STORAGE RING LATTICE MODELING AND ITS APPLICATIONS

Lingyun Yang

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirement
for the degree
Doctor of Philosophy
in the Department of Physics,
Indiana University

September 2007
Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirement for the degree of Doctor of Philosophy.

Shyh-Yuan Lee, Ph.D.

Steven Gottlieb, Ph.D.

Mark Hess, Ph.D.

September 20th, 2007

Paul W. Purdom, Ph.D.
To Jing and our parents

献给静，以及我们的父亲母亲
Acknowledgments

This page is in the very first few pages, but it is the last page I wrote, because I know this page is used to thank people who have helped me and even a few minutes ago, I was getting help from engineers at the NSLS (National Synchrotron Light Source), BNL.

First of all, I would like to thank my advisor, Prof. S.Y. Lee. Without him, this dissertation would simply not exist. I treasure the enjoyable experience of doing research with him. His encouragement has been like an RF cavity in a storage ring, keeping me both energized and in a proper phase. I will never forget the Thanksgiving dinners at his home. Because of him, his turkey and beef, although we were tens of thousands of miles away from home, we felt that we were not.

I would also like to thank Dr. Boris Podobedov, together with Prof. Lee, he provided the opportunity for me to go into the control room and do the hands-on experiments. During the VUV ring beam lifetime studies, his insights about physics and organization made the research smooth and fruitful. He turned the formulas in my head into the real knobs in the control room. Whenever I had any question about the ring, he could always find the right person to help me. Stephen L. Kramer taught me a lot about the control system of the VUV ring, and many other measurement setups he made. His rich experience made the beam study more efficient and easier. I am very grateful for his careful correction of my writings. Pauline Pearson, Susila Ramamoorthy, Yong Tang and Cheo Teng helped me with the control and analysis software. I thank them for writing more software extensions to facilitate my beam study.

When simulating the VUV ring beam lifetime, Dr. Michael Borland and Dr. Weiming Guo gave me a lot of help on elegant and Tcl/Tk. The replies from them
are always timely. I thank Dr. Aimin Xiao for her effort on implementing Piwinski’s formula; it has been a useful tool in my research.

The discussions with Dr. Xiaobiao Huang are productive all the time. His earlier experience on the ORM (Orbit Response Matrix) analysis and the ICA greatly shortened the time I need to develop the new code.

I Thank my fellow student Zhengzheng Liu. She helped me with converting the MAD lattice file, and her confidence on YAOC gives me a lot of encouragement.

Renkai Li deserves my special thanks, with his sharp mind, carefulness and patience, he served as my editor. His suggestions on both language and physics put this dissertation into a better shape. Of course, all the errors would be mine.

During the years, I have received various kinds of help. It is hard to record all the details, and I apologize for not being able to thank them all by name.

Words fail me to express my appreciation to my family. It is a great comfort knowing that they are always there whenever I need them.
Abstract

Lingyun Yang

Storage Ring Lattice Modeling and Its Applications

Small amplitude particle motion around its closed orbit in an accelerator can be well approximated by the linear betatron oscillation. This linearized betatron oscillation is simple harmonic motion governed by Hill’s equation. The closed orbit may be distorted by a dipole error or an orbit bump, and the observed distortion divided by the orbit bump strength (dipole error) is the Green’s function. Measurement of Green’s function can uniquely determine the linear part of the dynamical system.

The Orbit Response Matrix (ORM) measure the Green’s function at BPM (Beam Position Monitor) location with orbit bumpers at various locations. This measurement has been shown to be a powerful tool in accelerator modeling in finding gradient errors, quadrupole tilt angles, BPM gain factors and the tilt angles.

A new orbit code named YAOCC (Yet Another Orbit Code) is developed to reconstruct the ring lattice based on the ORM data. The code can be used to model the errors of quadrupoles and BPMs of a running machine, and to restore the symmetry of the linear lattice. Using a powerful optimization algorithm called Scaled Levenberg-Marquardt algorithm, we get a much better convergence in simulations than the earlier code developed at Indiana University. This new code has been used to study the Fermilab Booster, the Taiwan Photon Source (TPS) design lattice and the NSLS VUV ring at BNL. Thousands of accelerator error models were randomly generated and successfully reconstructed. The symmetry of lattice can thus be restored, except for very few cases where the random error distribution accidentally put
the lattice tunes too close to the betatron resonance such that no finite ORM data exists.

