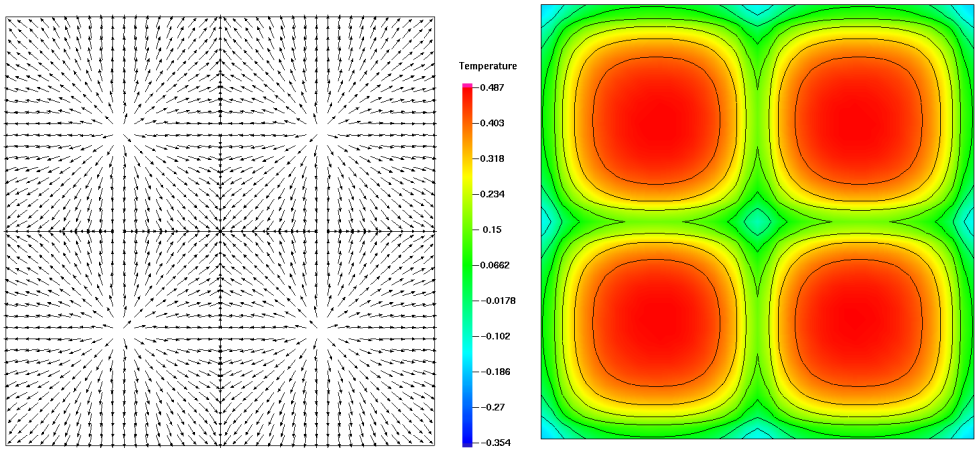
Velocity Vectors Temperature Contours

Figure 6.22: Solution 6 on the branch depicted in Fig. 6.15.



Velocity Vectors Temperature Contours

Figure 6.23: Solution 7 on the branch depicted in Fig. 6.15.



Velocity Vectors Temperature Contours

Figure 6.24: Solution 8 on the branch depicted in Fig. 6.15.

6.3 Symmetric 3×3 Cell Solution

Regarding the symmetric 3×3 square-cell pattern, Koschmieder [113] states:

> "An easy way for the fluid to fill the square container with equal cells would have been the nine square-cell pattern. With [Aspect Ratio] $A = 8.75$ we did not, however, obtain nine square cells, but rather an eight-cell solution ... We assume that a nine-cellular square pattern will eventually appear with larger aspect ratios, which we could not realize because the critical temperature differences would then have exceeded safe operating conditions in our apparatus ($\Delta T \approx 40°\text{C}$)."

In our calculations, we do in fact obtain symmetric (or nearly so) nine square-cell patterns for several different aspect ratios using our other "standard" parametric values of $Ra = 30$, $Ma = 92$, $Pr = 880$, and $Bi = 0.2$. Subsequent linear stability analysis, however, predicted that these solutions (located on the Γ-continuation branch in Fig. 6.25 and shown in detail in Fig. 6.26) were linearly unstable. We may conjecture that Koschmieder did not observe the nine-cell configuration because it is linearly unstable, as these calculations show. However, these results are still not conclusive because there may in fact be a stable nearby (in one of the other problem parameters) nine-cell solution.

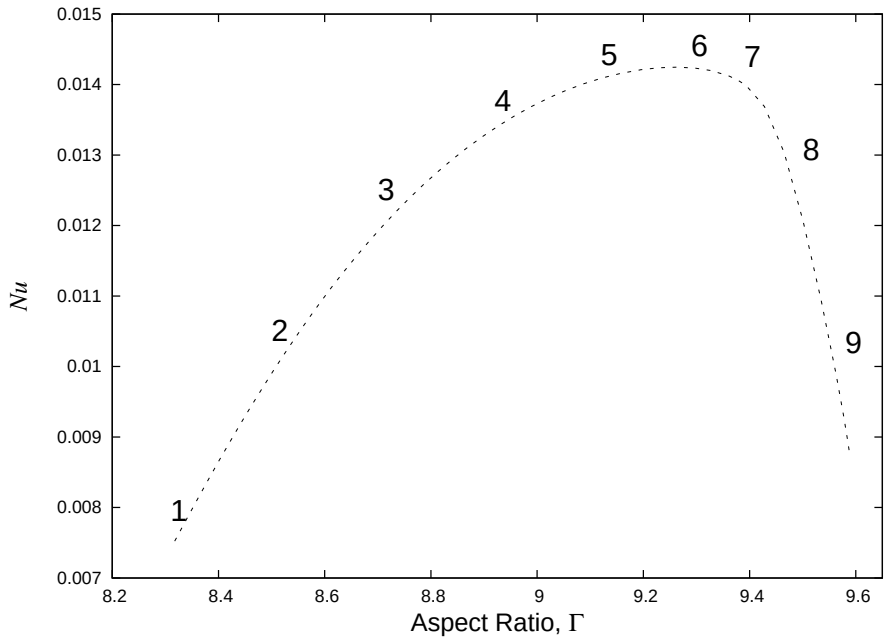


Figure 6.25: Solution branches computed via arclength continuation of *unstable* 3×3 square-cell solutions. Koschmieder conjectured that such a solution would exist for aspect ratios $\Gamma > 8.75$ but limitations of the experimental apparatus may have prevented this configuration from being observed.

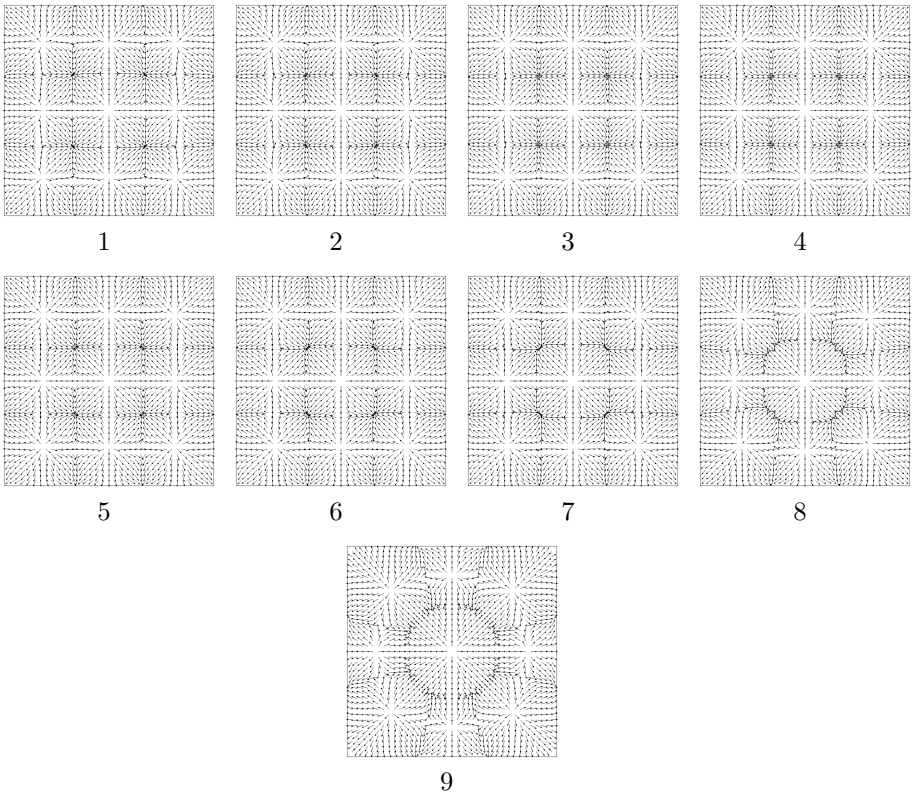


Figure 6.26: Velocity vectors for solutions on the branch depicted in Fig. 6.25.

6.3.1 Continuation in Ma for $\Gamma = 8.3176$

In order to determine if these nine-cell solutions would indeed be observable in a laboratory setting, we computed a new solution branch by performing pseudo-arclength continuation in Ma while holding Γ at a fixed value of approximately 8.3176. This value of Γ corresponds to the 3×3 symmetric solution marked as number 1 in Fig. 6.25, and shown in more detail in Fig. 6.26. The resulting Ma-continuation branch is shown in Fig. 6.27, where we have also marked several solutions of interest with numbers 1–7. The surface velocity fields of these seven solutions are shown in detail in Fig. 6.28.

We observe the following behavior on the Ma solution branch: the symmetric 3×3 cell solution does indeed persist for a fairly large range of Ma, up until about solution 3, which occurs at $Ma \approx 131$. However, whereas the nine convection cells are of roughly equal size in solution 1, by the time solution 3 is reached, the central cell has been reduced in size relative to the other 8. We also observe, as we move from solution 3 to solution 4, the breaking of the x and y symmetries, although the "diagonal" $y = \pm x$ symmetries are still maintained. The apparent "kink" in the solution branch of Fig. 6.27 which occurs between solutions 3 and 4 may thus be interpreted as a symmetry-breaking bifurcation point, though it is difficult to see this from a casual observation of the branch. We have continued the solution numbering 4–7 on the underside of the branch to signify this change to $y = \pm x$ symmetric solutions.

The cellular configuration does not change dramatically at the points

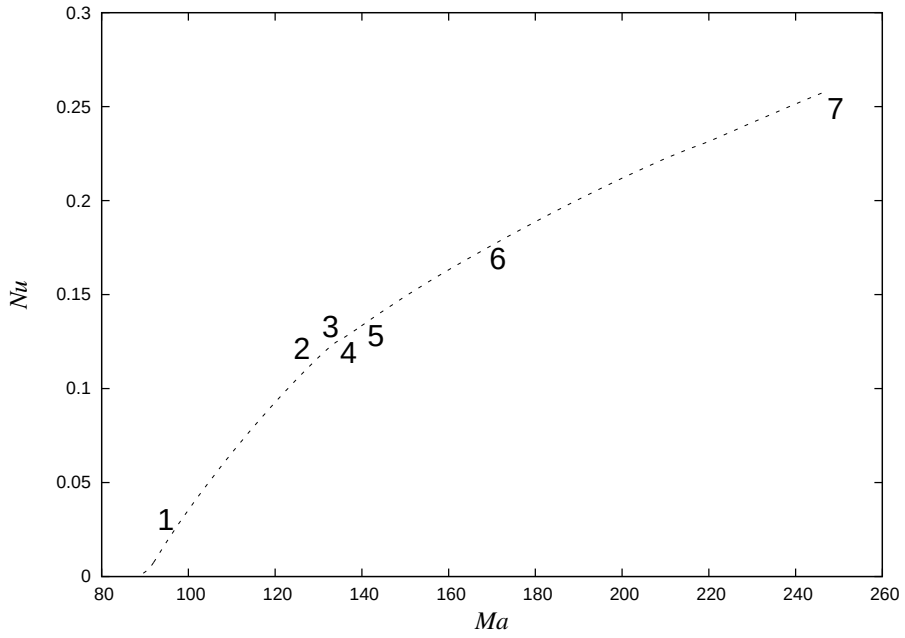


Figure 6.27: Solution branch computed via arclength continuation in Ma starting from the unstable $\Gamma \approx 8.32$ solution on the branch shown in Fig. 6.25.

marked 5, 6, and 7 on the solution branch of Fig. 6.27. This is in spite of the fact that the Ma is nearly doubled from 140 to about 260 in this interval. We can interpret this behavior as an insensitivity of the solution to the Ma in this regime. We do, however, observe the continued shrinking of the central cell, and it appears that the solution may eventually approach the six-cell configuration originally observed at much lower Ma and slightly higher $\Gamma \approx 8.605$, which is marked as solution in 11 in Fig. 5.47.

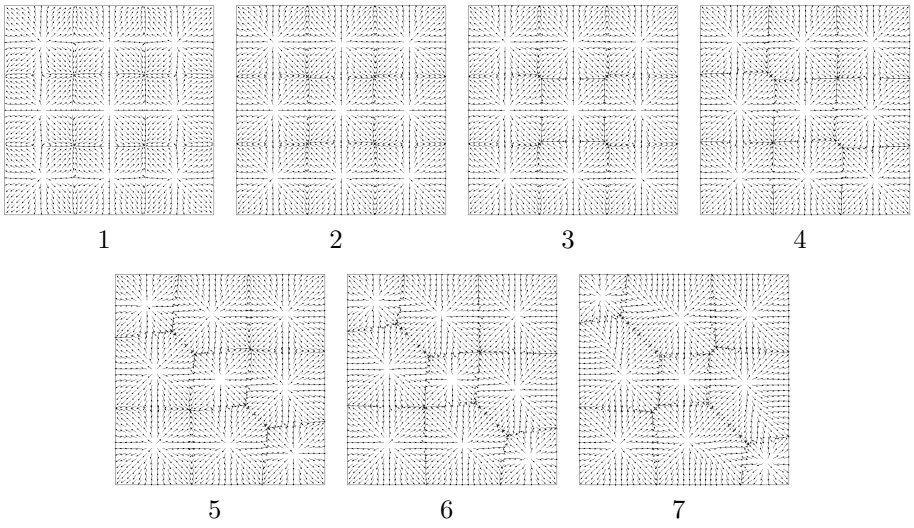
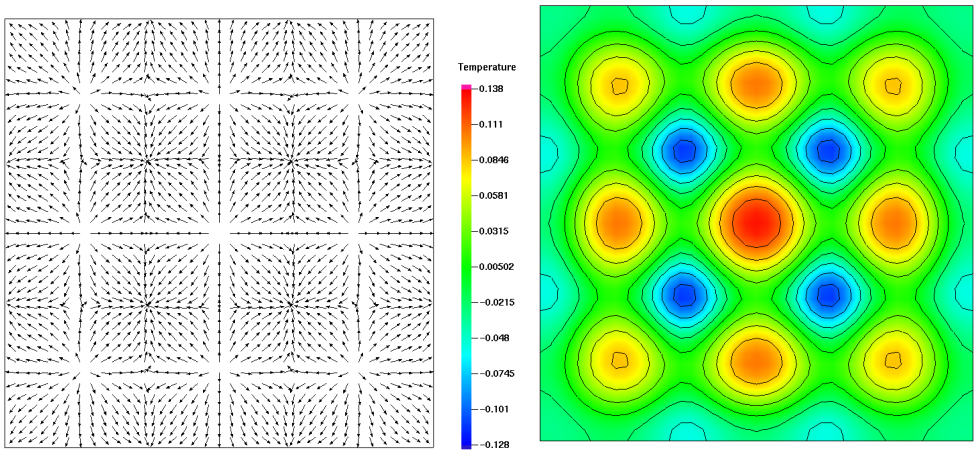


Figure 6.28: Velocity vectors for solutions on the branch depicted in Fig. 6.27 computed via continuation in Ma.

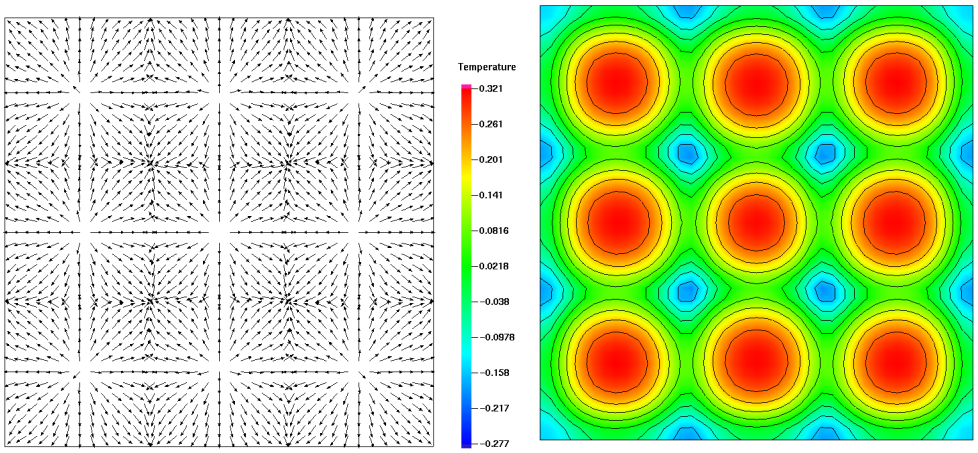
Regarding the stability of the solutions computed on this branch: recall that all of the symmetric 3×3 convection cell solutions (shown originally in Fig. 6.26) computed with varying aspect ratios were determined to be unstable to infinitesimal perturbations. Our initial hypothesis in performing the simulations of the present section was that a nearby (in Ma) stable 3×3-cell solution would be found, thus implying it would be observable experimentally. This turned out not to be the case: as shown in Table 6.3, we computed at least one positive eigenvalue for several selected values of Ma along the branch. It is still certainly possible that a stable, symmetric 3×3-cell solution exists for a different aspect ratio, just not the particular value of $\Gamma = 8.3176$ and the range of Ma attempted here.

Detailed images of the surface temperature contours and velocity fie.ds for solutions 1–7 are given in Figs. 6.29–6.35.



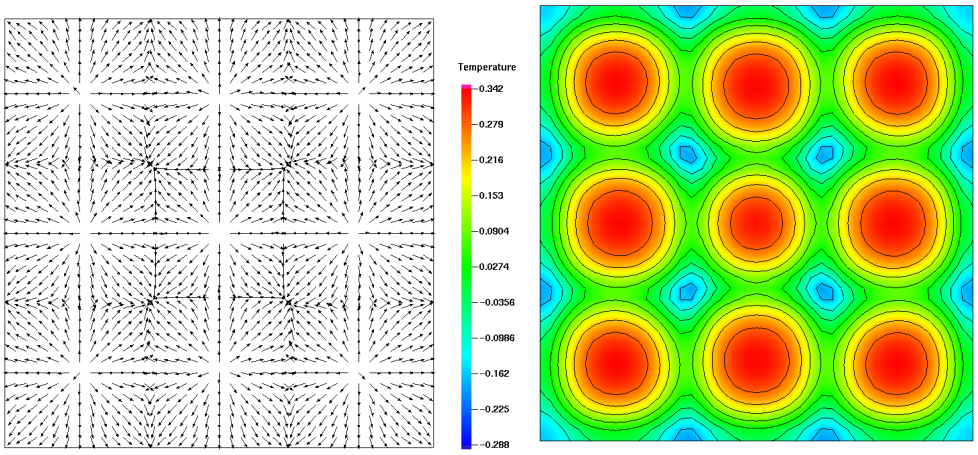
Velocity Vectors Temperature Contours

Figure 6.29: Solution 1 on the branch depicted in Fig. 6.27.



Velocity Vectors Temperature Contours

Figure 6.30: Solution 2 on the branch depicted in Fig. 6.27.

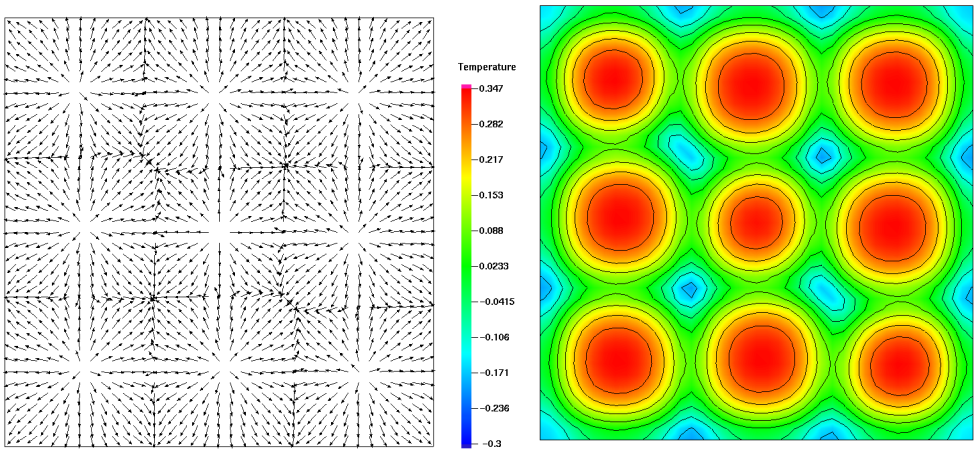


Velocity Vectors Temperature Contours

Figure 6.31: Solution 3 on the branch depicted in Fig. 6.27.

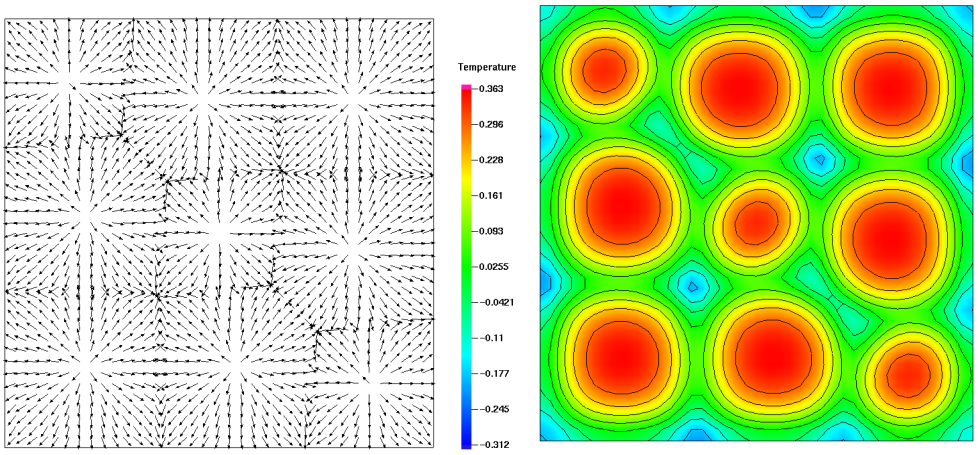
Table 6.3: Numerically-computed eigenvalues σ_i (nearest in magnitude to zero) for the increasing-Ma branch depicted in Fig. 6.27. At least one positive eigenvalue was found for every value of Ma tested. In addition, these cases were very challenging for the eigensolver, due to what appears to be two real-valued eigenvalues very close together in most cases. The eigensolver has difficulty converging whenever two such eigenvalues exist.

Ma	σ_i		
92.500	25.031	30.9715	30.9719
94.948	24.176	31.68661	31.68662
100.186	−16.2333	−16.2335	17.192
105.076	12.938	−19.3117	−19.3121
110.311	9.638	−21.299	−21.300
115.193	7.063	18.151	18.152
120.071	4.758	12.025	14.878
124.255	2.965	8.985	11.779
134.025	1.042	1.411	3.598
145.484	2.143	20.294	−24.703
151.130	3.749	−21.761	24.679
159.602	5.121	−17.942	−30.459
173.633	5.565	−11.240	−30.389
247.244	4.505	−20.347	−29.245



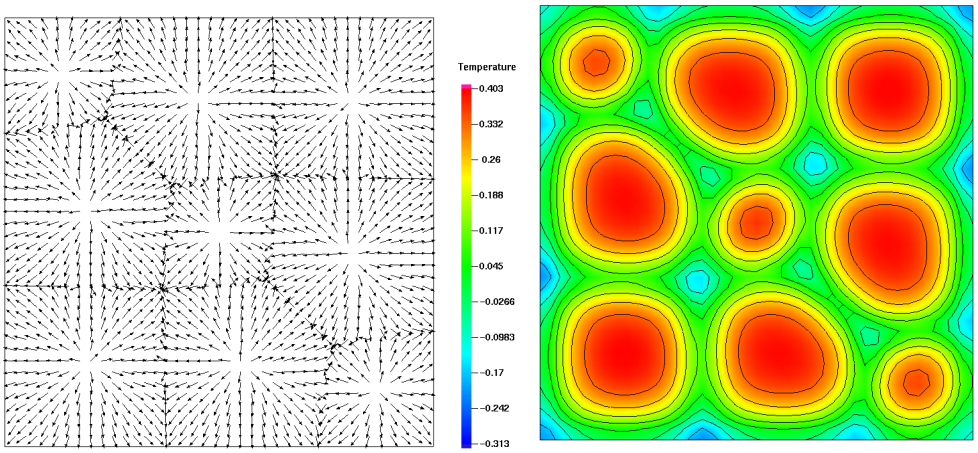
Velocity Vectors Temperature Contours

Figure 6.32: Solution 4 on the branch depicted in Fig. 6.27.



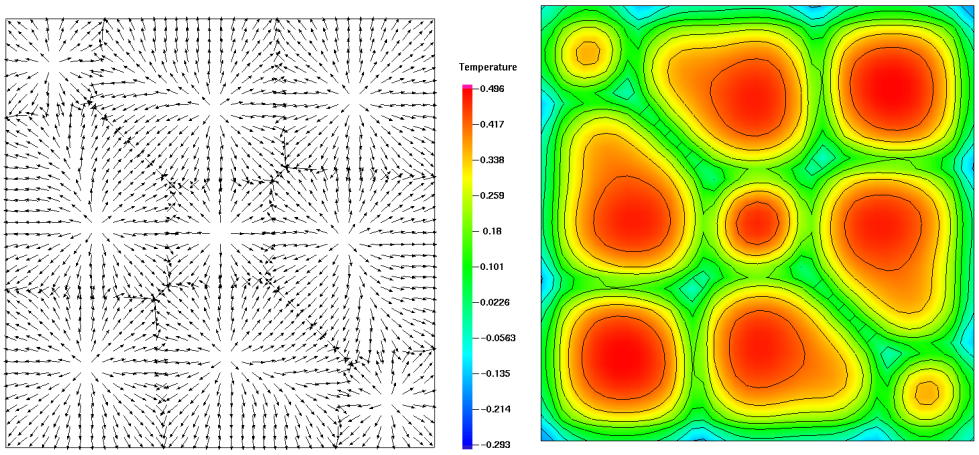
Velocity Vectors Temperature Contours

Figure 6.33: Solution 5 on the branch depicted in Fig. 6.27.



Velocity Vectors Temperature Contours

Figure 6.34: Solution 6 on the branch depicted in Fig. 6.27.



Velocity Vectors Temperature Contours

Figure 6.35: Solution 7 on the branch depicted in Fig. 6.27.

Chapter 7

Parallel Adaptive Simulations

The foregoing RBM convection simulations have focused solely on uniform grid finite element discretizations, but h-adaptive finite elements can also be useful in these types of problems depending on the goal of the simulation and the problem constraints. Here we focus exclusively on h-adaptive finite element methods, and their implementation in the context of the open-source library, `libMesh` [108].

This chapter is arranged in the following manner: in §7.1 we give a general overview of the `libMesh` finite element library. In §7.2 we discuss domain decomposition, the primary way in which `libMesh` achieves parallelism and scales up to handle larger problems. Then, in §7.3 we discuss the adaptive mesh refinement/coarsening (AMR/C) strategies and methodology employed by the library. In §7.4 we discuss how element-level constraints are handled in parallel in the library.

Then, in §7.5 we discuss error estimation and detail a generic, physics-independent gradient jump-based error indicator which is implemented in `libMesh` and used in the present work. In §7.6 we discuss the often-overlooked aspect of element flagging schemes which are used for actually selecting ele-

ments for refinement and coarsening. Finally, in §§7.7 and 7.8, the results of several parallel, adaptive simulations for steady and unsteady RBM convection problems are described.

7.1 Library Overview

All the adaptive and uniform grid calculations discussed in this work were performed using a code based on the open source finite element library `libMesh` [108] developed by the author and collaborators primarily in the CFD-Lab at the University of Texas at Austin. The `libMesh` library was created to facilitate parallel, adaptive, multiscale, multiphysics finite element simulations of increasing variety and difficulty in a reliable, reusable way. In this section, we briefly describe some of the parallel, adaptive algorithms implemented in the library, and their use in the present work.

The design of the `libMesh` library has been influenced from the start by other high-performance parallel C++ libraries, most notably `deal.II` [21, 22], `Alegra` [40], `UG` [28], `PZ` [71], and `hp90` [68]. Since January 2003, `libMesh` has been distributed under an open-source software license and hosted by the popular Sourceforge.net free software portal. Geographically dispersed development is managed with the Subversion (`SVN`) source code management software. To date, the online documentation for the library has averaged approximately 40,000 hits a month. The library itself is downloaded approximately 150 times a month.

The library fosters a spirit of scientific discovery and discussion. New

developers are sometimes recruited from the user community, either by being formally granted write access to the SVN repository or through the submission of patches, which are reviewed and applied to the development branch by the core developers. Documentation, an integral and time-consuming aspect of any C++ class library, is available online and is generated somewhat automatically from the comments in the code itself using the `doxygen` [95] utility. Support for and "hooks" into other well-known libraries such as CUBIT [128], `boost` C++ (boost.org), `Triangle` [188], `tetgen` [189], METIS/PARMETIS [103, 104], PETSc [18], SLEPc [93], LASPack [190], and others are also provided.

There is a key difference between a stand-alone, parallel application-specific finite element code with AMR/C capabilities, and a code such as the one developed in this work for solving RBM convection problems. The code described here is built "on top of" an existing class library with a well-defined application programming interface (API). The benefit to the user, in exchange for conforming to the library's API, is that he or she is freed from the necessity of implementing the associated infrastructure for a parallel finite element analysis capability.

