

# Applications of the Wavelet Transform to $B$ Mixing Analysis

by

Adam Samuel Cadien

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Science (Hons)

in

The Faculty of Science

(Physics)

The University Of British Columbia

(Vancouver, Canada)

April, 2008

© Adam Samuel Cadien 2008

# Abstract

The neutral  $B$  mesons  $B^0$  and  $B_s^0$  can undergo flavor changing oscillations due to interactions by the weak force. Experiments which measure the frequency of these state transitions produce extremely noisy results that are difficult to analyse. A method for extracting the frequency of  $B$  mesons oscillations using the continuous wavelet transform is developed here.

In this paper the physics of  $B$  meson mixing is related, leading to the derivation of a function describing the expected amount of mixing present in  $B^0$  and  $B_s^0$  meson decays. This result is then used to develop a new method for analysing the underlying frequency of oscillation in  $B$  mixing. An introduction to wavelet theory is provided in addition to details on interpreting daughter wavelet coefficient diagrams. Finally, the effectiveness of the analysis technique produced, referred to as the Template Fitting Method, is investigated through an application to data generated using Monte Carlo methods.

Adam Cadien  
adamcadien@gmail.com

# Table of Contents

<b>Abstract</b>	ii
<b>Table of Contents</b>	iii
<b>List of Figures</b>	v
<b>Acknowledgements</b>	vi
<b>1 Introduction</b>	1
<b>2 <math>B</math>-<math>\bar{B}</math> Oscillations</b>	3
2.1 CP Violation	3
2.1.1 $K$ Mixing	3
2.2 $B$ Mixing	4
2.2.1 Quantum Treatment of the $B$ $\bar{B}$ System	4
2.2.2 The Probabilities of Mixed and Un-mixed Decay	5
2.2.3 Sources of Error in Detection	7
2.3 The CKM Matrix	7
2.3.1 Finding CKM matrix elements $V_{td}$ and $V_{ts}$	8
<b>3 Applications of the Wavelet Transform in Data Analysis</b>	10
3.1 Mother Wavelets	10
3.2 The Continuous Wavelet Transform	10
3.2.1 Wavelet Coefficient Diagrams	11
<b>4 Finding <math>\Delta m</math> with the Continuous Wavelet Transform</b>	15
4.1 Data Generation and Error Simulation	15
4.1.1 Selecting the Mother Wavelet	15
4.2 The Template Fitting Method	18
4.2.1 The Bounding Box	18
4.2.2 Calculating $\chi$ for $\Delta m$ Extraction	19
4.3 Results & Discussion	20
<b>5 Conclusion</b>	25
<b>Bibliography</b>	26

## Appendices

<b>A Programs</b>	27
A.1 Program to Generate $P_{u,m}(t)$	27
A.2 Program that Applies Simulated Detection Errors	29
A.3 Program to Extract $\Delta m$ using the Template Fitting Method	32

# List of Figures

2.1	Lowest Order $B$ Mixing Diagrams . . . . .	5
2.2	The Unitarity Triangle with $\Delta m$ constraints. . . . .	9
3.1	The Morlet and Meyer mother wavelets. . . . .	11
3.2	Applying the Continuous Wavelet Transform. . . . .	13
3.3	Coping with noise in the continuous wavelet transform. . . . .	14
4.1	Simulated $B$ decay data using the function $P_{u,m}(t)$ . . . . .	16
4.2	Simulated $B$ decay data with detection errors. . . . .	16
4.3	Coefficient diagrams using the Morlet and Meyer mother wavelets. . . . .	17
4.4	The decomposing the $P_{u,m}(t)$ coefficient diagrams. . . . .	19
4.5	An example of how to set the bounding box on a coefficient diagram. . . . .	20
4.6	Setting the Bounding box on test data. . . . .	22
4.7	Results: First Iteration of the Template Fitting Method . . . . .	23
4.8	Results: Second Iteration of the Template Fitting Method. . . . .	24

# Acknowledgements

Colin Gay, under whose supervision I have learned what it means to be a researcher. In addition to his support of my undertaking this project, Colin has been an essential source of advice and a guide to all of my work.

Alex Muir for always showing interest in my work, and doing his best to answer my constant barrage of questions regarding everything from the inner workings of  $B$  detectors to the proper methods of statistical analysis.

Cheng-Wei Loh and Bill Mills who proof-read all of the derivations presented in Chapter 2 and who both presented me with great advice on how to write this paper.

Bryce Baril for proof reading the entire paper.

Everyone at UBC's ATLAS TRT lab, who provided me with a friendly and productive working environment.

# Chapter 1

## Introduction

Particle physics appears to characterize our world through symmetries. When evidence for something as elegant as Charge Parity ( $CP$ ) symmetry is found it is typically embraced by the scientific community at large. So in 1964 when Val Fitch and James Cronin discovered proof of  $CP$  violation in the decay of the  $K$  mesons it had profound effects. The effects of  $CP$  violation are described experimentally by approximately 30 quantities. Accurately measuring such values fuels the creation of many particle accelerators, and is an active area of research for thousands of physicists today.

The Cabbibo-Kobayashi-Maskawa (CKM) matrix is one way of representing the amount of  $CP$  violation that exists in quark to quark interactions. The neutral  $B$  meson<sup>1</sup> exhibits  $CP$  violation when it oscillates between its particle-antiparticle states which is used to bound three of the CKM matrix values, the magnitude of each element is proportional to the oscillation frequency of the  $B$  which is determined by the mass difference ( $\Delta m$ ) between the two states. The mass difference between the  $B$  and  $\bar{B}$  is the largest uncertainty currently present in the CKM Matrix.

Specifically the measurement of  $\Delta m_s$  is very difficult to obtain since the  $B_s$  oscillates several times during its lifetime. An experiment involving  $B$  mixing measurements may measure upwards of a half-million events [1] before seeing  $B$  particle/anti-particle transitions. The initial measurement of  $\Delta m_s$  took nearly 20 years to gather sufficient statistics to be able to extract the frequency of oscillation. Precision on the current measurement of  $\Delta m_s = 17.32^{+0.35}_{-0.18} \pm 0.07 ps^{-1}$  [2] needs to be greatly increased in order to verify the unitarity of the CKM matrix. This is the current method used by experimentalists for verifying the Standard Model, which may also lead to the discovery of new particles or interactions.

The extraction of  $\Delta m$  from measurements of the  $B$ 's lifetime is complicated because of its high decay rate and unusually high frequency of oscillation. Upwards of 30% of the measurements taken are due to errors in the detection process or background produced in a collision. The typical methods of frequency extraction, Fourier and Discrete Wavelet analysis, are ineffective at identifying the oscillation frequency to a high degree of certainty. Fourier methods are most effective when transforming a signal which has a large number of periods. The dataset analyzed contains only a few visible periods, meaning that when transformed errors are introduced. Similar problems arise when applying the discrete wavelet transform which also has difficulties working with finite length signals.

The Template Fitting Method is introduced here, a new data analysis technique for extracting the frequency of curves with low statistics and a low signal to noise ratio which is directly applicable to the extraction of  $\Delta m$ . The continuous wavelet transform is used to decompose a signal into its frequency content and a fitting process which uses predefined templates then

---

<sup>1</sup>The symbol  $B$  will be used when results pertaining to both the  $B^0$  and  $B_s^0$  are discussed, the same applies to  $\bar{B}$ . Similarly  $\Delta m$  refers to both  $\Delta m_s$  and  $\Delta m_d$ .

extracts  $\Delta m$ . Using Monte Carlo data it is demonstrated that the Template Fitting Method can extract the mixing frequency to the same degree of accuracy using half of the statistics used in current analysis methods.

This document is split primarily into 3 parts. Chapter 2 provides the background and theory of  $B$  mixing. While in Chapter 3 the process behind and effects of the wavelet transform are discussed. The Template Fitting Method along with results demonstrating its effectiveness are presented in Chapter 4. Finally a conclusion summarizing the information presented is given in Chapter 5.



# Chapter 2

## $B$ - $\bar{B}$ Oscillations

The instability of some neutral mesons due to the weak force causes them to undergo a change of state into their antiparticle. This rare event is the only source of CP violation that has been experimentally observed. The  $B$  meson's rate of oscillation is of interest because it is a measurable quantity of the amount of CP violation that takes place in some quark to quark interactions.

This chapter begins with a background on CP Violation, whose effect on  $K$  mesons is given in section 2.1.1. The  $K$  experiments are used as a basis for providing details of  $B$  mixing. An analysis of the  $B$ - $\bar{B}$  system is provided in section 2.2.1, which leads into the derivation of the probability of mixing presented in section 2.2.2. Details on the errors introduced in  $B$  detection are given in section 2.2.3. Finally how these results are used to describe the likeliness of CP Violation in quark interactions is in section 2.3.

### 2.1 CP Violation

$CP$  is the combined operation of charge and parity conjugation.  $C$  is the charge conjugation operator, which represents the exchange of a particle with its anti-particle.  $P$ , parity conjugation, is a simultaneous reflection about all axes.

Before 1955 it was thought that all particles obey symmetry through parity and charge individually. When Gell-Mann and Pais pointed out that interactions between the  $K^0$  and  $\bar{K}^0$  occur through a weak interaction[3] an investigation into the properties of  $K$  mesons ensued. Shortly after in 1956 it was demonstrated that parity symmetry is violated by the  $K^+$ [4]. So the theory of the time changed so that symmetry should truly be conserved under the combination of charge and parity ( $CP$  symmetry). However even this too was found to be broken in 1964 when  $CP$  violation was found to occur in decaying neutral  $K$  mesons, which due to weak interactions initially discovered by Gell-Mann and Pais, can mix in a unique way [5].

#### 2.1.1 $K$ Mixing

Under  $CP$  symmetry the  $K$  meson has the following normalized  $CP$  eigenstates

$$|K_1\rangle = \frac{1}{\sqrt{2}}(|K^0\rangle - |\bar{K}^0\rangle) \quad \text{and} \quad |K_2\rangle = \frac{1}{\sqrt{2}}(|K^0\rangle + |\bar{K}^0\rangle) \quad (2.1)$$

The  $K_1$ , with a  $CP$  of  $+1$ , decays into two pions while the  $K_2$ , with a  $CP$  of  $-1$ , decays into three pions, which conserves  $CP$  symmetry. Accordingly the  $K_1$  when decaying purely in the two-pion configuration has a much shorter lifetime than the  $K_2$  decaying purely in its long living three-pion configuration. The  $K$  meson doesn't always decay into its  $K_1$  or  $K_2$  states. It has been observed that a small fraction (represented by  $\epsilon$ ) of the long living  $K_L$  decay as  $K_1$  and

conversely some of the short living  $K_S$  decay as  $K_2$ . When the  $K$  decays, it decays not as an eigenstate of the  $CP$  but as a combination of these states:

$$|K_L\rangle = \frac{1}{\sqrt{1+|\epsilon|^2}}(\epsilon|K_1\rangle + |K_2\rangle) \quad \text{and} \quad |K_S\rangle = \frac{1}{\sqrt{1+|\epsilon|^2}}(|K_1\rangle + \epsilon|K_2\rangle) \quad (2.2)$$

Similarly  $CP$  violation is seen in the particle-antiparticle mixing which also occurs<sup>2</sup> in  $B^0$  and  $B_s^0$ .

The lifetimes of the  $K_1$  and  $K_2$  states differ by approximately [6]  $\Delta\tau_{K_{2-1}} = 5.2 \times 10^4 ps$ . Consequently experimentally detecting whether a  $K$  has decayed as a  $K_1$  or  $K_2$  is simplified. A beam of  $K_1$ 's will decay over a few centimeters and  $K_2$ 's over several meters thus making it very easy to detect an event with  $CP$  violation. In contrast the  $B_s$  meson has a mean lifetime of  $\tau_B = 1.5 ps$  and a lifetime difference on the order of  $\Delta\tau_B \propto 1e^{-2} ps$  resulting in a difference of decay distances on the order of  $\mu m$ 's.