YAOC uses a similar beam transport matrix as TRANSPORT, MAD and Elegant up to the 3rd order expansion. Users can implement their own elements with the virtual matrix (VMatrix). The operations between transfer matrices are straightforward due to operator overloading. The modeling and optimization algorithm of YAOC is based on the Scaled Levenberg-Marquardt algorithm, which provides better convergence property and is more robust to avoid the betatron resonance in iterative fittings. YAOC is also extensible to include the misalignment errors.

We have simulated Fermilab Booster with YAOC. In thousands of random lattice, YAOC reconstructed most of them correctly. We have also simulated different levels of errors, up to 25% of the nominal value. YAOC can reconstruct the lattices even when the ORM elements have noise or only part of the BPMs are used. Besides the proton ring, YAOC was also benchmarked on electron storage rings, such as the TPS design lattice and the NSLS VUV ring at BNL. Another application for identifying the source of vertical orbit oscillation is presented.

In this dissertation, I will also discuss the beam lifetime of the VUV ring. Both theoretical and experiment results are presented and compared. Some most fundamental beam qualities are measured, and calibrated, such as bunch length, vacuum pressure, lifetime and momentum acceptance. The effect of vacuum chamber dimensions on beam lifetime will be discussed.
Contents

Acceptance i

Acknowledgments iv

Abstract vi

1 Introduction 1

2 Beam Dynamics 5
   2.1 Curvilinear Coordinate System ......................... 5
   2.2 Linear Betatron Motion .................................. 7
   2.3 Synchrotron Motion ..................................... 13
   2.4 Basic Beam Observation and Instruments ................. 16
      2.4.1 Beam Position Monitor .............................. 16

3 Orbit Response Matrix and Lattice Modeling 19
   3.1 Dipole Error and Closed Orbit .......................... 20
   3.2 Orbit Response Matrix ................................. 22
   3.3 Lattice Modeling with Orbit Response Matrix ............ 25
      3.3.1 Motivation for a New Code ......................... 26
      3.3.2 Least-Square Modeling with ORM .................... 28
3.3.3 Algorithm of Fitting ............................................ 30

4 Benchmark and Application of YAOC .................................. 37
  4.1 Introduction ......................................................... 37
  4.2 Benchmarks on Fermilab Booster .................................. 40
    4.2.1 Gradient Error ................................................. 41
    4.2.2 Gradient and Tilt .............................................. 46
    4.2.3 Statistics ..................................................... 51
  4.3 The Effects of BPMs ............................................... 56
    4.3.1 BPM Noise ..................................................... 56
    4.3.2 Consecutively Disabled BPMs ................................. 59
    4.3.3 BPM Numbers ................................................. 63
    4.3.4 Conclusion on BPM effects .................................. 65
  4.4 Taiwan Photon Source .............................................. 66
  4.5 NSLS VUV Ring .................................................... 75

5 NSLS VUV Ring Lifetime ................................................ 81
  5.1 The NSLS VUV Ring ................................................ 81
  5.2 Beam-gas interaction ............................................. 84
  5.3 Touschek Lifetime ................................................ 90
    5.3.1 Bunch Length Measurement ................................. 93
    5.3.2 Momentum Acceptance ..................................... 93
    5.3.3 Effect of Physical Aperture ............................... 96
  5.4 Conclusion ....................................................... 98

6 Conclusion .............................................................. 99
  6.1 YAOC and ORM analysis .......................................... 99
  6.2 VUV ring beam lifetime study .................................. 100
A Matrix Decomposition

A.1 LU Decomposition ............................................. 105
A.2 Cholesky Decomposition ....................................... 106
A.3 QR Decomposition ............................................. 106
A.4 SVD .......................................................... 106

B Algorithms for Optimization

B.1 Steepest descent method ....................................... 109
B.2 Gauss-Newton method .......................................... 110
B.3 Levenberg-Marquardt Algorithm .............................. 113
  B.3.1 Theoretical Basis of Levenberg-Marquardt Algorithm .. 113
  B.3.2 Classical Levenberg-Marquardt Algorithm ................. 114
  B.3.3 Scaled Levenberg-Marquardt Algorithm .................... 118
List of Figures

2.1 Curvilinear coordinate system for particle motion in synchrotron. . . 6
2.2 FODO cell . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
2.3 Courant-Snyder invariant ellipse . . . . . . . . . . . . . . . . . . . . . . 12
2.4 Phase stability condition . . . . . . . . . . . . . . . . . . . . . . . . . . 15
2.5 Typical buttons BPM . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
2.6 Typical strip-lines BPM . . . . . . . . . . . . . . . . . . . . . . . . . . 17

