Appendix A

The continuation technique

A.1 Method description

The continuation method is an important tool in parameter studies as it allows steady state solutions to be traced systematically in parameter space. The method was originally proposed by *Keller* [1977], and since then extensions and numerous applications have been published (for example; Marangoni convection in small aspect-ratio containers by *Dijkstra* [1992], vortex shedding past variously shaped bodies by *Jackson* [1987], porous medium convection by *Riley and Winters* [1989]). General aspects of continuation techniques are described in *Seydel* [1994], the description of the implementation used here stems largely from *Dijkstra* [1995].

First we turn to some notational conventions. We assume that the finite-difference discretisation technique has been applied to the model equations of Chapter 2, using a grid with n points in the horizontal (x) direction and m points in the vertical (z) direction. The solution vector $\vec{u}$ has $N = 4 * nm$ components and can be written as

$$\begin{aligned} \vec{u} &= (\omega_{1,1}, \psi_{1,1}, T_{1,1}, S_{1,1}, \ldots, \omega_{n,m}, \psi_{n,m}, T_{n,m}, S_{n,m})^T \\ &= (u_1, \ldots, u_N)^T. \end{aligned} \tag{A.1}$$

The parameters in our model can be combined into one vector $\vec{p}$:

$$\vec{p} = (Ra_T, R_\rho, Pr, Le, A)^T. \tag{A.2}$$

The solution vector $\vec{u} \in \Re^N$ is to satisfy the following set of N equations that arises after discretisation of the model equations:

$$\Phi_{n,m}(\vec{u}, \vec{p}) = \vec{0}, \tag{A.3}$$

with $\Phi_{n,m} : \Re^N \times \Re^5 \rightarrow \Re^N$. In the sequel we will write Φ instead of $\Phi_{n,m}$ for convenience.

The continuation technique allows us to follow branches of steady state solutions as a function of one of the parameters in $\vec{p}$, from now on referred to as λ. A solution branch must be parameterized in some way, and at this point we introduce the commonly used arclength parameter s. A solution branch is then geometrically represented by the curve

$$\vec{\gamma}(s) = (\vec{u}(s), \lambda(s))^T, \tag{A.4}$$

where s gets its meaning of arclength by the requirement that the tangent to the curve in a point s_0, $\dot{\vec{\gamma}}(s_0)$, has unit length:

$$\dot{\vec{u}}^T(s_0)\dot{\vec{u}}(s_0) + \dot{\lambda}^2(s_0) = 1. \tag{A.5}$$

Equation (A.5) provides the extra constraint needed due to the addition of s. In computations we generally use a simple linearization in s_0 of (A.5) (known as pseudo-arclength parameterization):

$$\dot{\vec{u}}(s_0)\{\vec{u}(s) - \vec{u}(s_0)\} + \dot{\lambda}(s_0)\{\lambda(s) - \lambda(s_0)\} - \Delta S = 0, \tag{A.6}$$

where $\Delta S = s - s_0$ is a steplength along the branch to be chosen by the user.

We now restate the problem as follows; given a solution $\vec{u}_0 = \vec{u}(s_0)$ of (A.3) at a particular parameter value $\lambda_0 = \lambda(s_0)$ and a user-selected steplength ΔS, find a new solution on the branch by solving the following system:

$$\Phi(\vec{u}(s), \lambda(s)) = \vec{0} \tag{A.7a}$$

$$\dot{\vec{u}}_0^T(\vec{u} - \vec{u}_0) + \dot{\lambda}_0(\lambda - \lambda_0) - \Delta S = 0. \tag{A.7b}$$

The Jacobian corresponding to (A.7) reads:

$$J(s) = \begin{bmatrix} \Phi_{\vec{u}} & \Phi_\lambda \\ \dot{\vec{u}}^T & \dot{\lambda} \end{bmatrix}. \tag{A.8}$$

According to the properties of J in a point $(\vec{u}, \lambda)$ we define the following classification:

1. *Regular point*: $\Phi_{\vec{u}}$ is non-singular.
2. *Limit point*: dim(ker($\Phi_{\vec{u}}$)) = 1,
   $\Phi_\lambda \notin$ range($\Phi_{\vec{u}}$).
3. *Simple bifurcation point*: dim(ker($\Phi_{\vec{u}}$)) = 1,
   $\Phi_\lambda \in$ range($\Phi_{\vec{u}}$).

In the first two cases J is non-singular which implies that the branch can be continued in a unique way. The pseudo-arclength parameterization allows continuation through a limit point without problems. In a bifurcation point the solution is not unique; in the next section is described how the secondary branches emanating from this point are reached.
Although the classification is far from complete, the three mentioned cases are the most important ones, and currently these are the only cases we can identify.