## 2.2 $B$ Mixing

### 2.2.1 Quantum Treatment of the $B$ $\bar{B}$ System

Much like the  $K^0$ - $\bar{K}^0$  system, the Hamiltonian of the  $B$ - $\bar{B}$  system under the strong force is simply

$$\mathcal{H}_0 = \begin{pmatrix} m_0 & 0 \\ 0 & m_0 \end{pmatrix}. \quad (2.3)$$

$B$  mesons are made up of a  $b$  quark and  $d$  or  $s$  quark with flavor eigenstates (where  $g = (d, s)$  for the  $B_d^0$  and  $B_s^0$  respectively):

$$|B\rangle = |\bar{b}g\rangle \quad \text{and} \quad |\bar{B}\rangle = |b\bar{g}\rangle \quad (2.4)$$

Now taking into account weak interactions results in a Hamiltonian which is the sum of  $\mathcal{H}_0$  with terms due to an energy shift of the particle-antiparticle system and a set of new states accessible through the coupling of  $|B\rangle$  and  $|\bar{B}\rangle$  (defined by the element  $W_{12}$ ). It has the form<sup>3</sup>:

$$\mathcal{H} = \begin{pmatrix} M & M_{12} \\ M_{12}^* & M \end{pmatrix} - \frac{i}{2} \begin{pmatrix} \Gamma & \Gamma_{12} \\ \Gamma_{12}^* & \Gamma \end{pmatrix} \quad (2.5)$$

where  $M = m_0 + \delta E$  is the mass eigenstate of the strong force summed with the mass shift due to the weak force. The off diagonal elements  $M_{12} = W_{12} + \delta E_{12}$  and  $\Gamma_{12}$  is the mass and decay rate of the coupled  $|B\rangle$  and  $|\bar{B}\rangle$  states. Its these off diagonal elements which explain the energy shift during transition between  $B$  and  $\bar{B}$ . This is shown in figure 2.1 as the exchange in  $W$  bosons as expected from the Standard Model.

The  $B$ 's mass eigenstates are not states with definite flavor. These mixed eigenstates are conventionally named light and heavy in parallel to the  $K$ 's short and long states. The eigenstates are

$$|B_{\mathcal{H},\mathcal{L}}\rangle = p|B\rangle \mp q|\bar{B}\rangle, \quad (2.6)$$

<sup>2</sup>It is expected that the  $D^0$  meson will also exhibit this phenomena in measurements to be taken at the LHC.

<sup>3</sup>A detailed derivation of this Hamiltonian is available in [7].

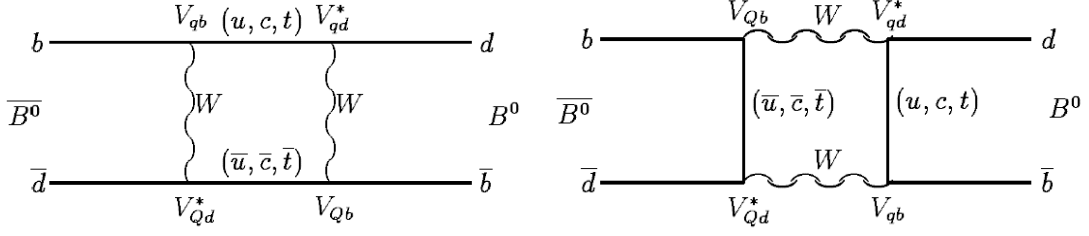


Figure 2.1: The Feynman diagrams describing the coupling of the  $B^0$  and  $\bar{B}^0$  particles. The  $B_s^0$  and  $\bar{B}_s^0$  interact in a similar manner.

with

$$\frac{q}{p} = -\sqrt{\frac{M_{12} - \frac{i}{2}\Gamma_{12}}{M_{12}^* - \frac{i}{2}\Gamma_{12}^*}}, \quad (2.7)$$

and with eigenvalues

$$\lambda_{\mathcal{H}, \mathcal{L}} = M - \frac{i}{2}\Gamma \pm \sqrt{(M_{12} - \frac{i}{2}\Gamma_{12})(M_{12}^* - \frac{i}{2}\Gamma_{12}^*)}. \quad (2.8)$$

Knowing that  $\mathcal{H} = M - \frac{i}{2}\Gamma$  and then diagonalizing  $\mathcal{H}$  reveals

$$\mathcal{H}_{diag} = \begin{pmatrix} \lambda_{\mathcal{H}} & 0 \\ 0 & \lambda_{\mathcal{L}} \end{pmatrix} = M_{diag} - \frac{i}{2}\Gamma_{diag}, \quad (2.9)$$

which means that

$$M_{diag} = \text{Re}(\mathcal{H}_{diag}) \quad \text{and} \quad \Gamma_{diag} = -2\text{Im}(\mathcal{H}_{diag}). \quad (2.10)$$

We can now define

$$\Delta m = \text{Re}(\lambda_{\mathcal{H}}) - \text{Re}(\lambda_{\mathcal{L}}) \quad \text{and} \quad \Delta\Gamma = -2\text{Im}(\lambda_{\mathcal{H}}) + 2\text{Im}(\lambda_{\mathcal{L}}). \quad (2.11)$$

These values are needed to describe the wave function of particles, so that we can see how they evolve with time through Schrodinger's Equation.

### 2.2.2 The Probabilities of Mixed and Un-mixed Decay

We are now prepared to derive the probabilities of a  $B/\bar{B}$  decaying as its anti-particle, referred to as the probability of mixing. The probability of mixed/un-mixed decay is used for data generation purposes, presented in section 4.1, since it matches experimental results for the probability of mixed and un-mixed  $B$  decays.

We begin with the time independent states of  $B$  and  $\bar{B}$ :

$$|B\rangle = \frac{1}{2p}|B_{\mathcal{H}}\rangle + \frac{1}{2p}|B_{\mathcal{L}}\rangle \quad \text{and} \quad |\bar{B}\rangle = -\frac{1}{2q}|B_{\mathcal{H}}\rangle + \frac{1}{2q}|B_{\mathcal{L}}\rangle \quad (2.12)$$

which were found by inverting equation 2.6 and substituting in  $|q|^2 + |p|^2 = 1$ . The time dependence for  $B_{\mathcal{H},\mathcal{L}}$  is then

$$\phi_{\mathcal{H},\mathcal{L}}(t) = e^{-i\lambda_{\mathcal{H},\mathcal{L}}t} = e^{-iMt} e^{\frac{\Gamma}{2}t} e^{\pm \frac{\Delta\Gamma}{4}t} e^{\pm i\frac{\Delta m}{2}t}, \quad (2.13)$$

which is tacked onto the stationary state to obtain the time dependent wave equations

$$\begin{aligned} |B(t)\rangle &= \frac{1}{2}e^{-iMt}e^{\frac{\Gamma}{2}t} \left[ \left( e^{\frac{\Delta\Gamma}{4}t} e^{i\frac{\Delta m}{2}t} + e^{-\frac{\Delta\Gamma}{4}t} e^{-i\frac{\Delta m}{2}t} \right) |B\rangle \right. \\ &\quad \left. + \frac{q}{p} \left( e^{\frac{\Delta\Gamma}{4}t} e^{i\frac{\Delta m}{2}t} - e^{-\frac{\Delta\Gamma}{4}t} e^{-i\frac{\Delta m}{2}t} \right) |\bar{B}\rangle \right] \\ |\bar{B}(t)\rangle &= \frac{1}{2}e^{-iMt}e^{\frac{\Gamma}{2}t} \left[ \frac{p}{q} \left( e^{\frac{\Delta\Gamma}{4}t} e^{i\frac{\Delta m}{2}t} - e^{-\frac{\Delta\Gamma}{4}t} e^{-i\frac{\Delta m}{2}t} \right) |B\rangle \right. \\ &\quad \left. + \left( e^{\frac{\Delta\Gamma}{4}t} e^{i\frac{\Delta m}{2}t} + e^{-\frac{\Delta\Gamma}{4}t} e^{-i\frac{\Delta m}{2}t} \right) |\bar{B}\rangle \right]. \end{aligned} \quad (2.14)$$

The normalization of these states is then

$$\begin{aligned} \eta^2 &= \int_0^\infty \langle B(t)|B(t)\rangle dt = \frac{\Gamma}{2} \left[ \frac{1 + |q/p|^2}{\Gamma^2 - \Delta\Gamma^2/4} + \frac{1 - |q/p|^2}{\Gamma^2 + \Delta m^2} \right] \\ \bar{\eta}^2 &= \int_0^\infty \langle \bar{B}(t)|\bar{B}(t)\rangle dt = \frac{\Gamma}{2} \left| \frac{q}{p} \right|^2 \left[ \frac{1 + |q/p|^2}{\Gamma^2 - \Delta\Gamma^2/4} + \frac{1 - |q/p|^2}{\Gamma^2 + \Delta m^2} \right]. \end{aligned} \quad (2.15)$$

The likeliness of a particle-antiparticle transition is described by

$$\begin{aligned} P_m(B \Leftrightarrow \bar{B})(t) &= \frac{1}{\bar{\eta}^2} |\langle \bar{B}|B(t)\rangle|^2 = \frac{1}{2\bar{\eta}^2} e^{-\Gamma t} (\cosh \frac{\Delta\Gamma t}{2} - \cos \Delta m t) \quad \text{and} \\ P_m(\bar{B} \Leftrightarrow B)(t) &= \frac{1}{\eta^2} |\langle B|\bar{B}(t)\rangle|^2 = \frac{1}{2\eta^2} e^{-\Gamma t} (\cosh \frac{\Delta\Gamma t}{2} - \cos \Delta m t), \end{aligned} \quad (2.16)$$

since the cross-terms cancel out in both cases these are greatly simplified. The corresponding probability that a transition does not occur and the particle is then un-mixed is

$$\begin{aligned} P_u(B \Leftrightarrow B)(t) &= \frac{1}{\eta^2} |\langle B|B(t)\rangle|^2 = \frac{1}{2\eta^2} e^{-\Gamma t} (\cosh \frac{\Delta\Gamma t}{2} + \cos \Delta m t) \quad \text{and} \\ P_u(\bar{B} \Leftrightarrow \bar{B})(t) &= \frac{1}{\bar{\eta}^2} |\langle \bar{B}|\bar{B}(t)\rangle|^2 = \frac{1}{2\bar{\eta}^2} e^{-\Gamma t} (\cosh \frac{\Delta\Gamma t}{2} + \cos \Delta m t). \end{aligned} \quad (2.17)$$

The current measured values[2] of the ratio of decay difference (between the particle-antiparticle states) to mean decay rate is

$$\frac{\Delta\Gamma_{B_s^0}}{\Gamma_{B_s^0}} \approx 0.31 \quad , \quad \frac{\Delta\Gamma_{B^0}}{\Gamma_{B^0}} \approx 0.08 \quad (2.18)$$

and

$$x_s^0 = \frac{\Delta m_{B_s^0}}{\Gamma_{B_s^0}} > 19.9 \quad , \quad x^0 = \frac{\Delta m_{B^0}}{\Gamma_{B^0}} > 0.78. \quad (2.19)$$

It follows that  $\Delta\Gamma \ll \Delta m$  for both  $B$  mesons [8]. Therefore  $\Gamma_{12} \ll M_{12}$  and from equation 2.7  $|q| = |p|$  thus neglecting  $CP$  violation in  $B$  mixing, this turns out to be a good approximation. In this limit  $\Delta\Gamma = 0$ , simplifying the  $\eta$  term such that  $\eta = \bar{\eta}$  and the mixed and un-mixed decay

probabilities become equal for the  $B$  and  $\bar{B}$  and equations 2.16, 2.17 become

$$P_{u,m}(t) = \frac{1}{2}\Gamma e^{-\Gamma t}(1 \pm \cos\Delta m t). \quad (2.20)$$

The lifetime,  $\Gamma$ , of a particular  $B$  or  $\bar{B}$  is typically measured to be on the order of  $\Gamma = 1.5ps$ . The value of interest is however the oscillation frequency,  $\Delta m$ . It is the mass difference between the heavy and light states of the  $B$  which is the frequency of oscillation between the *matter* $\leftrightarrow$ *anti-matter* states of the  $B$  meson. As discussed in section 2.3.1 the value  $\Delta m$  is relevant to determining the CKM elements  $V_{td}$  and  $V_{ts}$ .

### 2.2.3 Sources of Error in Detection

When a collision occurs inside of a detector, given that the proper particles<sup>4</sup> are used and if there is enough energy between them, there is a small possibility that  $B$  or  $\bar{B}$  mesons may be produced.

For every collision that occurs within the detector's range the position and energy of any emitted particles from the collision are tracked. Should a  $B$  or  $\bar{B}$  be produced, reconstruction of the event using emission data produces the presumed points of production and decay of the meson. Similar reconstruction methods are used to find the momentum of the particle and its decay products. Knowing the production and decay points gives the total distance,  $L$ , traveled by the meson before decaying. It is then possible to calculate the lifetime of the particle:  $t = LM_B/p_B$  where  $M_B$  is the mass of the meson.