While the library approach provides the flexibility of treating a diverse number of applications in a generic and reusable framework, the application-specific approach may allow a researcher to more easily implement specialized error indicators and refinement strategies which are only suitable for a specific application class. Here, we are concerned primarily with describing some different aspects of the `libMesh` library, including the parallel algorithms for

handling hanging nodes, parallel solution projection/restriction, and simultaneous refinement and coarsening strategies.

7.2 Domain Decomposition

A standard non-overlapping domain decomposition approach is used in `libMesh` to achieve data distribution on parallel computers [45]. The discrete domain Ω_h is partitioned into a collection of subdomains: $\{\Omega_h^p\}$ such that $\bigcup \Omega_h^p = \Omega_h$ and $\bigcap \Omega_h^p = \emptyset$. The elements in each subdomain are assigned to an individual processor, and common tasks such as finite element assembly are performed only on these subsets of elements.

The two primary metrics in judging the quality of a partition are the subdomain mesh size and the number of "edge cuts" in the resulting partition. For a mesh composed of a single type of element, each subdomain should contain an equal number of elements so that the resulting domain decomposition is load balanced across all available processors. The edge cut metric, on the other hand, is designed to minimize the inter-processor communication required by the parallel solver. For an overview of several domain decomposition strategies which are available, see [70, 96].

The domain decomposition approach assigns disjoint groups of elements to individual processors. This allows the element-based degrees of freedom to be assigned uniquely to the processor which owns the element, but requires some shared distribution of vertex, edge, and face degrees of freedom. Figure 7.1 illustrates the approach which is used in the library.

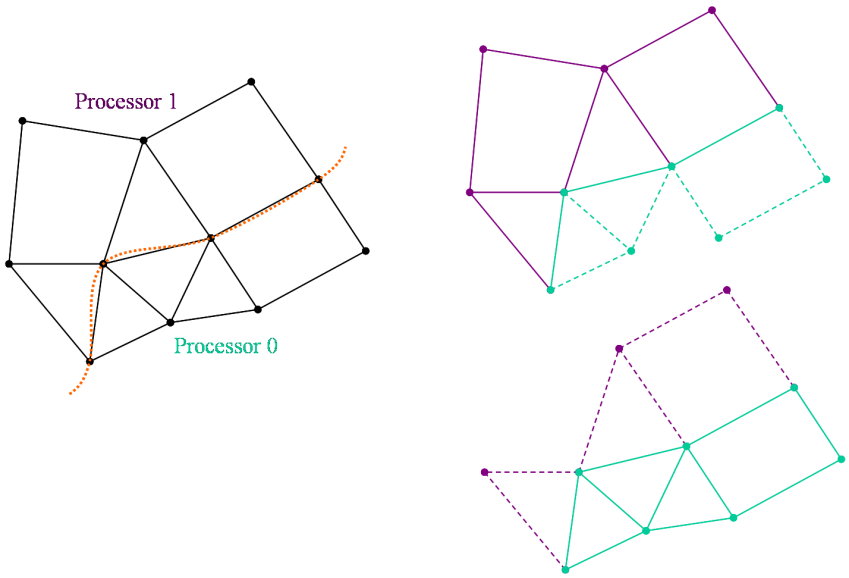


Figure 7.1: Element partitioning and degree of freedom distribution. Disjoint element sets are divided between processors, while boundary nodes are assigned to the processor with lower ID.

In this approach, any degrees of freedom on the border between subdomains are owned by the processor of lowest global index. This is evident from the figure, where the nodes on the shared interface (curved orange dotted line) have been assigned to processor 0. This is represented on the bottom right-hand side of Figure 7.1 by the elements drawn with solid lines, while elements not assigned to processor 0 are depicted by dotted lines. This approach for assigning degrees of freedom to processors also fits well with the sparse matrix partitioning scheme employed in `PETSc`, where complete rows of the sparse matrix are assigned to individual processors [17]. This is the natural matrix decomposition that results from the degree of freedom distribution used in the library.

7.3 AMR/C Methodology

The `libMesh` library employs a standard cell-based discretization discussed in many introductory texts on finite elements (c.f. [29]), using adaptive mesh refinement to produce efficient meshes which resolve small solution features (c.f. [13, 14, 82]). The adaptive technology utilizes element subdivision to locally refine the mesh and thereby resolve different scales such as boundary layers and interior shock layers [46]. Similar concepts have been used in finite element analysis codes since at least the mid-1970's [42, 43].

The focus in `libMesh` is on local subdivision (h-refinement) with hanging nodes and local coarsening by h-restitution of subelements. A major benefit of allowing hanging nodes in the h-refinement scheme is that it simplifies

the adaptation of general hybrid meshes which may be composed of multiple geometric element types. `libMesh` also permits h-refinement of elements with uniformly elevated polynomial degree, and the developmental branch supports adaptively p-refined and hp-refined meshes with some element types.

"Conforming" AMR strategies (without hanging nodes) are also possible. For example, the `PLTMG` [23] package enforces mesh conformity on 2D triangles by connecting a mid-edge node to the opposite vertex of a neighbor element. This idea has been extended by others to progressive longest-edge bisection of tetrahedral meshes [156, 157]. Constraints at mid-edge nodes can also be conveniently handled by algebraic techniques or multiplier methods [44, 45]. Although not currently implemented in `libMesh`, the use of conforming adaptation strategies is not precluded by the library.

The coarsening aspect of AMR/C merits further discussion. Coarsening occurs whenever a "parent" element is reactivated due to the deactivation of all of its "child" elements. This scheme assumes the existence of an initially conforming mesh, whose elements are never coarsened. There are many practical issues related to the selection of elements for refinement and coarsening, notwithstanding the calculation of an accurate error indicator. Special rules such as "refine any element which is already coarser than a neighbor chosen for refinement" or "refine any element which would otherwise become coarser than all its neighbors" may be chosen to smooth the mesh grading and force additional refinement in regions with otherwise small error.

7.4 Parallel Hanging Node Constraints

On edges (faces) which share a hanging node or nodes, some degrees of freedom must be constrained in order to ensure C^r (typically $r = 0$) continuity across the interface. In `libMesh`, we have taken special care to ensure that the application of these constraints:

1. Is a local computation: for efficiency reasons a global projection step, or other, similar large-scale calculation, is to be avoided.

2. Is element-independent: the same code which applies to two neighboring quadrilateral elements also applies to (say) a quadrilateral element neighboring a triangular element.

3. Requires minimal communication: In a parallel computing environment, extra communication steps introduce unnecessary latency and computational inefficiency.

In general, requirement 1 above is not difficult to satisfy: the computations are inherently local and thus computationally efficient from this standpoint. Requirement 2 (element-independence) is satisfied by the application of local L_2-projection problems for all the continuous solution values and derivatives across the interface.

Let us now consider in more detail the local L_2 projection procedure implemented in the library. We assume C^0 continuity, let the fine-scale finite

element solution be u^F, and let u^C be a coarser finite element solution to which we wish to constrain u^F along a shared interface γ. Then, we impose

$$(u^F, \phi_k^F)_\gamma = (u^C, \phi_k^F)_\gamma \quad \forall\, k = 1, \ldots, N_F \tag{7.1}$$

where $(\cdot, \cdot)_\gamma$ is the standard L_2 inner product on γ and N_F is the number of fine basis functions. Expanding the finite element solutions in the appropriate bases, Eqn. (7.1) yields

$$\sum_i (u_i^F \phi_i^F, \phi_k^F)_\gamma = \sum_j (u_j^C \phi_j^C, \phi_k^F)_\gamma \quad \forall\, k \tag{7.2}$$

which implies the (small) linear system of equations

$$Au^F = Bu^C \tag{7.3}$$

where

$$A_{ki} := (\phi_i^F, \phi_k^F)_\gamma \tag{7.4}$$

is a square matrix and

$$B_{kj} := (\phi_j^C, \phi_k^F)_\gamma \tag{7.5}$$

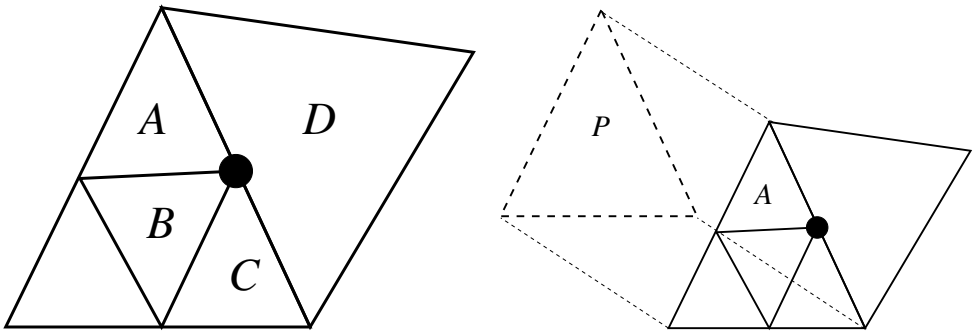
is rectangular, having more rows than columns since, by definition, there are more degrees of freedom in the fine space. We can solve for the fine degrees of freedom by numerically inverting the square matrix A to obtain

$$u^F = A^{-1} B u^C \tag{7.6}$$

Fig. 7.3 shows an example of a shared interface γ between elements having different refinement levels. This step requires the construction and

numerical inversion of small matrices, and is therefore less efficient than specialized constraint matrix construction based on specific element degree of freedom equations, but the resulting code is much cleaner and simpler, and it facilitates the development of new finite element types, since the developer need not also consider the possibility of constraining his new element with all previous element types.

Requirement 3 is satisfied by considering the degrees of freedom of a refined element to be constrained in terms of its parent, rather than the more traditional notion of being constrained in terms of a neighboring element. This idea is illustrated in Fig. 7.2a, in which the hanging node (black dot in the figure) constrains the refined elements A, B, and C to take on the mid-edge value of neighboring element D at this point. This type of constraint may violate the minimal communication requirement if D lives on a different processor than A, B, and C. An improved situation is obtained by considering



(a) Neighbor Constraint (b) Parent Constraint

Figure 7.2: Hanging node (black dot) constraints based on neighbors 7.2a and based on parents 7.2b.

only e.g. refined element A and its parent element P, denoted by dashed lines in Fig. 7.2b. Element A holds pointers to its parent due to the natural structure of the refinement tree, and the hanging node value can be constrained simply by considering the finite element solution space of the parent as the "coarse" space. The neighboring element is no longer involved, and the L_2 constraint equations now apply to child and parent rather than child and coarse neighbor. In cases where more than one refinement level separates neighboring elements (meshes violating the "level-1" rule) the same procedure can be applied; we exploit here the fact that the grid is conforming at some level, and the constraints are applied recursively.

7.5 Error Indicators

We wish to distinguish here between *a posteriori* error *estimator*s and *indicator*s. The main difference is that an error estimator is designed to converge to the *true* global error in the approximate solution (under refinement) whereas a local error indicator is employed to discern the error for a given element *relative to* other elements in the mesh. Error indicators are intended to be local, cheaply computable quantities, not to provide rigorous theoretical bounds on the error.

The field of *a posteriori* error estimation for finite elements has been a fertile one for more than thirty years, with numerous papers and several books on various topics having been produced in that time. Here we do not attempt to provide an exhaustive review of this large body of literature, but

rather to touch on a few publications which have influenced the present work and related work on the `libMesh` library [108].

A great deal of the *a posteriori* error estimation literature is based on linear, elliptic problems such as Laplace's equation, convection-diffusion, Stokes flow, and the Oseen equations. See, for example, the references [3, 5, 24, 80, 99, 105, 137, 145, 178, 179, 197, 200, 201]. The literature on *a posteriori* error estimation for nonlinear problems is more sparse than for the linear case. The papers [85, 139, 198, 199, 212] are relevant for this class of problem. General reference works with more complete literature reviews include [6, 115].

A commonly-used class of error indicators are the "gradient-recovery" or averaging indicators. Examples include the superconvergent patch recovery (SPR) method of Zienkiewicz and Zhu [217–219]. Their work was based on earlier contributions by the solid mechanics finite element analysis community (superconvergent stress field recovery) and mathematical analysis [15, 220, 221], and was later analyzed, generalized, and extended [4, 7, 34, 48, 206, 207, 213, 214]. Gradient-recovery error indicators are implemented in the `libMesh` library, although they are not used in the present work.

Error indicators can also be designed to improve accuracy in the computation of a quantity of interest. This class of indicators, known as goal-oriented or adjoint indicators, and the method of dual-weighted residuals (DWR) [20, 30, 78, 79] can be customized to derive optimal meshes for computing e.g. the solution value or gradient at a point, integrated fluxes, etc.

Such indicators are typically difficult to implement in practice, especially in the time-dependent, nonlinear case, and expensive to compute.

7.5.1 A Local Gradient-Jump Error Indicator

To introduce the primary local error indicator used in `libMesh` for the present work, let us consider the model elliptic PDE

$$\begin{aligned} -\nabla \cdot k\nabla u &= f && \in \Omega \\ u &= g_1 && \in \partial\Omega_D \\ k\frac{\partial u}{\partial n} &= g_2 u + g_3 && \in \partial\Omega_M \end{aligned} \tag{7.7}$$

where k is a given (possibly spatially varying) parameter. For example, Eqn. (7.7) may represent the flow of heat in a material with variable thermal conductivity. Dirichlet boundary conditions are specified on $\partial\Omega_D$, and mixed boundary conditions are specified on $\partial\Omega_M$. The boundary functions g_1, g_2, and g_3 are given. The entire boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_M$. Here we consider the scalar case, but this discussion can be generalized to systems of equations as well as equations involving non-symmetric (convection) terms as well as source terms.

It can be shown that the error $e := u - u^h$ satisfies

$$B(e, v) = \int_\Omega \mathscr{R}(u^h) v \, dx + \int_{\partial\Omega_M} r(u^h) v \, ds + \sum_e \int_{\partial\Omega'_e} \left[\!\left[k\frac{\partial u^h}{\partial n} \right]\!\right] v \, ds \tag{7.8}$$

for every v in some admissible space of functions $\mathcal{V}$, where $B(\cdot,\cdot)$ is the standard bilinear form for problem Eqn. (7.7), $[\![\cdot]\!]$ represents the jump in a discontinuous quantity, $\frac{\partial}{\partial n}$ represents the normal directional derivative, $\partial\Omega'_e$ are

the non-boundary edges (faces) of element e, $\mathscr{R}(u^h) := f + \nabla \cdot k\nabla u^h$ is the element interior residual and

$$r(u^h) := \left(g_2 u^h + g_3\right) - k\frac{\partial u^h}{\partial n} \tag{7.9}$$

is the exterior boundary residual due to the general mixed boundary condition [107]. We define the jump in $k\nabla u^h \cdot n$ across the interface of two neighboring elements Ω_e and Ω_f as

$$\left[\!\left[k\frac{\partial u^h}{\partial n}\right]\!\right] := \left(k\nabla u^h|_{\Omega_e} - k\nabla u^h|_{\Omega_f}\right) \cdot n_e \tag{7.10}$$

where n_e is a unit vector which is normal to internal edge (face) $\partial\Omega'_e$ and points in the direction of Ω_f.

According to the *a posteriori* error estimation theory for coercive elliptic problems (see e.g. [88], here we are coercive as long as a is a solenoidal velocity field) the remaining steps in the analysis are to

1. Apply the Cauchy-Schwarz inequality element-wise to find an upper bound on $|B(e, v)|$.

2. Using Galerkin orthogonality and the properties of the finite element interpolant, bound the arbitrary functions v on the right-hand side of Eqn. (7.8) in terms of the natural (energy) norm for the problem.

3. Choose, as a particular v, the error $e := u - u^h$.

The result of following these steps is an upper bound on the error in the energy norm in terms of the norms of the residuals on the element interiors and the

flux jumps across the inter-element boundaries having the form

$$\|e\|_E^2 \leq C \left(\sum_e h_e^2 \|\mathscr{R}(u^h)\|_{L_2(\Omega_e)}^2 + \sum_e h_e \|J\|_{L_2(\partial\Omega_e')}^2 \right) \tag{7.11}$$

where J is the jump term as defined in Eqn. (7.10). Before discussing how the global error bound of Eqn. (7.11) will be used for general problems, we will first mention some of the theory's limitations.

First, the bound of Eqn. (7.11) only directly applies to *linear*, coercive problems which come equipped with a natural energy norm. Obviously, this hypothesis is violated for the nonlinear natural convection problems of interest here. Coercivity is not trivial to show for the nonsymmetric problem (7.7) unless $\nabla \cdot a = 0$ and $\partial\Omega_M = \emptyset$ [81, 160]. In fact, even the Galerkin orthogonality property is not so straightforward in the nonlinear case.

Second, in going from Eqn. (7.8) to Eqn. (7.11), any possible cancellation of error has been removed via the use of the Cauchy-Schwarz inequality, thus we expect the bound to be overly-pessimistic, i.e. to over-predict the error and possibly trigger unnecessary refinement. Third, the bound of Eqn. (7.11) is not physics-independent, in the sense that its computation requires knowledge of the residual (i.e., the governing PDEs) as well as any coefficients which enter the jump term calculation. In addition, we cannot state *a priori* whether the element interior residual term or interface flux jump term will dominate the error estimate, or if they will be of comparable size.

Finally, in systems of multiple equations, the physical units (i.e. meters, seconds, etc.) of each equation need not be the same. Thus one must take

extra care in such situations that the equations are suitably normalized and that the residuals from separate equations are properly combined. Even under a suitable non-dimensionalization, the absolute scale of the different variables in a system of equations can still be widely varying, for example the non-dimensional pressure can easily be $\mathcal{O}(10 - 1000)$ while the non-dimensional velocity magnitude is $\mathcal{O}(0.1 - 1)$ in natural convection problems. This implies that the size of the error in a particular component *relative to* the size of the component itself is the only measure of the error which can be meaningfully compared.

To obtain an error indicator for general second-order problems, we start from Eqn. (7.8), set $v = v^h$ and apply the Galerkin orthogonality property. Then we obtain

$$\begin{aligned} 0 &= B(e, v^h) \\ &= \int_\Omega \mathscr{R}(u^h) v^h \, dx + \int_{\partial\Omega_M} r(u^h) v^h \, ds + \sum_e \int_{\partial\Omega_e'} \left[\!\left[k \frac{\partial u^h}{\partial n} \right]\!\right] v^h \, ds \end{aligned} \tag{7.12}$$

In a generic finite element code, we assume only that the finite element solution u^h and its gradient are available to the error indicator, so, for example, knowledge of how to compute the strong residual $\mathscr{R}(u^h)$ is not assumed, nor is knowledge of the boundary conditions for a particular problem. In addition, knowledge of the problem coefficients (such as the coefficient k) is not assumed for the error estimation procedure. These assumptions are by no means absolute, however they greatly simplify the implementation of the estimator.

Accordingly, the explicit physics-independent error indicator imple-

mented in our finite element codes is (drawing inspiration from the formally-derived bound of Eqn. (7.11))

$$\eta_e^2 := h_e \left\| \left[\!\left[\nabla u^h \cdot n \right]\!\right] \right\|^2_{L_2(\partial\Omega_e)} \tag{7.13}$$

Given the number and severity of the assumptions and restrictions leading up to Eqn. (7.13), it is hardly surprising that there are situations where it performs poorly. One such situation arises when solving elliptic problems with so-called "bi-even" (e.g. bi-quadratic, and presumably "tri-even" in three dimensions) finite element polynomial spaces. In such cases, as was shown initially by Babuška and Yu [16, 210, 211] and extended to parabolic problems [1, 2], the interior element residual error dominates the interface flux-jumps.

An inaccurate error indicator is likely to be obtained when the interior residuals are not computed. In fact, the use of the gradient-jump based indicator of Eqn. (7.13) can actually indicate the incorrect relative error between elements of a given mesh, thereby leading to *incorrect* adaptive refinement patterns. Keeping these limitations in mind, the totally generic nature of the gradient-jump indicator still makes it attractive for implementation in numerical libraries. For this reason, we include it in the RBM convection results shown later.

7.5.2 Gradient-Jump Error Indicator Algorithm

The generic gradient-jump error indicator discussed in §7.5.1 is well-suited for mesh refinement strategies employing hanging nodes, and can be computed in parallel provided one layer of "ghost" degrees of freedom are available. The basic steps for estimating the error using the gradient-jump indicator are sketched out in Algorithm 3.

Algorithm 3 Algorithm for estimating the error using generic, gradient-based error indicator described in §7.5.1. In the interior loop, a neighbor f is defined as "suitable" if it is not on the boundary and is either (a) active and of the same level as the current element or (b) coarser than the current element. The interface shared by the current element and a suitable neighbor is denoted by γ. The error contribution to element e is scaled by a mesh size parameter h_e appropriate to that element. The normal vector n is uniquely defined as being from the lower ID element toward the higher (in case (a)) or from the fine element to the coarse element (in case (b)).

```
for each variable v do
  Initialize FE and quadrature rule for variable v
  for each active, local element e do
    for each "suitable" neighbor f do
      Compute η² := ∫_γ ( ∂u_e/∂n − ∂u_f/∂n )² ds
      Accumulate η_e² += h_e η²
      Accumulate η_f² += h_f η²
    end for
  end for
end for
```

Consider again the case shown in Fig. 7.2a, and suppose we have looped to element A. The normal vector will be defined as in Fig. 7.3 pointing from element A to element D, and the interface γ will be the intersection of the edge of A with the shared edge of D. When the error estimates η_A^2 and η_D^2 are

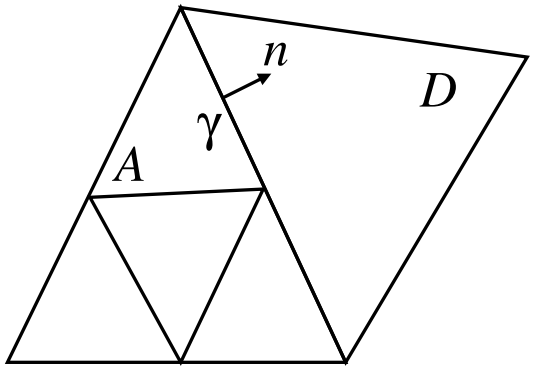


Figure 7.3: Definition of the normal vector n and interface γ for computing the error estimator on a hanging edge.

accumulated, η_A^2 will be scaled by a representative mesh size h_A, for example the diameter of the circle which circumscribes A, and η_D^2 will be scaled by a similar quantity representative of element D. In this way, each contribution to element D will be scaled by the appropriate constant, regardless of how many hanging nodes (refinement level difference relative to the initial grid) there are on a given edge of D. Slight modifications to Algorithm 3 are possible to handle mixed or Neumann boundary data, although this violates the encapsulation of the error estimator from the particular problem's physics.

7.6 A Statistical Element Flagging Strategy

Once the error indicator has been computed on every element, it is necessary to "flag" or mark different elements for refinement and coarsening based on their relative error values. One approach we have employed in the present work is a statistical strategy for proportioning cells between refinement and

coarsening. The ideas are related to earlier approaches [45] in which the mean μ and standard deviation σ of the indicator "population" are computed. Then, based on refinement and coarsening fractions r_f and c_f (either default values or specified by the user), the elements are flagged for refinement and coarsening. This scheme, depicted graphically in Figure 7.4, is beneficial in evolution problems where, at early times, the error is small and equidistributed and no elements need to be flagged for refinement. Later, as interesting features develop, the statistical distribution spreads and refinement and coarsening begins. As the steady solution is approached, the distribution of the error reaches a steady state as well, effectively stopping the AMR/C process.

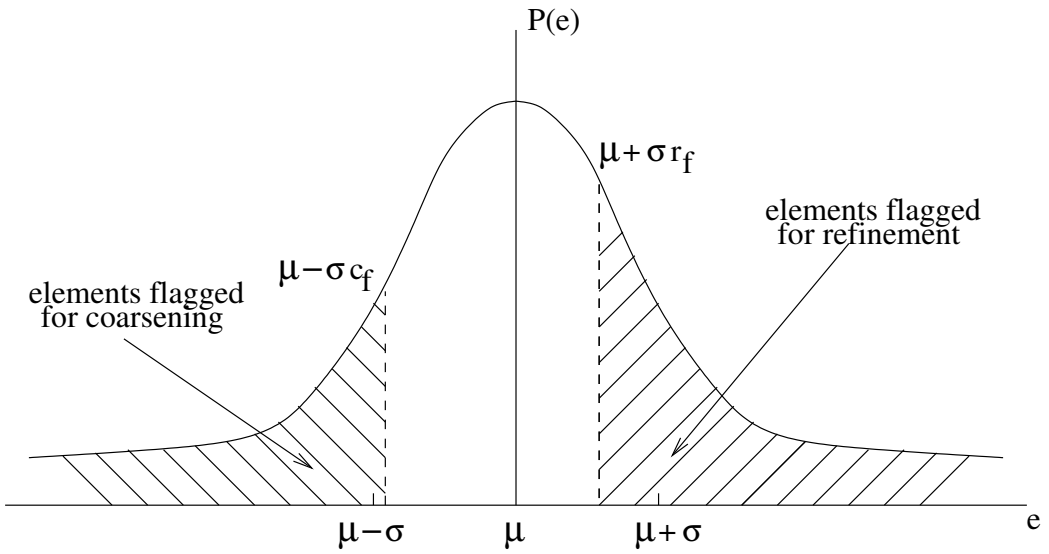


Figure 7.4: In the statistical refinement scheme, the element error e is assumed to have an approximately normal probability density function $P(e)$ with mean μ and standard deviation σ. Elements whose error is larger than $\mu + \sigma r_f$ are flagged for refinement while those with errors less than $e < \mu - \sigma c_f$ are flagged for coarsening.

7.7 Transient AMR for a Stick-Slip Boundary

In this and the following section, we report on our AMR results for unsteady and steady problems (respectively) in RBM convection. First, we discuss AMR/C for a transient RBM stick-slip boundary problem. In this case, the Navier-slip boundary condition (which is related to the penalty boundary condition formulation for essential data and is described in more detail in Appendix §1.3) is applied to various quadrants of the bottom surface of a container of fixed aspect ratio.

In the results that follow, the strength of the flow in the slip region is proportional to the strength of the local tangential stress field. This type of boundary condition introduces a singularity in the velocity gradient components at the stick-slip interface, and the gradient-jump based error indicator described in §7.5 appears to work fairly well for this application, despite the fact that it was not designed with exactly this problem in mind.

The sections in this part are arranged as follows: in §7.7.1 we give the algorithm used to combine both adaptive timestepping and AMR for a time-dependent solution procedure. In §7.7.2 we provide several "reference" solutions for RBM convection when slip is allowed in various quadrants. Then, in §7.7.3 we give a representative result from an intermediate step of the solution procedure. In §7.7.4 we give a detailed comparison of the adaptive grids generated for three particular refinement parameter settings. In §§7.7.5 and 7.7.6 we give an even more detailed view into the characteristics of the adaptive solution by plotting along representative cutlines which intersect the

stick-slip interface. Finally, in §7.7.7, we remark briefly on the nature of the stick-slip singularity and show how well several uniform and adaptive grid solutions approximate the analytical solution behavior near this interface.

7.7.1 Combined AMR and Adaptive Timestepping

A key aspect of any time-dependent mesh adaptation algorithm is the selection of the frequency at which the mesh is refined/coarsened. Clearly, refining the mesh at every timestep provides the most accurate result (least amount of lag between the solution evolution and the mesh evolution) but this strategy is also the most expensive, since every mesh adaptation step requires a subsequent mesh projection and solve step. When, as in the present work, time-dependent AMR is also combined with adaptive timestep selection, the relative expense of each nonlinear solve becomes even greater.

In order to handle these competing requirements in the simplest possible manner while still providing flexibility to the user, we have employed an approach using both an h-adaptive timestep interval and an independent adaptive timestep selection interval. The scheme, which is detailed in Algorithm 4, makes use of the `FEMSystem`'s timestepping-scheme independent programming paradigm, and employs the interchangeable `EulerSolver` (a standard θ-method solver) and `AdaptiveTimeSolver` objects which are derived from the generic `TimeSolver` class.