A.2 Numerical implementation aspects

Given a solution on the branch, $(\vec{u}_0, \lambda_0)$, a new solution $(\vec{u}_1, \lambda_1)$ is found using a predictor-corrector technique. The predictor can be derived from the tangent to the branch at s_0 which is calculated by taking the total derivative of (A.7a) to s in s_0:

$$\begin{aligned} \frac{\mathbf{D}}{\mathbf{D}s}\Phi(\vec{u}(s_0), \lambda(s_0)) &= [\Phi_{\vec{\gamma}}]\,\dot{\vec{\gamma}}_0 \\ &= [\Phi_{\vec{u}}\ \Phi_\lambda] \begin{bmatrix} \dot{\vec{u}}_0 \\ \dot{\lambda}_0 \end{bmatrix} = 0. \end{aligned} \tag{A.9}$$

In computations we first set $\dot{\lambda}_0 = 1$, solve $\Phi_{\vec{u}}\,\dot{\vec{u}}_0 = -\Phi_\lambda$, and then scale $\dot{\lambda}_0$ and $\dot{\vec{u}}_0$ by application of the normalization condition (A.5). Next, the predictor is defined as $\vec{u}^1 = \vec{u}_0 + \Delta S\dot{\vec{u}}_0, \lambda^1 = \lambda_0 + \Delta S\dot{\lambda}_0$.

For the Newton corrector equations (A.7) are linearized using

$$\vec{u}^{n+1} = \vec{u}^n + \Delta\vec{u}^{n+1}, \lambda^{n+1} = \lambda^n + \Delta\lambda^{n+1}, \tag{A.10}$$

the linearization of (A.7b) reading:

$$\dot{\vec{u}}_0^T(\vec{u}^n - \vec{u}_0) + \dot{\vec{u}}_0^T\,\Delta\vec{u}^{n+1} + (\lambda^n - \lambda_0)\dot{\lambda}_0 + \Delta\lambda^{n+1}\dot{\lambda}_0 - \Delta S = 0. \tag{A.11}$$

As a result the following set of equations is solved for $(\Delta\vec{u}^{n+1}, \Delta\lambda^{n+1})$, starting with the predictor $(\vec{u}^1, \lambda^1)$:

$$\begin{bmatrix} \Phi_{\vec{u}}(\vec{u}^n, \lambda^n) & \Phi_\lambda(\vec{u}^n, \lambda^n) \\ \dot{\vec{u}}_0^T & \dot{\lambda}_0 \end{bmatrix} \begin{bmatrix} \Delta\vec{u}^{n+1} \\ \Delta\lambda^{n+1} \end{bmatrix} = \begin{bmatrix} -\Phi(\vec{u}^n, \lambda^n) \\ r^n \end{bmatrix}, \tag{A.12}$$

where

$$r^n = \Delta S - \dot{\vec{u}}_0^T(\vec{u}^n - \vec{u}_0) - \dot{\lambda}_0(\lambda^n - \lambda_0). \tag{A.13}$$

If we define $B = \Phi_{\vec{u}}(\vec{u}^n, \lambda^n)$ and solve $B\vec{z} = -\Phi(\vec{u}^n, \lambda^n), B\vec{y} = \Phi_\lambda(\vec{u}^n, \lambda^n)$ the solution of (A.12) is given by:

$$\Delta\lambda^{n+1} = \frac{r^n - \dot{\vec{u}}_0^T \vec{z}}{\dot{\lambda}_0 - \dot{\vec{u}}_0^T \vec{y}}, \Delta\vec{u}^{n+1} = \vec{z} - \Delta\lambda^{n+1}\vec{y}. \tag{A.14}$$

If ΔS was chosen not too large, $(\vec{u}^n, \lambda^n)$ converged to $(\vec{u}_1, \lambda_1)$ after repeated application of (A.10) and (A.14). As a convergence criterium we used $\max|\vec{u}^{n+1}(i,j) - \vec{u}^n(i,j)| < \varepsilon = 10^{-6}$ except near limit and bifurcation points where a larger value had to be used (up to $\varepsilon = 10^{-1}$) due to B becoming nearly singular. The intermediate vectors $\vec{y}$ and $\vec{z}$ were calculated by LU-decomposition of B with partial pivoting and Gauss-elimination. In practical calculations with small to moderate gridsize $(< 100 * 100)$ this direct method proved to be more efficient than iterative methods.