Faster moving particles produce tracks in the detector which require a position resolution greater than what most detectors are able to provide. This decreases the accuracy in the position measurement of the particle, an effect known as momentum smearing.

Errors are also introduced through false detection of  $B$  mesons. This background error is caused by misidentification of the decay products of the  $B$ . Incorrect tags can occur from improperly detecting the lepton present in the decay resulting in detection of the opposite charge that is actually present. Depending on the method of detection and particle of interest, fake rate is typically responsible for 5% of  $B$  detections.

Determining whether the particle detected is a  $B$  or  $\bar{B}$  is typically done by flavor tagging the  $b$  quark at production and decay points. Of course the  $B$  will be detected using whatever method is designed by the experiment. The SLD group uses the angle between the  $b$  quark and incoming electron to tag the flavor of the quark, with an accuracy of 66%[7]. At BaBar and Belle the combined tagging efficiency has been estimated to be 28% [10], this is the percentage of swapped events used in generating simulation data.

## 2.3 The CKM Matrix

The CKM matrix holds experimentally found values which describe the amplitude of flavour-changing weak interactions. Each element is defined as  $V_{ij}$ , where  $i$  is one of  $u, c$ , or  $t$  and  $j$  is  $d, s$ , or  $b$ , and the value  $|V_{ij}|^2$  is the probability for a  $j$ -quark to transition to a  $i$ -quark. Applying

---

<sup>4</sup>Experimentally  $B$  mesons are most commonly produced from an electron positron collision, although CDF uses a proton anti-proton collision

the unitary constraint to the matrix, meaning

$$V_{CKM}V_{CKM}^\dagger = 1, \quad (2.21)$$

requires that the probabilities of quark transitions be normalizable. This stands to reason since each value squared is a probability, the sum of any row or column should be 1.

There are numerous ways of representing the CKM matrix, most common is Wolfenstein's parameterisation [9] shown here to  $\mathcal{O}(\lambda^3)$ :

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} = \begin{pmatrix} 1 - \frac{1}{2}\lambda^2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \frac{1}{2}\lambda^2 & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} \quad (2.22)$$

where  $\lambda$  is a function of the Cabbibo angle ( $\lambda = \sin(\theta_c)$ ), and  $A, \rho$  and  $\eta$  are real parameters. Specifically of interest here is  $\eta$  which defines the CP violating phase present in the flavor changing transitions. Imposing unitarity of  $V_{CKM}$ , the CKM matrix is constrained by 6 conditions. One of these normalization constraints,

$$V_{ud}V_{ub}^* = -V_{cd}V_{cb}^* - V_{td}V_{tb}^*, \quad (2.23)$$

is directly related to observables found from  $B$  meson decay. Each of the 6 equations arising from 2.21 are of the form of equation 2.23 and each makes a *Unitarity Triangle*. The magnitude of each term from equation 2.23 is the length of one side of a *Unitarity Triangle* which exists in the complex plane.

The area of each *Unitarity Triangle* is the same. Existing in the complex plane means that only terms with a complex phase contribute to the area of a triangle. Thus only the elements  $V_{td}$  and  $V_{ub}$ , which have a complex phase due to CP Violation, contribute to the area.

The *Unitarity Triangle's* area is a measure of the magnitude of CP Violation which occurs in a quark-quark transition. It also allows for a visual representation of the constraints placed on the CKM matrix elements from experimental results. The experimental constraints on the triangle (described by equation 2.23) are shown in figure 2.2. The uncertainty of the measured value of  $\Delta m_s$  and  $\Delta m_d$  directly correlate to errors in the lengths and angles of the triangle. By decreasing the uncertainty of an experimental result it is possible to put tighter constraints on the *Unitarity Triangle* which may show that the CKM matrix does not actually meet the unitary condition (equation 2.21, in this way the standard model is verified. Should the CKM elements not meet the unitary condition this would violate the current structure of the standard model.

### 2.3.1 Finding CKM matrix elements $V_{td}$ and $V_{ts}$

The  $B$  oscillation frequency has been shown to be the mass difference between the  $B$  and its anti-particle  $\bar{B}$ .

Omitting lengthy calculations beyond the scope of this paper the end result is

$$\Delta m_s = \mathcal{K}_s m_{B_s} |V_{ts}^* V_{tb}|^2, \quad (2.24)$$

$$\Delta m_d = \mathcal{K}_d m_{B_d} |V_{td}^* V_{tb}|^2, \quad (2.25)$$

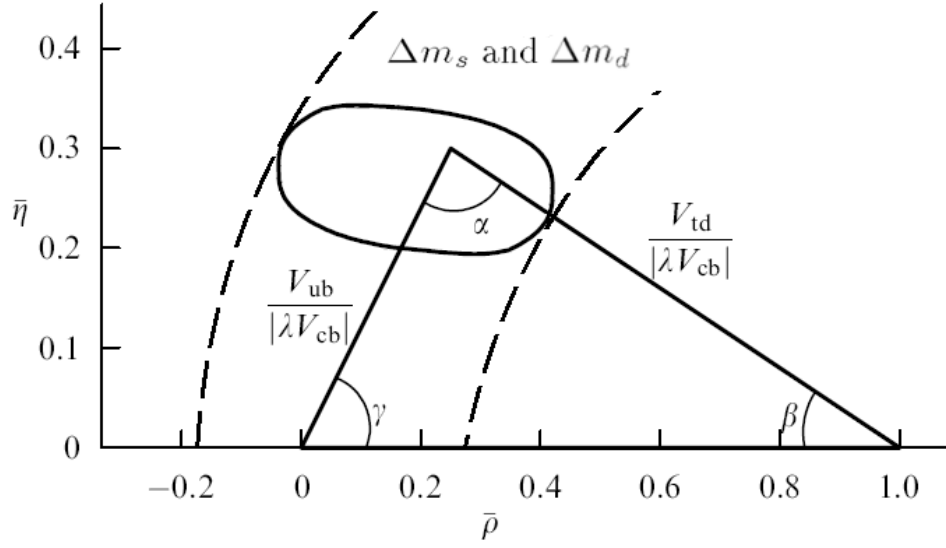


Figure 2.2: The *Unitarity Triangle* plotted in the complex plane. This is one of the two *Unitarity Triangle* whose sides all have lengths that are of the same order and are sensible to represent graphically. The dashed line represents the bounds placed on the error of the current placement of the vertex with angle  $\alpha$ . The experimentally measured values of  $\Delta m_s$  and  $\Delta m_d$  are responsible for setting the bounds on the CKM matrix elements  $V_{td}$  and  $V_{ts}$ .

where the values  $m_{B_s}$  and  $m_{B_d}$  are the mass of the  $B_s$  and  $B_d$  mesons respectively. The values  $\mathcal{K}_s$  and  $\mathcal{K}_d$  are functions of the bag parameter and the  $B$  decay constant [7], theoretically derived values which will be treated as constants here. Additionally the ratio of  $|V_{ts}/V_{td}|^2$  can be related when approximating the diagonal element  $|V_{tb}| \approx 1$ . So from equations 2.24 and 2.25

$$\frac{\Delta m_s}{\Delta m_d} = \frac{\mathcal{K}_s m_{B_s}}{\mathcal{K}_d m_{B_d}} \left| \frac{V_{ts}}{V_{td}} \right|^2. \quad (2.26)$$

This is the relationship between the  $B$  mixing oscillation frequency  $\Delta m$  and the CKM matrix elements  $V_{td}$ .

## Chapter 3

# Applications of the Wavelet Transform in Data Analysis

Although wavelet theory has been a topic of study since the 1930's, their application to data analysis began only 20 years ago. When Grossman and Morlet proposed the wavelet transform [11] it was used as an improvement to windowed Fourier analysis in searching for oil using sonar methods. Because it provides a sturdy platform to carry out frequency analysis of previously unusable datasets (due to a high noise ratio, discontinuities, or low statistics) the wavelet transform has become a popular technique for data analysis.

### 3.1 Mother Wavelets

The wavelet transform involves the use of a master wavelet function, referred to as the mother wavelet, and some function or dataset of interest. Mother wavelets must exist in the Hilbert Space for square integrable functions, that is

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty. \quad (3.1)$$

They must also be differentiable, this ensures the wavelet will not have any discontinuities when evaluated at any scale or position  $(a, b, t)$ . Any function that satisfies these constraints can be used for the continuous wavelet transform. Several wavelet functions arise that satisfy these constraints, the mother wavelets shown in figure 3.1 are the primary candidates for use in the data analysis section and are discussed more there.

### 3.2 The Continuous Wavelet Transform

The only constraint on a the function to be transformed  $f(x)$  is that it must occupy the same functional space as the wavelet  $\psi$ . The wavelet transform of  $f(x)$  is defined as

$$W_\psi = \langle \psi, f(x) \rangle, \quad (3.2)$$

where the inner product is

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x)^* g(x) dx. \quad (3.3)$$

The transformed function  $W_\psi$  is called a daughter wavelet. The daughter wavelet is meaningless however, without a specified scale and position of the mother wavelet. These scale and translation parameters are labeled  $a$  and  $b$  respectively, and the daughter wavelet is a function of these



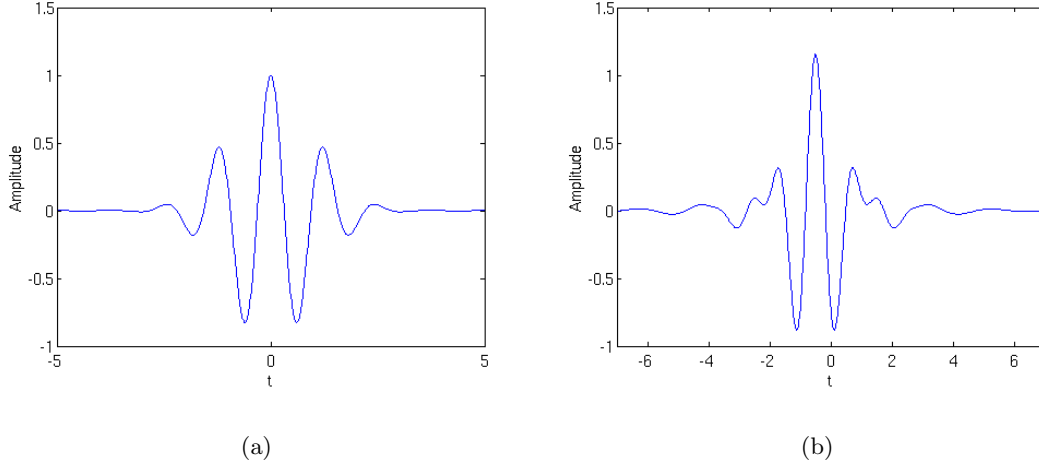


Figure 3.1: (a): The Morlet mother wavelet. (b): The Meyer mother wavelet, the accompanying scaling function is not shown.

$W_\psi(a, b)$ . A wavelet coefficient  $C_W(a, b)$  is then defined

$$C_W(a, b) = \int_{-\infty}^{\infty} W_\psi(a, b) dx. \quad (3.4)$$

The continuous wavelet transform results in a set of wavelet coefficients corresponding to each daughter wavelet within the bounds  $x_0 \leq b \leq x_\ell$  and  $0 \leq a \leq h_\ell$ . The wavelet coefficient is a measure of how well the mother wavelet fits the function at a certain scale and position.

### 3.2.1 Wavelet Coefficient Diagrams

Evaluating a 1-D function using the wavelet transform results in a 2-D description of the function now as a function of scale and position. Larger scale values correspond to a wider analysing function, that is  $\psi_a(x) = \psi(x/a)$ , and lower frequencies are detected. These dilated wavelets have no dependence on the high frequency content of the function being analysed. The opposite is true for small scale values, any low frequency content is lost when the transform is taken. So the daughter wavelet's coefficient is maximized when it's central frequency is near some frequency content of the dataset and the wavelet is in phase with these oscillations. These are the primary considerations taken into account when evaluating the 2 dimensional wavelet coefficient diagram.