3.1 Closed orbit distortion . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
3.2 Horizontal COD and Green’s function. (Fermilab Booster lattice) . . 23
3.3 ORM Simulation by X. Huang[14] . . . . . . . . . . . . . . . . . . . . . 27
3.4 One benchmark result of YAOC (based on MAD) . . . . . . . . . . . . 28

4.1 Scheme of benchmark . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.2 One of the 24 FODO cells. Courtesy of X. Huang . . . . . . . . . . . . 40
4.3 Lattice functions of the Fermilab Booster. . . . . . . . . . . . . . . . . 41
4.4 One benchmark example of fitting gradient error. . . . . . . . . . . . . 45
4.5 Benchmark of gradient and tilt angle error. Result of an example from
   Ref. [14]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
4.6 Benchmark of BPM errors. Quadrupole errors are also considered in
   the model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
LIST OF FIGURES

4.7 Gaussian distribution with $\sigma = 2\% K_1$, 1002 random cases in total.  
4.8 Statistical result of gradient fit.  
4.9 Quadrupole strength. The model fits gradient and tilt angle errors.  
4.10 Quadrupole tilt angle. The model fits gradient and tilt angle errors.  
4.11 BPM noise and quadrupole errors  
4.12 BPM noise and BPM errors  
4.13 Gradient error with $10^{-2}$ relative BPM noise  
4.14 Fitted gradient error, when the first 5 BPMs are disabled.  
4.15 Fitted gradient error, when the BPM no.20-24 are disabled  
4.16 Fitting quadrupole errors with 16 BPMs  
4.17 Fitting quadrupole errors with 24 BPMs  
4.18 RMS difference of quadrupole and BPM errors  
4.19 Lattice function of NSRRC TPS Storage Ring design lattice  
4.20 Quadrupole errors  
4.21 BPM errors  
4.22 Vertical $\beta$-function before and after fit  
4.23 One unsuccessful example on quadrupole errors, BPM gain factors look like shifted.  
4.24 Lattice functions  
4.25 RMS of betatron function and gradient  
4.26 $\chi^2$ of different perturbation locations  
4.27 Vertical orbit oscillation at two BPMs  

5.1 VUV ring lattice functions  
5.2 Gas scattering lifetime, ($Z = 7.0$)  
5.3 Lifetime drop when vacuum leaks  
5.4 Lifetime drop when vacuum leaks
5.5 Average vacuum pressure .......................................................... 89
5.6 Lifetime change due to vacuum leak .............................................. 90
5.7 RMS bunch length. Single bunch operation at $I = 5mA$ .................. 94
5.8 Momentum aperture. ................................................................. 95
5.9 Lifetime variation with closed orbit locally bumped near injection septum. ................................................................. 96
5.10 Touschek lifetime simulation and experiment. Single bunch operation, $\approx 130$ mA ................................................................. 97
Chapter 1

Introduction

Accelerator modeling is crucial for understanding the beam dynamics and improving machine performance. Although many codes have been developed to calculate the lattice properties of an accelerator and simulate the beam dynamics within it, e.g. MAD/MAD-X [22, 21], Elegant [1], all of them are based on an assumption that we know all the magnets very well, and the magnet parameters used in our simulations are the same as the operating machine. These codes are very successful in the accelerator design and its concept proof simulations, but there are difficulties with them done the reverse way, i.e. modeling an operating machine from experiment data instead of predicting the experiment data with known accelerator parameters. Especially when the experiment is done under a different operating condition in which the magnets are not as well understood as at a normal operating point. The lattice of the storage ring is optimized for beam quality at the design stage; the magnets and diagnostic components of the ring are carefully chosen to achieve optimum performance. But the ring may be slightly different from what we have designed, either from a long time shutdown, or that it is newly built not yet tuned, or we are doing an operation under very different energy/lattice condition. Some of the components can be calibrated
during maintenance, while some other parameters may be hard to measure directly after the installation. A fast and easy operating method would be very helpful for beam study and daily operation.

The Orbit Response Matrix (ORM) modeling method can be used to find these indirectly measurable parameters for a storage ring. A code, different from the lattice calculation code mentioned above, which can reconstruct the operating lattice from experiment data, would be very helpful. The reversely derived lattice, if very close to a real operating machine, would bring more insight to both accelerator operation and beam physics research. For this reason YAOC (Yet Another Orbit Code) was developed and bench marked.

The ORM is a set of data coming from the closed orbit distortion (COD) due to an orbit bump at another location in the storage ring. Each element of the ORM is the ratio of COD to orbit bump strength. In dynamics, it corresponds to a Green’s function between the orbit bumper and BPM (Beam Position Monitor). Green’s function closely depends on the betatron amplitude functions at the bumper and BPM locations, and the phase advance between them. In an experiment, it is easy to set the bump strength and measure the COD. These can be done automatically by the control system or a MATLAB package such as Accelerator Toolbox (AT) [34, 7].