Since the method of adaptive timestepping employed here is step-doubling (see §2.2) and employs an additional two solves per timestep, it is undesirable

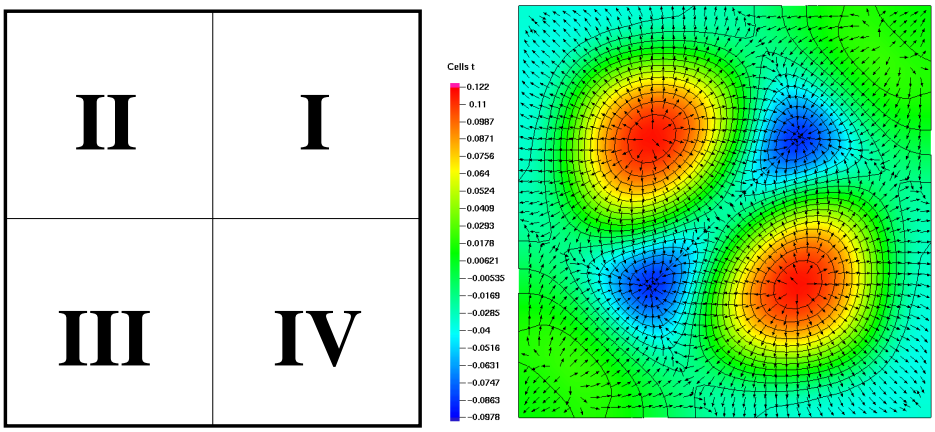
to use an adaptive timestep selection during the adaptive mesh refinement loop. Another reason for choosing an adaptive timestep selection interval is that, especially at early times, large initial transients arising from particular choices of initial conditions may generate an artificially small timestep which will cause unnecessary extra work at the beginning of the simulation.

Algorithm 4 Timestepping loop combining adaptive mesh refinement and adaptive timestepping at user-defined timestep intervals.

```
Initialize AdaptiveTimeSolver and EulerSolver objects
for each timestep n do
  Set EulerSolver
  if at the h-adaptation interval then
    for each adaptive step r do
      Solve nonlinear system
      Estimate error and flag elements
      Refine/Coarsen elements and project solution
    end for
  end if
  if at the adaptive timestep interval then
    Set AdaptiveTimeSolver
  end if
  Solve nonlinear system
end for
```

7.7.2 Navier-Slip Reference Results

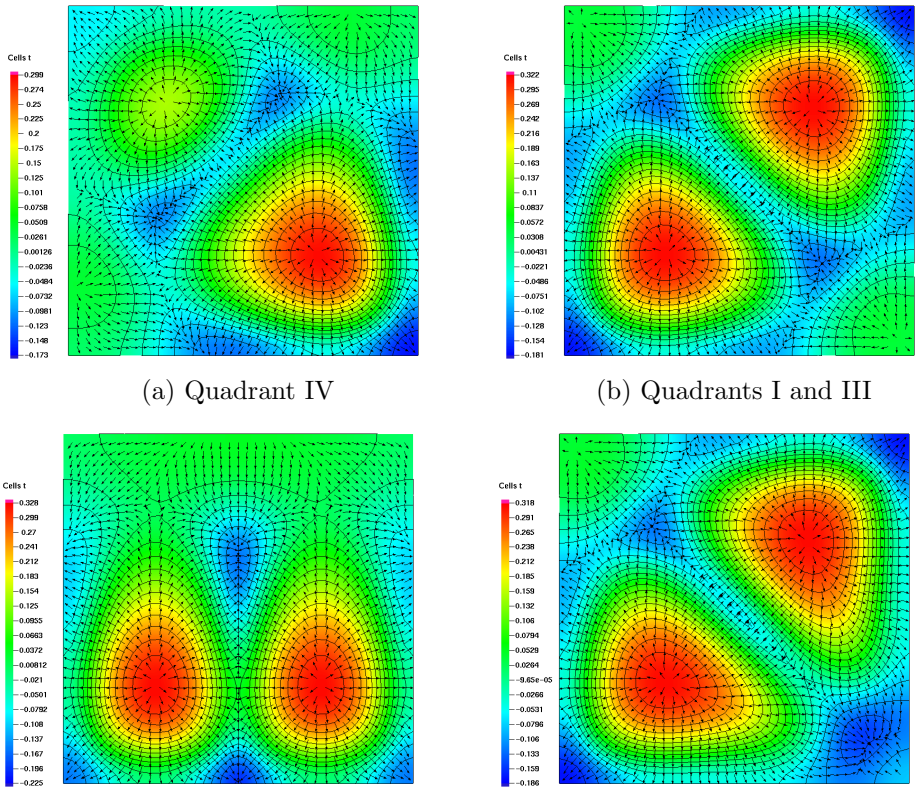
For reference, we provide in this section several results for a single aspect ratio ($\Gamma = 6$) with slip allowed in different quadrants. We number the quadrants I, II, III, and IV in the standard way, as shown in Fig. 7.5a.



(a) Quadrant labels (b) No-slip reference solution

Figure 7.5: Numbering of the different quadrants where slip will be turned on/off 7.5a, and representative no-slip solution, 7.5b.

The steady-state results shown in Fig. 7.6 were computed on uniform $20 \times 20 \times 2$ grids using the adaptive timestepping algorithm discussed elsewhere in this work. The results show the effect that allowing slip in different quadrants on the bottom surface has on the entire flow field. In particular, we observe higher peak temperatures when more slip is allowed, and consequently higher heat transfer rates. Slip generally has the effect of reinforcing the other instabilities in the problem which tend to set up circulating flow patterns, and therefore invigorates these circulating flows.



(a) Quadrant IV

(b) Quadrants I and III

(c) Quadrants III and IV

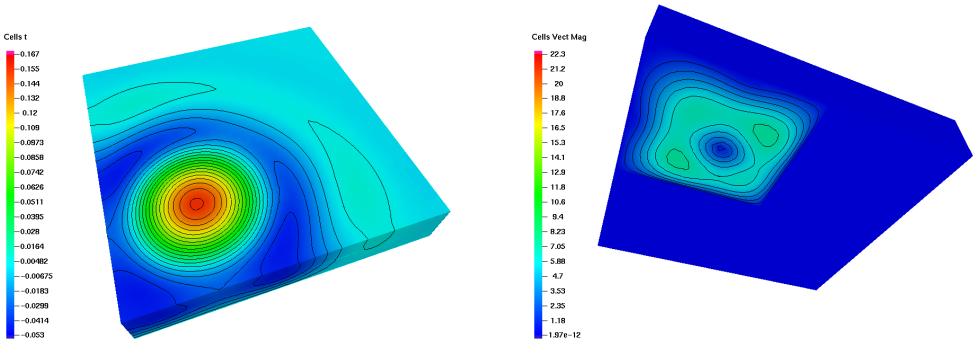
(d) Quadrants I, III, and IV

Figure 7.6: Steady state results when slip is allowed in quadrants IV (7.6a), I and III (7.6b), III and IV (7.6c), and I, III, and IV (7.6d). The trend is generally toward higher overall heat transfer rates when additional slip is allowed. The steady solution for a full Navier-slip bottom surface is a symmetric four-cell configuration. When compared to the reference no-slip solution in Fig. 7.5b, all of the results with slip show higher surface peak heating rates.

7.7.3 Representative AMR Results: Slip in Quadrant I

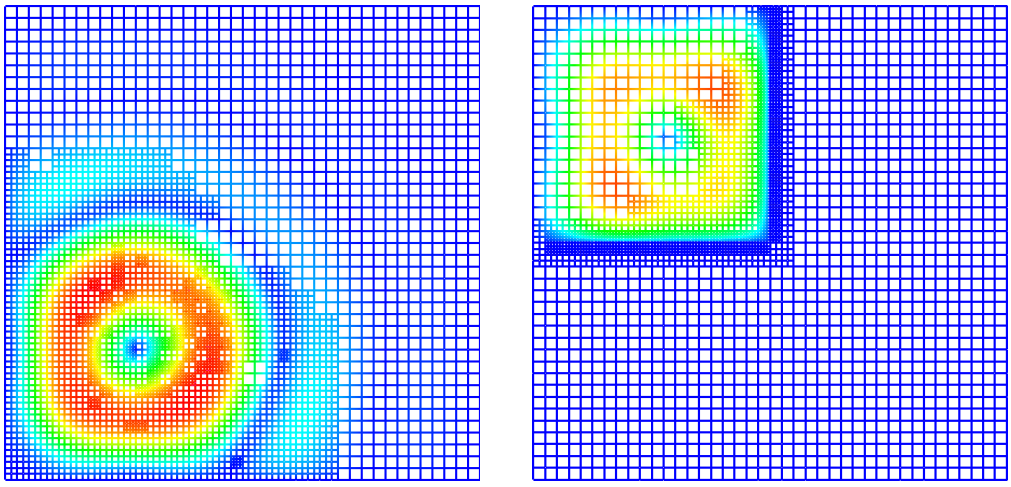
We now present results from a representative timestep for a simulation employing both adaptive timestepping and adaptive mesh refinement and coarsening. Navier-type slip is allowed only in the upper-right quadrant of the domain bottom. The problem parameters here are $\Gamma = 6$, $Ma = 92$, $Ra = 30$, $Bi = 0.2$, $Pr = 880$. In Fig 7.7, we show the temperature (7.7a) and velocity magnitude (7.7b) contours, at a representative intermediate timestep, on the top and bottom of the domain, respectively. The error indicator for this case is the generic gradient-based indicator, considering jumps in the horizontal velocity components u_1 and u_2 only.

The maximum refinement level was limited to two levels beyond the coarse grid. Inspecting the adaptively-refined grids in Figs 7.8a (free surface) and 7.8b, (stick-slip interface) we observe that the boundary between the slip and no-slip regions is targeted for refinement, as well as the quadrant of the domain where the convection cell is located. The sporadic refinement pattern observed in some places on the surface can be caused by too strict a tolerance for coarsening and the imprecise nature of the error indicator.



(a) Temperature Contours (b) Velocity Magnitude Contours

Figure 7.7: Contours of temperature (7.7a) and velocity magnitude (7.7b), viewed from above and below, respectively.



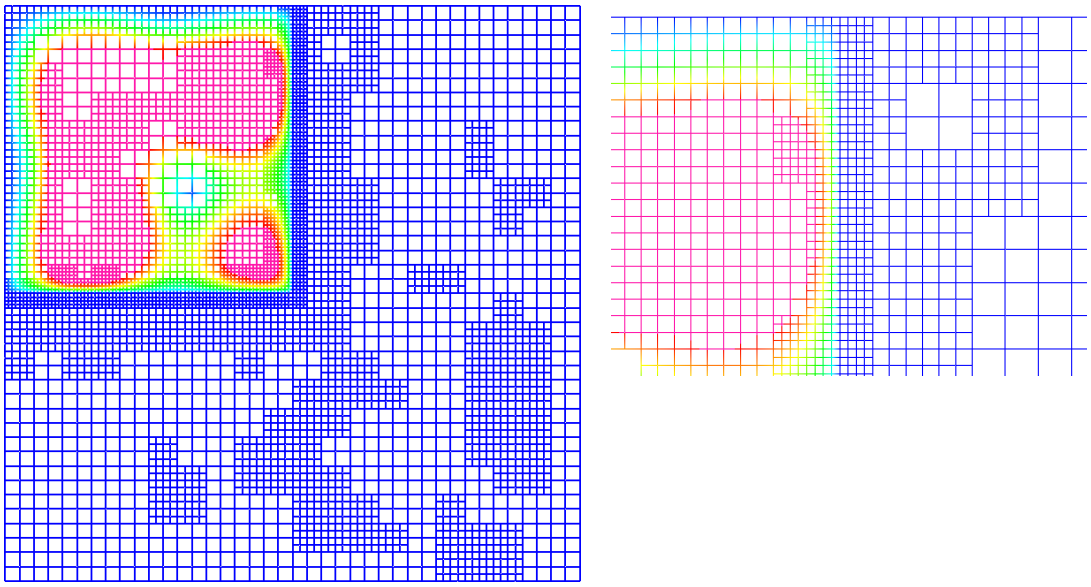
(a) AMR mesh, top surface (b) AMR mesh, bottom surface

Figure 7.8: Adapted grid from above showing temperature contours (7.8a) and below (7.8b) showing velocity magnitude contours as colors overlaid on the mesh lines.

7.7.4 Comparison of Steady State Adapted Grids

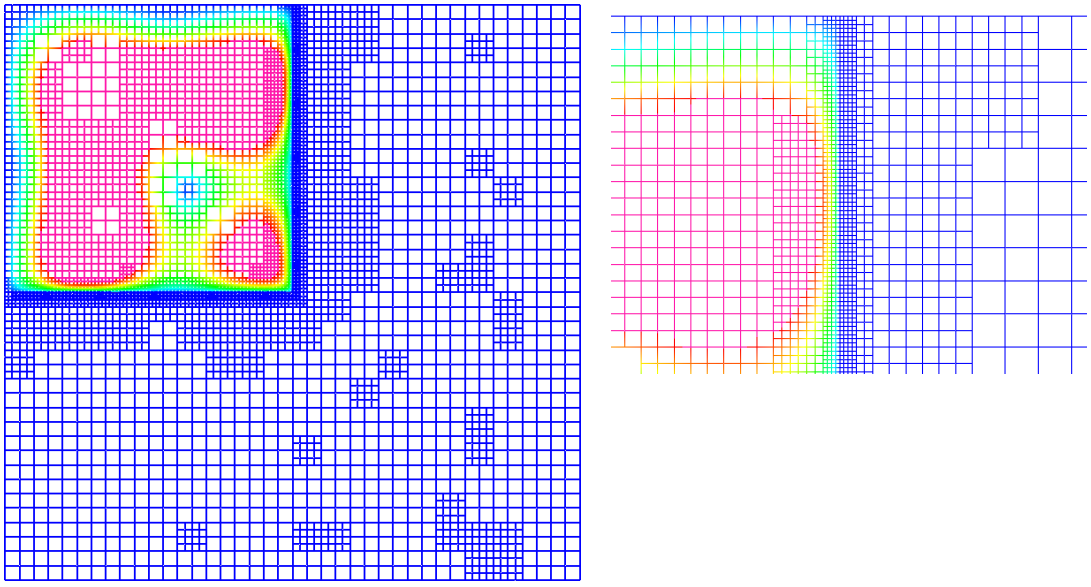
In this section, we provide a detailed comparison of automatically-generated meshes for the converged steady-state solutions of the time-dependent calculations. We ran three particular time-dependent simulations having respectively two, three, and four maximum levels of h-refinement allowed. Due to the nature of the solution in this problem, limiting the maximum refinement level is a practical necessity. Otherwise, the indicator would refine endlessly into the stick-slip interface (where the solution gradients are singular) thereby quickly exhausting the system memory available for the computation.

In Figs. 7.9, 7.10, and 7.11, we present the adaptively-refined steady state meshes with two, three, and four maximum levels of refinement allowed. In each of the preceding figures, we have shown (on the left-hand side, in part (a)) the adapted mesh which appears on the bottom of the computation domain. On the right-hand side (part (b)) a close-up view of the refinement patterns near the stick-slip interface at the top of the domain is provided. As expected, the refinement patterns are at their finest level at the stick-slip interface. The slip region itself is, in general, also more refined than the surrounding no-slip region.



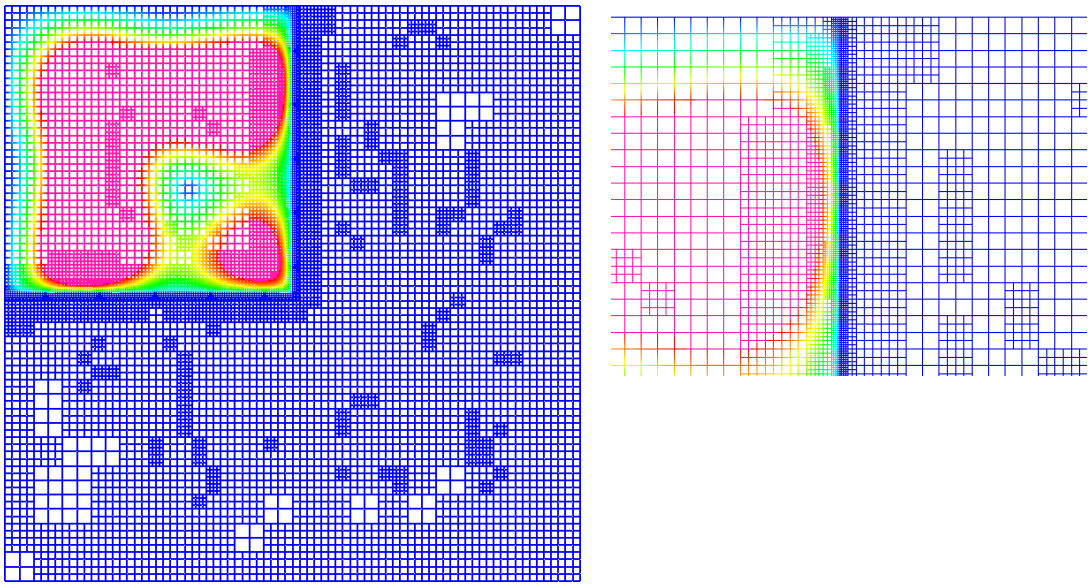
(a) Bottom surface mesh (b) Zoomed view, stick-slip interface

Figure 7.9: AMR grid, two levels of refinement, domain bottom.



(a) Bottom surface mesh (b) Zoomed view, stick-slip interface

Figure 7.10: AMR grid, three levels of refinement, domain bottom.

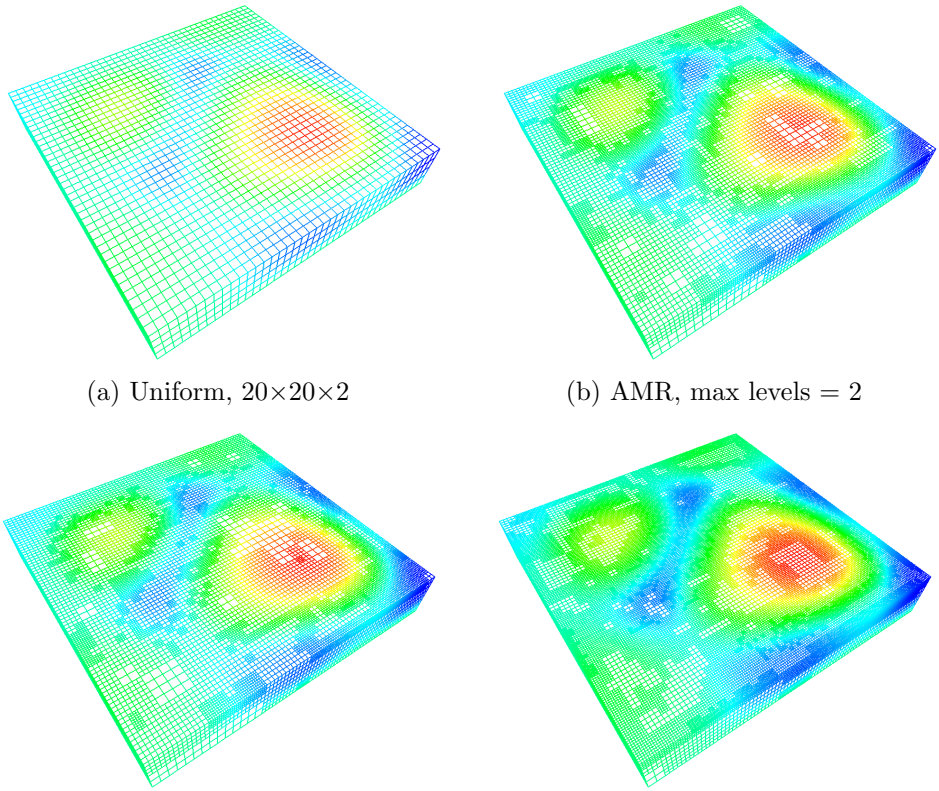


(a) Bottom surface mesh (b) Zoomed view, stick-slip interface

Figure 7.11: AMR grid, four levels of refinement, domain bottom.

In Fig. 7.12 we have once again shown all three of the adapted, steady state solutions, this time with a coarse-level $20 \times 20 \times 2$ mesh included for reference. Here we are showing a "3/4" view of the top (free) surface, and the effect of the slip region is clear from the asymmetric nature of the solution. The presence of slip on the bottom surface is positively correlated with the relative strength of the convection cell which forms above this region. This makes sense if we consider that less momentum is lost from the fluid (there is less drag) as it traverses the slip region and therefore the overall strength of the circulating flow is enhanced.

In terms of the refinement patterns in each case, here we see more refinement near the periphery of the convection cells, with coarser elements generally located at the center (warm) area of each cell. In addition, there is refinement at the stronger of the two convection cells near the lateral walls of the container in each case. This refinement pattern can be contrasted with the mesh configurations observed on the bottom of the domain, where the singular nature of the stick-slip interface drives the adaptive procedure. Finally, we note that although the steady-state solution displays a single $y = x$ line of symmetry, the refinement patterns themselves are not symmetric. This is due to the coarsening tolerance issues mentioned previously, and also to the fact that the mesh evolved along with the solution in a time-dependent manner, and not all intermediate stages exhibited the symmetry of the final state.



(a) Uniform, 20×20×2

(b) AMR, max levels = 2

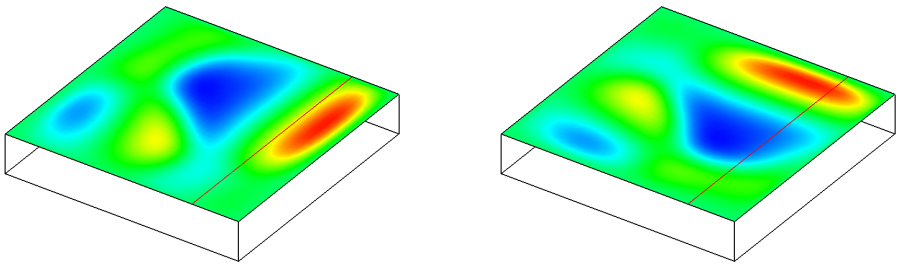
(c) AMR, max levels = 3

(d) AMR, max levels = 4

Figure 7.12: Steady state uniform and adaptive grid comparison, temperature contours overlaid on grid lines.

7.7.5 Comparison of Steady State Free Surface Cutlines

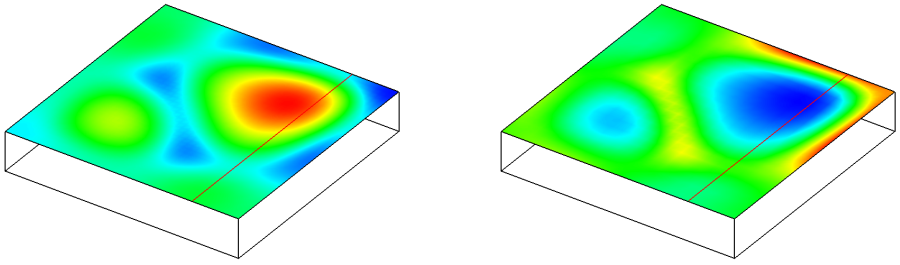
In order to examine in detail the differences in the approximate solutions on different grids, we now plot the steady state solution on the free surface along the line $x_1 = 4.8$. These cutlines are illustrated on cutplanes from representative solution fields in Figs. 7.13 and 7.14. In this entire section, we will refer to and plot the x_1 and x_2 axes in their "unscaled" form, i.e. the following figures will be displayed in their "true" domain aspect ratio, even though they were actually solved on the nondimensionalized cube domain.



(a) u_1 (b) u_2

Figure 7.13: Cutplanes of x_1 (7.13a) and x_2 (7.13b) velocity (u_1 and u_2) on the free surface. Shown superimposed on the cutplanes are red cutlines located at $x_1 = 4.8$, along which detailed line plots of the velocity are given in Figs. 7.15a and 7.15b. The color bars denoting the magnitudes are not included here, actual values are given on the cutline plots themselves.

In Fig. 7.15, we show the solution behavior for the x_1 and x_2 components of velocity on the free surface, along the line $x_1 = 4.8$. In the Figure, we compare several uniform grids with an adaptive grid having a maximum of



(a) $\tilde{T}$ (b) p

Figure 7.14: Cutplanes of temperature $\tilde{T}$ (7.14a) and pressure p (7.14b) on the free surface. Shown superimposed on the cutplanes are red cutlines located at $x_1 = 4.8$, along which detailed line plots of the temperature and pressure are given in Figs. 7.16a and 7.16b. The color bars denoting the magnitudes are not included here, the actual values are given on the cutline plots themselves.

three levels of refinement present. At its finest-allowable level, the "AMR(3)" grid is actually allowed to be one level finer than the finest $80 \times 80 \times 8$ uniform grid. In Figs. 7.15a and 7.15b, we have highlighted the approximate solutions in the peak flow region which occurs near the main convection cell in this configuration. We observe that, although the AMR grid has roughly the same horizontal resolution as either the $40 \times 40 \times 4$ or $80 \times 80 \times 8$ uniform grids in these regions, it does not do quite as well as the uniform grids at capturing the peak velocity value. (It generally does better than the coarsest-level $20 \times 20 \times 2$ mesh.)

This may be due to e.g. errors from nearby under-resolved solution

features or insufficient vertical resolution in the AMR solution. This illustrates the difficulty of using generic AMR schemes to approximate specific quantities of interest, such as the velocity at a point. Without more sophisticated error indicators it is difficult to determine which elements most strongly influence the computation of that quantity. On a global scale, we observe generally good qualitative agreement between the uniform- and adaptive-grid solutions. Unlike the stick-slip boundary, there are no singularities present and the scale of the free-surface flow is well-resolved by each of the present meshes.

In Fig. 7.16, we show the solution behavior for the temperature $\tilde{T}$ and pressure p on the free surface, along the line $x_1 = 4.8$. Here, the observations are roughly the same as for the velocity profiles of Fig. 7.15: the adaptive solution generally lies somewhere in between the $20 \times 20 \times 2$ and $40 \times 40 \times 4$ grids in capturing the minimum pressure value (which is correlated with the maximum temperature value found in the center of the convection cell). In Fig. 7.16a, we have highlighted the region where temperature contours intersect the wall at $x_2 = 6$. The contours here should satisfy the insulated boundary condition (in a weak sense) and indeed we observe that, under refinement, the slope of the contour lines at the boundary gets smaller. In this region, the adaptive solution tracks the finest-level $80 \times 80 \times 8$ mesh solution quite faithfully.

Some remarks on why the non-dimensional pressure variable is on a much different scale (as seen in Fig. 7.16b and other plots of the pressure) than either the velocity ($\mathcal{O}(10)$) or the temperature ($\mathcal{O}(1)$) variables are in order. First, let us consider the physical pressure value which is indicated by

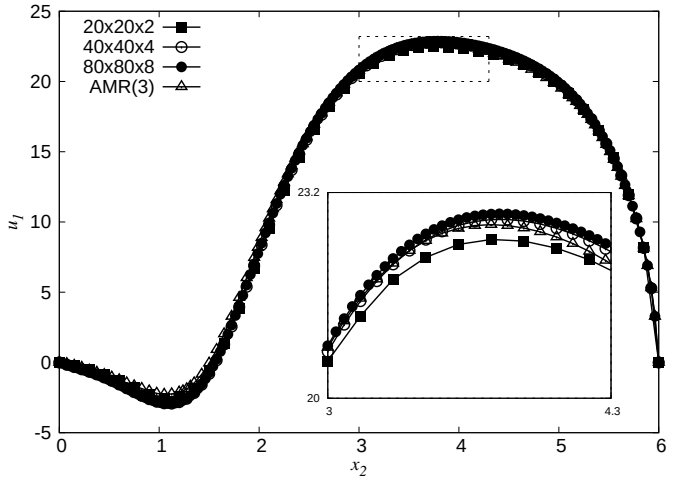
a non-dimensional pressure of $p^* = \mathcal{O}(10^3)$. Recall that the non-dimensional pressure $p^* := \frac{t_s}{\mu} p$, where p is the physical pressure, μ is the molecular viscosity and $t_s := \frac{\Gamma^2 d^2}{\alpha}$ is the time scale. The physical pressure is therefore given by (c.f. the non-dimensionalization discussion in §2.6)

$$p = \frac{\mu}{t_s} p^* = \frac{\mu\alpha}{\Gamma^2 d^2} p^* \tag{7.14}$$

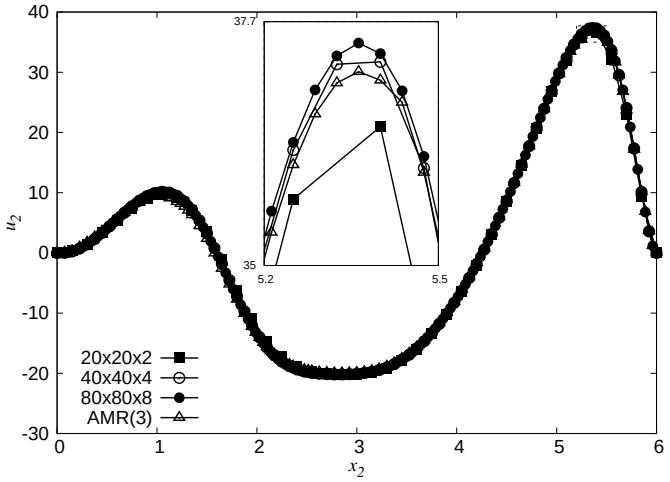
For the fluid and configuration of interest here ($\mu \approx 0.986$ g/cm-s, $\alpha \approx 0.001$ cm^2/s, $\Gamma = 6$, $d \approx 0.1$ cm), we have, for the physical pressure

$$p \approx 2.739 \times 10^{-3} p^* \qquad \left[\frac{\text{dyn}}{\text{cm}^2}\right] \tag{7.15}$$

Therefore, a nondimensional pressure value of say $p^* \approx 10^3$ yields a physical pressure of $\mathcal{O}(1)$ dyn/cm^2, or about 10^{-5} psi, a very small physical pressure deviation away from the hydrostatic field. This scaling presents a few challenges when attempting to compare the error in different variables, and when attempting to minimize nonlinear residual vectors whose components may differ by several orders of magnitude.