An example of finding the wavelet coefficients for a continuous wavelet transform is given in figure 3.2. Shown in figure 3.2(a), are the initial wavelet (dashed) with its parameters with  $[a, b] = [1, 0]$  and the sinusoid (solid). The wavelet is then shifted through all of the combinations of translation and scale finding the coefficient  $C_W(a, b)$ . In figure 3.2(b) the mother wavelet has parameters  $[a, b] = [5, 0]$ . This appears to be the central frequency of the sinusoid, which then becomes clear in figure 3.2(c) when the wavelet is drawn with parameters  $[a, b] = [5, \pi/2]$ . When the wavelet is transitioned to be in phase with the function and matches its frequency, the largest values of  $C_W(a, b)$  are found. With the wavelet completely out of phase, as in figure

3.2(d) where  $[a, b] = [1, \pi]$ , the smallest coefficients are produced. As the scale of the wavelet is increased, coefficients will never be as greater than when it was perfectly fit at  $a = 5$ . The wavelet coefficient diagram is then shown in figure 3.2(f), which is made by plotting all of the coefficients as a function of their scale and position. The light areas represent large coefficients, black areas are smaller values of the daughter wavelet coefficient and they represent a poor fit. Peaks exist when the wavelet has a scale near  $a = 5$  and when it is in phase with the sinusoid.

The coefficient diagram alone makes it possible to interpret much about a dataset. Periodic content which may have been hidden in noise or background becomes apparent when frequency matching occurs. This property is especially useful when there are multiple periodic functions in a single dataset, as shown in figure 3.3. Clearly the continuous wavelet transform is still applicable for datasets with low statistics and high noise content.

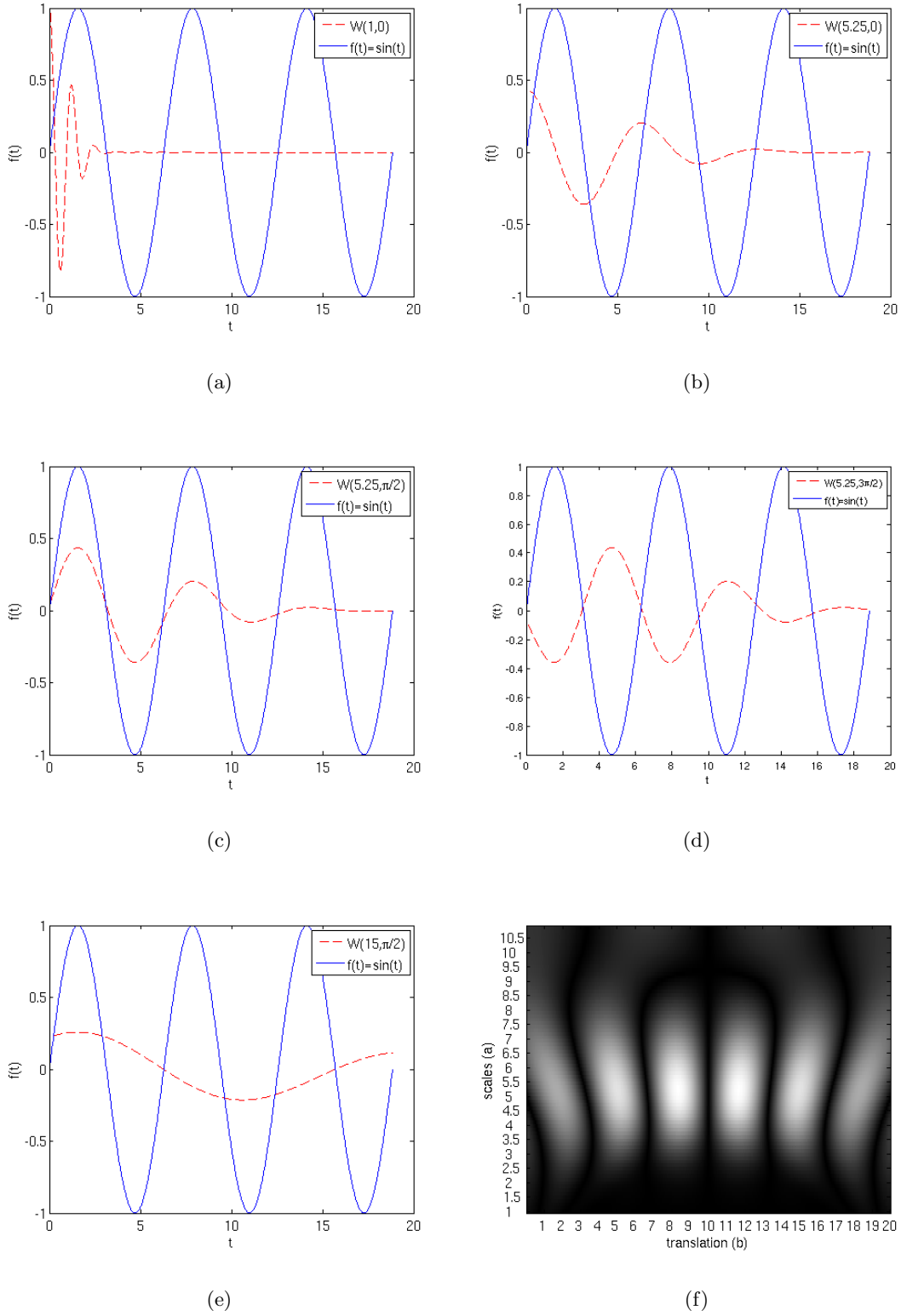


Figure 3.2: (a)-(e) The mother (dashed line) and a sinusoid function (solid line). (f) The continuous wavelet transform coefficient diagram. Light areas represent large coefficients, black areas are small coefficients  $C_W(a, b)$ .

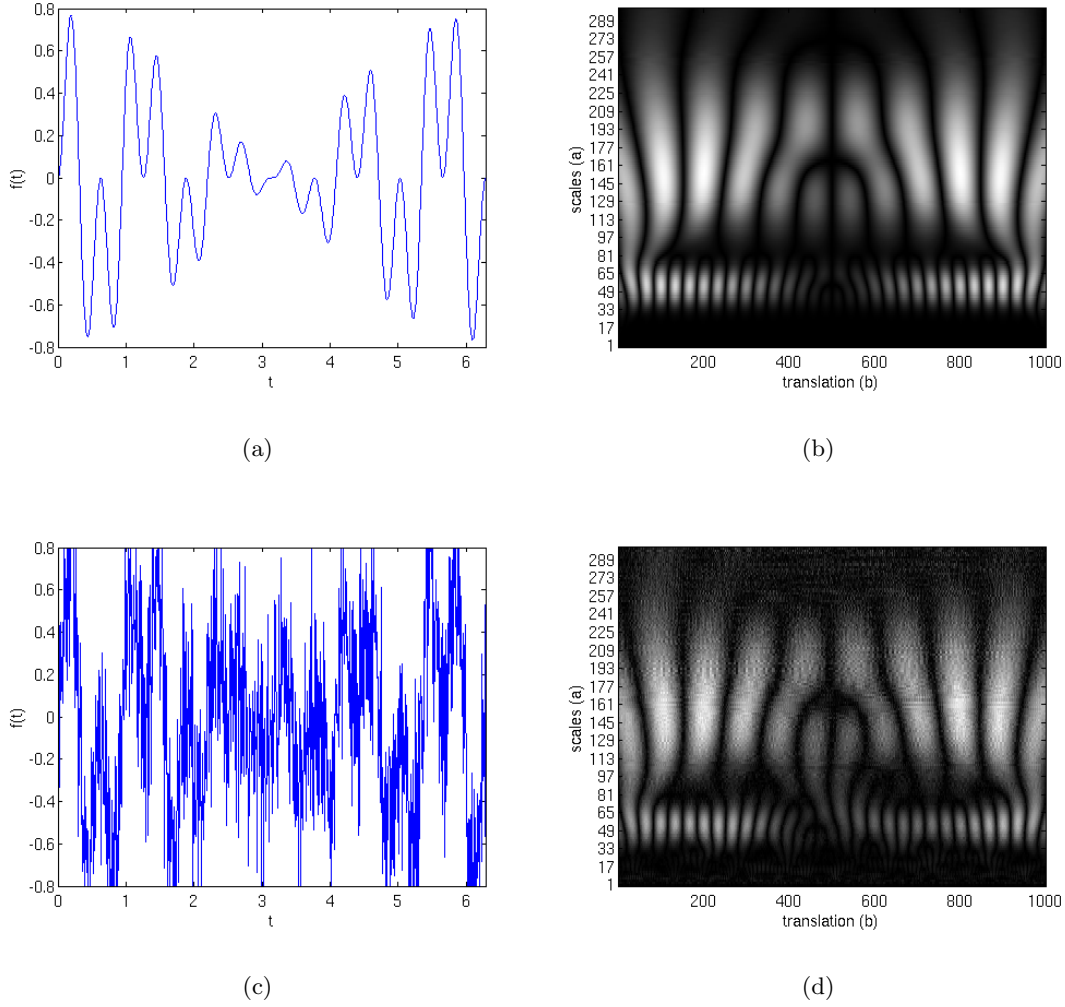


Figure 3.3: (a) The function  $f(t) = \cos(t/2)\sin(5t)\sin(10t)$ . (b) The continuous wavelet transform of  $f(t)$ , each component of the function is visible in the diagram. The low frequency cosine term is present in the broad scale, its inflection point is visible at  $t = \pi$ . The higher frequency term  $\sin(5t)$  exists over  $90 < a < 240$ , the slope inwards due to interactions with the cosine term. The highest frequency term  $\sin(10t)$  exists in the small peaks at  $25 < a < 65$ . (c) The function  $f(t) = \cos(t/2)\sin(5t)\sin(10t)$  with each point perturbed by a random Gaussian number with width  $\sigma = 0.1$ . (d) The coefficient diagram of the data from (c). Although not as smooth as diagram (b) all frequency content in the transform of the noisy function is still clearly visible. The function  $f(t)$  could easily be extracted from this.

## Chapter 4

# Finding $\Delta m$ with the Continuous Wavelet Transform

The data analysis method for extracting the oscillation frequencies of mixed and un-mixed  $B$  mesons is described in this chapter. A brief overview of data generation methods is given first, followed by a discussion on how a mother wavelet was selected in section 4.1.1. In section 4.2 the data analysis method developed using the wavelet coefficient diagrams is presented <sup>5</sup>.

### 4.1 Data Generation and Error Simulation

Experimental datasets for  $B$  meson oscillations are small and too noisy to work with while developing a frequency extraction method. However an extremely accurate model for the probability of decay as a mixed or un-mixed  $B$ , equation 2.20 presents the opportunity for an alternative method for algorithm development. For a given frequency  $\Delta m$  a large master set of data is generated using Monte Carlo methods and the master function  $P_{u,m}(t)$ , shown in figure 4.1. This is also performed by the Matlab script given in appendix sections A.1.

To ensure that an algorithm developed on test data will work equally as well on experimental data it is necessary to mimic the errors involved in detection. Background detection errors are introduced by sampling multiple curves which dilute the dataset with false  $B$  detections. The largest source of errors, the swapping of  $B$  and  $\bar{B}$  events, is achieved in the test data by simply switching a number of mixed points for un-mixed. The finite resolution of a particular detector is mimicked by disturbing every sample point by a Gaussian random number centered at the measured time of decay. Additionally, to account for the smearing affect caused by the wide range of time dilation experienced by the particles, the width of the Gaussian used to shift each sample is increased as a function of the decay time. An implementation of error simulation and injection is provided in the appendix section A.2. The data presented in figure 4.1 convoluted with the simulation errors described above is shown in figure 4.2.

#### 4.1.1 Selecting the Mother Wavelet

Choosing which mother wavelet is appropriate for an analysis is dependent upon the dataset to be analyzed and the type of analysis to be performed. A wavelet which is able to extract the underlying function in a dataset is the primary goal in choosing the mother wavelet. For the purpose of frequency extraction from a dataset with a large background ratio, selecting a wavelet which closely matches the underlying function will increase the likeliness of a good decomposition. A mother wavelet which is close in shape to the underlying curve will highlight its broad structure

---

<sup>5</sup>The code required for obtaining these results is given in the Appendices.

