before merging them together to produce an optimized third image. First, the method is demonstrated on a 2D line from the Gulf of Mexico. Then, the method is applied to a 3D seismic volume from a different part of the Gulf of Mexico.

Chapter 5 discusses another application of improving migration resolution by approximating the least-squares Hessian using non-stationary data matching operations. An approximation to the least-squares Hessian can be calculated by solving a data matching problem between two conventionally migrated images, and the Hessian can be represented by the combination of amplitude and frequency balancing operations. An example is applied to a 2D synthetic Sigsbee data set.

FUTURE WORK

In the future, the work presented in Chapter 5 should be extended to involve real data and 3D examples. It also could benefit by comparing the results of the proposed approach taken in the chapter to other previous approaches presented to approximate the least-squares Hessian (Hu and Schuster, 1998; Dong et al., 2012; Casasanta et al., 2017; Dai and Schuster, 2013; Sacchi et al., 2007; Aoki and Schuster, 2009; Yu et al., 2006; Hu et al., 2001), to see how it compares in different situations.

Another extension of this data matching procedure may be to incorporate the phase of the signal to be matched. Negligible improvements were made when trying to incorporate phase corrections into the high-resolution and legacy data matching problem of Chapter 4, but other data matching problems could benefit from these corrections.

Several applications of data matching were discussed in this thesis. However, many applications remain unaddressed from a data matching standpoint. Problems such as seismic and well-log tying, deconvolution, automatic gain control (AGC), and surface-related multiple elimination (SRME) can also be recast as data matching problems. Looking at these problems in a new light may bring advancements to computational geophysics.

Appendix

CODE

The examples in this thesis were implemented with the Madagascar open-source software environment for reproducible computational experiments (Fomel et al., 2013). The package is available at `http://www.ahay.org/`.

The reproducible document for the results in this thesis, including code, is available at `http://www.sygreer.com/research/honorsThesis`. However, some of the data used in this thesis are proprietary, so those results may not directly be reproducible.

For brevity in this thesis, code is only included for one example of the main frequency balancing algorithm presented in Chapter 3. The code for the rest of the examples in this thesis are available online at the URL above.

Table 1: List of figures in this thesis and the locations of scripts and programs to generate them

Figures	**Directory**	**Listings**
2.1	chapter-locfreq/merge/	1, 2, 3
2.2, 2.3	chapter-background/dmExample/	—
3.1, 3.2, 3.3, 3.4	chapter-merge/apache/	—
3.5, 3.6, 3.7, 3.8	chapter-locfreq/merge/	1, 2, 3
3.9, 3.10	chapter-locfreq/vecta/	—
3.11	chapter-locfreq/convergence/	—
4.1, 4.2, 4.3, 4.4	chapter-merge/apache/	—
4.5, 4.6	chapter-merge/pcable/	—
4.7	chapter-merge/pcable2/	—
5.1, 5.2, 5.3, 5.5, 5.6	chapter-merge/mighes/	—
5.4,	chapter-merge/triop/	—

Listing 1: chapter-locfreq/merge/SConstruct

```
from rsf.proj import *
from radius import radius

# must have 'legacy.rsf' and 'hires.rsf' initial data sets in same directory

# Initial figures
```