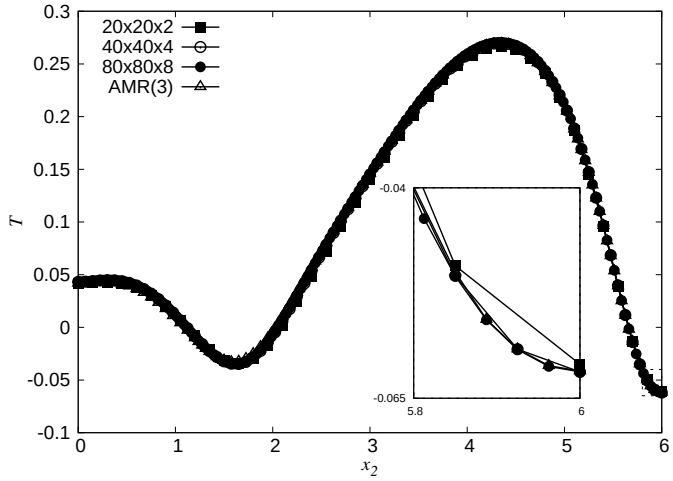


(a) Velocity component u_1, $x_1 = \text{const} = 4.8$, free surface

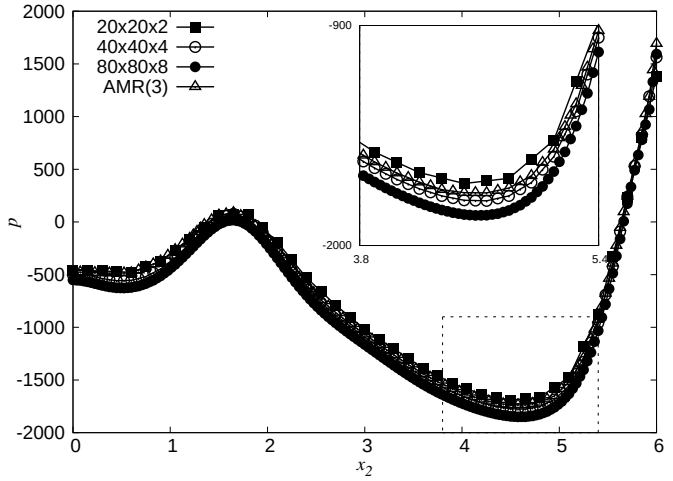


(b) Velocity component u_2, $x_1 = \text{const} = 4.8$, free surface

Figure 7.15: Plot of u_1 vs. x_2 (7.15a) and u_2 vs. x_2 (7.15b) for $x_3 = 1$ (on the free surface). The plot is taken the line $x_1 = 4.8$. The results for three uniform grids: $20 \times 20 \times 2$ (■), $40 \times 40 \times 4$ (◯), and $80 \times 80 \times 8$ (●) are shown along with an adaptive case "AMR(3)" (△) in which a maximum of three levels of adaptive refinement were allowed.



(a) Velocity component $\tilde{T}$, $x_1 = \text{const} = 4.8$, free surface

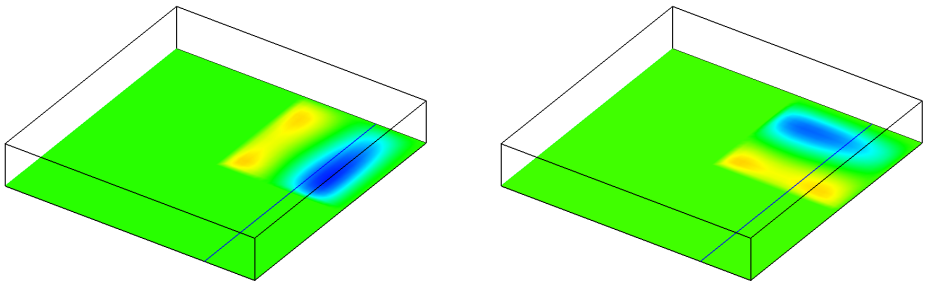


(b) Velocity component u_2, $x_1 = \text{const} = 4.8$, free surface

Figure 7.16: Plot of u_1 vs. x_2 (7.16a) and u_2 vs. x_2 (7.16b) for $x_3 = 1$ (on the free surface). The plot is taken the line $x_1 = 4.8$. The results for three uniform grids: $20 \times 20 \times 2$ (■), $40 \times 40 \times 4$ (◯), and $80 \times 80 \times 8$ (●) are shown along with an adaptive case "AMR(3)" (△) in which a maximum of three levels of adaptive refinement were allowed.

7.7.6 Comparison of Steady State Bottom Surface Cutlines

We now turn to a similar examination of steady state cutline variables, this time following a cutline in the bottom surface of the domain which intersects the stick-slip interface. Plots showing the cutline position within cutplanes of the x_1 and x_2 velocity components are given in Figs. 7.17a and 7.17b respectively, while a plot showing the position of the cutline in a pressure cutplane on the bottom surface is given in Fig. 7.18. The bottom surface is a homogeneous Dirichlet boundary for the temperature variable. The bottom surface cutplane for $\tilde{T}$ is therefore trivial and we have not included it here.



(a) u_1 (b) u_2

Figure 7.17: Cutplanes of x_1 (7.17a) and x_2 (7.17b) velocity (u_1 and u_2) on the bottom surface. Shown superimposed on the cutplanes are blue cutlines located at $x_1 = 4.8$, along which detailed line plots of the velocity are given in Figs. 7.19a and 7.19b. The color bars denoting the magnitudes are not included here, actual values are given on the cutline plots themselves. Non-zero "slip" flow appears only in the upper-right quadrant of the domain.

In Fig. 7.19, we give the solution behavior for the x_1 and x_2 velocity components on the bottom of the domain, along the line $x_1 = 4.8$. This

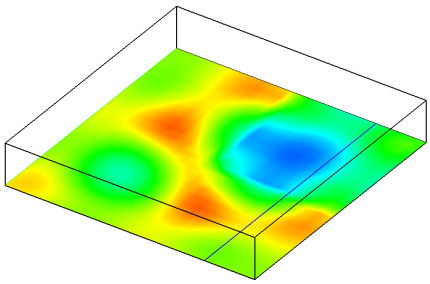
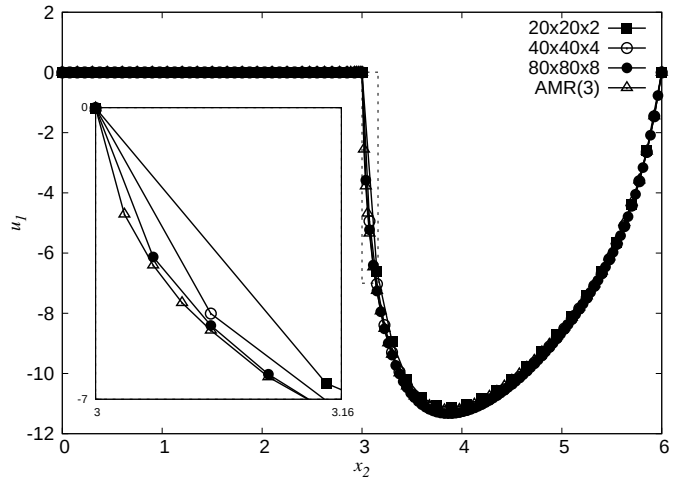


Figure 7.18: Cutplane of pressure p on the bottom surface. Shown superimposed on the cutplane is a blue cutline located at $x_1 = 4.8$, along which a detailed line plot of the pressure is given in Fig. 7.22. The color bar denoting the magnitude is not included here, the actual values are given on the cutline plot itself.

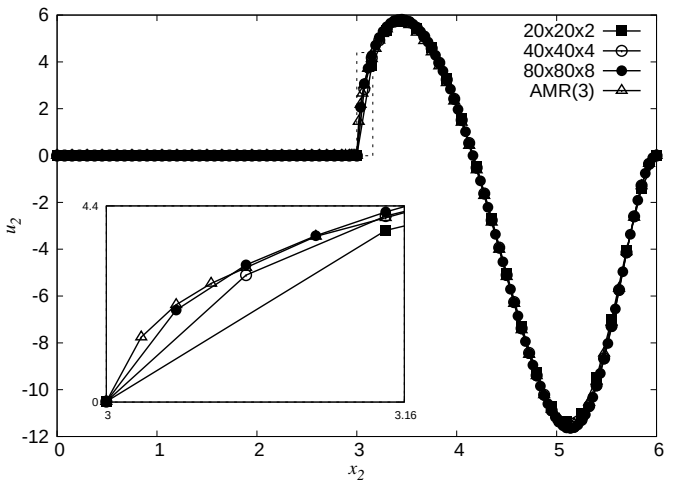
plot shows that the velocity fields change rapidly as we cross the line of "discontinuous" slip velocities at $x_2 = 3$. The inset plots in Figs. 7.19a and 7.19b are magnifications of the dashed line regions shown in the main figure. The uniform-grid solutions depart most drastically from the adaptive solutions here, due to the rapidly-changing nature of the velocity field components in these regions.

In the rest of the domain, good qualitative agreement between the uniform and adaptive solutions is obtained. This particular case illustrates the usefulness of adaptivity in automatically resolving localized solution features without incurring the excessive costs of uniform grid refinement. Similar results are obtained all along the line of discontinuous slip lengths. For example, in Fig. 7.20 we present another set of plots for the two horizontal velocity com-

ponents across the stick-slip interface, this time for $x_1 = \text{const} = 3.3$.

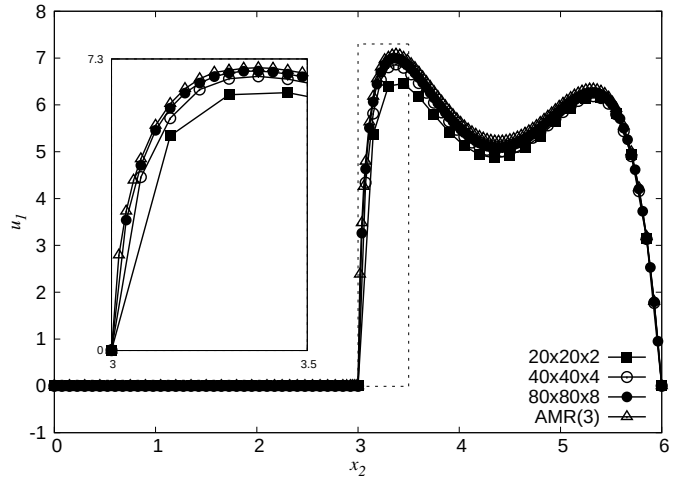


(a) Velocity component u_1, $x_1 = \text{const} = 4.8$, bottom surface

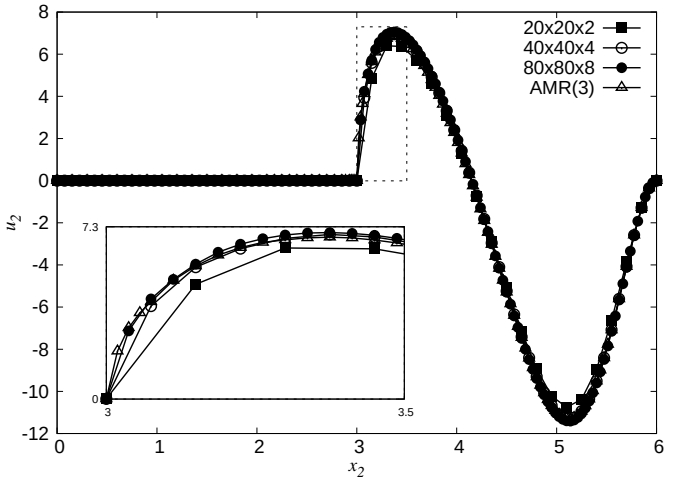


(b) Velocity component u_2, $x_1 = \text{const} = 4.8$, bottom surface

Figure 7.19: Plot of u_1 vs. x_2 (7.19a) and u_2 vs. x_2 (7.19b) across the stick-slip interface for $x_3 = 0$ (on the bottom surface). The plot is taken the line $x_1 = 4.8$. The results for three uniform grids: $20 \times 20 \times 2$ (■), $40 \times 40 \times 4$ (◯), and $80 \times 80 \times 8$ (●) are shown along with an adaptive case "AMR(3)" (△) in which a maximum of three levels of adaptive refinement were allowed.



(a) Velocity component u_1, $x_1 = \text{const} = 3.3$, bottom surface



(b) Velocity component u_2, $x_1 = \text{const} = 3.3$, bottom surface

Figure 7.20: Plot of u_1 vs. x_2 (7.20a) and u_2 vs. x_2 (7.20b) across the stick-slip interface for $x_3 = 0$ (on the bottom surface). The plot is taken the line $x_1 = 3.3$. The results for three uniform grids: $20 \times 20 \times 2$ (■), $40 \times 40 \times 4$ (◯), and $80 \times 80 \times 8$ (●) are shown along with an adaptive case "AMR(3)" (△) in which a maximum of three levels of adaptive refinement were allowed.

7.7.7 Some Remarks on the Nature of the Stick-Slip Singularity

It is well-documented that the flow field very near a two-dimensional "stick-slip" singularity [66, 84, 129, 130, 166] and in three-dimensional cases where a sliding wall intersects two non-moving walls [8, 87] is well-approximated by the steady, linear Stokes equations. In the two-dimensional case, it is possible to obtain an analytical solution to the Stokes equations which, in the limit $r \to 0$ (r here is the radial distance from the stick-slip interface), the velocity field behaves as $u \approx \sqrt{r}$. This result in turn implies that the tangential stress and the pressure field near such points exhibit $\frac{1}{\sqrt{r}}$ type (i.e. integrable) singularities.

Although the present flow has nonlinear convection and is coupled to an energy conservation equation, we might reasonably expect singular solution behavior similar to the Stokes case to result here as well. In addition, the gradient-jump based error indicator should be expected to react strongly to such a solution (exact solution not C^1 continuous) since the gradient jumps will not approach zero under h-refinement in this case. In Fig. 7.21 we present a log-log plot of the x_1 velocity variable near the stick-slip interface along a line of constant $x_1 = 4.8$.

While even the coarsest-level grid is sufficient to capture the global characteristics of the flow field (see other line plots) we observe that the uniform grids do not come very close to resolving the solution behavior near the stick-slip interface. As additional levels of adaptivity are allowed, the solution begins to more closely match the analytically-obtained $\sqrt{r}$ behavior, which on

Fig. 7.21 should appear as a straight line having a slope of $\frac{1}{2}$.

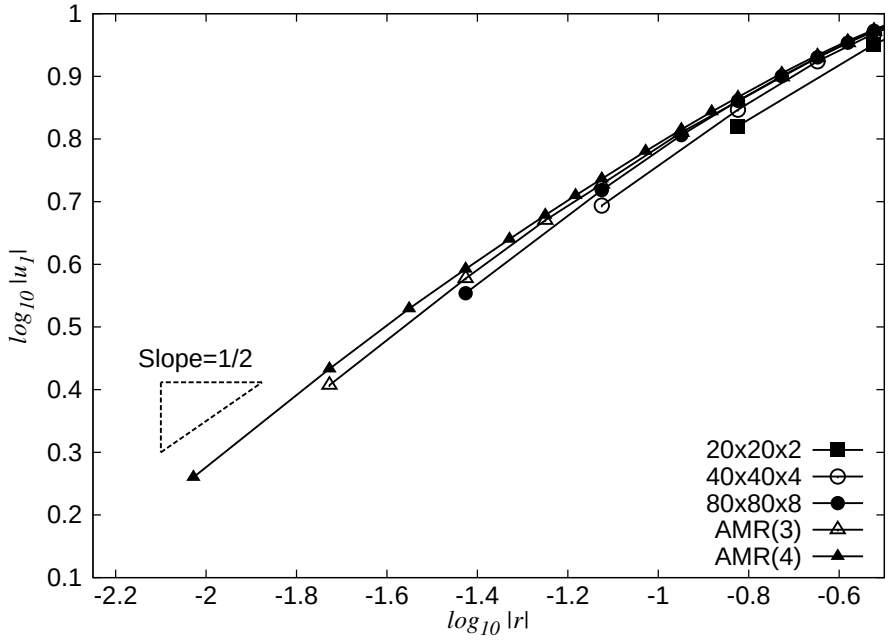


Figure 7.21: Log-log plot of u_1 vs. distance r from the stick-slip interface for $x_2 > 3, x_3 = 0$ (on the bottom surface). The plot is taken the line $x_1 = 4.8$. The results for three uniform grids: $20 \times 20 \times 2$ (■), $40 \times 40 \times 4$ (◯), and $80 \times 80 \times 8$ (●) are shown along with two adaptive cases in which maximums of three "AMR(3)" (△) and four "AMR(4)" (▲) levels of adaptive refinement were allowed.

We also note that the pressure p is a singular variable in this case with $\frac{1}{\sqrt{r}}$ type singularity across the discontinuous slip length interface. (Note: p and the velocity component gradients are still in L_2 (square-integrable) due to the additional r factor arising in the Jacobian of the integral in polar coordinates.) In general, attempting to approximate such a singular function with finite element methods yields a solution with degraded convergence properties. In Fig. 7.22 we give a plot of the pressure for several different grids to show the difficulties in attempting to approximate a singular function. In this case, the coarsest level $20 \times 20 \times 2$ mesh is too coarse to "see" the singularity and the solution does not exhibit any sharp over- or under-shoots.

Under uniform refinement and in the AMR case, the solution appears to tend toward $\pm\infty$ on either side of the singular interface while still remaining continuous. In a future work, it might be interesting to investigate the effects of using a discontinuous pressure "Q_2P_{-1}" [89] basis in approximating this problem. We also note that, far from the singularity, the AMR solution more closely tracks the coarse-grid pressure values than the finer grid values. This is most likely due to insufficient resolution of the AMR solution in these regions, but also suggests that improving the singularity approximation (which AMR appears to do) does not always guarantee an improved global approximation. This occurs because too much refinement becomes concentrated in the singular region (for the present error indicator) while the rest of the domain is under-refined.

To attempt to show the singular behavior of the pressure near the

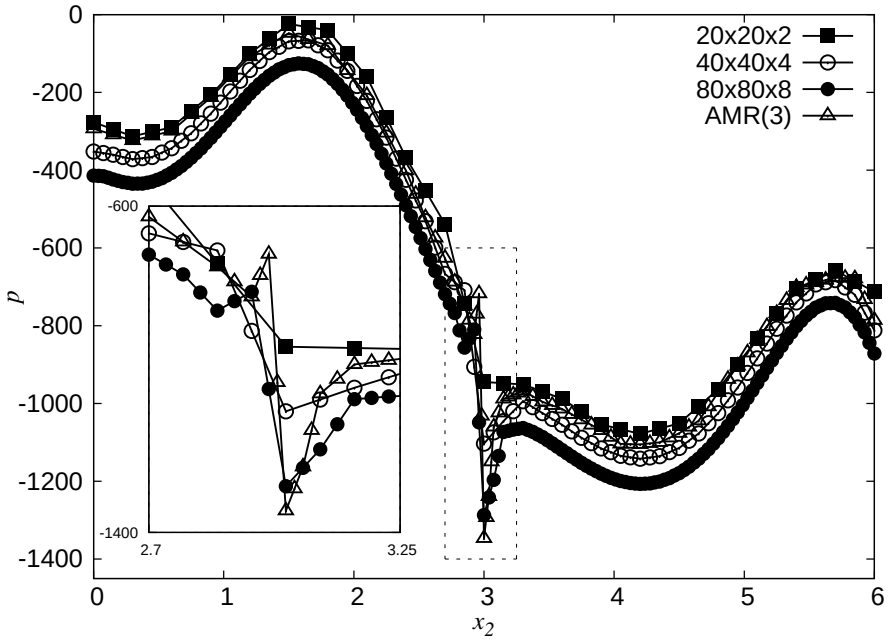


Figure 7.22: Plot of p vs. x_2 (7.22) across the stick-slip interface for $x_3 = 0$ (on the bottom surface). The plot is taken the line $x_1 = 4.8$. The results for three uniform grids: $20 \times 20 \times 2$ (■), $40 \times 40 \times 4$ (◯), and $80 \times 80 \times 8$ (●) are shown along with an adaptive case "AMR(3)" (△) in which a maximum of three levels of adaptive refinement were allowed.

discontinuous slip length interface, in Fig. 7.23 we have plotted $\log|p|$ vs. $\log|r|$, where $r := |x_2 - 3|$ is the distance away from the singular interface, for $x_2 > 3$. Unfortunately, in this case the data do not line up quite as well with the analytically-predicted slope of $-\frac{1}{2}$ as they do for the velocity case. This may be due to insufficient resolution, even with four levels of adaptivity, near the singular point.

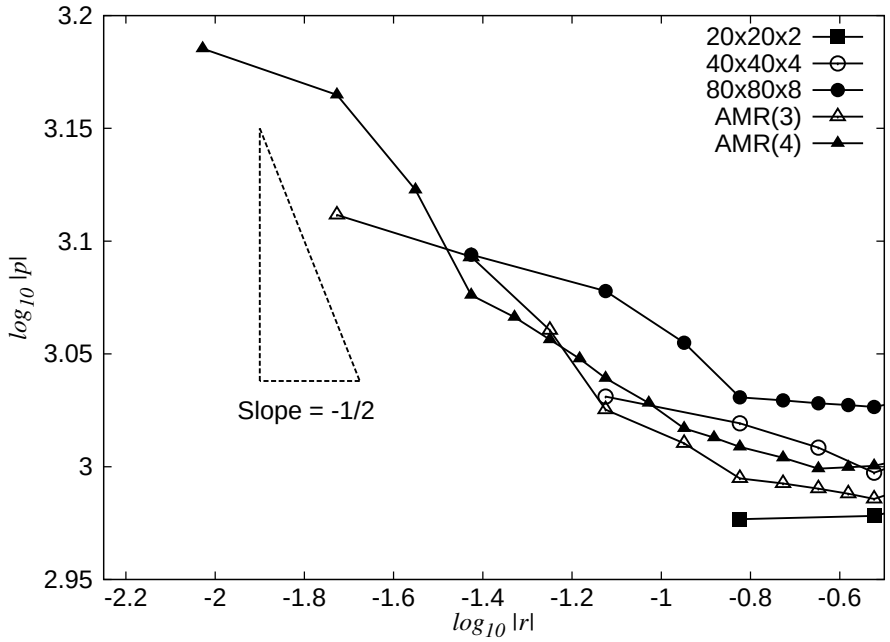


Figure 7.23: Log-log plot of p vs. distance r from the stick-slip interface for $x_2 > 3, x_3 = 0$ (on the bottom surface). The plot is taken the line $x_1 = 4.8$. The results for three uniform grids: $20 \times 20 \times 2$ (■), $40 \times 40 \times 4$ (◯), and $80 \times 80 \times 8$ (●) are shown along with two adaptive cases in which maximums of three "AMR(3)" (△) and four "AMR(4)" (▲) levels of adaptive refinement were allowed.

7.8 Steady AMR Results for Quantity of Interest

In order to test how well the adapted grids produced by our generic gradient-jump based error indicator would compute a quantity of interest, we applied AMR to the relatively challenging nine-cell steady state observed for $\Gamma \approx 8.928$. Our other "standard" values of $Ma = 92$, $Ra = 30$, $Bi = 0.2$, $Pr = 880$ were once again used. From this steady state, we computed several adaptive steps with coarsening turned off, using jumps in different solution variables to indicate the error. At each step, we kept track of two quantities of interest: the Nusselt number Nu, and the kinetic energy KE.

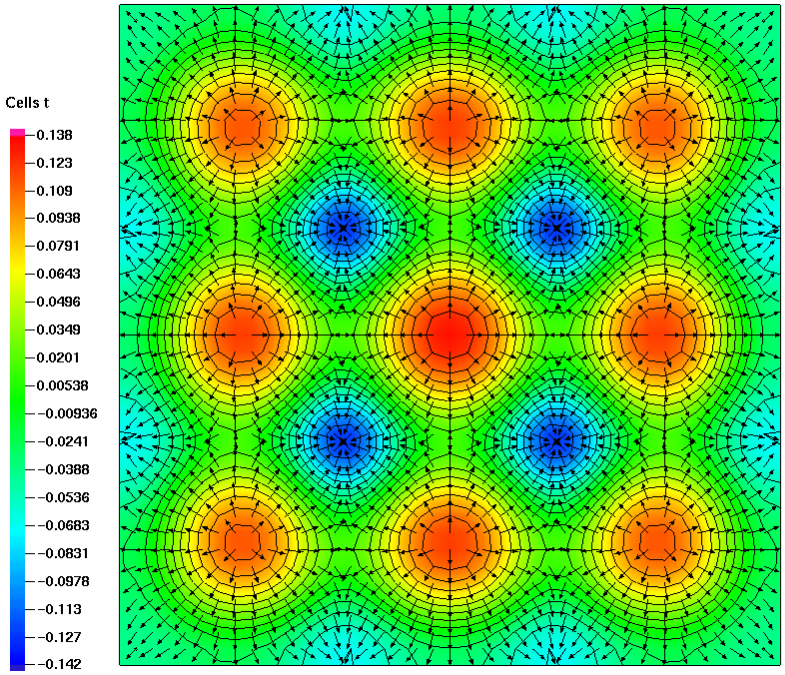


Figure 7.24: Symmetric 3×3-cell solution selected for uniform and adaptive refinement study comparing Nu and KE quantity of interest calculations.

We initially performed a large-scale uniform grid refinement study to achieve a solution accurate to four significant digits in Nu and about three digits in KE. The results of this study are presented in tabular form in Table 7.1. The largest of these uniform grid simulations employed PETSc's parallel block Jacobi preconditioner with ILU(t) on subdomains, and ran on 128 Xeon 5100 series dual-core 2.66 GHz CPUs for a little over 10 hours of wall time. The preconditioner in these cases was very important, since the number of iterations grows quite rapidly in three dimensions without the use of a high-quality preconditioner. Of course, there was also a trade-off in the amount of time required to compute the preconditioner.

Table 7.1: Convergence of the quantities of interest Nu and KE under uniform mesh refinement.

Mesh Size	N. Dofs	Nu	KE
10×10×10	38,375	0.0132632465177091	20.42543048964289
20×20×20	284,945	0.0168667414275728	29.94553204108120
40×40×40	2,194,685	0.0169420157344057	30.14719432161064
45×45×45	3,111,620	0.0169436730562605	30.14971090554227
50×50×50	4,253,855	0.0169446215119628	30.15096877982507

In Table 7.2 we present the results for refining on jumps in velocity components u_1, u_2, and u_3. At each step, a fixed percentage of elements with the highest indicated error is flagged for refinement. The results in this case are reasonable, with the quantity of interest achieving the same number of significant digits with only around 1.1 million elements. Unfortunately, the

results did not appear to be particularly robust: under repeated refinements, the error in the quantities of interest increases (see horizontal line separating steps 7 and 8) before eventually settling back down again at much higher-resolution grids. By step 14 of Table 7.2, there are likely elements in the mesh with smaller h than the finest uniform mesh attempted. Therefore, the results should be compared with some caution, as the adaptive mesh solution may actually become more accurate than the uniform grid solution at some point.

Table 7.2: Convergence of the quantities of interest Nu and KE under adaptive mesh refinement. The error indicator is the flux-jump based indicator described §7.5.1, considering jumps in the velocity components u_1, u_2, and u_3. There is reasonable convergence in the quantities of interest up to about step 7 (see horizontal line) but just beyond this point we see little to no improvement in the quantity of interest calculations until many more degrees of freedom are present in the solution.