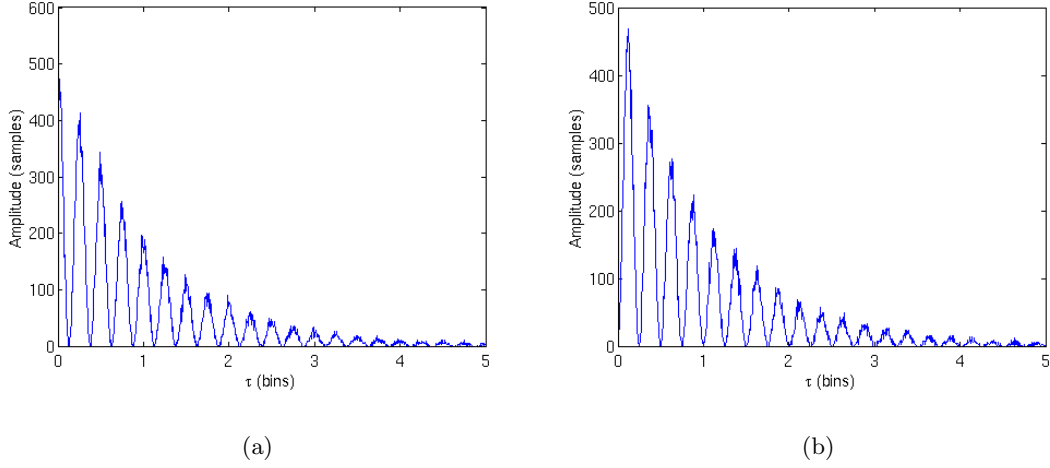


Figure 4.1: (a) Histogram of the simulated probability of decay for un-mixed  $B$  mesons,  $P_u(t)$ . (b) The corresponding histogram of the probability of decay as a mixed particle  $P_m(t)$ . Both plots contain 50,000 points divided into 1,000 evenly spaced bins. These samples were generated using Monte Carlo methods and have no simulated errors injected.

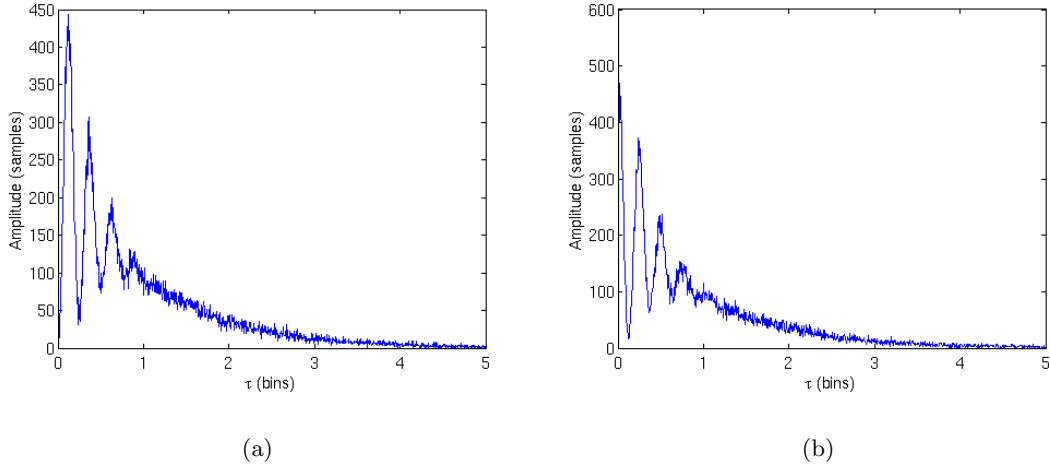


Figure 4.2: (a) The  $P_u(t)$  data with detection errors. (b) The corresponding  $P_m(t)$  curve with the similar errors. Both datasets contain 30% swapped events, which is introduced by switching  $P_m(t)$  samples with  $P_u(t)$  samples. Additionally 5% of the points are replaced by samples from randomly generated functions, this simulates background errors. Finally all of the points are shifted by a gaussian random number with a varying width  $\sigma = 0.1t$ , this is a function of the time of their decay and mimics the effect of smearing described in section 2.2.3. Both of these plots were made using the program in appendix section A.2.

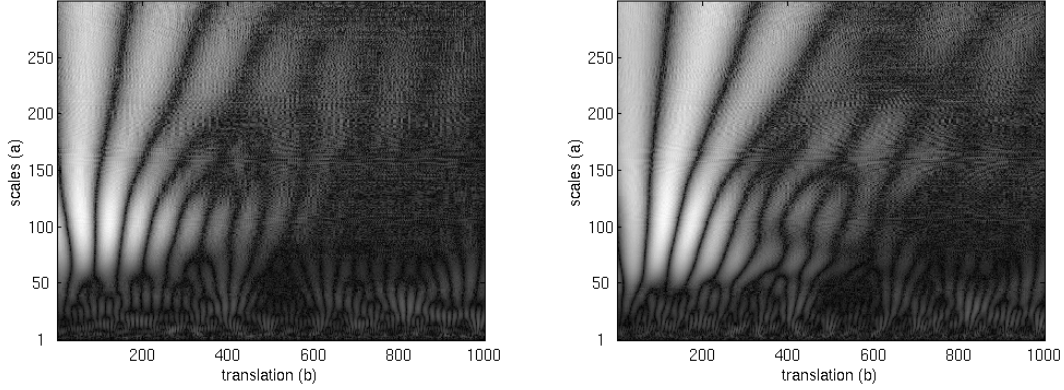


Figure 4.3: (a) Coefficient diagram made using the Morlet wavelet. (b) Coefficient diagram made using the Meyer wavelet. Both are transforms of the noise injected  $P_m t$  data given in figure 4.2(b).

resulting in clear sections of periodic behaviour in the wavelet coefficient diagram. The oscillations present in the coefficient diagram are caused by the mother wavelet coming in and out of phase with the function being transformed.

There are some subtle qualities of mother wavelets which may not become clear for a certain dataset until an inspection of the coefficient diagram is done. In figure 4.3 the wavelet transform for the sample dataset from figure 4.2, using both the Morlet and Meyer mother wavelets, is given.

The Morlet wavelet proves to be effective in providing a stable series of oscillations in the frequency regime of interest. The Morlet wavelet is defined by the function

$$\psi(x) = Ce^{-x^2/2}\cos(5x) \quad (4.1)$$

where  $C$  is the normalization constant. This clearly bears a resemblance to the mixed and un-mixed decay probability (function 2.20) as shown in figure 3.1(a).

A second mother wavelet candidate, the Meyer wavelet, is ineffective at high swapping rates, see figure 4.3(b). When more than  $\sim 30\%$  of the points become swapped the mixing amplitude is not visible to the eye. For the transform to be effective with such a large amount of background, a wavelet which very closely matches the function to be extracted is needed. Additional properties such as existing for several periods at a significant amplitude also makes the Morlet wavelet more sensitive local periodic behaviour.

## 4.2 The Template Fitting Method

To begin with, a template is defined here as the array of wavelet coefficients for a particular daughter wavelet. The quality of interest in these templates is the frequency of the master curve used in data generation. The extraction of  $\Delta m$  is performed following the procedure referred to as the Template Fitting Method:

1. Take the Continuous Wavelet Transform of the dataset to be analysed.
2. Define a bounding box which isolates the frequency peaks of the  $\Delta m$  oscillations.
3. Generate a set of templates over a desired frequency range.
4. Calculate the  $\chi$  value for each template against the experimental data template, within the bounding box.
5. Iterate over steps 3 and 4 reducing the step size of the frequency and the bounds as necessary, concentrating on the frequencies near the minimum  $\chi^2$  value.
6. The oscillation frequency is the frequency of the template with the minimum  $\chi$  value.

The underlying concept is that signals with like frequencies will result in similar peak distributions in the same scale region of the coefficient diagrams. We can generate datasets which very closely mimic the experimentally sampled one, this is taken advantage of by generating templates at varying frequencies. By taking a point by point difference between the wavelet coefficients inside of a predetermined bounding box, exactly how well the generated data matches the experimental data is found. The template with the closest frequency to the experimental data should then have the smallest  $\chi$  value.

Data generation for the templates can take a significant amount of time<sup>6</sup>, sometimes up to an hour for each template. When a high degree of accuracy is desired it has been shown that performing Template Fitting Method in a recursive manner reduces the number of templates needed. With each iteration the bounds for the frequencies of the next iteration should be set to be  $\sim \pm 10\%$  of the current frequency with the minimum  $\chi$  value. As the minimum and maximum frequency approach a common value, the stepsize  $\Delta f_{tmp}$  of the template frequency is reduced to provide a sufficient number of templates to compare with, typically on the order of 10 templates per iteration. All results shown in this paper were generated using the Matlab scripts given in the appendix. The program in appendix section A.3 provides an example implementation of the Template Fitting Method.

### 4.2.1 The Bounding Box

The most crucial portion of the Template Fitting Method is setting the bounding box on the experimental data properly. The goal is to completely isolate any part of the template which corresponds to the oscillation frequency. This is the only parameter that the fit should be dependent on. Setting the bounds too large will result in the fit being dependent on random noise or a divergent exponential, thus skewing the measured frequency. Bounding boxes which are too small or off-center may not provide enough data for an accurate fit.

---

<sup>6</sup>This is dependent on sample size, area of the bounding box and speed of the computer used.



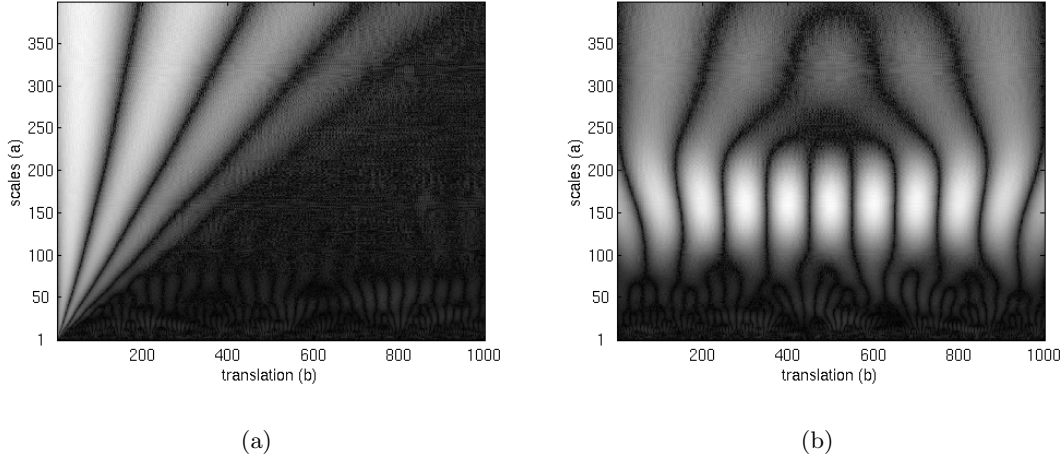


Figure 4.4: (a) The function  $P_{u,m}(t)$  is decomposed into an exponential term in addition to (b) a sinusoidal term. The peaks in the coefficient diagram which are purely dependent on the  $\cos(\Delta mt)$  term are what need to be isolated by setting the bounding box.

The template created by the transform of  $P_{u,m}(t)$  has 3 components. High frequency noise can be described at smaller scales. At larger scales the exponential portion of  $P_{u,m}(t)$  which diverges as the scale increases (see figure 4.4(a)) can be characterized. Between these is the sinusoid that is  $\Delta m$  dependent (see figure 4.4(b)). The bounding box should encompass the peaks in this third frequency range. Since the data contains a minimum of %25 swapped samples only the first few peaks will clearly be visible in the coefficient diagram. Although it is possible to successfully extract the frequency using a fit on just the first peak, it will be much more sensitive to any bias in detection at short lifetimes,  $\tau$ . Figure 4.5 shows an appropriate bounding box which contains the minimum number of periods, 3, at the scale with the largest coefficient values.

#### 4.2.2 Calculating $\chi$ for $\Delta m$ Extraction

By truth matching the templates with the experimental data template a measure of the goodness of the fit is calculated. In this case  $\chi$  is used

$$\chi_{tmp} = \sqrt{\sum_{a=1}^{a_n} \sum_{b=0}^{b_n} \frac{(W_{a,b,tmp} - X_{a,b})^2}{\sigma_b^2}}, \quad (4.2)$$

where  $tmp$  specifies the template,  $a$  and  $b$  are the wavelet scale and position coefficients with  $a_n$  and  $b_n$  being their largest values.  $X$  is the wavelet coefficient of the experimental data and  $W$  is the coefficient for the template. The variance  $\sigma_b^2$  is constant over all scale values and is set at  $\sigma_b = \sqrt{n_b}$ , where  $n_b$  is the number of samples that decayed at the time corresponding to the center of the translated wavelet (b).