In the ORM modeling, one tries to minimize the difference between the measured ORM data and those derived from theoretical models. A successful modeling can uniquely determine the quadrupole strength (gradient), BPM gain factors, and the tilt angle of these magnets. After measuring the ORM data, a proper lattice is chosen as the starting point (usually the design lattice), but with some magnet parameters or diagnostic instrument parameters left unfixed; The modeling is carried out in the least square fitting to minimize the difference between the modeling ORM and the experiment ORM data. This difference is characterized by a merit function $\chi^2$. By iteratively tuning these unfixed parameters of our model, a stable point is reached
when the merit function achieves its extreme. At this point a successful model is very close or even equivalent to the operational accelerator from where the ORM data derived. This conclusion is drawn from the fact that the ORM is an ensemble of Green’s functions.

A MATLAB code called LOCO (Linear Optics of Closed Orbit) [31, 30, 29] is already developed to model lattice of an accelerator, such as quadrupole strength, BPM gain factors, corrector scaling factors and their tilt angles. But the convergence property in the iterative search is limited by the numerical algorithm, i.e. Newton-Gauss algorithm. Another code by Dr. Xiaobiao Huang was developed based on the vanilla version of Levenberg-Marquardt algorithm. It is very successful for analyzing Proton Storage Ring (PSR) lattice, but could not resolve the coupling between the neighbor magnets of Fermilab Booster [14]. In this dissertation, a code named YAOC is developed, using a robust and efficient numerical algorithm called the Scaled Levenberg-Marquardt algorithm, to improve the speed and convergence. Extensive benchmarks were done on various accelerators, such as the Fermilab booster, the Taiwan Photon Source (TPS) design lattice and the National Synchrotron Light Source (NSLS) VUV ring at Brookhaven National Laboratory (BNL) to study the uniqueness of the reconstructed lattices. The BPM effects including noise and layout are also discussed based on the Fermilab Booster lattice.

This code can reconstruct the linear lattice from experiment, and can be used to look for the source of orbit perturbation. An application on the NSLS VUV ring shows that it can precisely determine the location and amplitude of the vertical orbit perturbation source.
Chapter 2

Beam Dynamics

In this chapter, I start with the coordinate system, and introduce the basic beam dynamics for particle motion in an accelerator. The single particle motion is discussed in both transverse and longitudinal direction. After that, the Beam Position Monitor (BPM), one of the most common beam instruments for transverse beam measurement, is introduced.

2.1 Curvilinear Coordinate System

In the beam dynamics of accelerator physics, the Curvilinear Coordinate System is used to describe the motion of particles. Let \( r_0 \) be the reference orbit on a plane, the coordinate system in Fig. 2.1 is constructed in the following way

1. The tangent unit vector

\[
\hat{s}(s) = \frac{dr_0(s)}{ds}
\]  

(2.1.1)

2. The normal unit vector

\[
\hat{x}(s) = -\rho(s)\frac{d\hat{s}(s)}{ds}
\]  

(2.1.2)

where \( \rho(s) \) defines the radius of curvature.
3. The bi-normal unit vector is the cross product of $\hat{x}(s)$ and $\hat{s}(s)$

$$\hat{z}(s) = \hat{x}(s) \times \hat{s}(s) \quad (2.1.3)$$

The position of a particle can be decomposed in the coordinate system as shown in Fig. 2.1.

$$\mathbf{r} = \mathbf{r}_0 + x\hat{x} + z\hat{z} \quad (2.1.4)$$

where $x$ and $z$ are betatron coordinates. A canonical transformation is then performed to express the equation of motion in terms of the reference orbit coordinate system. With $s$ as the independent variable instead of time $t$, the new Hamiltonian up to the second order is given by [20]

$$H \approx -p \left(1 + \frac{x}{\rho} \right) + \frac{1 + x/\rho}{2p} [(p_x - eA_x)^2 + (p_z - eA_z)^2] - eA_s \quad (2.1.5)$$

where $\vec{A} = (A_x, A_z, A_s)$ are vector potential of magnetic field, $p = \sqrt{E^2/c^2 - m^2c^2}$ is the total particle momentum and $(x, p_x, z, p_z)$ are conjugate phase-space coordinates.
2.2 Linear Betatron Motion

The transverse motion of a single particle is the sum of two components: the closed orbit and a small-amplitude betatron oscillation. The closed orbit in a synchrotron is defined as a particle trajectory that comes back to itself after one complete revolution. Betatron oscillation around the closed orbit is governed by an arrangement of quadrupoles, called the accelerator lattice.