Step No.	N. Dofs	Nu	KE
0	35,876	0.0163131241424295	28.69531970757616
1	65,189	0.0163461749938360	28.76202541543948
2	109,883	0.0167319564318534	29.43947428988276
3	184,508	0.0177792116614755	31.43168673870631
4	323,675	0.0171759441735921	30.49356328829025
5	587,471	0.0169095198555241	30.08513971371139
6	813,104	0.0169211853872427	30.10375638233517
7	1,112,603	0.0169420845257666	30.14094997862168
8	1,552,034	0.0169566517162645	30.16648807967294
9	2,146,364	0.0169715380331441	30.19443288268158
10	2,535,767	0.0169646079257598	30.18318757665123
11	3,012,641	0.0169478644813658	30.15545745334799
12	3,321,029	0.0169454723194000	30.15159483815279
13	3,656,444	0.0169454375086756	30.15159427633592
14	4,032,317	0.0169455076489806	30.15160179351342

In Table 7.3, we give the results for computing an error indicator based on jumps in the gradient of pressure, p. In this case, we did not see much of an improvement in the computation of the quantity of interest over the uniform case at all. We did not conduct additional calculations beyond about half a million degrees of freedom because of this.

Table 7.3: Convergence of the quantities of interest Nu and KE under adaptive mesh refinement. The error indicator is the gradient-jump based indicator described in §7.5.1, considering jumps in the pressure, p.

Step No.	N. Dofs	Nu	KE
0	35,876	0.0163131241424295	28.69531970757616
1	63,839	0.0166790780655216	29.43938391254895
2	107,492	0.0171816538594203	30.40755467265051
3	179,081	0.0167739027837273	29.76792587416629
4	318,575	0.0168692408473895	30.01308292041674
5	583,850	0.0169133159266340	30.09686495626169

In Table 7.4, we give the results for computing an error indicator based only on jumps in the gradient of temperature, $\tilde{T}$. As in the previous cases, although there is an initial improvement in the quantity of interest calculation, eventually the adapted grids are no better than the uniform grid solution. Refining on jumps in $\tilde{T}$ is about as good or slightly better than refining on jumps in the velocity components for this problem. The adaptive grids are actually somewhat worse, since they have an associated additional cost of computing the error indicator, projecting solutions, etc.

Table 7.4: Convergence of the quantities of interest Nu and KE under adaptive mesh refinement. The error indicator is the gradient-jump based indicator described in §7.5.1, considering jumps in the temperature, $\tilde{T}$.

Step No.	N. Dofs	Nu	KE
0	35,876	0.0163131241424295	28.69531970757616
1	63,452	0.0165481559141895	29.20528075398439
2	108,209	0.0167968754108370	29.77325272370992
3	182,645	0.0171353692138777	30.43087230151457
4	313,607	0.0169094413882463	30.09213359649814
5	566,771	0.0169365028517080	30.13612625640094
6	994,847	0.0169317432028866	30.12758711176039
7	1,763,192	0.0169452896789618	30.15043287458374

In Figs. 7.25 and 7.26 we plot the results of the preceding tables on a log-log scale in terms of the relative error in Nu (Fig. 7.25) and KE (Fig. 7.26) vs. the number of degrees of freedom. The uniform refinement data as well as the AMR data for jumps in gradients of velocity, temperature, and pressure are included. In general, the adaptive grids obtained with this indicator do not provide a compelling improvement over simple uniform refinement for the quantities of interest chosen here, though refining to jumps in $\tilde{T}$ appears to be slightly better than the other variables. Adaptively refining based on pressure appears to be the least useful, as it approximately tracks the uniform refinement data. Methods for automatically generating grids which improve the calculations of these and other quantities of interest (through the use of e.g. dual error indicators) remain an interesting area of future research.

In Figures 7.27–7.29 we present the first 12 adaptive mesh refinement steps for the case where refinement is driven by jumps in gradients of u_1, u_2, and u_3. The free surface mesh is shown in each figure, with the understanding that it plays an important role in the computation of Nu. In the early refinement steps, we observe the most adaptation taking place on the "cold" points on the surface, and a little around the top and bottom periphery of the warm spots. This trend continues under further refinement: the warm spots are consistently refined less aggressively than the cold spots, though once the mesh reaches a certain level it is difficult to say exactly where the new refinement is occurring since the mesh lines are quite close together. Also, recall that beyond adaptive step 7, the Nu quantity of interest had ceased converging to the "true" solution. It is possible that, because the warm spots play just as important a role in the computation of Nu, the fact that they are under-refined relative to the cold spots is one reason the accuracy of the Nu calculation suffers.

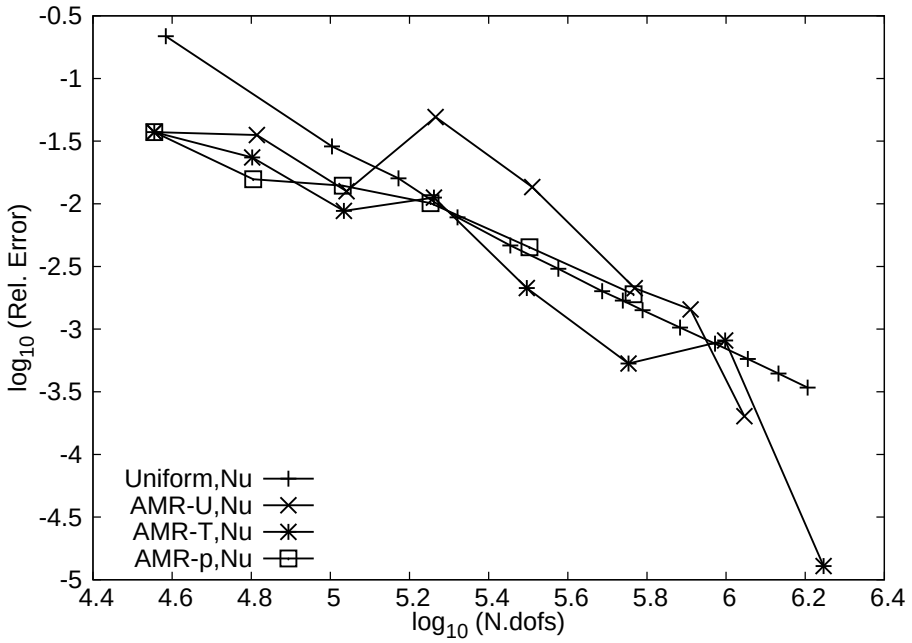


Figure 7.25: Comparison of relative error in the quantity of interest Nu under uniform and adaptive refinement. The adaptive mesh refinement procedure is driven by the generic gradient-jump based indicator described previously in this chapter, using the jumps in different variables to select elements for refinement. Coarsening was inactive during the calculations.

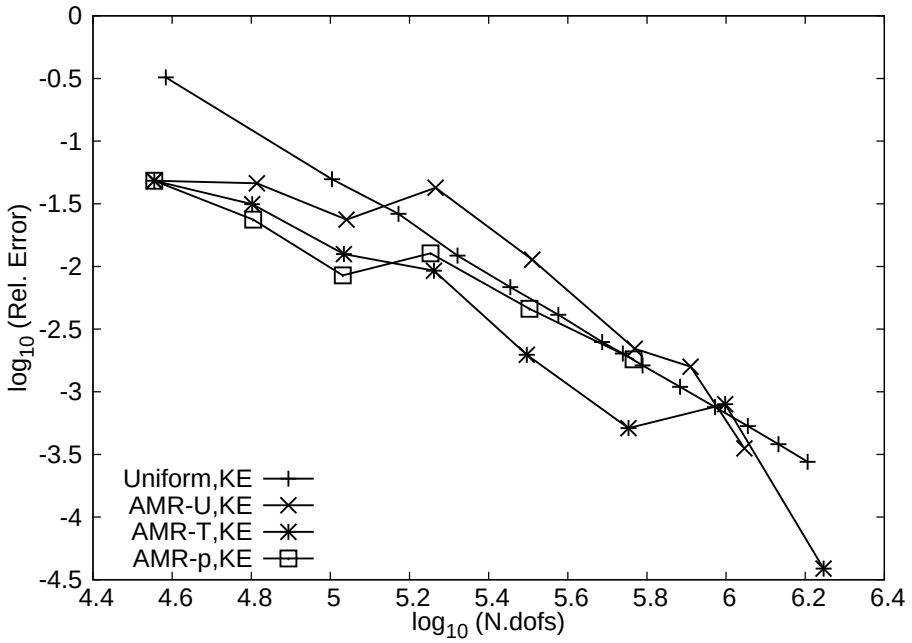
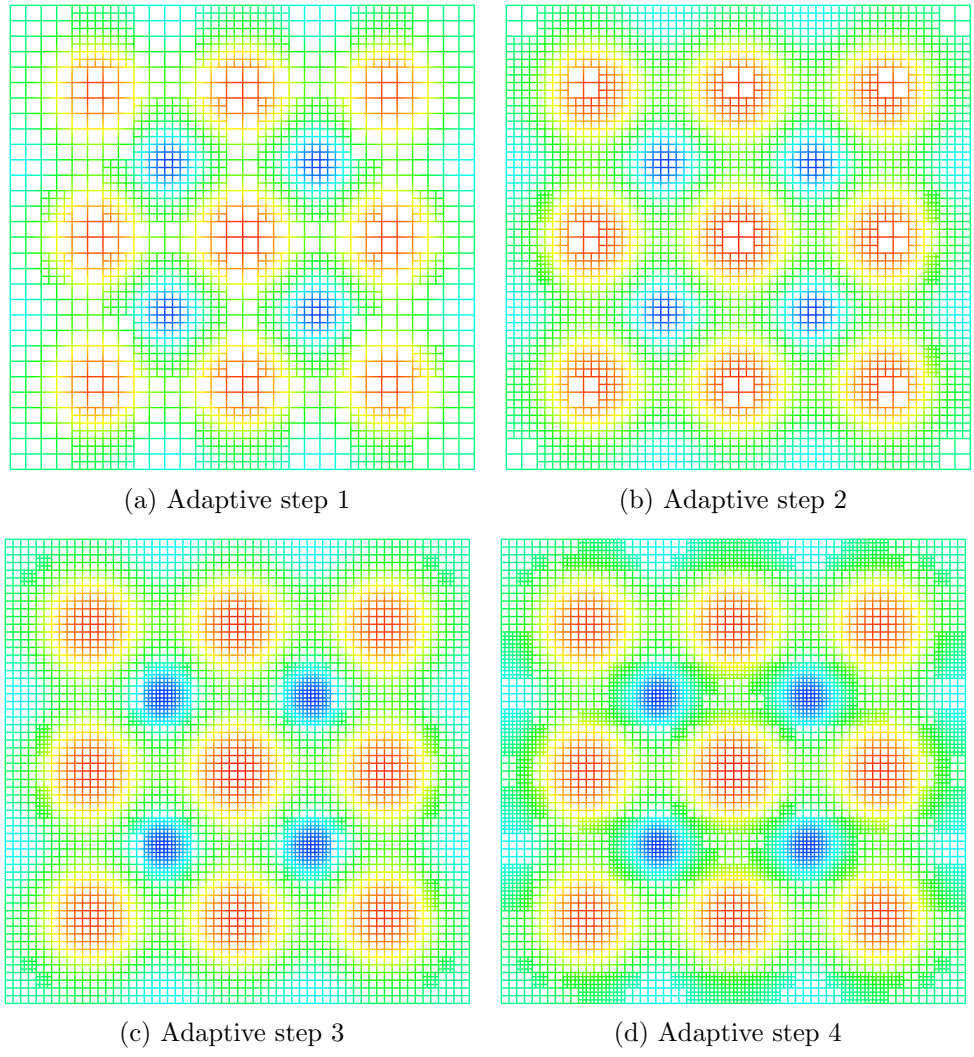


Figure 7.26: Comparison of relative error in the quantity of interest KE under uniform and adaptive refinement. The adaptive mesh refinement procedure is driven by the generic gradient-jump based indicator described previously in this chapter, using the jumps in different variables to select elements for refinement. Coarsening was inactive during the calculations.



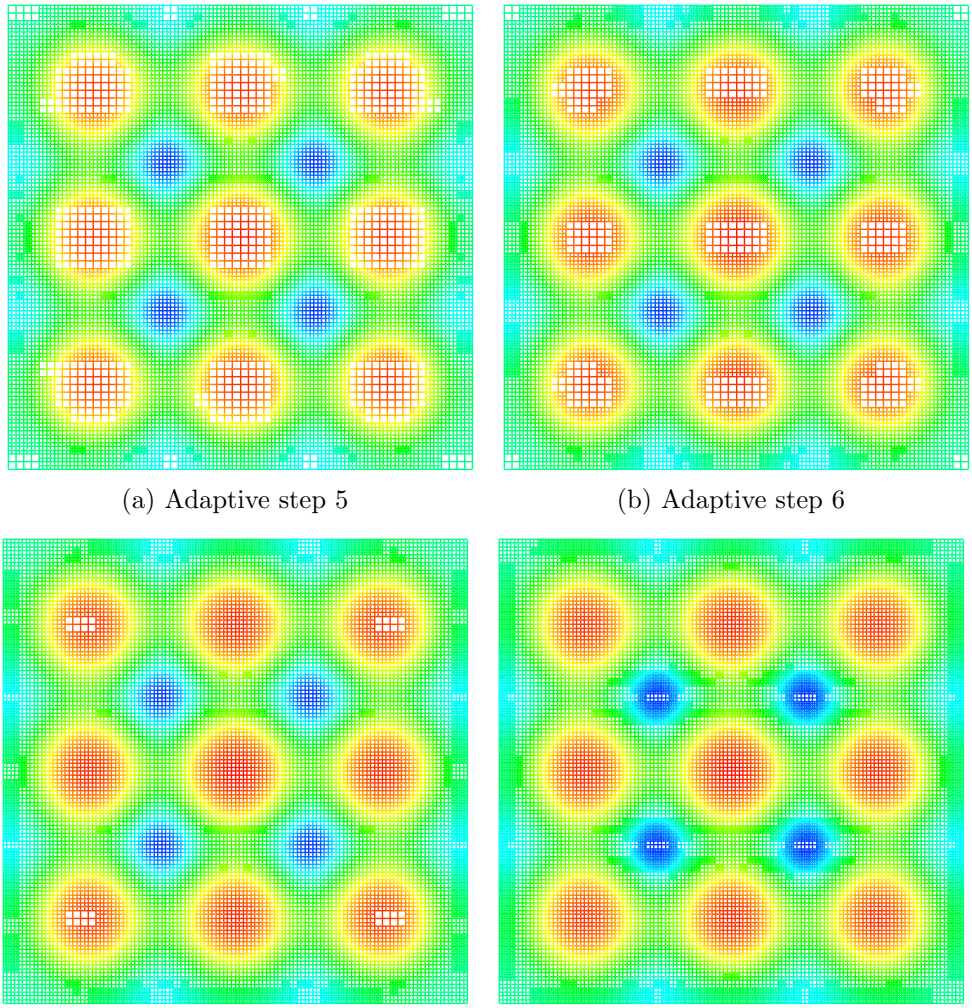
(a) Adaptive step 1

(b) Adaptive step 2

(c) Adaptive step 3

(d) Adaptive step 4

Figure 7.27: Sequence of adapted grids, AMR based on jumps in the gradients of u_1, u_2, and u_3.



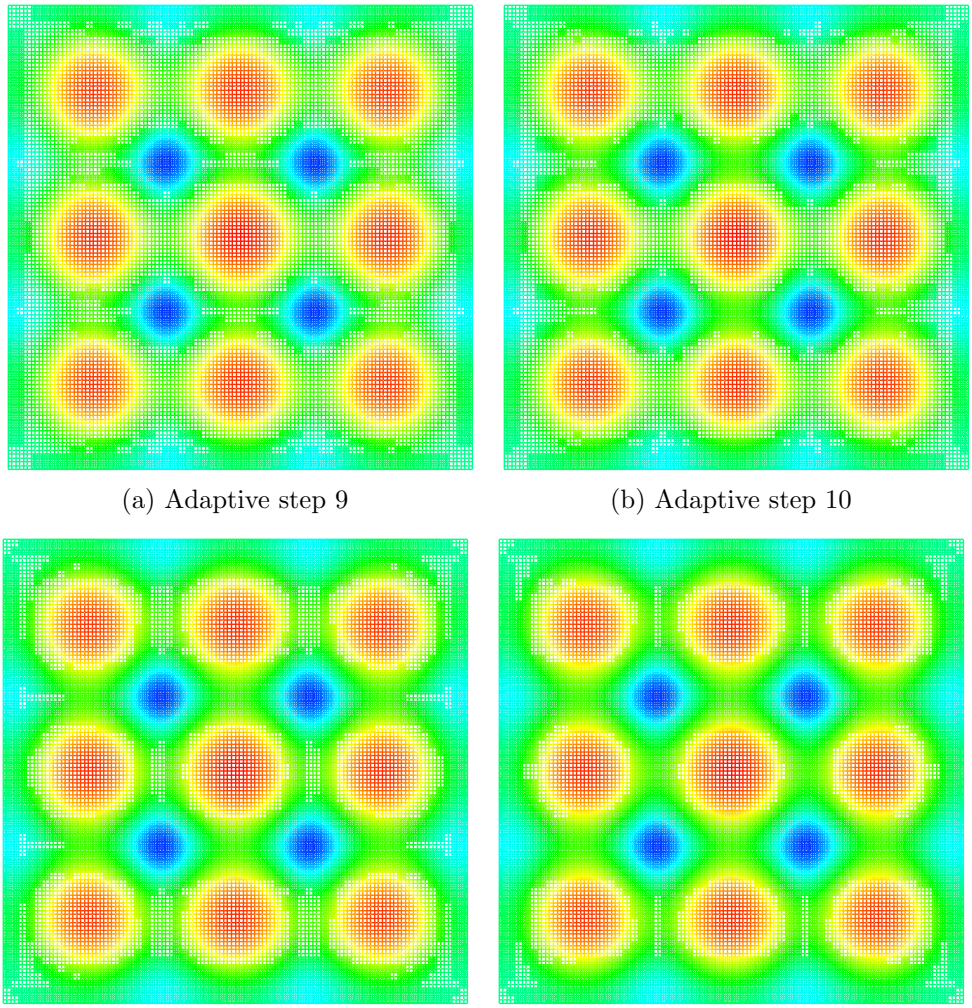
(a) Adaptive step 5

(b) Adaptive step 6

(c) Adaptive step 7

(d) Adaptive step 8

Figure 7.28: Sequence of adapted grids, AMR based on jumps in the gradients of u_1, u_2, and u_3.



(a) Adaptive step 9

(b) Adaptive step 10

(c) Adaptive step 11

(d) Adaptive step 12

Figure 7.29: Sequence of adapted grids, AMR based on jumps in the gradients of u_1, u_2, and u_3.

Chapter 8

Conclusions and Future Work

We believe the results of this research have shown that the scaled form of the coupled incompressible Navier-Stokes and energy conservation equations with the aspect ratio Γ as an independent parameter exhibit a great deal of promise and utility for computing solution branches in large regions of parametric space. During the course of our study, we observed computationally several experimentally obtained convection patterns, and also computed others not yet seen experimentally. Solutions having anywhere from one to eight convection cells were documented fully, and configurations with even higher numbers of convection cells were obtained as well.

Key results include the calculation of eigenvalues for determining the stability of steady state solutions found on branches in parametric space, as well as an extensive catalog of the relationship between thermal transfer (as measured by Nu) and aspect ratio for these configurations. In the solution branches themselves, we observed classical nonlinear dynamics features such as bifurcations, turning points, and hysteresis. The number and variability of the solutions to the governing equations in a geometry as simple as a box was quite interesting.

The preliminary adaptive mesh refinement results for the related problem involving Navier-type slip at the bottom boundary of the domain also appears to be quite interesting. The possibility of controlling these natural convection flows by modifying the boundary condition is an interesting one. The combined adaptive timestepping and AMR algorithm with custom adaptation intervals developed for this aspect of the work also appears to be a promising technique for simulating related flows with a similar stick-slip type singularity.

8.1 Future Work

In this section, we discuss some of the promising avenues of future work opened up by the present research which were either too time-consuming to tackle in the present work or require additional theoretical development before they can be attempted. This section is arranged in the following manner: in §8.1.1 we briefly discuss some additional simulations which we feel would extend upon the base of results generated in the present work. Then, in §8.1.2 we discuss what benefits might be gained from additional work on improved error indicators for the RBM convection problem. Finally, we give a brief discussion of the relatively open-ended topic of optimal shape design in the context of RBM convection in §8.1.3.

8.1.1 Further Simulations

Although the present work examined a fairly large region of the Γ (and to a lesser extent, Ma) parameter space, we feel there is still a great deal of interesting work to be done tracing out solution branches. For example, several of the branches in this work which the eigensolver predicted to be linearly unstable, such as the symmetric 2×2 solution branch, certainly bear further investigation. This would include but not be limited to uniform mesh refinement studies (especially in terms of the convergence of eigenvalues) as well as algorithms for performing adaptive mesh refinement while tracing solution branches.

Other simulations of interest would include non-square (yet still suitably scaled) polygonal geometries such as the equilateral triangle and circular domains investigated by Koschmieder. One parameter not investigated at all in this work is the Biot number Bi. We suspect the solutions are very sensitive to this number, and therefore experimental uncertainties in the determination of this number justify additional simulations to investigate that parameter space. There is also the important technique of extended systems and perturbations away from critical points. In this case, the position of a turning point in (say) (Γ, Nu) space is tracked as a function of the variation of one of the other parameters, say Ma. Methods for solving such extended systems have been available in the literature [172] for some time and would be a natural extension of the finite element pseudo-arclength continuation method developed for this work.

A topic which may be of considerable importance, and which has been touched on only briefly several times throughout the present discussion, is that of the deforming surface. It is expected that, in small aspect ratio containers, surface deformations will play a much more important role than they do in their large aspect ratio counterparts. In this work, we have assumed an undeformed surface not only for the sake of implementational simplicity, but also because, at least in a qualitative sense, the absence of a deforming surface did not seem to adversely affect the convection patterns arising in the calculations. A novel and promising avenue of future inquiry would be the development of a numerical scheme solving for the fully-coupled, two-dimensional deforming surface and comparing those results to the ones obtained in the present work.

Also, it appears that we have only scratched the surface on a number of interesting simulations which could be conducted using the Navier-slip boundary condition. Some examples include time-dependent Navier-slip and what periodic solution structures could be induced by such a boundary condition, slip on the container sides, and the use of Navier-slip to generate new staring points for solution branches in parameter space. For example, it appears that the four-cell solution for $\Gamma = 6$ is stable when we approach it starting from a Navier-slip four-cell solution by suddenly turning off the slip. This appears to be equivalent to approaching the solution from a more "energetic" level, and it is possible that the steady, stable states in such cases are different from those obtained when approaching from a less energetic (perturbed from zero) state.

8.1.2 Improved Error Indicators

An area where the results obtained in this work could be extended and improved upon is that of improved error indicators for the quantities of interest KE and Nu. One popular technique is the so-called adjoint or dual-based error indicators. These indicators are still an area of open research, especially for nonlinear, time-dependent problems. A particularly difficult aspect is making these indicators cost effective: in the worst case the dual problem can be even more expensive to solve than the primal one, requiring e.g. a higher-order discretization.

As observed in this work, the explicit, generic gradient-jump based error indicator works reasonably well in problems where there is an obvious "shock" or discontinuous feature, and computing a particular quantity of interest is not a goal of the simulation. On the other hand, when improved quantity of interest calculations *are* required, the generic indicator obviously falls short. An interesting avenue of future research would be the development of an adjoint-based error indicator for nonlinear, time-dependent applications such as RBM convection, which is computationally efficient, and can be applied in a somewhat generic manner, making it suitable for library use.

8.1.3 Optimal Shape Design

Another interesting field which may have a promising intersection with RBM convection is that of optimal shape design. Several authors have published extensively in this area, including Pironneau [131–133, 152–154] who

focuses primarily on applied optimal shape design in the area of fluid mechanics (e.g. the design of optimal airfoils). More mathematical and theoretical introductions are provided in the books by Haslinger and Mäkinen [92] and Dorin Bucur [39] in the area of variational methods. Kohli and Carey [109] considered the problem of reducing the computational effort required in the numerical computation of gradients for sensitivity information by using adaptive techniques.

According to Bucur, "A shape optimization problem is a minimization problem where the unknown variable runs over a class of domains." In his notation, we seek to minimize a cost function $F(A)$ over a class $\mathcal{A}$ of admissible domains A, i.e.

$$\min_{A \in \mathcal{A}} F(A) \tag{8.1}$$

The cost functional typically depends on the solution to a PDE constraint which in turn depends on the shape of the domain itself. There is generally no guarantee that an optimal domain exists unless the problem is sufficiently geometrically constrained.

A classic example is described by Haslinger [92] for Laplace's problem on a "curved rectangle," a domain with three straight sides and one side free to be "designed," which is parametrized by Bézier functions. The PDE and boundary conditions correspond to the temperature distribution in a two-dimensional body heated by a uniform, constant heat source. There is a fixed (Dirichlet) temperature specified along the curved boundary and the straight sides are insulated. The functional to minimize is the "energy" or L_2-norm

of the solution. The problem has no solution even when a "total volume" constraint is imposed on the domain: the curved boundary segment simply becomes more and more oscillatory as the number of design variables (Bézier segments) used to discretize it is increased.

From a physical point of view this makes sense: the most heat will be dissipated by a "fin" with infinite surface "area" (perimeter in two dimensions). To maintain the fixed volume constraint, the domain must oscillate more and more rapidly to increase its perimeter. The problem can be "solved" by further constraining the admissible geometry, for example by limiting the maximum slope at any point on the designed side, or by specifying a max-allowable perimeter for the side in addition to the volume constraint. This simple example shows that, even for a linear problem, one must be extremely careful when posing the problem in order to ensure that it will have a solution. According to Bucur [39], a general problem will be more likely to have an "optimal" solution if there are "severe" geometrical constraints imposed on the class of admissible domains. Some examples include convexity and the maximum perimeter constraint for the fin problem posed previously.

The present work on RBM convection has focused primarily on domains of square cross-section, but it raises the question of whether there may be an optimal (polygonal, etc.) shape for facilitating heat transfer through small aspect ratio fluid layers. Any investigation into the possibility of an optimal shape would of course first require a well-posed problem. For example, shapes are typically defined by a finite set of Bézier coefficients, but this approach

does not allow one to study directly the behavior in polygonal domains with varying numbers of sides. The problem would also need to be appropriately constrained (as discussed above) for a solution to exist. Nevertheless we feel this is an exciting area of further development, and the high-performance, parallel finite element code designed in the present work provides a reasonable starting point for such an investigation.

Appendices

Appendix 1

Rayleigh-Bénard-Marangoni Flow

In this appendix, additional details on the Rayleigh-Bénard-Marangoni equations and their solution are given. The outline of the sections is as follows. In §1.1 we describe briefly the classical development of the incompressible energy conservation equation. In §1.2 the precise form of the governing equations required for working with traction boundary conditions is discussed. In §1.4 we give a short description of how the free surface thermal boundary condition is derived and used in the governing equations. Finally, in §1.5, various derivatives of the governing equations with respect to problem parameters are given for reference.

1.1 Conservation of Energy for Incompressible Fluids

The incompressible conservation of energy equation, written in terms of the total energy per unit mass e, is

$$\rho \frac{De}{Dt} = \nabla \cdot (\boldsymbol{\sigma} \boldsymbol{u}) + \rho \boldsymbol{g} \cdot \boldsymbol{u} + \nabla \cdot (k \nabla T) \qquad \text{(A-1.1)}$$

where D/Dt is the material or substantial derivative, and the total energy per unit mass is defined as

$$e = \check{U} + \frac{1}{2}|\boldsymbol{u}|^2 \tag{A-1.2}$$

where $\check{U}$ is the "internal" energy per unit mass and $\frac{1}{2}|\boldsymbol{u}|^2$ is the total kinetic energy per unit mass. By "dotting" the incompressible conservation of momentum equation (Eqn. (2.1)) with the velocity $\boldsymbol{u}$ and subtracting it from Eqn. (A-1.1), we obtain, for the conservation of internal energy

$$\rho \frac{D\check{U}}{Dt} = \boldsymbol{\sigma} : \nabla \boldsymbol{u} + \nabla \cdot (k \nabla T) \tag{A-1.3}$$

Now, for an incompressible fluid, it is well-known that the internal energy per unit mass is a function of the temperature and the entropy per unit mass S, $\check{U} = \check{U}(S, T)$, and furthermore that $\frac{\partial \check{U}}{\partial T} = c$, where c is the (constant) heat capacity [69]. For an incompressible fluid, we need not differentiate between specific heat at constant volume c_v and constant pressure c_p, since they are the same. Employing these thermodynamic relationships, the internal energy conservation equation becomes

$$\rho c \frac{DT}{Dt} = \nabla \cdot (k \nabla T) + \Phi \tag{A-1.4}$$

where $\Phi := \mu \boldsymbol{\varepsilon}(\boldsymbol{u}) : \nabla \boldsymbol{u}$ is the known as the "viscous dissipation" term, and is negligible for the types of flows considered in this work. Dropping Φ from Eqn. (A-1.4) leads directly to Eqn. (2.3), the form of conservation of energy employed here.