Because the experimental data doesn't always have the same number of samples as generated

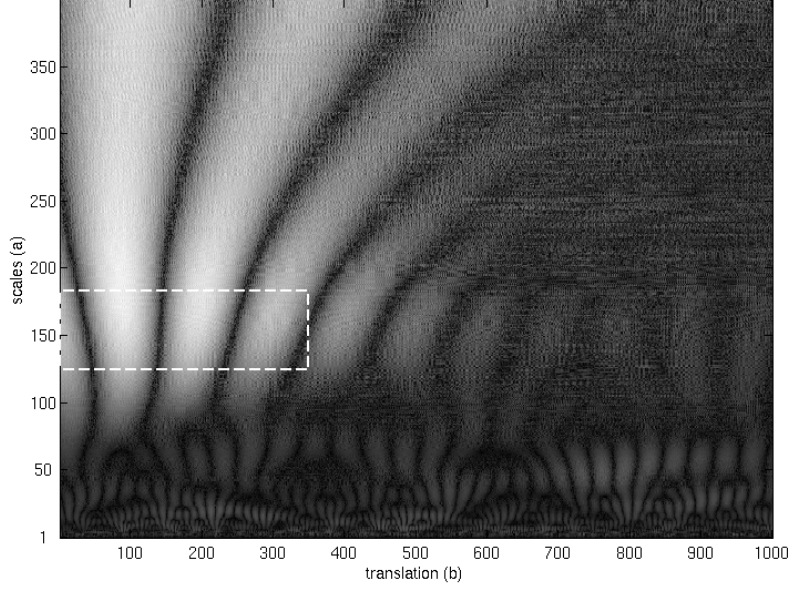


Figure 4.5: The combination of the cosine and exponential terms result in the coefficient diagram shown here. The bounding box, represented by the dashed line, isolates out the cosine term from the high frequency noise and diverging exponential term.

data it is also necessary to first minimize the function

$$\chi_{tmp} = \sqrt{\sum_{a=1}^{a_n} \sum_{b=0}^{b_n} \frac{(W_{a,b,tmp} - \alpha_{tmp} X_{a,b})^2}{\sigma_b^2}} \quad (4.3)$$

for the value of  $\alpha_{tmp}$ . This fixes any problems that may arise from an inherent skew in the values of  $W_{a,b,tmp}$  and  $X_{a,b}$ .

### 4.3 Results & Discussion

The results presented here provide a proof of concept for the Template Fitting Method and demonstrate its degree of accuracy along with any limitations found in the fitting process. All experimental data presented here was generated using the technique described in section 4.1.

Plotting  $\chi_{tmp}$  as a function of the underlying oscillation frequency for the template, as shown in figure 4.8, will typically result in a parabolic curve centered about the frequency of interest. The frequency  $\Delta m$  is taken to be the minimum of this curve with error bounds set at the 95% confidence level (CL) bounds of a cubic polynomial fit.

The experimental data set whose frequency is to be extracted is generated with 20,000 sample points and a randomly selected oscillation frequency of  $\Delta m = 6.418$ . Detection errors are simulated with 28% of the mixed and un-mixed points being swapped, as well as 5% of the samples being due to injected background errors. Once generated, the samples are binned and the

continuous wavelet transform is taken over a broad sweep of scales to assist in deciding upon the placement of the bounding box. The transform of both the  $P_u$  and  $P_m$  datasets is shown in figure 4.6 with the bounding box highlighted.

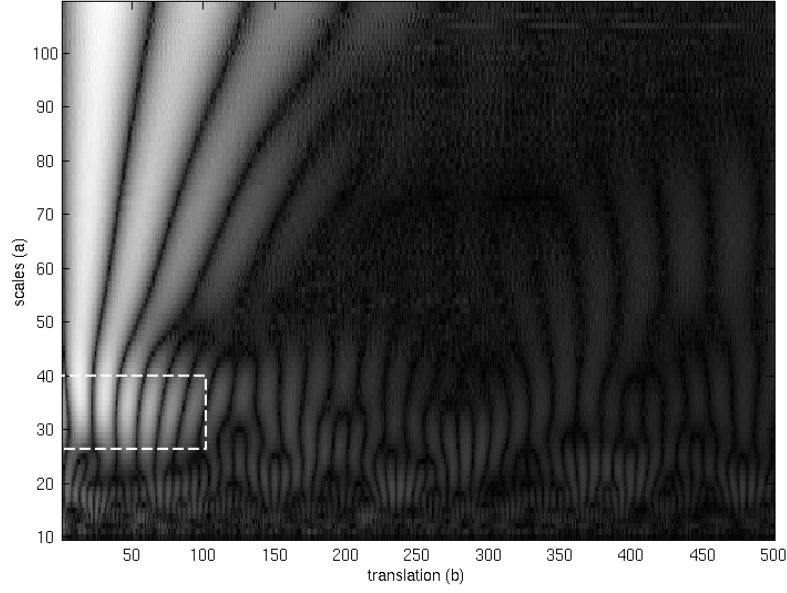
Since the Template Fitting Method is used in a recursive manner, two iterations are necessary to isolate  $\Delta m$ . The first iteration is performed using nine templates evenly spaced over a frequency range of  $3 < \Delta m < 12$  oscillations per lifetime. The goal of the first iteration is to find a general area over which a more accurate second iteration can be performed. The templates are generated using the same number of sampling points as the experimental data, but with 30% swapped events and 2% background.

The calculated value of  $\chi_{tmp}$  is found to be minimized within a frequency range of  $5 < \Delta m < 7$  oscillations per lifetime, as seen in figure 4.7. This is the range of frequencies used for the second iteration, now using 100 evenly spaced templates to obtain a higher degree of accuracy when extrapolating the minimum. The results from the second iteration are summarized in figure 4.8. The extracted frequencies are  $6.52 \pm 0.14$  for the  $P_m$  data and  $6.44 \pm 0.14$  for the  $P_u$  data, with a mean frequency of  $6.48 \pm 0.14$  oscillations per lifetime. The extracted frequency shows a good agreement with the experimental dataset, demonstrating that the Template Fitting Method is applicable to this kind of dataset.

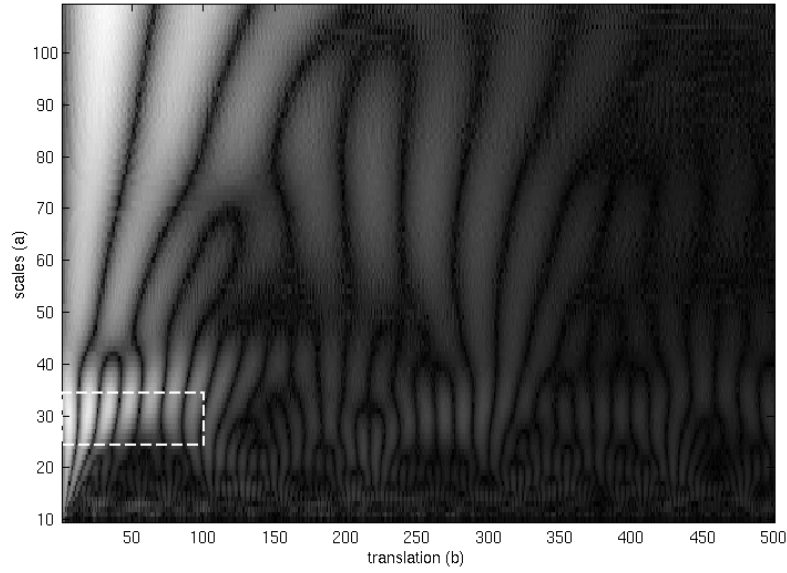
The results produced here are at the same (scaled) level of precision shown by the experimental results produced by CDF from Spring 2006 which place the  $B_s^0$  mass difference at  $\Delta m_{B_s^0} = 17.33 ps^{-1}$  with a statistical error of  $^{+0.42}_{-0.21}$  [2]. However it is necessary that the Template Fitting Method be compared directly to such methods on the same platform to ensure that it produces results which are comparable.

Test analyses produced thus far do not provide enough evidence that the Template Fitting Method is applicable to data sets with a high frequency. Results using the method demonstrated here show that the precision begins to degrade at  $\Delta m > 12$ , at which point this method fails to extract a relevant frequency. This issue is caused by the overlay of noise with the mixing oscillations in the continuous wavelet transform's coefficient diagrams. This makes it impossible to set a bounding box which isolates the  $\cos(\Delta mt)$  term (from the equation 2.20 for  $P_{u,m}$ ) from the background errors.

There is some hope that these problems may be overcome by averaging the amplitude of the coefficient diagrams made at the same frequency but with varying levels of noise. In this way it may be possible to isolate out those characteristics in the coefficient diagrams which are unique to a specific frequency no matter what level noise is injected. It is also possible that templates generated with no noise, or generated from the cosine term alone may provide a better fit. The adaptability of the Template Fitting Method is a promising quality, it also means that a significant amount of time is required to perfect the method used for analysing a dataset as complex as the flavor oscillations of  $B$  mesons.

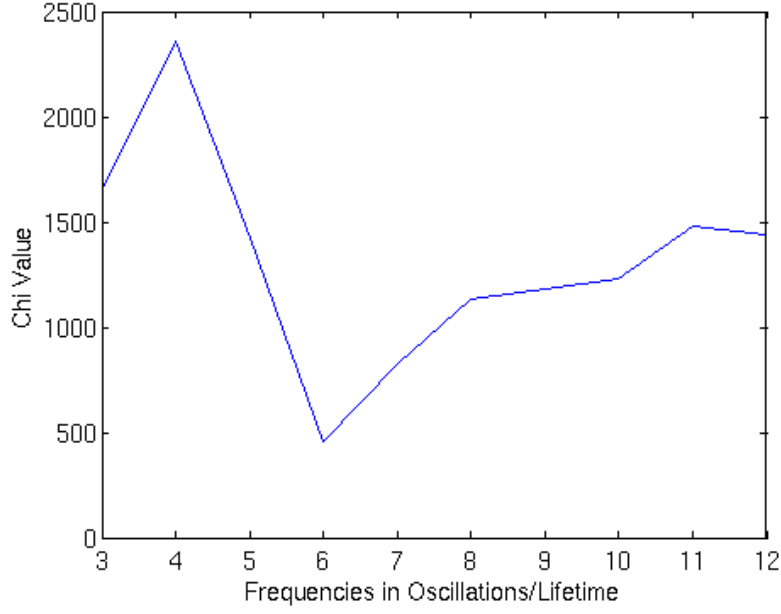


(a)

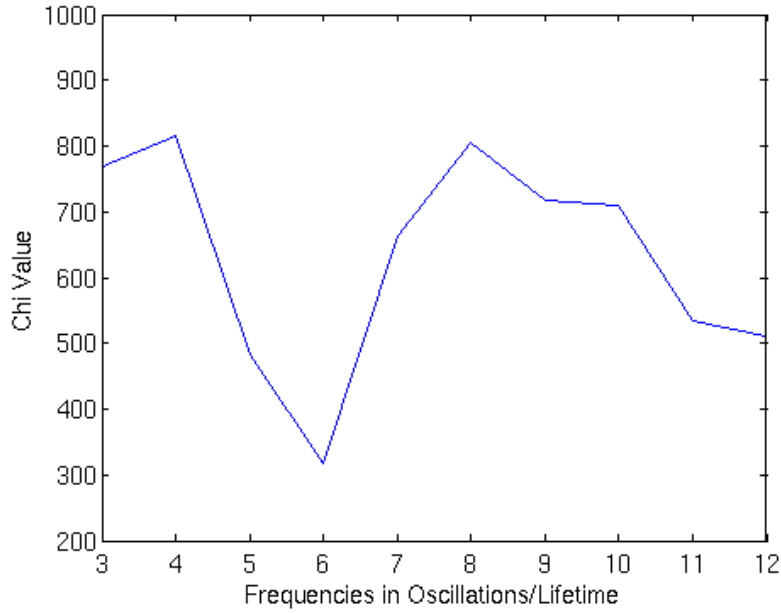


(b)

Figure 4.6: The bounding box for both the mixed, figure (a), and un-mixed, figure (b) test datasets. The scales selected are  $26 < a_m < 40$  and  $23 < a_u < 33$  for the mixed and un-mixed datasets respectively. In order to provide enough points to get a good measure of  $\chi_{tmp}$  a scale spacing of  $\Delta a = 0.5$  is used to double the number of points generated.