In synchrotron, bending magnets are needed to provide the complete revolution of particle beam, but for a fixed bending field the bending angle depends on the particle momentum. In linear approximation, the closed orbit deviation is proportional to the fractional off-momentum deviation \( (p - p_0)/p_0 \), where \( p_0 \) is the momentum of a reference particle, and the ratio of closed orbit deviation and fractional off-momentum deviation is called dispersion. It is location dependent.

Neglecting higher-order terms, the betatron equations of motion is [20]

\[
\begin{align*}
    x'' - \frac{\rho + x}{\rho^2} &= \frac{B_x p_0}{B \rho p} \left(1 + \frac{x}{\rho}\right)^2 \\
    z'' &= -\frac{B_x p_0}{B \rho p} \left(1 + \frac{x}{\rho}\right)^2
\end{align*}
\]  

(2.2.1)

where \( B \rho = p_0/e \) is called magnet rigidity. A more practical formula would be

\[
B \rho \ [Tm] = 3.3357p/e \ [GeV/c]
\]  

(2.2.2)

In dipole and quadrupole magnets, we can express the two-dimensional magnetic field as

\[
B = B_x(x, z)\hat{x} + B_z(x, z)\hat{z}
\]  

(2.2.3)

and use the Maxwell’s equation \( \nabla \times \mathbf{B} = 0 \) to solve for \( B_x, B_z \). The equation of motion, so called Hill’s equation therefore becomes

\[
\begin{align*}
    x'' + K_x(s)x &= 0, \quad K_x = 1/\rho^2 - K_1(s) \\
    z'' + K_z(s)z &= 0, \quad K_z = K_1(s)
\end{align*}
\]  

(2.2.4)
where
\[ K_1(s) = \frac{B_1(s)}{B_\rho} \equiv \frac{\partial B_z}{\partial x} \frac{\partial x}{B_\rho} \]

With periodic condition \( K_y(s + L) = K_y(s) \), where \( y \) represents either horizontal or vertical phase space coordinates, the solution to Eq. (2.2.4) becomes
\[
y(s) = \begin{cases} 
a \cos(\sqrt{K} s + b), & K > 0, 

a + b, & K = 0, 
a \cosh(\sqrt{-K} s + b), & K < 0.
\end{cases} \tag{2.2.5}
\]

Considering the initial values of \( y(s_0) \) and \( y'(s_0) \), the solution to can be expressed as
\[
\begin{pmatrix} y(s) \\
y'(s) \end{pmatrix} = M(s|s_0) \begin{pmatrix} y(s_0) \\
y'(s_0) \end{pmatrix} \tag{2.2.6}
\]
where \( M(s|s_0) \) is the betatron transfer matrix connecting two points at \( s_0 \) and \( s \) in phase space. For any two linearly independent solutions \( y_1, y_2 \) of Hill’s equation, the Wronskian is independent of time, i.e.
\[
W(s) = y_1 y'_2 - y'_1 y_2, \quad \frac{dW}{ds} = 0 \tag{2.2.7}
\]

The transfer matrix for a constant focusing function \( K \) is
\[
M(s|s_0) = \begin{cases} 
\begin{pmatrix} \cos(\sqrt{K} \ell) & \frac{1}{\sqrt{K}} \sin(\sqrt{K} \ell) \\
-\sqrt{K} \sin(\sqrt{K} \ell) & \cos(\sqrt{K} \ell) \end{pmatrix} & K > 0 : \text{focusing quad.} \\
\begin{pmatrix} 1 & \ell \\
0 & 1 \end{pmatrix} & K = 0 : \text{drift space} \\
\begin{pmatrix} \cosh(\sqrt{-K} \ell) & \frac{1}{\sqrt{-K}} \sinh(\sqrt{-K} \ell) \\
\sqrt{-K} \sinh(\sqrt{-K} \ell) & \cosh(\sqrt{-K} \ell) \end{pmatrix} & K < 0 : \text{defocusing quad.}
\end{cases} \tag{2.2.8}
\]
where $\ell = s - s_0$ is the length of the magnet.