1.2 Traction Boundary Conditions

In this section, we attempt to motivate the form of the traction boundary conditions for the Rayleigh-Bénard-Marangoni problem using the equations for steady Stokes flow as an example (see e.g. [144] for a derivation)

$$-\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{f} \tag{A-1.5}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{A-1.6}$$

where, as before, $\boldsymbol{u}$ is the (unknown) velocity field and $\boldsymbol{f}$ is a generic body force. The total stress tensor, $\boldsymbol{\sigma}$, is defined by

$$\boldsymbol{\sigma} := -p\boldsymbol{I} + \boldsymbol{\tau} \tag{A-1.7}$$

where viscous stress tensor, $\boldsymbol{\tau}$ is (for a Newtonian fluid)

$$\boldsymbol{\tau} := -\frac{2}{3}\mu(\nabla \cdot \boldsymbol{u})\boldsymbol{I} + \mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{A-1.8}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{u}) := \nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^t$ is twice the symmetric part of the velocity gradient. We use here the standard "row-wise" definition for the divergence of a tensor, i.e. in index notation, we have

$$\nabla \cdot \boldsymbol{\sigma} := \frac{\partial \sigma_{ij}}{\partial x_j} \tag{A-1.9}$$

where σ_{ij} is the $(i,j)^{th}$ entry of $\boldsymbol{\sigma}$ and summation over the repeated index j is implied. Note that we have retained the $\nabla \cdot \boldsymbol{u}$ term in the viscous stress tensor definition, even though the continuity Eqn. (A-1.6) implies it should drop. The viscous stress tensor with the $\nabla \cdot \boldsymbol{u}$ term dropped is the form used

in Eqn. (2.4) in §2.2. The reason for keeping this term will be discussed in more depth later. At this point, the derivation of the variational form of the Stokes equations can go in one of at least two different directions:

1. Laplacian form — the divergence operator is "multiplied through" before dotting the equations with a test function. This is the most common case, and yields a specific form of the natural boundary condition.

2. Traction form — Eqns. (A-1.5) and (A-1.6) are first dotted with a test function and the divergence theorem is applied. This formulation yields the full traction boundary condition as well as some additional terms in the governing equations which are not present in the previous case.

We will first discuss the more common Laplacian form before moving on to the more general traction form of the problem.

1.2.1 Laplacian Form

Expanding the divergence operator in Eqn. (A-1.5) yields

$$\nabla p + \mu \left[\frac{2}{3} \nabla(\nabla \cdot \boldsymbol{u}) - \Delta \boldsymbol{u} - \nabla \cdot (\nabla \boldsymbol{u})^t \right] = \boldsymbol{f} \qquad \text{(A-1.10)}$$

For the last term in square brackets on the left-hand side, we have

$$\begin{aligned} (\nabla \cdot (\nabla \boldsymbol{u})^t)_i &= \frac{\partial}{\partial x_j} \frac{\partial u_j}{\partial x_i} \\ &= \frac{\partial}{\partial x_i} \frac{\partial u_j}{\partial x_j} \\ &= \frac{\partial}{\partial x_i} (\nabla \cdot \boldsymbol{u}) \end{aligned}$$

which can be combined with the first term in square brackets to obtain

$$\nabla p - \mu \left[\frac{1}{3} \nabla (\nabla \cdot \boldsymbol{u}) + \Delta \boldsymbol{u} \right] = \boldsymbol{f} \tag{A-1.11}$$

The variational form of Eqn. (A-1.11) is now obtained by dotting with an appropriate test function $\boldsymbol{v}$, integrating over the domain Ω, and employing the divergence theorem

$$\int_\Omega \mu \left[\nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \frac{1}{3} (\nabla \cdot \boldsymbol{u})(\nabla \cdot \boldsymbol{v}) \right] - p(\nabla \cdot \boldsymbol{v}) \, dx =$$
$$\int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, dx + \int_{\partial\Omega} \left[\mu \nabla \boldsymbol{u} \cdot \boldsymbol{v} \cdot \boldsymbol{n} + \left(\frac{1}{3} \mu (\nabla \cdot \boldsymbol{u}) - p \right) (\boldsymbol{v} \cdot \boldsymbol{n}) \right] ds \tag{A-1.12}$$

where the matrix dyadic product notation $A{:}B := A_{ij} B_{ij}$ is used and where we have used the identity

$$\Delta \boldsymbol{u} \cdot \boldsymbol{v} = \nabla \cdot (\nabla \boldsymbol{u} \cdot \boldsymbol{v}) - \nabla \boldsymbol{u} : \nabla \boldsymbol{v} \tag{A-1.13}$$

which may be easily-verified using index notation. Then, through the vector identity

$$\begin{aligned} \nabla \boldsymbol{u} \cdot \boldsymbol{v} \cdot \boldsymbol{n} &= \left((\nabla \boldsymbol{u})^t \boldsymbol{v} \right)^t \boldsymbol{n} \\ &= \boldsymbol{v}^t (\nabla \boldsymbol{u} \boldsymbol{n}) \\ &= (\nabla \boldsymbol{u} \boldsymbol{n}) \cdot \boldsymbol{v} \end{aligned}$$

and by making the substitution $\nabla \cdot \boldsymbol{u} = 0$ we obtain the variational equation for the Laplacian form of the Stokes momentum equations

$$\int_\Omega \mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} - p(\nabla \cdot \boldsymbol{v}) \, dx =$$
$$\int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, dx + \int_{\partial\Omega} \left[\mu \nabla \boldsymbol{u} - p \boldsymbol{I} \right] \boldsymbol{n} \cdot \boldsymbol{v} \, ds \tag{A-1.14}$$

In particular, note that the boundary term in Eqn. (A-1.14) is *not* $\int_{\partial\Omega}(\boldsymbol{\sigma}\cdot\boldsymbol{n})\cdot\boldsymbol{v}$, which may be crucial depending on the type of boundary conditions to be applied.

1.2.2 Traction Form

To construct the traction form of the Stokes equations, the strong form of the momentum Eqn. (A-1.5) is dotted with a test function $\boldsymbol{v}$ and we integrate over the domain Ω to obtain

$$-\int_{\Omega}(\nabla\cdot\boldsymbol{\sigma})\cdot\boldsymbol{v}\,dx=\int_{\Omega}\boldsymbol{f}\cdot\boldsymbol{v}\,dx \tag{A-1.15}$$

For tensor $\boldsymbol{\sigma}$ and vector $\boldsymbol{v}$ we have the identity

$$\nabla\cdot(\boldsymbol{\sigma}\boldsymbol{v})=\left(\nabla\cdot\boldsymbol{\sigma}^{t}\right)\cdot\boldsymbol{v}+\boldsymbol{\sigma}^{t}:\nabla\boldsymbol{v} \tag{A-1.16}$$

This identity, combined with the fact that $\boldsymbol{\sigma}$ is symmetric in the present problem, allows us to substitute for $(\nabla\cdot\boldsymbol{\sigma})\cdot\boldsymbol{v}$ in Eqn. (A-1.15). Employing the divergence theorem, we obtain

$$\int_{\Omega}\boldsymbol{\sigma}:\nabla\boldsymbol{v}\,dx=\int_{\Omega}\boldsymbol{f}\cdot\boldsymbol{v}\,dx+\int_{\partial\Omega}(\boldsymbol{\sigma}\cdot\boldsymbol{n})\cdot\boldsymbol{v}\,ds \tag{A-1.17}$$

(On the boundary, we used the identity $\boldsymbol{\sigma}\boldsymbol{v}\cdot\boldsymbol{n}=(\boldsymbol{\sigma}\cdot\boldsymbol{n})\cdot\boldsymbol{v}$ as in the Laplacian form derivation.) Substituting for $\boldsymbol{\sigma}$ in the interior integral, we obtain

$$\int_{\Omega}\mu\left[\nabla\boldsymbol{u}+(\nabla\boldsymbol{u})^{t}\right]:\nabla\boldsymbol{v}-p(\nabla\cdot\boldsymbol{v})-\frac{2}{3}\mu(\nabla\cdot\boldsymbol{u})(\nabla\cdot\boldsymbol{v})\,dx=\int_{\Omega}\boldsymbol{f}\cdot\boldsymbol{v}\,dx+\int_{\partial\Omega}(\boldsymbol{\sigma}\cdot\boldsymbol{n})\cdot\boldsymbol{v}\,ds \tag{A-1.18}$$

Finally, substituting for the divergence-free condition, we obtain the full traction boundary condition form of the Stokes equations

$$\int_\Omega \mu \left[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^t\right] : \nabla \boldsymbol{v} - p(\nabla \cdot \boldsymbol{v})\, dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, dx + \int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \boldsymbol{v}\, ds \tag{A-1.19}$$

In contrast to Eqn. (A-1.14), we note that in Eqn. (A-1.19) we have the full traction boundary condition, and we also have the additional term $(\nabla \boldsymbol{u})^t : \nabla \boldsymbol{v}$. We can show that Eqns. (A-1.14) and (A-1.19) are *identical* under the divergence-free condition, since we have that

$$\begin{aligned} \int_\Omega (\nabla \boldsymbol{u})^t : \nabla \boldsymbol{v}\, dx &= -\int_\Omega \left(\nabla \cdot (\nabla \boldsymbol{u})^t\right) \cdot \boldsymbol{v}\, dx + \int_{\partial\Omega} (\nabla \boldsymbol{u})^t \cdot \boldsymbol{v} \cdot \boldsymbol{n}\, ds \\ &= -\int_\Omega \nabla(\nabla \cdot \boldsymbol{u}) \cdot \boldsymbol{v}\, dx + \int_{\partial\Omega} \nabla \boldsymbol{u} \boldsymbol{v} \cdot \boldsymbol{n}\, ds \\ &= 0 + \int_{\partial\Omega} (\nabla \boldsymbol{u})^t \boldsymbol{n} \cdot \boldsymbol{v}\, ds \end{aligned} \tag{A-1.20}$$

Subtracting this term from the boundary term in Eqn. (A-1.19), we arrive at the boundary term in Eqn. (A-1.14). If a Dirichlet boundary condition for $\boldsymbol{u}$ is provided on the entire $\partial\Omega$, then $\boldsymbol{v}|_{\partial\Omega} = \boldsymbol{0}$ will be imposed, and it is possible to use Eqns. (A-1.14) and (A-1.19) interchangeably (even though there will be a (possibly undesirable) cross-coupling between the velocity degrees of freedom for Eqn. (A-1.19)). If, on the other hand, full traction boundary conditions are meant to be imposed (e.g. in the case of thermocapillary surface tension boundary conditions) it seems Eqn. (A-1.19) must be used so that $\boldsymbol{\sigma} \cdot \boldsymbol{n}$ can be specified directly.

1.3 Penalty Boundary Conditions

We again employ the Stokes flow equations

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma} &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \tag{A-1.21}$$

in our discussion of the application of penalty boundary conditions and their relationship to Navier's slip condition. In addition to requiring Eqn. (A-1.21) hold in the interior Ω, we also assume a specified velocity $\boldsymbol{u} = \boldsymbol{g}$ is imposed on the boundary $\partial\Omega$. The "standard" Galerkin formulation is then: find $(\boldsymbol{u}, p) \in \mathcal{S} \times \mathcal{Q}$ such that

$$\int_\Omega \boldsymbol{\sigma} : \nabla \boldsymbol{v} \, dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, dx + \int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \boldsymbol{v} \, ds \tag{A-1.22}$$

$$\int_\Omega (\nabla \cdot \boldsymbol{u}) \, q \, dx = 0 \tag{A-1.23}$$

holds for all $(\boldsymbol{v}, q) \in \mathcal{V} \times \mathcal{Q}$. The spaces $\mathcal{V}$ and $\mathcal{Q}$ are related and cannot be chosen totally arbitrarily, but here we will focus only on the velocity space and how its boundary conditions are set. The test space $\mathcal{V}$ is usually chosen as the set of vector functions whose components are in $H_0^1(\Omega)$, that is, have square-integrable first derivatives and vanish on $\partial\Omega$. The trial or solution space $\mathcal{S}$ is then defined as $\mathcal{S} = \boldsymbol{g} + \mathcal{V}$, i.e. vector functions from the same class as the test functions and which satisfy the boundary condition. In this formulation, the boundary term in Eqn. (A-1.22) vanishes due to the choice of the space $\mathcal{V}$, and in practice the degrees of freedom associated with the Dirichlet boundary nodes are either condensed out of the final system matrix or left in with appropriately-modified row and column entries.

An alternative formulation is obtained by not restricting $\boldsymbol{v} \in \mathcal{V}$ to vanish on the boundary, and by replacing the boundary term in Eqn. (A-1.22) by

$$\int_{\partial\Omega} (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \boldsymbol{v} \, ds = \frac{1}{\epsilon} \int_{\partial\Omega} (\boldsymbol{u} - \boldsymbol{g}) \cdot \boldsymbol{v} \, ds \tag{A-1.24}$$

where $\epsilon \ll 1$ is the penalty parameter. This formulation ensures that any deviation in $\boldsymbol{u}$ away from $\boldsymbol{g}$ on the boundary will generate a large residual and therefore fail to satisfy the variational statement. The full penalty boundary condition variational statement is then: find $(\boldsymbol{u}, p) \in \mathcal{V} \times \mathcal{Q}$ such that

$$\int_{\Omega} \boldsymbol{\sigma} : \nabla \boldsymbol{v} \, dx = \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{v} \, dx + \frac{1}{\epsilon} \int_{\partial\Omega} (\boldsymbol{u} - \boldsymbol{g}) \cdot \boldsymbol{v} \, ds \tag{A-1.25}$$

$$\int_{\Omega} (\nabla \cdot \boldsymbol{u}) \, q \, dx = 0 \tag{A-1.26}$$

holds for all $(\boldsymbol{v}, q) \in \mathcal{V} \times \mathcal{Q}$, where now $\mathcal{V}$ consists of vector functions with components in $H^1(\Omega)$, that is, which do not necessarily vanish on the boundary.

While the penalty boundary condition may be applied relatively generically to many systems of PDE, for Stokes and Navier-Stokes flow it turns out to be directly related to what is known as the Navier slip condition. First reported by Navier in 1823 [134] and later independently confirmed by Maxwell in 1879 [124], the Navier slip condition states simply that the amount of tangential "slip" at a solid boundary (i.e. the amount by which the velocity of a solid boundary and the adjacent fluid differ) is directly proportional to the rate of shear at the boundary, with constant of proportionality $\mathcal{L}$, known as the slip length. Letting $\boldsymbol{t}$ be a vector in the tangent plane of the solid boundary,

for the Navier slip condition we have the relation

$$(\boldsymbol{u} - \boldsymbol{u}_{wall}) \cdot \boldsymbol{t} = \mathcal{L}\,(\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \boldsymbol{t} \tag{A-1.27}$$

(We note that the condition Eqn. (A-1.27) can be generalized to apply in the wall-normal direction as well, though with a different (much smaller) slip length in that case, to prevent flow "through" the solid wall.) Comparing Eqn. (A-1.27) with Eqn. (A-1.24) we immediately observe an equivalency when the penalty parameter $\epsilon = \mathcal{L}$. Selecting $\epsilon \ll 1$ may thus be interpreted physically as specifying an arbitrarily small slip length. In some finite element codes it is much more convenient to impose the Navier slip/penalty boundary condition than it is to condense the boundary condition degrees of freedom or modify the matrix rows and columns, and thus it is encouraging that such a boundary condition has a physical justification.

The possibility of having a non-zero slip length naturally raises the following questions. When does such an approximation make physical sense? What is a reasonable value for $\mathcal{L}$? Some of the most common applications in which boundary slip is important are microscale and nanoscale flows. In such flows, the liquid film thickness near a solid boundary is often comparable to the size of the molecules which make up the liquid itself. For example, Koplik and Banavar [110] conducted a molecular dynamics (MD) simulation of a microscale version of the classical lid-driven cavity problem. The simulation was performed assuming liquid Argon molecules modeled by Lennard-Jones potentials having a characteristic length $L = 3.4$ angstroms (3.4×10^{-10} m),

characteristic mass of $m = 40$ AU (1 AU $= 9.1093826 \times 10^{-31}$ kg), and characteristic energy $e = 120\text{K} \times k_B$. These parameters yield a characteristic atomistic time scale of $\tau = L(m/e)^{1/2} \approx 2.12 \times 10^{-12}$ s.

The simulations were conducted in a very small container having a side length of only 58 angstroms. Koplik found that the no-slip assumption started to break down when the normalized shear stress $\dot{\gamma}\tau$ was of $\mathcal{O}(1)$, in other words, when the applied shear stress at the wall was very high indeed. Koplik described the behavior of the fluid in this case as being "analogous to shear-thinning behavior in polymeric fluids," but found that the standard Navier slip condition as we have stated it here was not perfectly correlated with the presence of slip in his calculations. He suggests that the technique of circumventing corner singularities using a fixed microscopic slip length is therefore not correct. Later work by Thompson and Troian [192] confirmed that the assumption of a constant slip length in such problems indeed breaks down at high (normalized) rates of shear and proposed a universal, singular, nonlinear formula for the slip length onto which their MD simulation data collapsed, given the correct empirical constants.

Other important applications of the Navier slip boundary condition include the moving contact line problem [59, 74, 159] which occurs when a liquid film is spreading across a solid surface, flows near sharp corners [122, 130], and flow through small capillaries [167]. The phenomenon of electroosmotically driven flow, in which apparent fluid slip occurs at the boundary of a microdevice due to the presence of a thin double layer of electrons may also be

thought of as an application of Navier slip. Zhao [215] describes a device for mixing a fluid employing periodically modulated electric potentials. Another class where boundary slip can be significant is flow through chemically-treated hydrophobic channels [155, 209, 216]. In particular, Tretheway [193] has observed significant fluid slip while conducting very careful measurements to within 450 nm of such a hydrophobic surface. The review article by Neto et al. [135] provides an excellent overview of the many different experimental studies on slip boundaries which have been conducted.

We are aware of no studies to date in which the combined effects of fluid slip at the walls and thermocapillary surface tension (RBM convection) on free surfaces have been investigated. However, it is our conjecture that this may indeed be an interesting class of multiphysics problem because surface forces and boundary slip can both be important in extremely thin liquid layers. For example, the combination of an electroosmotically driven flow with a free surface seems particularly interesting.

1.4 Free Surface Thermal Boundary Condition Derivation

Here we briefly discuss the derivation of the reduced temperature free surface thermal boundary condition. Because we assumed that the "air layer" above the liquid is always conducting, for any temperature profile T in the liquid layer (conducting or convecting) it holds that

$$-k\frac{\partial T}{\partial z}\bigg|_{z=d} = \frac{k_{\text{air}}}{d_{\text{air}}}\left(T|_{z=d} - T_c\right) \tag{A-1.28}$$

where $T|_{z=d}$ is the temperature at the liquid-gas interface. Starting from the definition of $\tilde{T}$

$$\tilde{T} := T - T_0 \tag{A-1.29}$$

we can multiply Eqn. (A-1.29) by $-k$, differentiate with respect to z, and evaluate the result at $z = d$ to obtain

$$-k\frac{\partial \tilde{T}}{\partial z}\bigg|_{z=d} = -k\frac{\partial T}{\partial z}\bigg|_{z=d} - \left(-k\frac{\partial T_0}{\partial z}\bigg|_{z=d}\right) \tag{A-1.30}$$

Substituting Eqn. (A-1.28) and the conducting air layer assumption of Eqn. (2.21) into the right-hand side of Eqn. (A-1.30) gives

$$\begin{aligned} -k\frac{\partial \tilde{T}}{\partial z}\bigg|_{z=d} &= \frac{k_{\text{air}}}{d_{\text{air}}}\left(T|_{z=d} - T_c\right) - \frac{k_{\text{air}}}{d_{\text{air}}}\left(T_0|_{z=d} - T_c\right) \\ &= \frac{k_{\text{air}}}{d_{\text{air}}}\left(T|_{z=d} - T_0|_{z=d}\right) \\ &= \frac{k_{\text{air}}}{d_{\text{air}}}\,\tilde{T}\bigg|_{z=d} \end{aligned} \tag{A-1.31}$$

as claimed.

1.5 Scaled Eqns. Derivative with Respect to Parameters

In this section, we give the derivatives (with respect to parameters Γ, Bi, and Ma) of the steady ($\theta = 1$) "scaled" RBM equations, i.e. the set of governing equations in which the vertical and horizontal length scales have been treated differently. These terms are of importance when computing improved initial guesses during continuation schemes, see e.g. §4. For the derivative with

respect to Γ, for $\alpha = 1, 2, 3$, we have

$$\begin{aligned}
\frac{\partial F_i^{(u_\alpha)}}{\partial \Gamma} &= \int_\Omega \frac{1}{Pr} u_3 \frac{\partial u_\alpha}{\partial x_3} \phi_i \, dx \\
&+ \int_\Omega 2\Gamma \frac{\partial u_\alpha}{\partial x_3} \frac{\partial \phi_i}{\partial x_3} \, dx \\
&+ \sum_{\beta=1}^{3} \int_\Omega \left(\delta_{\alpha 3} + \delta_{\beta 3}\right) \Gamma^{(\delta_{\alpha 3} + \delta_{\beta 3} - 1)} \frac{\partial u_\beta}{\partial x_\alpha} \frac{\partial \phi_i}{\partial x_\beta} \, dx \\
&+ \int_\Omega 3\Gamma^2 Ra \tilde{T} \hat{g}_\alpha \phi_i \, dx \\
&- \int_\Omega \delta_{\alpha 3} \Gamma^{(\delta_{\alpha 3} - 1)} p \frac{\partial \phi_i}{\partial x_\alpha} \, dx \\
&+ \int_{\partial \Omega_s} 2\Gamma Ma \frac{\partial \tilde{T}}{\partial x_\alpha} \phi_i \, ds
\end{aligned} \tag{A-1.32}$$

While, for the pressure equation, we obtain

$$\frac{\partial F_i^{(p)}}{\partial \Gamma} = \int_\Omega \frac{\partial u_3}{\partial x_3} \psi_i \, dx \tag{A-1.33}$$

and for the energy conservation equation

$$\begin{aligned}
\frac{\partial F_i^{(\tilde{T})}}{\partial \Gamma} &= \int_\Omega \left(u_3 \frac{\partial \tilde{T}}{\partial x_3} + u_3 \frac{\partial T_0}{\partial x_3} \right) \phi_i \, dx \\
&+ \int_\Omega 2\Gamma \frac{\partial \tilde{T}}{\partial x_3} \frac{\partial \phi_i}{\partial x_3} \, dx \\
&+ \int_{\partial \Omega_s} 2\Gamma Bi \, \tilde{T} \phi_i \, ds
\end{aligned} \tag{A-1.34}$$

For completeness, we also give the derivative, in terms of the Biot number, Bi, and Marangoni number, Ma, of the discrete equations.

$$\frac{\partial F_i^{(\tilde{T})}}{\partial \left(Bi\right)} = \int_{\partial \Omega_s} \Gamma^2 \tilde{T} \phi_i \, ds \tag{A-1.35}$$

$$\frac{\partial F_i^{(u_\alpha)}}{\partial \left(Ma\right)} = \int_{\partial \Omega_s} \Gamma^2 \frac{\partial \tilde{T}}{\partial x_\alpha} \phi_i \, ds \tag{A-1.36}$$

Appendix 2

Numerical Methods

Here we give additional details about well-known numerical methods which are commonly-used in conjunction with finite element codes.

2.1 The θ-Method

The θ-method is a standard technique for time-discretization of systems of first-order ODEs. While it is not usually considered "adaptive" in the strictest sense, in practice we can choose the parameter θ dynamically to achieve additional "stability" (implicit Euler, $\theta = 1$), or conversely additional accuracy (Crank-Nicolson, $\theta = 1/2$). (While all of the implicit θ-methods are unconditionally "A_0" stable, the Crank-Nicolson method lacks so-called "L_0" stability, and is known to suffer from the phenomenon of "ringing" (as is well-described in the reference work by Smith [191]) in cases where the boundary and initial conditions do not coincide. This tendency for unwanted oscillations to grow in size is exacerbated by decreasing the spatial grid size h — clearly an undesirable property on adaptively-refined grids.

While the θ-method (or any of the subsidiary methods derivable from

the θ-method) is typically derived for standard systems of ODEs in the form

$$\dot{\boldsymbol{U}} = \boldsymbol{f}(\boldsymbol{U}, t) \tag{B-2.1}$$

where $\boldsymbol{U}$ is an $N \times 1$ vector of unknowns and $\boldsymbol{f} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$ is a given function. In the course of studying systems of nonlinear natural convection PDEs discretized by the finite element method in space, it is actually much more common to deal with systems of ODEs of the form

$$\boldsymbol{M}(\boldsymbol{U})\dot{\boldsymbol{U}} = \boldsymbol{f}(\boldsymbol{U}, t) \tag{B-2.2}$$

where $\boldsymbol{M}$ is the so-called "mass matrix," which may depend on the unknown $\boldsymbol{U}$ (as in some stabilized schemes) and may not be invertible due to the presence of time-independent constraint equations, such as e.g. the incompressibility constraint. Obviously if $\boldsymbol{M}$ is constant and formally invertible then Eqn. (B-2.2) can be rewritten in the same form as Eqn. (B-2.1) and the usual techniques can be applied. Here, we focus entirely on deriving methods for the non-constant mass matrix case of Eqn. (B-2.2).

To motivate the θ-method, we consider time interval n, for which $t \in [t_n, t_{n+1}]$ and for which $t_{n+1} = t_n + \Delta t$ defines the timestep Δt. Assuming $\boldsymbol{U}$ is smooth enough in this time interval, we can expand $\boldsymbol{U}$ in independent Taylor series about t_n and t_{n+1} as

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t\dot{\boldsymbol{U}}^n + \frac{\Delta t^2}{2}\ddot{\boldsymbol{U}}^n + \mathcal{O}(\Delta t^3) \tag{B-2.3}$$

$$\boldsymbol{U}^n = \boldsymbol{U}^{n+1} - \Delta t\dot{\boldsymbol{U}}^{n+1} + \frac{\Delta t^2}{2}\ddot{\boldsymbol{U}}^{n+1} + \mathcal{O}(\Delta t^3) \tag{B-2.4}$$

The explicit (resp. implicit) Euler method is obtained from Eqn. (B-2.3) (resp. Eqn. (B-2.4)) by dropping terms of $\mathcal{O}(\Delta t^3)$ and higher, and substituting for $\dot{\boldsymbol{U}}$ from Eqn. (B-2.1) or (B-2.2). In the first-order case, the mass matrix inverse is used formally during this substitution step, but is subsequently "multiplied out" in the final step so that it does not appear in the final scheme. For completeness, we give the explicit and implicit Euler schemes for Eqn. (B-2.2) here

$$\text{Explicit:} \quad \boldsymbol{M}\left(\boldsymbol{U}^n\right)\left(\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^n}{\Delta t}\right) = \boldsymbol{f}(\boldsymbol{U}^n, t_n) + \mathcal{O}(\Delta t) \tag{B-2.5}$$

$$\text{Implicit:} \quad \boldsymbol{M}\left(\boldsymbol{U}^{n+1}\right)\left(\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^n}{\Delta t}\right) = \boldsymbol{f}(\boldsymbol{U}^{n+1}, t_{n+1}) + \mathcal{O}(\Delta t) \tag{B-2.6}$$

The θ-method is derived for the constant mass matrix case by observing that an advantageous cancellation of the Taylor series truncation error is obtained when selecting a particular linear combination of Eqns. (B-2.3) and (B-2.4). A similar procedure is possible in the non-constant mass matrix case as well, but additional requirements on $\boldsymbol{M}$ are necessary to ensure that the resulting method will indeed be second-order. For simplicity, let $\boldsymbol{M}^n := \boldsymbol{M}(\boldsymbol{U}^n)$ and $\boldsymbol{f}^n := \boldsymbol{f}(\boldsymbol{U}^n, t_n)$. Then, by multiplying Eqn. (B-2.3) by $(1-\theta)\boldsymbol{M}^n$ and Eqn. (B-2.4) by $-\theta\boldsymbol{M}^{n+1}$ and adding them together, we obtain

$$\begin{aligned}\left((1-\theta)\boldsymbol{M}^n + \theta\boldsymbol{M}^{n+1}\right)\left(\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^n}{\Delta t}\right) &= (1-\theta)\,\boldsymbol{f}^n + \theta\boldsymbol{f}^{n+1} \\ &+ \text{T.E.}\end{aligned} \tag{B-2.7}$$

where the truncation error has the particular form

$$\text{T.E.} := \Delta t \left[\left(\frac{(1-\theta)}{2} \boldsymbol{M}^n - \frac{\theta}{2} \boldsymbol{M}^{n+1} \right) \ddot{\boldsymbol{U}}^{n+1} + \mathcal{O}(\Delta t) \right] \tag{B-2.8}$$

Unlike the constant mass matrix case, setting $\theta = 1/2$ (the Crank-Nicolson scheme) is not sufficient to obtain a second-order accurate method in time unless we can also show that

$$\boldsymbol{M}^{n+1} = (1 + \mathcal{O}(\Delta t)) \boldsymbol{M}^n \tag{B-2.9}$$

For the nonlinear natural convection problems of interest here, the restriction of Eqn. (B-2.9) will obviously be satisfied for any constant mass matrix (or even mass matrices with zero blocks, as in the case of unstabilized Rayleigh-Bénard-Marangoni flows and double-diffusive convection in porous media.)