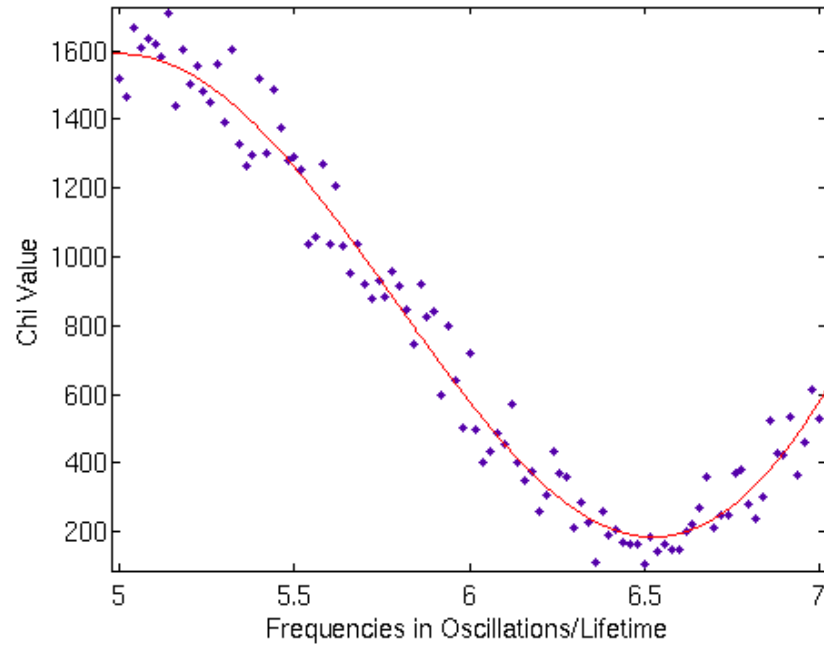


(a)

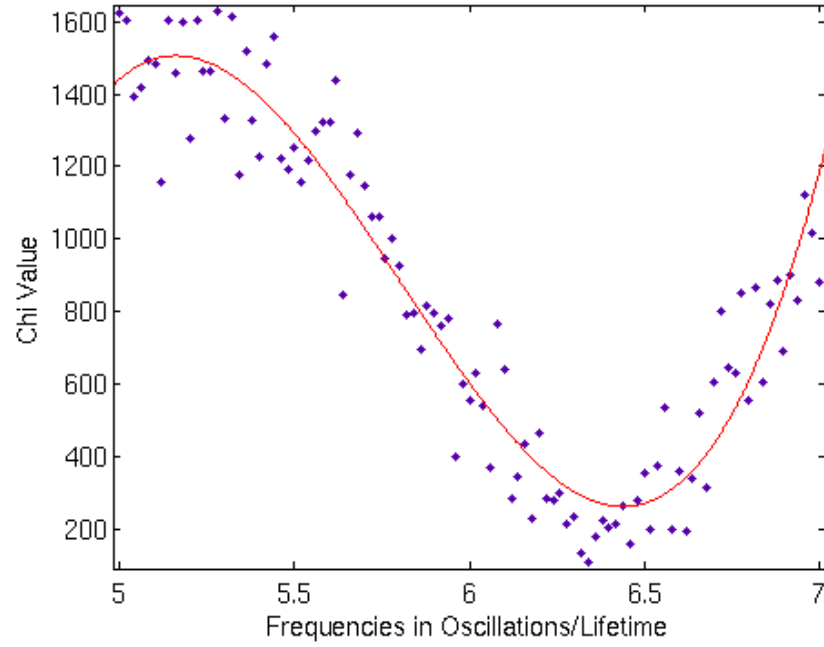


(b)

Figure 4.7: (a) The first iteration results of the Template Fitting Method, with  $\chi_{tmp}$  values as a function of the frequency for  $P_m(t)$ . (b) The corresponding results for the first iteration of  $P_u(t)$ . The Template Fitting Method shows a good fit in the vicinity of  $\Delta m = 6$ . The first iteration has successfully identified the closest frequency to the one present in the simulated experimental data.



(a)



(b)

Figure 4.8: (a) The second iteration results for  $P_m(t)$ . (b) The corresponding second iteration results for  $P_u(t)$ . Each point corresponds to a  $\chi_{tmp}$  value, the solid curve generated from a cubic polynomial fit. The minimum of this curve is the extracted frequency.

## Chapter 5

# Conclusion

$B$  mixing is a convenient event for measuring some of the CKM matrix elements. Because the mass difference between its heavy and light states is directly applicable to finding  $V_{td}$  and  $V_{ts}$  numerous experiments are dedicated to improving the statistics for measuring its value. However the meson's high rate of oscillation in addition to the difficulty involved in flavor tagging make the results of these experiments troublesome to analyse.

The Template Fitting Method has been developed for the purpose of improving the statistical error on the quantity  $\Delta m$ . In section 4.3 it has been shown that the Template Fitting Method can find the frequency of oscillation for data sets composed of 35% false points with a residual of 0.062 which is well within the error bounds of  $\pm 0.14$ . It is clear that the Template Fitting Method has applications in extracting the  $B$  mixing frequency, however it has not yet been developed to the point of completion. The scope of this project has not allowed for the development of a more fine tuned process.

In the future there are many improvements to the Template Fitting Method which may make it a far more accurate tool for data analysis. Using the de-noising capabilities of wavelets to extract out high frequency white noise may make it possible to correctly extract out larger values of  $\Delta m$ . Averaging a number of wavelets at the same frequency to create generic templates which highlight the noise independent qualities of the data set could reduce the effects of a low signal to noise ratio. Applying non-rectangular bounding areas that better fit the sinusoidal peaks found in the coefficient diagrams, may improve the calculation of  $\chi_{tmp}$ .

Experiments which are expensive to make and run commonly produce low statistics and contain large sources of error. This is a significant problem with many of the enormous experiments produced to verify the standard model. Measuring  $B$  meson decays requires an immense amount of resources. Despite that, it is possible that the biggest improvement to the accuracy of these measurements will not come from the expansion of the experiment, but from new methods for analysing the data they produce.

# Bibliography

- [1] Belle Group, Nature 452, 332-335 (2008) .
- [2] W.-M. Yao et al. (Particle Data Group), J. Phys. G **33**, 1 (2006).
- [3] M. Gell-Mann, A.Pais, Phys. Rev. 97, 1387 (1955) .
- [4] T.D. Lee and C. N. Yang, Phys, Rev, 104, 254 (1956).
- [5] J.H. Christenson et al., Phys. Rev. Lett. 13, 138 (1964).
- [6] D. Griffiths, “Introduction To Elementary Particles”, John Wiley & Sons (1987).
- [7] C. Gay, Annu. Rev. Nucl. Part. Sci. 50, 577-641 (2000).
- [8] J. Rademacker, “Evaluation of the LHCb RICH Detectors and a Measurement of the CKM Angle”, University of Oxford, 1-28 (2001)
- [9] L. Wolfenstein, Phys. Rev. D. 31, 2381 (1985).
- [10] A. Bondar, P. Pakhlov, A. Poluektov, Phys.-Usp. 50, 669-690 (2007).
- [11] A. Grossmann, J. Morlet, ”Decomposition of Hardy functions into square integrable wavelets of constant shape” SIAM J. Math., 15, 723-736 (1984)
- [12] Y. Meyer, R. Ryan, “Wavelets Algorithms & Applications”,SIAM (1993).
- [13] W. Hardle, G. Kerkycharian, D. Picard, A. Tsybakov, “Wavelets, Approximations, and Statistical Applications”, Springer (1998).
- [14] P. Addison, “The Illustrated Wavelet Transform Handbook”, IoP (2002).
- [15] C. Chui, “An Introduction to Wavelets”, Academic Press Inc. (1992).



# Appendix A

## Programs

All programs presented here are written in the Matlab scripting language and will run in any stable Matlab Version 7.0 or better environment with the Wavelet Toolbox. The Curve Fitting Toolbox also available for Matlab is of considerable use when analysing the  $\chi$ -frequency plots for a minimum with error bounds.

### A.1 Program to Generate $P_{u,m}(t)$

```
function [pu_data,pm_data]=generate_master_data(samples,mxtau,gamma,freq)
%Function [pu_data,pm_data]=generate_master_data(samples,mxtau,gamma,freq)
%Returns the master datasets (pu_data,pm_data) each with a length samples.
%The master curve is sampled anywhere from t=0 to t=mxtau.

%Seed the random number generator
rand('state',sum(100*clock));

%Initialization
pu_data=zeros(1,samples);
pm_data=zeros(1,samples);
pu_peakval=0.5*gamma*exp(-gamma*0/freq)*2;
pm_peakval=0.5*gamma*exp(-gamma*0.5/freq)*2;
pu_master=inline('0.5*g.*exp(-g*t).*(1+cos(2*pi*f.*t))');
pm_master=inline('0.5*g.*exp(-g*t).*(1-cos(2*pi*f.*t))');

%Monte Carlo Method for Data Generation
for i=1:samples,

    %Generate un-mixed Data.
    while pu_data(i) == 0,
        %Select a random sampling point.
        rndtime=rand*mxtau;
        %Evaluate the function at the sampling point.
        master_time=pu_master(freq,gamma,rndtime);

        %Randomly drop values
        if rand*pu_peakval<master_time,
```

```
        pu_data(i)=rndtime;
    end
end

%Generate Mixed Data, done in the same was as un-mixed data.
while pm_data(i) == 0,
    rndtime=rand*mxtau;
    master_time=pm_master(freq,gamma,rndtime);

    if rand*pm_peakval<master_time,
        pm_data(i)=rndtime;
    end
end
end
```

## A.2 Program that Applies Simulated Detection Errors

```
function [pu_final,pm_final]=sim_error(pu_data,pm_data,masterpnts,...
                                     bgapnts,bgbpnts,smearfun,swaprate)
%[pu_final,pm_final]=sim_error(pu_data,pm_data,masterpnts,...
%    bgapnts,bgbpnts,smearfun,swaprate)
%run after generating master sample data using gen_master_data
%datafile is the name of a .mat file containing at least:
%2 variables of min size <1x100000 (double)>, called pm_data and pu_data
%Arguments:
%masterpnts - <100000, the number of points to use from the pu/pm master sets
%bgapnts    - <10000, the number of points to take from the background a curve
%bgbpnts    - <10000, the number of points to take from the background b curve
%smearfun   - usually '0.1*t', the time dependent function for smearing data
%swaprate   - the % (0<sr<1) of misses that should occur between pu and pm data

rand('state',sum(100*clock));

%Initialize

load bga_samp %loads bga_noise_samp with 10000 points, the 1st background curve
load bgb_samp %loads bgb_noise_samp with 10000 points, the 2nd background curve
bga_master_sz=10000;
bgb_master_sz=10000;
bga_noise_samp=bga_noise_samp/max(abs(bga_noise_samp));
bgb_noise_samp=bgb_noise_samp/max(abs(bgb_noise_samp));
pm_master_sz=length(pm_data);
pm=zeros(masterpnts+bgapnts+bgbpnts,1);
pu=zeros(masterpnts+bgapnts+bgbpnts,1);

%select points to use at random from master noisy sets
pmnormfact=(1/2)*max(abs(pm_data));
punormfact=(1/2)*max(abs(pu_data));
for i=1:masterpnts,
    pm(i)=pm_data(ceil(rand*pm_master_sz));
    pu(i)=pu_data(ceil(rand*pm_master_sz));
end
for i=masterpnts+1:masterpnts+bgapnts,
    pm(i)=bga_noise_samp(ceil(rand*bga_master_sz))*pmnormfact;
    pu(i)=bga_noise_samp(ceil(rand*bga_master_sz))*punormfact;
end
for i=masterpnts+bgapnts+1:masterpnts+bgapnts+bgbpnts,
    pm(i)=bgb_noise_samp(ceil(rand*bgb_master_sz))*pmnormfact;
```

```

    pu(i)=bgb_noise_samp(ceil(rand*bgb_master_sz))*punormfact;
end

clear bga_noise_samp;
clear bgb_noise_samp;
clear pm_data;
clear pu_data;

%smear the data using the smear function arguement
sigmafunc=inline(smearfun);

shiftvals=randn(size(pm));
shiftvals=shiftvals.*sigmafunc(pm);
mxdat=max(pm);
for i=1:length(pm)
    while(pm(i)+shiftvals(i)>mxdat)
        shiftvals(i)=randn(1);
        shiftvals(i)=shiftvals(i).*sigmafunc(pm(i));
    end
end
pm_smear=pm+shiftvals;
clear pm;

shiftvals=randn(size(pu));
shiftvals=shiftvals.*sigmafunc(pu);
mxdat=max(pu);
for i=1:length(pu)
    while(pu(i)+shiftvals(i)>mxdat)
        shiftvals(i)=randn(1);
        shiftvals(i)=shiftvals(i).*sigmafunc(pu(i));
    end
end
pu_smear=pu+shiftvals;
clear pu;

%swap the data using miss rate arguement
k=0;
datsz=size(pm_smear);
for i=1:datsz(1),
    for j=1:datsz(2),
        if(rand<=swaprate),
            k=k+1;
            temp=pu_smear(i,j);
            pu_smear(i,j)=pm_smear(i,j);