In thin-lens approximation with $\ell \to 0$ while keeping $K\ell$ a constant, the transfer matrix for a quadrupole is

$$M_{QF} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{pmatrix}, \quad M_{QD} = \begin{pmatrix} 1 & 0 \\ \frac{1}{f} & 1 \end{pmatrix}$$

(2.2.9)

where $f$ is the focal length given by

$$f = \lim_{\ell \to 0} \frac{1}{|K| \ell}$$

(2.2.10)

Similarly, the transfer matrix for a pure sector dipole with $K_x = \frac{1}{\rho^2}$ is

$$M_x(s, s_0) = \begin{pmatrix} \cos \theta & \rho \sin \theta \\ \frac{1}{\rho} \sin \theta & \cos \theta \end{pmatrix}$$

(2.2.11)

where $\theta = \ell / \rho$ is the bending angle and $\rho$ is the bending radius. In thin lens approximation, the transfer matrix of a sector dipole becomes equivalent to that of a drift space in Eq. (2.2.8).

The transfer matrix for any intervals made up of subintervals is just the product of the transfer matrices of the subintervals, e.g.

$$M(s_2|s_0) = M(s_2|s_1)M(s_1|s_0)$$

(2.2.12)

The most general form for matrix $M$ with unit modulus can be parametrized as [8, 20, 5, 3, 2, 24]

$$M = \begin{pmatrix} \cos \Phi + \alpha \sin \Phi & \beta \sin \Phi \\ -\gamma \sin \Phi & \cos \Phi - \alpha \sin \Phi \end{pmatrix} = I \cos \Phi + J \sin \Phi$$

(2.2.13)

where $\alpha$, $\beta$ and $\gamma$ are Courant-Snyder parameters, $\Phi$ is the phase advance, $I$ is the unit matrix and

$$J = \begin{pmatrix} \alpha & \beta \\ -\gamma & -\alpha \end{pmatrix}, \quad \text{with } \text{Tr}(J) = 0, \quad J^2 = -I, \quad \text{or } \beta \gamma = 1 + \alpha^2$$

(2.2.14)
The Courant-Snyder parameters can be obtained from the transfer matrix $M$ as following

$$
\Phi = \begin{cases} 
\arccos \frac{M_{11} + M_{22}}{2} & M_{12} \geq 0 \\
2\pi - \arccos \frac{M_{11} + M_{22}}{2} & M_{12} < 0
\end{cases}
$$

(2.2.15)

$$
\beta = \frac{M_{12}}{\sin \Phi},
$$

(2.2.16)

$$
\alpha = \frac{M_{11} - M_{22}}{2 \sin \Phi},
$$

(2.2.17)

$$
\gamma = -\frac{M_{21}}{\sin \Phi}
$$

(2.2.18)

where $M_{ij}$ is the matrix element of a unit modulus. Since the betatron function $\beta(s)$ is always nonnegative, the fractional part of phase advance $\Phi$ is uniquely determined in $[0, 2\pi]$ by $\cos \Phi$ and the sign of $\sin \Phi$ which is the same as $M_{12}$.

Consider a FODO cell with quadrupole focal length $f_1$ and $-f_2$, where the $\pm$ sign designates the focusing and defocusing quadrupoles respectively. The transfer matrix of a FODO cell $\left\{\frac{1}{2} \text{QF} \ O \ \text{QD} \ O \ \frac{1}{2} \text{QF}\right\}$, as shown in Fig. 2.2, is

$$
M = \begin{pmatrix}
1 & 0 \\
-\frac{1}{2f_1} & 1
\end{pmatrix}
\begin{pmatrix}
1 & L \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
\frac{1}{f_2} & 1
\end{pmatrix}
\begin{pmatrix}
1 & L \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
-\frac{1}{2f_1} & 1
\end{pmatrix}
$$

(2.2.19)
2.2. LINEAR BETATRON MOTION

Since $K(s)$ is a periodic function, $K(s) = K(s + L)$ where $L$ is the length of repetitive modules. The Hill’s equation can be solved by using Floquet transformation

$$y_1(s) = aw(s)e^{i\psi(s)}, \quad y_2(s) = aw(s)e^{-i\psi(s)} \quad (2.2.20)$$

where $a$ is a constant. The equations for envelope function and the phase function are

$$w'' + Kw + \frac{1}{w^3} = 0 \quad (2.2.21)$$
$$\psi' = 1/w^2 \quad (2.2.22)$$

Any solution to Hill’s equation (2.2.4) is a linear superposition of $y_1$ and $y_2$, i.e.