During continuation of a branch the occurrence of a limit point or a bifurcation point was monitored by $q(s) = \det(B)$; q changes sign as the singularity is passed. Discrimination between limit points and bifurcation points is simple: if $\frac{\partial\lambda}{\partial s}$ changes sign too a limit point is passed, else a bifurcation point is encountered. The determinant can be calculated directly as the product of the diagonal elements of the LU-decomposition. After a singularity has been detected its position is accurately determined using a secant algorithm to approximate s_1 satisfying $\det(q(s_1)) = 0$. Let s^0 and s^1 the positions on the branch before and after the singularity respectively, then the following algorithm is applied starting with $i = 1$:

$$s^{i+1} = s^i - q(s^i)\frac{s^i - s^{i-1}}{q(s^i) - q(s^{i-1})}. \tag{A.15}$$

Iteration is stopped if $\max|\vec{u}^{n+1}(i,j) - \vec{u}^n(i,j)| < 10^{-2}$ because convergence becomes cumbersome near the singularity.

In case of a bifurcation point a vector $(\vec{u}_{ort}, \lambda_{ort})$ orthogonal to the solution branch is calculated as a first guess to a position on the secondary branch. Let $(\tilde{\vec{u}}, \tilde{\lambda})$ be the point on the branch as the secant iteration has finished, and $\tilde{B}$ and $\tilde{\Phi}_\lambda$ the corresponding matrix and righthandside vector. Then we have to solve the following system:

$$\begin{bmatrix} \tilde{B} & \tilde{\Phi}_\lambda \\ \dot{\vec{u}}_0^T & \dot{\lambda}_0 \end{bmatrix} \begin{bmatrix} \vec{u}_{ort} \\ \lambda_{ort} \end{bmatrix} = \begin{bmatrix} \vec{0} \\ 0 \end{bmatrix}. \tag{A.16}$$

Since $\tilde{B}$ is nearly singular, a close approximation of the eigenvector $\vec{\phi}$ corresponding to the zero eigenvalue can be solved from $\tilde{B}\vec{\phi} = 0$ using inverse iteration (see for example *Golub and Van Loan* [1989]) in a few (2 to 3) steps. A solution to (A.16) can then be expressed in terms of $\vec{\phi}$:

$$\lambda_{ort} = -\frac{\dot{\vec{u}}_0^T \vec{\phi}}{\dot{\lambda}_0 - \dot{\vec{u}}_0^T \tilde{B}^{-1}\tilde{\Phi}_\lambda}, \vec{u}_{ort} = \vec{\phi} - \lambda_{ort}\tilde{B}^{-1}\tilde{\Phi}_\lambda. \tag{A.17}$$

The predictors $\vec{\tilde{u}}^1 = \vec{\tilde{u}} \pm \Delta S \vec{u}_{ort}, \tilde{\lambda}^1 = \tilde{\lambda} \pm \Delta S \lambda_{ort}$ are used to start the continuation of both parts of the secondary branch.

Appendix B

Linear stability

For the determination of the linear stability of a solution of (A.3) a generalized eigenvalue problem has to be solved of the form

$$A\vec{u} = \chi D\vec{u} \tag{B.1}$$

where diagonal matrix D is singular due to Dirichlet boundary conditions.

We only need to calculate those eigenvalues that are close to the imaginary axis since a change in them might change the stability of the solution. To calculate the eigenvalues first a mapping

$$\chi = \frac{\tau - 1}{\tau + 1}, \tau \in \mathcal{C}/\{-1\} \tag{B.2}$$

is applied which maps $\mathcal{C}^-$, the imaginary axis and $\mathcal{C}^+$ onto $|\tau| < 1$, $|\tau| = 1$ and $|\tau| > 1$ respectively. The dominant eigenvalues τ are then calculated from the transformed eigenvalue problem

$$-(A + D)\vec{u} = \tau(A - D)\vec{u}. \tag{B.3}$$

The calculation is started with a number of independent vectors equal to the desired number of eigenvalues. Infinite eigenvectors of (B.3) are first filtered out from the starting vectors by inverse iteration. Next the dominant eigenvalues are calculated using simultaneous iteration.

The solution is linearly stable if all eigenvalues have negative real parts. If at least one of the eigenvalues has a positive real part the solution is unstable. A Hopf bifurcation point is detected if a complex pair of eigenvalues $\sigma = \zeta \pm i\tau$ crosses the imaginary axis. Its position can be accurately determined using the previously described secant algorithm with $q(s) = \zeta(s)$.