```

```
        pm_smear(i,j)=temp;
    end
end
end
pm_final=pm_smear;
pu_final=pu_smear;
```

### A.3 Program to Extract $\Delta m$ using the Template Fitting Method

```
function frq = template_extraction(tfreqs,tbox_pm,tbox_pu,xfreq,dirc,...
                                basename,generate_data,with_plots)
%[frq,err] = template_extraction(template_freq,template_box,...
%                               experimental_freq,directory,base_filename,generate_data,with_plots)
%Generates or loads experimental data and templates to perform the template
%fitting method to extract the oscillation frequency in the experimental data.
%
%Parameters:
%frq,err: The frequency extracted along with the error associated with the
%         measurement.
%tbox: A 3 vector specifying the bounding box to be used in generating the
%       templates and experimental data. Has the form:
%       [init_scale,del_scale,last_scale; init_posit,del_posit,last_posit]
%tfreqs: A vector specifying the template frequencies to make the templates at
%xfreq: The experimental frequency to attempt to find
%dirc,basename: Strings. if generate_data==0 then the templates are loaded
%               from the directory/file 'dirc/basename_[pu|pm]_f<template freq>'
%               Otherwise this is the base filename used to save the
%               templates under.
%generate_data: An int, if==3 then generates JUST templates & loads exp.
%               if ==2 then the script generates BOTH the template data and
%               experimental data required and saves for re-use. if==1, then generates
%               JUST the experimental data, and loads the template files. if==0, then
%               BOTH the template and experimental data are loaded.
%with_plots: A boolean, if==1 then plots of each template are saved,
%             along with plots corresponding to the fitting process

%Initialization
scales_pm=tbox_pm(1,1):tbox_pm(1,2):tbox_pm(1,3);
scpm_len=length(scales_pm);
posits_pm=tbox_pm(2,1):tbox_pm(2,2):tbox_pm(2,3);
pspm_len=length(posits_pm);
scales_pu=tbox_pu(1,1):tbox_pu(1,2):tbox_pu(1,3);
scpu_len=length(scales_pu);
posits_pu=tbox_pu(2,1):tbox_pu(2,2):tbox_pu(2,3);
pspu_len=length(posits_pu);
prefix=strcat('./',dirc,'/',basename,'_');

%Template Generation or Loading
switch generate_data
```

```

case 3 %gen temp, load exp
    %Load the experimental data
    suffix=sprintf('xdata_f%i',dub2int(xfreq));
    name=strcat(prefix,suffix);
    load(name);
    %Generate temapltes
    templates=gen_templates(tfreqs,scales_pm,posits_pm,...
                            scales_pu,posits_pu,with_plots,prefix);

case 2 %gen temp and exp
    %Generate experimental data
    xdata=gen_xdata(xfreq,scales_pm,posits_pm,...
                    scales_pu,posits_pu,with_plots,prefix);

    %Generate templates
    templates=gen_templates(tfreqs,scales_pm,posits_pm,...
                            scales_pu,posits_pu,with_plots,prefix);

case 1 %load temp, gen exp
    %Generate experimental data
    xdata=gen_xdata(xfreq,scales_pm,posits_pm,...
                    scales_pu,posits_pu,with_plots,prefix);

    %Load the template Data
    suffix='templatedata';
    name=strcat(prefix,suffix);
    load(name);

case 0 %load temp and exp
    %Load Experimental Data
    suffix=sprintf('xdata_f%i',dub2int(xfreq));
    name=strcat(prefix,suffix);
    load(name);
    %Load the template Data
    suffix='templatedata';
    name=strcat(prefix,suffix);
    load(name);

otherwise
    return;

end

%Now the experimental data and template data are present, in:
%  xdata(pum,sc,pos) and templates(pum,fq,sc,pos)

%Perfrom the Frequency Extraction using template truth matching.
tcoef_vals=zeros(2,length(tfreqs)); %The template coefficients
tchisq=zeros(2,length(tfreqs)); %Chi-square template-xdata fit value.

%Precalculate the uncertainty for finding chi-squared

```

```

sigpm=sqrt(squeeze(xdata(1,1:scpm_len,1:pspm_len)*1.25));
sigpu=sqrt(squeeze(xdata(2,1:scpu_len,1:pspu_len)*1.25));

%options used for minimizing the template coefficient.
opts=optimset('MaxFunEvals',1e5,'MaxIter',1e5);
cnt=0; %counter used for indexing
tmp=zeros(max(scpm_len,scpu_len),max(pspm_len,pspu_len));
for i=1:length(tfreqs)
    cnt=cnt+1;
    tmp(1:scpm_len,1:pspm_len)=squeeze(templates(1,i,1:scpm_len,1:pspm_len));
    [tcoef_vals(1,cnt),tchisq(1,cnt)]=fminsearch(@(tcoef)...
        sqrt(sum(sum(((tmp(1:scpm_len,1:pspm_len)-...
            tcoef.*squeeze(xdata(1,1:scpm_len,1:pspm_len))))...
./sigpm(1:scpm_len,1:pspm_len)).^2))),1,opts);
    tmp(1:scpu_len,1:pspu_len)=squeeze(templates(2,i,1:scpu_len,1:pspu_len));
    [tcoef_vals(2,cnt),tchisq(2,cnt)]=fminsearch(@(tcoef)...
        sqrt(sum(sum(((tmp(1:scpu_len,1:pspu_len)-...
tcoef.*squeeze(xdata(2,1:scpu_len,1:pspu_len))))...
./sigpu(1:scpu_len,1:pspu_len)).^2))),1,opts);
end
%Save the template comparison data.
suffix=sprintf('templatechisq_f%i',dub2int(xfreq));
name=strcat(prefix,suffix);
save(name,'tchisq');

[val,pm_ind]=min(tchisq(1,:));
[val,pu_ind]=min(tchisq(2,:));

if(with_plots)
    mn=0;
    mx1=max(tchisq(1,:));
    mx2=max(tchisq(2,:));
    figure; plot(tfreqs,tchisq(1,:), 'o');
    hold on; plot([xfreq,xfreq],[mn,mx1], '--');
    xlabel('Frequency (\Delta m) in Oscillations/Lifetime');
    ylabel('Chi-squared');
    suffix='pm_extractchisq_f';
    name=strcat(prefix,suffix);
    saveas(gcf,name,'fig');
    close(gcf);
    figure; plot(tfreqs,tchisq(2,:), 'o');
    hold on; plot([xfreq,xfreq],[mn,mx2], '--');
    xlabel('Frequency (\Delta m) in Oscillations/Lifetime');
    ylabel('Chi-squared');

```



```

    suffix='pu_extractchisq_f';
    name=strcat(prefix,suffix);
    saveas(gcf,name,'fig');
    close(gcf);
end

frq(1)=tfreqs(pm_ind);
frq(2)=tfreqs(pu_ind);

end

function integer = dub2int(dub)
%function integer = dub2int(double)
%Converts a double to an integer by multiplying by 10 to get rid of decimal,
%and takes the absolute value.
    while(mod(dub,1)>0 && dub<1e3)
        dub=dub*10;
    end
    integer=abs(dub);
end

function xdata=gen_xdata(xfreq,scales_pm,posits_pm,...
                        scales_pu,posits_pu,with_plots,prefix)

    gen_samp=100000;
    datasetsamp=49000;
    bgasamp=500;
    bgbsamp=500;
    hpnts=2000;
    mxtau=5;
    gamma=1;
    smear='0.1*t';
    swap=0.3;

    [pu_data,pm_data]=generate_master_data(gen_samp,mxtau,gamma,xfreq);
    [pu_td,pm_td]=sim_error(pu_data,pm_data,datasetsamp,...
                            bgasamp,bgbsamp,smear,swap);
    xdata=zeros(2,max(length(scales_pm),length(scales_pu)),...
                max(length(posits_pm),length(posits_pu)));

    xdata_hist(1,:)=hist(pm_td,hpnts);
    xdata_hist(2,:)=hist(pu_td,hpnts);
    abs(cwt(xdata_hist(1,posits_pm),scales_pm,'morl'));

```

```

xdata(1,1:length(scales_pm),1:length(posits_pm))=...
    abs(cwt(xdata_hist(1,posits_pm),scales_pm,'morl'));
xdata(2,1:length(scales_pu),1:length(posits_pu))=...
    abs(cwt(xdata_hist(2,posits_pu),scales_pu,'morl'));

if(with_plots)
    %Generate P_m plot and save/close
    figure;
imagesc(squeeze(xdata(1,1:length(scales_pm),1:length(posits_pm))));
    xlabel('position (unitless)');
    ylabel('scales');
    suffix=sprintf('xdata_pm_f%d',dub2int(xfreq));
    name=strcat(prefix,suffix);
    saveas(gcf,name,'fig');
    close(gcf);
    %Generate P_u plot and save/close
    figure;
imagesc(squeeze(xdata(2,1:length(scales_pu),1:length(posits_pu))));
    xlabel('position (unitless)');
    ylabel('scales');
    suffix=sprintf('xdata_pu_f%d',dub2int(xfreq));
    name=strcat(prefix,suffix);
    saveas(gcf,name,'fig');
    close(gcf);
end
suffix=sprintf('xdata_f%i',dub2int(xfreq));
name=strcat(prefix,suffix);
save(name,'xdata_hist','xdata');
end

function templates=gen_templates(tfreqs,scales_pm,posits_pm,...
                                scales_pu,posits_pu,with_plots,prefix)

    gen_samp=100000;
    datasetsamp=49000;
    bgasamp=500;
    bgbsamp=500;
    hpnts=2000;
    mxtau=5;
    gamma=1;
    smear='0.1*t';
    swap=0.3;
    mxscal=max(length(scales_pm),length(scales_pu));
    mxposi=max(length(scales_pm),length(scales_pu));
    %Generate the templates

```

```

tdata_hist=zeros(2,length(tfreqs),hpnts); %[pm or pu, frequency, tau]
templates=zeros(2,length(tfreqs),mxscal,mxposi); %[pm|pu,freq,scale,position]

%Generate P_u and P_m curves.
for i=1:length(tfreqs)
    [pu_data,pm_data]=generate_master_data(gen_samp,mxtau,gamma,...
                                           tfreqs(i));
    [pu_td,pm_td]=sim_error(pu_data,pm_data,datasetsamp,...
                           bgasamp,bgbsamp,smear,swap);
    tdata_hist(i,1,:)=hist(pm_td,hpnts);
    tdata_hist(i,2,:)=hist(pu_td,hpnts);
end
%Free up some space.
clear pu_data pm_data pu_td pm_td;

%Generate the templates for each P_u and P_m curve.
for i=1:length(tfreqs)
    templates(1,i,1:length(scales_pm),1:length(posits_pm))=...
        abs(cwt(tdata_hist(i,1,posits_pm),scales_pm,'morl'));
    templates(2,i,1:length(scales_pu),1:length(posits_pu))=
        abs(cwt(tdata_hist(i,2,posits_pu),scales_pu,'morl'));
end

if(with_plots)
    for i=1:length(tfreqs)
        %Generate P_m plot and save/close
        figure;
        imagesc(squeeze(templates(1,i,1:length(scales_pm),...
                                1:length(posits_pm))));
        xlabel('position (unitless)');
        ylabel('scales');
        suffix=sprintf('template_pm_f%d',dub2int(tfreqs(i)));
        name=strcat(prefix,suffix);
        saveas(gcf,name,'fig');
        close(gcf);

        %Generate P_u plot and save/close
        figure;
        imagesc(squeeze(templates(2,i,1:length(scales_pu),...
                                1:length(posits_pu))));
        xlabel('position (unitless)');
        ylabel('scales');
        suffix=sprintf('template_pu_f%i',dub2int(tfreqs(i)));

```

```
        name=strcat(prefix,suffix);
        saveas(gcf,name,'fig');
        close(gcf);
    end
end

suffix='templatedata';
name=strcat(prefix,suffix);
save(name,'templates');
clear tdata_hist;
end
```