It is more difficult to analyze the mass matrices which arise due to nonlinear stabilization terms. In general, the stabilization parameter τ can depend on the unknown $\boldsymbol{U}$ in a highly-nonlinear way, and these effects must be analyzed on a case-by-case basis. We note that these considerations on $\boldsymbol{M}$ in no way affect the accuracy of the first-order schemes, and so one must carefully justify the expense of assembling the additional right-hand side terms for the Crank-Nicolson method by showing (either analytically or by numerical experimentation) that the resulting scheme is truly second-order accurate.

A variation on the preceding scheme is obtained by considering Taylor series *not* about the end points $t \in [t_n, t_{n+1}]$ of the time interval, but instead about a somewhat arbitrary intermediate time

$$t_\theta := \theta t_{n+1} + (1-\theta)\, t_n \tag{B-2.10}$$

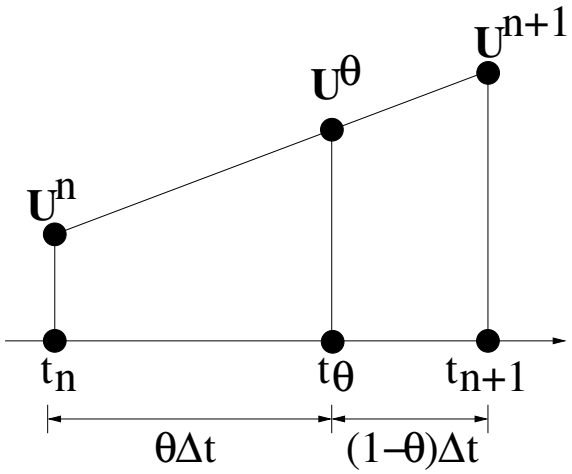


Figure B-2.1: Definition of the intermediate value $\boldsymbol{U}^\theta$.

where $0 \leq \theta \leq 1$ as before. By assuming the solution dependence is linear between the two endpoint values (see Fig. B-2.1) it follows that the solution at time t_θ is given by

$$\boldsymbol{U}^\theta := \theta \boldsymbol{U}^{n+1} + (1-\theta)\,\boldsymbol{U}^n \tag{B-2.11}$$

The selection of t_θ in this manner splits the full timestep Δt up into two parts of size $\theta\Delta t$ and $(1-\theta)\Delta t$, as shown in the figure. We may now once again construct two Taylor series expansions of $\boldsymbol{U}$, this time both will be centered about the $\boldsymbol{U}^\theta$ solution.

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^\theta + (1-\theta)\Delta t \dot{\boldsymbol{U}}^\theta + \frac{(1-\theta)^2\Delta t^2}{2}\ddot{\boldsymbol{U}}^\theta + \mathcal{O}(\Delta t^3) \tag{B-2.12}$$

$$\boldsymbol{U}^{n} = \boldsymbol{U}^\theta - \theta\ \Delta t \dot{\boldsymbol{U}}^\theta + \frac{\theta^2\Delta t^2}{2}\ddot{\boldsymbol{U}}^\theta + \mathcal{O}(\Delta t^3) \tag{B-2.13}$$

Subtracting Eqn. (B-2.13) from Eqn. (B-2.12) we find

$$\boldsymbol{U}^{n+1} - \boldsymbol{U}^{n} = \Delta t \dot{\boldsymbol{U}}^{\theta} + \frac{\Delta t^2}{2}(1-2\theta)\ddot{\boldsymbol{U}}^{\theta} + \mathcal{O}(\Delta t^3) \tag{B-2.14}$$

Finally, multiplying Eqn. (B-2.14) through by $\frac{1}{\Delta t}\boldsymbol{M}_{\theta}$, and substituting

$$\boldsymbol{M}_{\theta}\dot{\boldsymbol{U}}^{\theta} = \boldsymbol{f}(\boldsymbol{U}^{\theta}, t_{\theta}) \tag{B-2.15}$$

from the original ODE of Eqn. (B-2.2), we obtain

$$\boldsymbol{M}_{\theta}\left(\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^{n}}{\Delta t}\right) = \boldsymbol{f}(\boldsymbol{U}^{\theta}, t_{\theta}) + (1-2\theta)\mathcal{O}(\Delta t) + \mathcal{O}(\Delta t^2) \tag{B-2.16}$$

Clearly, the method is second-order accurate in time only for the specific choice of $\theta = 1/2$, the truncation error here having a much simpler form than for Eqn. (B-2.8). This would appear to be the more useful of the two schemes, especially for problems with nonlinear mass matrices.

2.2 Step-Doubling Methods

An interesting aspect of "step-doubling" methods is that they do not depend on a particular underlying timestepping scheme (such as in the case of predictor-corrector methods like ABTR) and they do not require larger "stencils" i.e. more saved old solutions, which some predictors require in order to maintain a reasonable level of accuracy.

Step-doubling methods have been used with some success with explicit Runge-Kutta methods (see §16.2 of the Numerical Recipes book [158],

or Gear's book [83]) where they are sometimes called RK45 methods. Step-doubling methods are particularly efficient in this case due to their ability to directly reuse computed information effectively. In this work, we have employed step-doubling primarily in the context of an underlying implicit (first-order) Euler scheme. For implicit methods, step-doubling schemes are undoubtedly more expensive than predictor-corrector schemes, but their adaptive timestep selection is also frequently much more robust. In order to keep the present discussion relatively general, we will not target any particular method, but just assume that our chosen timestepping scheme is locally accurate to order p, that is

$$u(t_{n+1}) - U^{n+1} = C \left. \frac{\partial^p u}{\partial t^p} \right|_{t_\star} \Delta t_{n+1}^p + H.O.T. \tag{B-2.17}$$

where u is the exact solution, U the numerical approximation to it, $C \in \mathbb{R}$ is a constant independent of Δt, Δt_{n+1} is the most recent timestep, and $t_\star \in [t_n, t_{n+1}]$. For example, implicit Euler (a "first"-order method) is *locally* second-order accurate, and thus $p = 2$. We typically think of u and U interchangeably as scalars, continuous functions, or what is the most common case, vectors in $\mathbb{R}^n$.

To define the step selection method, let U_1^{n+1} and U_2^{n+1} be two approximations to the exact solution $u(t_{n+1})$ at time level $n+1$, with U_1^{n+1} being obtained via single timestep of size Δt_{n+1} and U_2^{n+1} obtained via two timesteps of size $\frac{\Delta t_{n+1}}{2}$. Since we assumed the method satisfies the truncation

error relation of Eqn. (B-2.17), we can show that

$$u(t_{n+1}) - U_1^{n+1} = C_1 \left.\frac{\partial^p u}{\partial t^p}\right|_{t_\star} \Delta t_{n+1}^p + H.O.T. \tag{B-2.18}$$

$$u(t_{n+1}) - U_2^{n+1} = C_2 \left.\frac{\partial^p u}{\partial t^p}\right|_{t_\star} \Delta t_{n+1}^p + H.O.T. \tag{B-2.19}$$

where $C_1 \neq C_2$. In general C_1 and C_2 depend on the underlying timestepping scheme or schemes and the number of sub-timesteps used to obtain U_2^{n+1}. Indeed, there is nothing special about taking a half-step, any number of substeps are possible. We note that, because of the smaller stepsizes used to obtain it, solution U_2^{n+1} is in general a more accurate representation of the true solution than U_1^{n+1}.

Our next step is to use Eqns. (B-2.18) and (B-2.19) to obtain a computable estimate of the truncation error $\tau^{n+1} := \|u(t_{n+1}) - U_2^{n+1}\|$. Dropping higher-order terms and subtracting Eqn. (B-2.19) from Eqn. (B-2.18) yields

$$U_2^{n+1} - U_1^{n+1} = (C_1 - C_2) \left.\frac{\partial^p u}{\partial t^p}\right|_{t_\star} \Delta t_{n+1}^p \tag{B-2.20}$$

We may rearrange Eqn. (B-2.20) to read

$$\left.\frac{\partial^p u}{\partial t^p}\right|_{t_\star} \Delta t_{n+1}^p = \frac{U_2^{n+1} - U_1^{n+1}}{C_1 - C_2} \tag{B-2.21}$$

Then, plugging Eqn. (B-2.21) back into Eqn. (B-2.19) we obtain

$$\tau^{n+1} \approx \left|\frac{C_2}{C_1 - C_2}\right| \|U_2^{n+1} - U_1^{n+1}\| \tag{B-2.22}$$

In other words, the two computed approximations provide an estimate to the truncation error itself, up to multiplication by a constant. By specifying a particular method (or methods) for the full-step and substep methods,

it is possible to determine the constant (which does not depend on the solution) explicitly. For example, using a first-order (Euler) scheme with two half-steps leads to the particularly simple result: $\frac{C_2}{C_1 - C_2} = 1$. For a second-order (e.g. trapezoidal) method (whose single-step truncation error constant is $C_1 = -1/12$) and two substeps, we obtain $\frac{C_2}{C_1 - C_2} = 1/3$.

We now turn our attention to using the truncation error estimate to select the next timestep Δt_{n+2}. Clearly, from Eqns. (B-2.18) and (B-2.19), the ratio of successive truncation errors is related by

$$\frac{\tau^{n+2}}{\tau^{n+1}} \approx \frac{\Delta t_{n+2}^p}{\Delta t_{n+1}^p} \tag{B-2.23}$$

We can rearrange Eqn. (B-2.23) to solve for Δt_{n+2} as

$$\Delta t_{n+2} \approx \Delta t_{n+1} \left(\frac{\tau^{n+2}}{\tau^{n+1}} \right)^{1/p} \tag{B-2.24}$$

Now, suppose that the user has specified a tolerance ϵ, such that the truncation error at step m is to satisfy $\tau^m \approx \epsilon \|U^m\|$. Then using this and our previous estimate for τ^{n+1} we have the following simple relation for selecting the next timestep

$$\Delta t_{n+2} = \Delta t_{n+1} \left(\frac{\epsilon \|U_2^{n+1}\|}{\left| \frac{C_2}{C_1 - C_2} \right| \|U_2^{n+1} - U_1^{n+1}\|} \right)^{1/p} \tag{B-2.25}$$

We stress that the development of the method has not been rigorous, but the "proof" of these techniques is in their usefulness in practice. Additional variations such as adding "hysteresis" (i.e. a constant multiple of Δt_{n+2} as defined in Eqn. (B-2.25)) or slightly changing the power $1/p$ depending on

whether we are growing or shrinking the next step have also been used in practical applications, but we found the basic formula of Eqn. (B-2.25) suitable for our purposes.

2.3 The Inexact Newton Method

A standard technique for solving nonlinear finite element problems is the Inexact Newton (IN) Method. It is adaptive in the sense that it provides a systematic way for automatically selecting linear system tolerances which maintain nearly optimal rates of convergence for the nonlinear system iterations while providing a practical implementation for large linear systems. Such techniques have become the de facto standard for solving systems of nonlinear natural convection problems, and are especially important in the solution of steady equations. Here we provide some of the background theory and nomenclature which will be referred to later in the application sections.

The theory was developed beginning with Dembo and Eisenstat in 1982 [67], expanded to globally convergent Newton schemes with backtracking by Eisenstat and Walker in 1994 [76], and again in 1996 [77] where they discuss adaptive selection of the method's forcing term. The concept is very general, being applicable whenever Newton's Method itself can be applied, and it has been shown to be *more* efficient for certain classes of problems which are susceptible to the phenomenon of oversolving [194]. To fix ideas, we will assume as in [77] that we are given $F : \mathbb{R}^n \to \mathbb{R}^n$ which is continuously differentiable near x_* where $F(x_*) = 0$, and where the Jacobian $F'(x_*)$ is non-singular. Then

the standard IN method is: for each Newton step k, given a forcing term η_k, find a Newton step s_k which satisfies

$$||F(x_k) + F'(x_k)s_k|| \leq \eta_k ||F(x_k)|| \tag{B-2.26}$$

then set $x_{k+1} = x_k + s_k$ and continue until $||F(x_k)||$ is less than some pre-selected tolerance. The forcing term η_k can be selected by a variety of methods, but the one recommended by [77] is

$$\eta_k = \begin{cases} 1/2 & , \quad k = 0 \\ \gamma \left(\frac{||F(x_k)||}{||F(x_{k-1})||} \right)^{\alpha} & , \quad k > 0 \end{cases} \tag{B-2.27}$$

where $\gamma = 0.9$ and $\alpha = \frac{1+\sqrt{5}}{2}$ are typically selected as parameters which work well with a broad class of applications.

The condition in Eqn. (B-2.26) is particularly convenient, when we consider that the linear system of equations being solved by the iterative method is $F'(x_k)s_k = -F(x_k)$, and we recognize that the residual of this equation is $r_k := F(x_k) + F'(x_k)s_k$, we see that Eqn. (B-2.26) is essentially a relative residual reduction condition imposed on the iterative solver. In practice, this is the condition actually used by most numerical software libraries (e.g. Petsc [17]) to signal convergence, and so fits easily into most existing codes.

The forcing terms in Eqn. (B-2.27) are chosen with at least two design criteria in mind:

- For the initial ($k = 0$) iterate, we seek only a modest reduction in the relative nonlinear residual with the purpose being to avoid reducing the

residual of the linear system when it is not necessarily a good fit to the nonlinear functional.

- For $k > 1$ when we are approaching the root, η_k is designed to capture the so-called q-quadratic convergence of Newton's Method.

The goal of the IN method is to optimize the work done in the iterative (e.g. GMRES) solver, by focusing the iterations on systems which are closer to the root. In practice, the method typically requires more Newton steps than Newton's method with a constant small iterative tolerance. The trade-off is that fewer linear solver iterations are required at each Newton step, and thus the possibility of making the overall method more efficient and robust arises.

Robustness gains are the greatest for steady-state problems in which the initial iterate is far from the true solution and a steep reduction in the initial relative linear system residual does not result in a comparable reduction in the relative nonlinear residual. For time-dependent problems (with the timestep chosen small enough) the initial guess is typically "good" enough that full benefit is realized from reducing the initial linear system residual as much as possible.

We also note that it is possible to achieve an IN-like algorithm by setting a low relative linear residual tolerance (e.g. 10^{-12}) and then limiting the maximum number of GMRES iterations to a fixed small amount at each nonlinear step. Unfortunately, this technique can also preclude the achievement of

full q-quadratic convergence rates if the maximum-allowed number of iterative solver iterations is set too small.

Another variation of this technique is to grow the number of allowed GMRES (or other iterative solver) iterations with some type of geometric progression, so that most of the iterations are focused near the point where the local linear model is in good agreement with the nonlinear function. We have experimented with growing the maximum number of iterations "quadratically" (i.e. 1,2,4,8,...) as well as "quartically" (1,4,16,64,...), with the usual result being that more nonlinear iterations are performed with fewer overall linear iterations. In some cases, this simple change results in a few percent reduction in the "wall time" required to solve the whole problem. This last point is an important one, since more linear systems will invariably needed to be assembled using the IN method. The total CPU time required for the calculation, not just the total number of linear or nonlinear iterations required, is the final arbiter of the algorithm's efficiency.

2.3.1 Globalization

Whether or not Newton's method converges (regardless of actual convergence rate) is a strong function of the initial guess x_0 which is provided to the system. For time-dependent problems, using an initial guess of the solution at the previous timestep is usually sufficient to provide monotone convergence (i.e. be in the so-called "ball of attraction"). However, for steady state problems, the initial guess is typically quite far (in terms of L_2-norm, say) from the

solution, and in fact even in time-dependent problems with large timesteps, this can be an issue. While it is usually effective to reduce the timestep to guarantee monotone convergence of Newton's method, this introduces unnecessary additional linear system solves in cases where time accuracy is not desired.

Another solution to this problem is known as "globalization" of Newton's method, and typically involves some form of Newton step size reduction designed to achieve monotone reduction in the nonlinear system residual. Eisenstat and Walker [76] introduced the concept of globalization in the context of both exact and inexact Newton methods, and developed a globally-convergent method particularly convenient to use in the framework of IN methods.

The globalized inexact Newton (GIN) method of Eisenstat and Walker can be summarized as a method which satisfies Eqn. (B-2.26), and furthermore, for $t \in (0, 1)$, it holds that

$$||F(x_k + s_k)|| \leq (1 - t(1 - \eta_k)) \, ||F(x_k)|| \tag{B-2.28}$$

The test Eqn. (B-2.28) is designed to ensure that a "sufficient" decrease in $||F||$ is achieved by step s_k. In the limit as $t \to 0$, we obtain the weakest possible condition, i.e.

$$||F(x_k + s_k)|| \leq ||F(x_k)|| \tag{B-2.29}$$

In some situations, it may not be possible to choose an s_k that satisfies Eqn. (B-2.28), even when t is arbitrarily small. In this case, there are several possible solutions: you can continue on with the next step in the hope

that $||F||$ improves, or, with additional knowledge about the problem, design a suitable continuation scheme (i.e. pseudo-timestepping for steady problems, or mollification of PDE coefficients). For time-dependent problems, it is also possible to implement a predictor scheme which improves the initial guess provided to Newton's method within each timestep.

2.3.2 Quotient Convergence Rates

An important concept when dealing with solutions to nonlinear systems of equations using root-finding techniques such as e.g. Newton's method or fixed-point iterations is rate of convergence. The rate of convergence is strongly tied to the efficiency of a given algorithm; it is not enough simply to know the cost of a single iteration, one must also have some idea of roughly how many iterations will be required. The proper framework for discussing such issues is that of quotient convergence rates. A sequence x_k which converges to x^* is said to be "convergent of q-order α" if there exists $M > 0$ such that

$$M_k := \frac{|x_{k+1} - x^*|}{|x_k - x^*|^\alpha} \leq M \tag{B-2.30}$$

for all k sufficiently large. Note that there is no restriction on the size of M in order for the definition to hold, only that the bound be uniform for large k. In fact, we say that the sequence x_k is "superconvergent of q-order α" if in addition to Eqn. (B-2.30) it also holds that

$$\lim_{k \to \infty} M_k = 0 \tag{B-2.31}$$

As a specific example, we note that Newton's method is convergent of q-order 2, and this fact can be used as a sanity check when programming the Jacobian equation for a nonlinear system.

Bibliography

[1] S. Adjerid, I. Babuška, and J. E. Flaherty. A posteriori error estimation for the finite element method-of-lines solution of parabolic systems. *Math. Models and Meths. in Appl. Sci*, 9(2):261–286, 1999.

[2] S. Adjerid, J. E. Flaherty, and Y. J. Wang. A posteriori error estimation with finite element methods of lines for one-dimensional parabolic systems. *Numer. Math.*, 65:1–21, 1993.

[3] A. N. Agarwal and P. M. Pinsky. Stabilized element residual method (SERM): a posteriori error estimation for the advection-diffusion equation. *J. Comput. Appl. Math.*, 74(1–2):3–17, 1996.

[4] M. Ainsworth and A. W. Craig. A posteriori error estimators in the finite element method. *Numeriche Mathematik*, 60:429–463, 1991.

[5] M. Ainsworth and J. T. Oden. A posteriori error estimates for Stokes' and Oseen's equations. *SIAM J. Numer. Anal.*, 34:228–245, 1997.

[6] M. Ainsworth and J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley Interscience Publishers, August 2000.

[7] M. Ainsworth, J. Z. Zhu, A. W. Craig, and O. C. Zienkiewicz. Analysis of the Zienkiewicz-Zhu Error Estimator in the Finite Element Method. *IJNME*, 28:2161–2174, 1989.

[8] S. Albensoeder and H. C. Kuhlmann. Accurate three-dimensional lid-driven cavity flow. *Journal of Computational Physics*, 206:536–558, July 2005.

[9] C. D. Andereck, S. S. Liu, and H. L. Swinney. Flow regimes in a circular Couette system with independently rotating cylinders. *Journal of Fluid Mechanics*, 164:155–183, 1986.

[10] D. K. Anson, T. Mullin, and K. A. Cliffe. A numerical and experimental investigation of a new solution in the Taylor vortex problem. *Journal of Fluid Mechanics*, 207:475–487, 1989.

[11] W. E. Arnoldi. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–25, 1951.

[12] P. Assemat, A. Bergeon, and E. Knobloch. Nonlinear Marangoni convection in circular and elliptical cylinders. *Physics of Fluids*, 19(10), 2007. Article No. 104101(17).

[13] I. Babuska and W. C. Rheinboldt. Computational error estimates and adaptive processes for some nonlinear structural problems. *Computer Methods in Applied Mechanics and Engineering*, 34:895–937, 1982.

[14] I. Babuška, O. C. Zienkiewicz, J. Gago, and E. R. A. Oliviera. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*. Wiley, 1986.

[15] I. Babuška and A. Miller. The Post-Processing Approach in the Finite Element Method. *IJNME*, 20:1085–1129, 2311–2324, 1984.

[16] I. Babuška and D. Yu. Asymptotically exact a-posteriori error estimator for biquadratic elements. *Finite Elements in Analysis and Design*, 3:341–354, 1987.

[17] S. Balay, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. Smith, and H. Zhang. *PETSc Users Manual*. Argonne National Laboratory, Mathematics and Computer Science Division, 2.3.0 edition, April 2005.

[18] S. Balay, V. Eijkhout, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[19] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[20] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Verlag, Basel, Switzerland, 2003.

[21] Wolfgang Bangerth. Using Modern Features of C++ for Adaptive Finite Element Methods: Dimension-Independent Programming in deal.II.

In Michel Deville and Robert Owens, editors, *Proceedings of the 16th IMACS World Congress 2000, Lausanne, Switzerland, 2000*, 2000. Document Sessions/118-1.

[22] Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

[23] R. E. Bank. *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide 8.0, Software, Environments and Tools, Vol. 5.* SIAM, Philadelphia, 1998.

[24] R. E. Bank and B. D. Welfert. A posteriori error estimates for the stokes problem. *SIAM J. Numerical Analysis*, 28:591–623, 1991.

[25] E. Barragy and G. F. Carey. A Partitioning Scheme and Iterative Solution for Sparse Bordered Systems. *CMAME*, 70:321–327, 1988.

[26] E. Barragy and G. F. Carey. Bifurcation Detection Using Lanczos Method and Imbedded Subspaces. *J. of Impact of Computing in Science and Engineering*, 3:76–92, 1991.

[27] W. Barth and G. F. Carey. On a Natural-Convection Benchmark Problem in Non-Newtonian Fluids. *Numerical Heat Transfer Part B: Fundamentals*, 50(3):193–216, 2006.

[28] P. Bastian and et. al. UG – A flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science*, 1:27–40, 1997.

[29] Eric B. Becker, Graham F. Carey, and J. Tinsley Oden. *Finite Elements – An Introduction*, volume 1. Prentice Hall, 1981.

[30] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

[31] H. Bénard. Les tourbillons cellulaires dans une nappe liquide. *Rev. Gén. Sciences Pure Appl.*, 11:1261–1271, 1309–1328, 1900.

[32] H. Bénard. Les tourbillons cellulaires dans une nappe liquide transportant dela chaleur par convection en regime permenant. *Ann. Chem. Phys.*, 23:62–144, 1901.

[33] A. Bergeon, D. Henry, and E. Knobloch. Three-dimensional Marangoni-Bénard flows in square and nearly square containers. *Physics of Fluids*, 13:92–98, January 2001.

[34] T. Blacker and T. Belytchko. Superconvergent patch recovery with equilibrium and conjoint interpolant enhancements. *IJNME*, 37:1517–1536, 1994.

[35] M. J. Block. Surface tension as the cause of Bénard cells and surface deformation in a liquid film. *Nature*, 178:650–651, 1956.

[36] J. Boussinesq. *Théorie Analytique de la Chaleur, Vol 2.* Gauthier-Villars, Paris, 1903.

[37] J. Bragard and M. G. Velarde. Bénard Marangoni convection: planforms and related theoretical predictions. *Journal of Fluid Mechanics*, 368:165–194, August 1998.

[38] P. Brunet, G. Amberg, and P. H. Alfredsson. Control of thermocapillary instabilities far from threshold. *Physics of Fluids*, 17, October 2005. Article No. 104109(12).

[39] D. Bucur and G. Buttazzo. *Variational methods in Shape Optimization Problems.* Progress in Nonlinear Differential Equations and Their Applications, v. 65, Birkhäuser, Basel, 2005.

[40] K. Budge and J. Peery. Experiences Developing ALEGRA: A C++ Coupled Physics Framework, 1996. In M.E Henderson, C. R. Anderson, and S. L. Lyons, editors, Object Oriented Methods for Interoperable Scientific and Engineering Computing.

[41] E. A. Burroughs, L. A. Romero, R. B. Lehoucq, and A. G. Salinger. Linear stability of flow in a differentially heated cavity via large-scale eigenvale calculations. *Int. J. Numer. Meth. Heat and Fluid Flow*, 14(6):803–822, 2004.

[42] G. F. Carey. A Mesh Refinement Scheme for Finite Element Computa-

tions. *Computer Methods in Applied Mechanics and Engineering*, 7:93–105, 1976.

[43] G. F. Carey. An Analysis of Finite Element Equations and Mesh Subdivision. *Computer Methods in Applied Mechanics and Engineering*, 9:165–179, 1976.

[44] G. F. Carey. Derivative calculation from finite element solutions. *CMAME*, 35:1–14, 1982.

[45] G. F. Carey. *Computational Grids: Generation, Adaptation, and Solution Strategies*. Taylor & Francis, 1997.

[46] G. F. Carey, M. Anderson, B. Carnes, and B. Kirk. Some aspects of adaptive grid technology related to boundary and interior layers. *J. Comput. Appl. Math.*, 166(1):55–86, 2004.

[47] G. F. Carey, R. McLay, G. Bicken, and W. Barth. Parallel Finite Element Solution of 3D Raleigh-Bénard-Marangoni Flows. *Int. J. Num. Meth. Fluids*, 31:37–52, 1999.

[48] V. Carey. *A Posteriori Error Estimation for the Finite Element Method via Local Averaging*. Ph.d. thesis, CAM dept., Cornell University, August 2005.

[49] B. R. Carnes and G. F. Carey. Estimating spatial and parameter error in parameterized nonlinear reaction-diffusion equations. *Communications in Numerical Methods in Engineering*, –:–, in press, 2006.

[50] P. Cerisier, C. Pérez-García, C. Jamond, and J. Pantaloni. Wavelength selection in Bénard-Marangoni convection. *Physical Review A*, 35(4):1949–1952, February 1987.

[51] T. F. Chan and Y. Saad. Iterative Methods for Solving Bordered Systems with Applications to Continuation Methods. *SIAM J. Sci. Stat. Comput*, 6(2):438–451, 1985.

[52] S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Oxford: Clarendon Press, 1961.

[53] A. J. Chapman. *Heat Transfer, 4th Edition*. Macmillan Publishing Co. New York, 1984.

[54] K. A. Cliffe. Numerical calculations of two-cell and single-cell Taylor flows. *Journal of Fluid Mechanics*, 135:219–233, 1983.

[55] K. A. Cliffe. Numerical calculations of the primary-flow exchange process in the Taylor problem. *Journal of Fluid Mechanics*, 197:57–79, 1988.

[56] K. A. Cliffe and T. Mullin. A numerical and experimental study of anomalous modes in the Taylor experiment. *Journal of Fluid Mechanics*, 153:243–258, 1986.

[57] K. A. Cliffe and S. J. Tavener. Marangoni-Bénard Convection with a Deformable Free Surface. *J. Computational Physics*, 145:193–227, 1998.

[58] A. Cloot and G. Lebon. A nonlinear stability analysis of the Bénard-Marangoni problem. *Journal of Fluid Mechanics*, 145:447–469, August 1984.

[59] R. G. Cox. The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow. *Journal of Fluid Mechanics*, 168:169–194, 1986.