$$y(s) = A_1y_1(s) + A_2y_2(s), \quad \text{and for any one of these superpositions, } (y(s_2), y'(s_2))^T = M(s_2|s_1)(y(s_1), y'(s_1))^T \text{ holds.}$$

i.e. the coefficients, after expand this equation, of $A_1$ and $A_2$ are zeros. This gives the transfer matrix $M(s_2|s_1)$ in terms of $w(s_1)$, $w(s_2)$, $\psi_2 - \psi_1$ and their derivatives with respect to $s$. Using the periodic boundary conditions when $s_2 - s_1 = L$, and comparing this transfer matrix with the modular matrix in Eq. (2.2.13), we have

$$w^2 = \beta \quad (2.2.23)$$
$$\alpha = -ww' = -\beta'/2 \quad (2.2.24)$$

and the transfer matrix $M(s_2|s_1)$ expressed in terms of Courant-Snyder parameters are

$$M(s_2|s_1) = \begin{pmatrix}
\sqrt{\beta_2} \cos \psi + \alpha_1 \sin \psi & \sqrt{\beta_1 \beta_2} \sin \psi \\
\frac{\sqrt{\beta_1 \beta_2}}{1 + \alpha_1 \alpha_2} \sin \psi + \frac{\alpha_1 - \alpha_2}{\sqrt{\beta_1 \beta_2}} \cos \psi & \sqrt{\beta_1} \sqrt{\beta_2} \cos \psi - \alpha_1 \sin \psi
\end{pmatrix} \quad (2.2.25)$$

$$= \begin{pmatrix}
\sqrt{\beta_2} & 0 \\
-\frac{1}{\sqrt{\beta_1}} & \frac{\alpha_2}{\sqrt{\beta_2}} \\
\frac{\sqrt{\beta_1}}{\sqrt{\beta_2}} & -\frac{\alpha_1}{\sqrt{\beta_2}}
\end{pmatrix} \begin{pmatrix}
\cos \psi & \sin \psi \\
-\sin \psi & \cos \psi
\end{pmatrix} \begin{pmatrix}
\frac{1}{\sqrt{\beta_1}} & 0 \\
\alpha_1 & \sqrt{\beta_1}
\end{pmatrix}$$
where $\beta_1 = \beta(s_1)$, $\alpha_1 = \alpha(s_1)$ and $\beta_2 = \beta(s_2)$, $\alpha_2 = \alpha(s_2)$ are values of betatron amplitude functions at $s_1$ and $s_2$. $\psi = \psi(s_2) - \psi(s_1)$.

The general solution becomes

$$y(s) = a \sqrt{\beta_y(s)} \cos[\psi_y(s) + \xi_y], \quad \psi_y(s) = \int_0^s \frac{1}{\beta_y(s)} \, ds$$

(2.2.26)

where $a$, $\xi_y$ are constants determined from initial conditions. The betatron tune $\nu_y$ or $Q_y$ is defined as the number of betatron oscillations in one revolution:

$$\nu_y = Q_y = \frac{1}{2\pi} \int_{s}^{s+C} \frac{ds}{\beta_y(s)}$$

(2.2.27)

The Hill’s equation (2.2.4) can also be derived from a pseudo-Hamiltonian

$$H = \frac{1}{2} y'^2 + \frac{1}{2} K(s) y'^2$$

(2.2.28)
where \((y, y')\) are conjugate phase space coordinates. Applying the canonical transformation with generating function \(F_1(y, \psi) = \int_0^y y'dy\), we obtain

\[
y = \sqrt{\beta J} \cos \psi
\]
(2.2.29)

\[
y' = -\sqrt{\frac{2J}{\beta}} [\sin \psi + \alpha \cos \psi]
\]
(2.2.30)

where \(\alpha = -\beta'/2\) and the action

\[
J = \frac{1}{2\beta} [y^2 + (\beta y' + \alpha y)^2]
\]
(2.2.31)

or equivalently

\[
2J = \gamma y^2 + 2\alpha yy' + \beta y'^2 \equiv \epsilon
\]
(2.2.32)

The phase space area enclosed by the invariant ellipse is equal to \(2\pi J(\epsilon)\). The ellipse is shown geometrically in Fig. 2.3.

The ellipse is traced out by a particle in \((y, y')\) phase space. As the particle moves, the shape of its ellipse evolves, but its area remains constant.

### 2.3 Synchrotron Motion

The reference orbit is designed for a particle with momentum \(p_0\). RF cavities that provide longitudinal electric field for particle acceleration or deceleration at a frequency \(f_{\text{rf}}\) that synchronized with the revolution frequency \(f_0\) of particle motion, i.e. \(f_{\text{rf}} = hf_0\), where \(h\) is called the harmonic number. Let \(\phi_s\) be the synchronous phase of this reference particle, a particle with momentum \(p\) will have phase \(\phi\) relative to the RF wave.