Appendix C

A flux law for vertical salt transport

In double-diffusive systems as studied in this thesis, the vertical salinity distibution has a characteristic "step"-structure, i.e. salt is well mixed within the convective layers, but strong salinity gradients exist within the diffusive interfaces. The "step"-structure allows the formulation of a flux law for the vertical salt flux which depends on the overall temperature and salinity differences. Flux laws of this type have been formulated first for a double-diffusive system heated from below, containing only one diffusive interface, but their applicability has been shown for systems of layers as well [*Huppert*, 1971; *Turner*, 1973]. Since the vertical salinity distribution in the laterally heated case is similar to the vertically heated case, we apply the theory in a similar way. We regard a diffusive interface extending infinitely into the horizontal, enclosed by a cool and fresh layer (top) and a warm and salty layer (bottom), where each of the layers has a thickness h (m).

The Sherwood number Sh is a measure of the vertical salt transport through the interface and is defined as follows (dimensional quantities are denoted by a star):

$$Sh = \frac{F_S^*}{\kappa_S(-\delta S^*)/h}, \tag{C.1}$$

where F_S^* (m s^{-1}) is the vertical salt flux, h is the height of a convective layer ($h = O(\eta)$ as shown in this thesis) and δS^* is the jump in salinity across the diffusive interface.

In a steady-state situation, Sh can be formally expressed as [*Turner*, 1973]:

$$Sh = Sh(Ra_{\delta T}, Ra_{\delta S}, Pr, Le), \tag{C.2}$$

where

$$Ra_{\delta T} = \frac{g\alpha(-\delta T^*)h^3}{\nu\kappa_T}. \tag{C.3}$$

$$Ra_{\delta S} = \frac{g\beta(-\delta S^*)h^3}{\nu\kappa_T}. \tag{C.4}$$

Since salt is uniformly distributed in the convective layers, the salt flux through the diffusive interfaces must be independent of the layer thickness h. This condition is satisfied if we combine (C.1) and (C.2) as follows (different combinations are also possible):

$$Sh = c_S\ (Ra_{\delta S})^{1/3}, \tag{C.5}$$

where the coefficient c_S may still depend on $R_\rho = R\delta S/\delta T$ (Pr and Le are kept constant in our simulations).

Eliminating h from (C.5), substituting $\delta S^* = \Delta S\ \delta S$ and applying (C.1) we arrive at the following expression for F_S^*:

$$F_S^* = \kappa_S \Delta S\ c_S\ (\frac{g\beta\Delta S}{\nu\kappa_T})^{1/3}\ (-\delta S)^{4/3}. \tag{C.6}$$

Now F_S^* is made dimensionless using $F_S = F_S^*\ H/\kappa_T\Delta S$, which results in (absorbing the ratio κ_S/κ_T into c_S):

$$F_S = c_S\ (\frac{g\beta\Delta S H^3}{\nu\kappa_T})^{1/3}\ (-\delta S)^{4/3}, \tag{C.7}$$

or, with $Ra_S = g\beta\Delta S H^3/\nu\kappa_T$, and $Ra_\eta = Ra_S R^{-4}$:

$$F_S = c_S\ (Ra_\eta R^4)^{1/3}\ (-\delta S)^{4/3}. \tag{C.8}$$

Expression (C.8) is used in Chapter 6 to investigate the validity of the flux-law assumptions for the presented simulations.

References

Dijkstra, H. A., On the structure of cellular solutions in Rayleigh-Bénard-Marangoni flows in small-aspect-ratio containers, *J. Fluid Mech.*, **243**, 73–102 (1992).

Dijkstra, H. A., An efficient code to compute non-parallel steady flows and their linear stability, *Computers & Fluids*, **24**(4), 415–434 (1995).

Golub, G. H., and Van Loan, C. F., Matrix Computations, The Johns Hopkins University Press (1989)

Huppert, H. E., On the stability of a series of double-diffusive layers, *Deep-Sea Res.*, 18, 1005-1021 (1971).

Jackson, C. P., A finite-element study of the onset of vortex shedding in flow past variously shaped bodies, *J. Fluid Mech.*, **182**, 23–45 (1987).

Keller, H. B., Numerical solutions of bifurcation and nonlinear eigenvalue problems, Applications of Bifurcation Theory, Academic, 359–384 (1977).

Riley, D. S., and Winters, K. H., Modal exchange mechanisms in Lapwood convection, *J. Fluid Mech.*, **204**, 325–358 (1989).

Turner, J. S., *Buoyancy effects in fluids*, Cambridge (1973).

Seydel, R., *Practical Bifurcation and Stability Analysis*, IAM **5**, Springer Verlag (1994).