[60] C. Cuvelier and J. M. Driessen. Thermocapillary free boundaries in crystal growth. *J. Fluid Mech.*, 169:1–26, 1986.

[61] P. C. Dauby and G. Lebon. Bénard-Marangoni Instability in Rigid Rectangular Containers. *J. Fluid Mech.*, 329:25–64, 1996.

[62] P. C. Dauby, G. Lebon, and E. Bouhy. Linear Bénard-Marangoni Instability in Rigid Circular Containers. *Physical Review E*, 56:520–530, 1997.

[63] S. H. Davis. Buoyancy-surface tension instability by the method of energy. *J. Fluid Mech*, 39:347–359, 1969.

[64] S. H. Davis and G. M. Homsy. Energy stability theory for free-surface problems - Buoyancy-thermocapillary layers. *Journal of Fluid Mechanics*, 98:527–553, June 1980.

[65] L. de Luca and C. Meola. Surfactant effects on the dynamics of a thin liquid sheet. *Journal of Fluid Mechanics*, 300:71–85, 1995.

[66] W. R. Dean and P. E. Montagnon. On the steady state motion of inviscid liquid in a corner. *Proc. Cambridge Phil. Soc*, 45:389, 1949.

[67] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton Methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[68] L. Demkowicz, W. Rachowicz, and P. R. B. Devloo. A fully automatic *hp*-adaptivity. *J. Scientific Computing*, 17:127–155, 2002.

[69] M.O. Deville, P. F. Fischer, and E. H. Mund. *Higher-Order Methods for Incompressible Fluid Flow*. Cambridge Univ. Press, 2002.

[70] Karen Devine, Erik Boman, Robert Heaphy, Bruce Hendrickson, and Courtenay Vaughan. Zoltan data management services for parallel dynamic applications. *Computing in Science and Engineering*, 4(2):90–97, 2002.

[71] P. R. B. Devloo and G. C. Longhin. Object oriented design philosophy for scientific computing. *Mathematical Modelling and Numerical Analysis*, 36:793–807, 2002.

[72] H. A. Dijkstra. Structure of cellular patterns in Rayleigh-Bénard-Marangoni flow in two-dimensional containers with rigid side-walls. *J. Fluid Mech.*, 243:73–102, 1992.

[73] H. A. Dijkstra. Surface tension driven cellular patterns in three-dimensional boxes – I: Linear stability. *Microgravity Science and Technology*, 7(4):307–312, 1995.

[74] E. B. Dussan V. The moving contact line: the slip boundary condition. *J. Fluid Mech.*, 77:665–684, 1976.

[75] K. Eckert, M. Bestehorn, and A. Thess. Square cells in surface-tension-driven Bénard convection: experiment and theory. *Journal of Fluid Mechanics*, 356:155–197, February 1998.

[76] S. C. Eisenstat and H. F. Walker. Globally Convergent Inexact Newton Methods. *SIAM Journal on Optimization*, 4(2):393–422, 1994.

[77] S. C. Eisenstat and H. F. Walker. Choosing the Forcing Terms in an Inexact Newton Method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996.

[78] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numerica*, pages 105–158, 1995.

[79] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Adaptive Finite Elements*. Springer-Verlag, Berlin, 1996.

[80] K. Eriksson and C. Johnson. An adaptive finite element method for linear elliptic problems. *Mathematics of Computation*, 50(182), 1988.

[81] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Springer, 2004.

[82] J. E. Flaherty, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis, editors. *Adaptive Methods for Partial Differential Equations*. SIAM, 1989.

[83] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[84] G. C. Georgiou, L. G. Olson, W. W. Schultz, and S. Sagan. A singular finite element for Stokes flow: The stick-slip problem. *Int. J. Num. Meth. Fluids*, 9(11):1353–1367, 1989.

[85] V. Girault and P. A. Raviart. *Finite Element Methods for Navier Stokes Equations*. Springer Series in Computational Mathematics, vol. 5, Springer-Verlag, 1986.

[86] R. Glowinski, H. B. Keller, and L. Reinhart. Continuation-Conjugate Gradient Methods for the Least Squares Solution of Nonlinear Boundary Value Problems. *SIAM J. Sci. Stat. Comput*, 6(4):793–832, 1985.

[87] A. M. Gomilko, V. S. Malyuga, and V. V. Meleshko. On steady Stokes flow in a trihedral rectangular corner. *Journal of Fluid Mechanics*, 476:159–177, February 2003.

[88] T. Grätsch and K.-J. Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & Structures*, 83:235–265, 2005.

[89] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method*. John Wiley and Sons, 1998.

[90] L. Hadji. Nonlinear analysis of the coupling between interface deflection and hexagonal patterns in Rayleigh-Bénard-Marangoni convection. *Physical Review E*, 53:5982–5992, June 1996.

[91] S. C. Hardy. The surface tension for liquid gallium. *J. Crystal Growth*, 71:602–606, 1985.

[92] J. Haslinger and R. A. E. Mäkinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation*. SIAM Advances in Design and Control Series, Philadelphia, 2003.

[93] V. Hernandez, J. E. Roman, and V. Vidal. Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Softw.*, 31(3):351–362, 2005.

[94] Vicente Hernandez, Jose E. Roman, and Vicente Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Transactions on Mathematical Software*, 31(3):351–362, September 2005.

[95] Michael Imamura. Using Doxygen: A quick guide to getting started and using the Doxygen inline documentation system for documenting source code. Technical report, Linux Users Group at Georgia Tech, May 2002.

[96] S. Iqbal and G. F. Carey. Performance analysis of dynamic load balancing algorithms with variable number of processors. *Journal of Parallel and Distributed Computing*, 65(8):934–948, 2005.

[97] H. Jeffreys. The stability of a layer of fluid heated from below. *Phil. Mag.*, 2:833–844, 1926.

[98] H. Jeffreys. Some cases of instability in fluid motion. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 118(779):195–208, March 1928.

[99] V. John. A numerical study of a posteriori error estimators for convection-diffusion equations. *Comput. Methods in Appl. Mech. and Engrg.*, 190(5–7):757–781, 2000.

[100] D. Johnson and R. Narayanan. Experimental observation of dynamic mode switching in interfacial-tension-driven convection near a codimension-two point. *Physical Review E*, 54(4):3102–+, October 1996.

[101] D. Johnson and R. Narayanan. Geometric effects on convective coupling and interfacial structures in bilayer convection. *Physical Review E*, 56:5462–5472, November 1997.

[102] A. Juel, J. M. Burgess, W. D. McCormick, J. B. Swift, and H. L. Swinney. Surface tension-driven convection patterns in two liquid layers. *Physica D Nonlinear Phenomena*, 143:169–186, September 2000.

[103] G. Karypis and V. Kumar. METIS unstructured graph partitioning and sparse matrix order. Technical report, University of Minnesota, Department of Computer Science, August 1995.

[104] G. Karypis and V. Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix reordering. *Parallel and Distributed Computing*, 48:71–95, 1998.

[105] D. Kay and D. Silvester. The reliability of local error estimators for convection-diffusion equations. *IMA J. Numer. Anal.*, 21(1):107–122, 2001.

[106] H. B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In *Applications of Bifurcation Theory, P. H. Rabinowitz (ed.)*, pages 359–389. Academic Press, 1977.

[107] D. W. Kelly, J. P. Gago, O. C. Zienkiewicz, and I. Babuska. A posteriori error analysis and adaptive processes in the finite element method: Part I Error analysis. *Int. J. Num. Meth. Engng.*, 19:1593–1619, 1983.

[108] B. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey. `libMesh`: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3–4):237–254, 2006. http://dx.doi.org/10.1007/s00366-006-0049-3.

[109] H. S. Kohli and G. F. Carey. Shape optimization using adaptive shape refinement. *International Journal for Numerical Methods in Engineering*, 36:2435–2451, July 1993.

[110] J. Koplik and J. R. Banavar. Corner flow in the sliding plate problem. *Physics of Fluids*, 7:3118–3125, December 1995.

[111] E. L. Koschmieder. *Bénard Cells and Taylor Vortices*. Cambridge University Press, New York, NY, 1993.

[112] E. L. Koschmieder and M. I. Biggerstaff. Onset of surface-tension-driven Bénard convection. *J. Fluid Mech.*, 167:49–64, 1986.

[113] E. L. Koschmieder and S. A. Prahl. Surface-tension-driven Bénard convection in small containers. *J. Fluid Mech.*, 215:571–583, 1990.

[114] M. Krishnan, N. Agrawal, M. A. Burns, and V. M. Ugaz. Reactions and Fluidics in Miniaturized Natural Convection Systems. *Analytical Chemistry*, 76(21):6254–6265, November 2004.

[115] M. Křížek, P. Neittaanmäki, and R. Stenberg, editors. *Finite Element Methods: Superconvergence, post-processing, and a posteriori estimates*. Lecture notes in pure and applied mathematics, v. 196, Marcel Dekker Inc., 1998.

[116] K. J. Lee, W. D. McCormick, Q. Ouyang, and H. L. Swinney. Pattern Formation by Interacting Chemical Fronts. *Science*, 261:192–194, July 1993.

[117] K.-J. Lee, W. D. McCormick, J. E. Pearson, and H. L. Swinney. Experimental observation of self-replicating spots in a reaction-diffusion system. *Nature*, 369:215–218, May 1994.

[118] R. B. Lehoucq and A. G. Salinger. Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers. *Int. J. Numer. Meth. Fluids*, 36:309–327, 2001.

[119] A. L. Lin, M. Bertram, K. Martinez, H. L. Swinney, A. Ardelea, and G. F. Carey. Resonant Phase Patterns in a Reaction-Diffusion System. *Physical Review Letters*, 84(18):4240–4243, 2000.

[120] H. Mancini and D. Maza. Bénard-Marangoni thermal oscillators: An experimental study. *Physical Review E*, 55:2757–2768, March 1997.

[121] C. G. M. Marangoni. On the expansion of a drop of liquid floating on the surface of another liquid. *Tipographia dei fratelli Fusi, Pavia*, 1865.

[122] M. T. Matthews and J. M. Hill. Micro/nano sliding plate problem with Navier boundary condition. *Zeitschrift Angewandte Mathematik und Physik*, 57:875–903, September 2006.

[123] P. C. Matthews. Hexagonal patterns in finite domains. *Physica D: Nonlinear Phenomena*, 116:81–94, May 1998.

[124] J. C. Maxwell. On stresses in rarefied gases arising from inequalities in temperature. *Phil. Trans. Roy. Soc. Lond*, 170:231–256, 1879.

[125] D. Maza, B. Echebarria, C. Pérez-García, and H. Mancini. Bénard-Marangoni convection in small aspect ratio containers. *Physica Scripta Volume T*, 67:82–85, 1996.

[126] M. Medale and P. Cerisier. Numerical simulation of Bénard-Marangoni convection in small aspect ratio containers. *Numerical Heat Transfer A*, 42:55–72, 2002.

[127] K. Meerbergen and D. Roose. Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems. *IMA J. of Numerical Analysis*, 16:297–346, 1996.

[128] R. J. Meyers., T. J. Tautges, and P. M. Tuchinsky. The hex-tet hex-dominant meshing algorithm as implemented in CUBIT. In *Proceedings of the 7th International Meshing Roundtable*, pages 151–158, October 1998.

[129] D. H. Michael. The separation of a viscous liquid at a straight edge. *Mathematica*, 5:82–84, 1958.

[130] H. K. Moffatt. Viscous and resistive eddies near a sharp corner. *Journal of Fluid Mechanics*, 18:1–18, 1964.

[131] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Clarendon Press, Oxford, 2001.

[132] B. Mohammadi and O. Pironneau. Applied optimal shape design. *Journal of Computational and Applied Mathematics*, 149(1):193–205, 2002.

[133] B. Mohammadi and O. Pironneau. Shape Optimization in Fluid Mechanics. *Annual Review of Fluid Mechanics*, 36:255–279, 2004.

[134] C. L. M. H. Navier. Mémoire sur les lois du mouvement des fluides. *Mémoires de l'Académie Royale des Sciences de l'Institut de France*, VI:389–440, 1823.

[135] C. Neto, D. R. Evans, E. Bonaccurso, H.-J. Butt, and V. S. J. Craig. Boundary slip in Newtonian liquids: a review of experimental studies. *Reports of Progress in Physics*, 68:2859–2897, December 2005.

[136] D. A. Nield. Surface tension and buoyancy effects in cellular convection. *J. Fluid Mech.*, 19:341–352, 1964.

[137] Nitsche, Joachim A. and Schatz, Alfred H. Interior Estimates for Ritz-Galerkin Methods. *Mathematics of Computation*, 28(128):937–958, oct 1974.

[138] A. Oberbeck. Über die bewegungserscheinungen der atmosphaere. *Sitzungsberichte königl. Preuss. Akad. Wissenschaften*, pages 383–395, 1888.

[139] J. T. Oden, W. Wu, and M. Ainsworth. A posteriori error estimators for the Navier-Stokes problem. *CMAME*, 111:185–202, 1994.

[140] T. Ondarçuhu, J. Millán-Rodríguez, H. L. Mancini, A. Garcimartín, and C. Pérez-García. Bénard-Marangoni convective patterns in small cylindrical layers. *Physical Review E*, 48:1051–1057, August 1993.

[141] T. Ondarçuhu, G. B. Mindlin, H. L. Mancini, and C. Perez Garcia. Dynamical patterns in Bénard-Marangoni convection in a square container. *Physical Review Letters*, 70:3892–3895, June 1993.

[142] Q. Ouyang and H. L. Swinney. Transition from a uniform state to hexagonal and striped Turing patterns. *Nature*, 352:610–612, August 1991.

[143] O. Ozen, E. Theisen, D. T. Johnson, P. C. Dauby, and R. Narayanan. Experiments on the role of gas height in the Rayleigh-Marangoni instability problem. *J. Colloid Interface Sci.*, 289(1):271–275, September 2005.

[144] R. L. Panton. *Incompressible Flow, 2nd ed.* John Wiley and Sons, Inc., Austin, Texas, 1996.

[145] A. Papastavrou and R. Verfürth. A posteriori error estimators for stationary convection-diffusion problems: a computational comparison. *Comput. Methods in Appl. Mech. and Engrg.*, 189(2):449–462, 2000.

[146] R. Pasquetti, P. Cerisier, and C. Le Niliot. Laboratory and numerical investigations on Bénard-Marangoni convection in circular vessels. *Physics of Fluids*, 14:277–288, January 2002.

[147] W. B. Patberg, A. Koers, W. D. E. Steenge, and A. A. H. Drinkenburg. Effectiveness of mass transfer in a packed distillation column in relation

to surface tension gradients. *Chemical Engineering Science*, 38:917–923, 1983.

[148] J. R. A. Pearson. On convection cells induced by surface tension. *J. Fluid Mech.*, 4:489–500, 1958.

[149] J. W. Peterson. A Numerical Investigation of Bénard Convection in Small Aspect Ratio Containers. Master's Report, ASE-EM dept., University of Texas at Austin, May 2003. `http://www.cfdlab.ae.utexas.edu/~peterson/masters.pdf`.

[150] J. W. Peterson, G. F. Carey, D. J. Knezevic, and B. T. Murray. Adaptive finite element methodology for tumor angiogenesis modeling. *Int. J. Numer. Meth. Eng.*, 69(6):1212–1238, 2007. http://dx.doi.org/10.1002/nme.1802.

[151] G. Pfister, H. Schmidt, K. A. Cliffe, and T. Mullin. Bifurcation phenomena in Taylor-Couette flow in a very short annulus. *Journal of Fluid Mechanics*, 191:1–18, 1988.

[152] O. Pironneau. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, 59:117–128, 1973.

[153] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.

[154] O. Pironneau. Optimal shape design for elliptic systems. In *Lecture Notes in Control and Information Sciences: System Modeling and Optimization*, volume 38, pages 42–66. Springer-Berlin, Heidelberg, 1982.

[155] R. Pit, H. Hervet, and L. Léger. Direct Experimental Evidence of Slip in Hexadecane: Solid Interfaces. *Physical Review Letters*, 85:980–983, July 2000.

[156] A. Plaza and G. F. Carey. Local refinement of simplicial grids based on the skeleton. *Applied Numerical Mathematics*, 32:195–218, 2000.

[157] A. Plaza, M. A. Padrón, and G. F. Carey. A 3D refinement/derefinement algorithm for solving evolution problems. *Applied Numerical Mathematics*, 32:401–418, 2000.

[158] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, 1988.

[159] T. Qian, X. P. Wang, and P. Sheng. Generalized Navier boundary condition for the moving contact line. *Communications in Mathematical Sciences*, 1(2):333–341, 2003.

[160] A. Quarteroni and A. Valli. Numerical Approximation of Partial Differential Equations. In *Springer Series in Computational Mathematics*, volume 23. Springer-Verlag, Berlin, 1994.

[161] S. Rahal, P. Cerisier, and H. Azuma. Bénard Marangoni convection in a small circular container: influence of the Biot and Prandtl numbers on pattern dynamics and free surface deformation. *Experiments in Fluids*, 43:547–554, October 2007.

[162] M. L. Ramon, D. Maza, and H. L. Mancini. Patterns in Small Aspect Ratio Bénard-Marangoni Convection. *Physical Review E*, 60:4193–4198, 1999.

[163] M. L. Ramon, D. M. Maza, H. L. Mancini, A. M. Mancho, and H. Herrero. Hexagonal structure in intermediate aspect ratio Bénard-Marangoni Convection. *Int. J. of Bifurcation and Chaos*, 11(11):2779–2787, November 2001.

[164] L. Rayleigh. On convection currents in a horizontal layer of fluid when the higher temperature is on the under side. *Philosophical Magazine*, 32:529–546, 1916.

[165] V. C. Regnier, P. C. Dauby, and G. Lebon. Linear and nonlinear Rayleigh-Bénard-Marangoni instability with surface deformations. *Physics of Fluids*, 12:2787–2799, November 2000.

[166] S. Richardson. A 'stick-slip' problem related to the motion of a free jet at low Reynolds numbers. *Proc. Cambridge Phil. Soc*, 67:477–489, 1970.

[167] S. Richardson. On the no-slip boundary condition. *Journal of Fluid Mechanics*, 59:707–719, 1973.

[168] D. S. Riley and K. H. Winters. Modal exchange mechanisms in Lapwood convection. *Journal of Fluid Mechanics*, 204:325–358, 1989.

[169] D. S. Riley and K. H. Winters. Time-periodic convection in porous media: the evolution of hopf bifurcations with aspect ratio. *J. Fluid Mechanics*, 223:457–474, 1991.

[170] S. Rosenblat, S. H. Davis, and G. M. Homsy. Nonlinear Marangoni convection in bounded layers. I - Circular cylindrical containers. II - Rectangular cylindrical containers. *Journal of Fluid Mechanics*, 120:91–138, July 1982.

[171] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, 1992.

[172] A. G. Salinger, N. M. Bou-Rabee, R. P. Pawlowski, E. D. Wilkes, E. A. Burroughs, R. B. Lehoucq, and L. A. Romero. Sand2002-0396: Loca 1.0 library of continuation algorithms: Theory and implementation manual. Technical report, Sandia National Laboratories, Albuquerque, NM, 2002.

[173] A. G. Salinger, R. B. Lehoucq, R. P. Pawlowski, and J. N. Shadid. Computational Bifurcation and Stability Studies of the 8:1 Thermal Cavity Problem. *Int. J. Numer. Meth. Fluids*, 40:1059–1073, 2002.

[174] J. W. Scanlon and L. A. Segel. Finite amplitude cellular convection induced by surface tension. *Journal of Fluid Mechanics*, 30:149–162, 1967.

[175] M. F. Schatz and K. Howden. Purification of silicone oils for fluid experiments. *Experiments in Fluids*, 19:359–361, September 1995.

[176] M. F. Schatz, S. J. Vanhook, W. D. McCormick, J. B. Swift, and H. L. Swinney. Onset of Surface-Tension-Driven Bénard Convection. *Physical Review Letters*, 75:1938–1941, September 1995.

[177] M. F. Schatz, S. J. VanHook, W. D. McCormick, J. B. Swift, and H. L. Swinney. Time-independent square patterns in surface-tension-driven Bénard convection. *Physics of Fluids*, 11(9):2577–2582, 1999.

[178] Schatz, A. H. and Wahlbin, L. B. Interior Maximum Norm Estimates for Finite Element Methods. *Mathematics of Computation*, 31(138):414–442, apr 1977.

[179] Schatz, A. H. and Wahlbin, L. B. On the Quasi-Optimality in L_∞ of the $\overset{\circ}{H}{}^1$-Projection into Finite Element Spaces. *Mathematics of Computation*, 38(157):1–22, jan 1982.

[180] D. Schwabe. The Bénard-Marangoni Instability in Small Circular Containers Under Microgravity. *Advances In Space Research*, 24(10):1347–1356, 1999.

[181] D. Schwabe. Marangoni instabilities in small circular containers under microgravity. *Experiments in Fluids*, 40:942–950, June 2006.

[182] L. E. Scriven and C. V. Sternling. The Marangoni effects. *Nature*, 187:186–188, 1960.

[183] L. E. Scriven and C. V. Sternling. On cellular convection driven by surface-tension gradients: effects of mean surface tension and surface viscosity. *Journal of Fluid Mechanics*, 19:321–340, 1964.

[184] R. Seydel. *Practical Bifurcation and Stability Analysis from Equilibrium to Chaos*. Springer-Verlag, New York, 1994.

[185] R. Seydel and W. N. Gill. Numerical solution of the Navier-Stokes equations related to a problem of Marangoni Convection. In J. Albrecht and L. Collatz, editors, *International Series of Numerical Mathematics v. 83 (Numerische Behandlung von Eigenwertaufgaben Band 4)*, pages 226–231. Birkhäuser-Verlag, 1987.

[186] J. N. Shadid, A. G. Salinger, R. P. Pawlowski, P. T. Lin, G. L. Hennigan, R. S. Tuminaro, and R. B. Lehoucq. Large-scale stabilized FE Computational Analysis of Nonlinear Steady-state transport/reaction systems. *CMAME*, 195:1846–1871, 2006.

[187] Y. Shen, G. P. Neitzel, D. F. Jankowski, and H. D. Mittelmann. Energy stability of thermocapillary convection in a model of the float-zone crystal-growth process. *J. Fluid Mech.*, 217:639–660, 1990.

[188] Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*,

pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[189] Hang Si. *TetGen - A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*. Weierstrass Institute for Applied Analysis and Stochastics, Berlin, June 2005.

[190] Tomáš Skalický. *LASPack Reference Manual*. Dresden University of Technology, January 1996.

[191] G. D. Smith. *Numerical Solution of Partial Differential Equations, 3^{rd} ed*. Oxford University Press, New York, 1996.

[192] P. A. Thompson and S. M. Troian. A general boundary condition for liquid flow at solid surfaces. *Nature*, 389:360–362, September 1997.

[193] D. C. Tretheway and C. D. Meinhart. Apparent fluid slip at hydrophobic microchannel walls. *Physics of Fluids*, 14:L9–L12, March 2002.

[194] R. S. Tuminaro, H. F. Walker, and J. N. Shadid. On Backtracking Failure in Newton-GMRES Methods with a Demonstration for the Navier-Stokes Equations. *J. Computational Physics*, 180:549–558, 2002.

[195] S. J. Vanhook, M. F. Schatz, W. D. McCormick, J. B. Swift, and H. L. Swinney. Long-wavelength instability in surface-tension-driven Bénard convection. *Physical Review Letters*, 75:4397–4400, December 1995.

[196] S. J. Vanhook, M. F. Schatz, J. B. Swift, W. D. McCormick, and H. L. Swinney. Long-wavelength surface-tension-driven Bénard convection: experiment and theory. *Journal of Fluid Mechanics*, 345:45–78, August 1997.

[197] R. Verfürth. A posteriori error estimators for the stokes equations. *Numerische Mathematik*, 55:309–325, 1989.

[198] R. Verfürth. A posteriori error estimates for nonlinear problems. finite element discretizations of elliptic equations. *Math. Comp.*, 62(206):445–475, 1994.

[199] R. Verfürth. A posteriori error estimates for nonlinear problems. L^r-estimates for finite element discretizations of elliptic equations. *RAIRO Modél. Math. Anal. Numér.*, 32(7):817–842, 1998.

[200] R. Verfürth. A posteriori error estimators for convection-diffusion equations. *Numerische Mathematik*, 80(4):641–663, 1998.

[201] R. Verfürth. Robust *a posteriori* error estimates for stationary convection diffusion equations. *SIAM J. Numer. Anal.*, 43(4):1766–1782, 2005.

[202] X. Wang and G. F. Carey. Finite-element Simulation of a Heated Thin Fluid layer. *Numerical Heat Transfer Part A: Applications*, 45(9):841–867, 2004.

[203] X. Wang and G. F. Carey. On Marangoni effects in a heated thin fluid layer with a monolayer surfactant. Part I: model development and stability analysis. *Int. J. Num. Meth. Fluids*, 48(1):1–16, 2005.

[204] X. Wang and G. F. Carey. On Marangoni effects in a heated thin fluid layer with a monolayer surfactant. Part II: finite element formulation and numerical studies. *Int. J. Num. Meth. Fluids*, 48(1):17–42, 2005.

[205] F. M. White. *Viscous Fluid Flow*. McGraw-Hill, 2nd edition, 1991.

[206] N.-E. Wiberg, F. Abdulwahab, and S. Ziukas. Improved Element Stresses for Node and Element Patches Using Superconvergent Patch Recovery. *CNME*, 11:619–627, 1995.

[207] N.-E. Wiberg and X. D. Li. Superconvergent Patch Recovery of Finite-Element Solution and a posteriori L_2 norm error estimate. *CNME*, 10:313–320, 1994.

[208] K. H. Winters. A bifurcation study of laminar flow in a curved tube of rectangular cross-section. *Journal of Fluid Mechanics*, 180:343–369, 1987.

[209] J. Yang and D. Y. Kwok. Effect of liquid slip in electrokinetic parallel-plate microchannel flow. *J. Colloid and Interface Science*, 260(1):225–233, 2003.

[210] D. Yu. Asymptotically exact a-posteriori error estimator for elements of bi-even degree. *Math. Numer. Sinica*, 13:89–101, 1991.

[211] D. Yu. Asymptotically exact a-posteriori error estimator for elements of bi-odd degree. *Math. Numer. Sinica*, 13:307–314, 1991.

[212] E. Zeidler. *Nonlinear functional analysis and applications.* Springer-Verlag, 1990.

[213] Z. Zhang and J. Zhu. Analysis of the Superconvergent Patch Recovery Technique and A Posteriori Error Estimator in the Finite Element Method (I). *Computer Methods in Applied Mechanics and Engineering*, 123(1–4):173–187, 1995.

[214] Z. Zhang and J. Zhu. Analysis of the Superconvergent Patch Recovery Technique and A Posteriori Error Estimator in the Finite Element Method (II). *Computer Methods in Applied Mechanics and Engineering*, 163(1–4):159–170, 1998.

[215] H. Zhao and H. H. Bau. Microfluidic chaotic stirrer utilizing induced-charge electro-osmosis. *Physical Review E*, 75, 2007. Article No. 066217(8).

[216] Y. Zhu and S. Granick. Rate-Dependent Slip of Newtonian Liquid at Smooth Surfaces. *Physical Review Letters*, 87(9), August 2001. Article No. 096105(4).

[217] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24:337–357, 1987.

[218] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a-posteriori error estimates. part I: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33:1331–1364, 1992.

[219] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a-posteriori error estimates. part II: error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33:1365–1382, 1992.

[220] M. Zlamal. Some superconvergence results in the finite element method. In A. Dold and B. Eckmann, editors, *Mathematical Aspects of Finite Element Methods no. 606, Springer Lecture Notes in Mathematics*, 1975.

[221] M. Zlamal. Superconvergence and reduced integration in the finite element method. *Mathematics of Computation*, 32:663–685, 1978.

[222] F. J. Zuiderweg and A. Harmens. The influence of surface phenomena on the performance of distillation columns. *Chemical Engineering Science*, 9:89–103, 1958.

Vita

During his career as a graduate student, John was a College of Engineering Thrust 2000 fellow as well as a NASA Graduate Student Researcher Program fellow. His research interests beyond Bénard convection include high-speed compressible flows, double-diffusive convection, tumor angiogenesis modeling, and the development of object-oriented finite element software in C++. He is an active developer of the `libMesh`[1] open source finite element library, and an application code built on this library was used to obtain the numerical results in this dissertation. After graduation he will join the Texas Advanced Computing Center staff in Austin as a Research Associate. He also has a long-term interest in a research or teaching position in academia.

Permanent address: 13305 Teton Ridge Cv
Austin, Texas 78727

This dissertation was typeset with LaTeX[†] by the author.

[1] libmesh.sourceforge.net

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.