The linearized synchrotron equation of motion is [20]

\[
\frac{d^2}{dt^2}(\phi - \phi_s) = \frac{h \omega_0^2 eV \eta_0 \cos \phi_s}{2\pi \beta^2 E}(\phi - \phi_s)
\]
(2.3.1)
where \( \phi - \phi_s \) is the RF phase relative to synchronous particle, \( h \) is RF harmonic number, \( V \) is RF peak voltage, \( \omega_0 \) and \( E \) are revolution frequency and energy of synchronous particle. \( \eta_0 \) is the phase slip factor which describes how revolution period changes with particle momentum

\[
\eta = \frac{\Delta T/T_0}{\Delta p/p_0} = \alpha_c - \frac{1}{\gamma^2} - \frac{1}{\gamma_t^2} - \frac{1}{\gamma^2}
\]  
(2.3.2)

where \( \Delta p = p - p_0 \), \( \Delta T = T - T_0 \), \( \gamma_T \) is called transition energy, \( \alpha_c \) is momentum compaction factor

\[
\alpha_c = \frac{\Delta C/C}{\Delta p/p}
\]  
(2.3.3)

It is a property of machine lattice characterizing the dependence of path length on particle momentum.

Below the transition energy, with \( \gamma < \gamma_T \) and \( \eta < 0 \), a higher momentum particle will have a revolution period shorter than that of the synchronous particle. Above the transition energy, with \( \gamma > \gamma_T \), the converse is true. At \( \gamma = \gamma_T \), the revolution period is independent of the particle momentum. This is the isochronous condition.

From Eq. (2.3.1), we know that a stable solution is in the form of harmonic oscillation, and the stability condition for synchrotron oscillation is \( \eta_0 \cos \phi_s < 0 \), or equivalently

\[
\begin{cases}
0 \leq \phi_s \leq \pi/2, & \text{if } \gamma < \gamma_T \text{ or } \eta < 0 \\
\pi/2 \leq \phi_s \leq \pi, & \text{if } \gamma > \gamma_T \text{ or } \eta > 0
\end{cases}
\]  
(2.3.4)

As shown in Fig. 2.4 for positive a charged particle, when \( \gamma > \gamma_T \) or \( \eta > 0 \) the low energy particles takes less time traveling one turn back to RF cavity, and arrive earlier than high energy particles. They gain more energy and moving their energy towards to synchronous particles, while the high energy particles arrive later and gain less energy. This condition of \( \phi_s \) and \( \gamma \) makes synchrotron oscillation stable. Similarly we can get the phase stability condition for negative charged particles.
The synchrotron tune, defined as the number of synchrotron oscillations per revolution, is

\[ Q_s = \frac{\omega_s}{\omega_0} = \sqrt{\frac{h e V |\eta_0 \cos \phi_s|}{2\pi^2 \beta^2 E}} \]  

(2.3.5)

where \( \omega_s \) is synchrotron oscillation frequency from Eq. (2.3.1).

In \((\phi, \delta)\) phase space, the trajectory traced out by the particle lies on a constant-H contour. The separatrix is defined to be the particular contour that passes through the Unstable Fixed Point (UFP) \((\pi - \phi_s, 0)\). The Hamiltonian value of the separatrix is [20, Chap. 3]

\[ H_{sx} = H(\phi = \pi - \phi_s, \delta = 0) \]

\[ = \frac{\omega_0 e V}{2\pi^2 \beta^2 E} [\cos \phi_s + (\pi - 2\phi_s) \sin \phi_s] \]  

(2.3.6)

Figure 2.4: Phase stability condition
The phase-space trajectory is
\[ \delta^2 + \frac{eV}{\pi \beta^2 E h \eta} [\cos \phi + \cos \phi_s - (\pi - \phi - \phi_s) \sin \phi_s] = 0 \quad (2.3.7) \]

The phase-space trajectory around the stable fixed point (SFP) is
\[ \psi = \hat{\phi} \cos(\omega_s t + \chi), \quad \delta = -\frac{Q_s}{h |\eta|} \hat{\phi} \sin(\omega_s t + \chi) \quad (2.3.8) \]

where \( \psi = \phi - \phi_s \). The maximum amplitudes of the phase-space ellipse \( \hat{\phi} \) and \( \hat{\delta} \) are related as
\[ \hat{\delta} = \frac{Q_s}{h |\eta|} \hat{\phi} \quad (2.3.9) \]

The bunch length in length units is \[32\]
\[ \sigma_l = \frac{c}{h \omega_0} \hat{\phi} = \frac{c |\eta|}{\omega_s} \hat{\delta} = \frac{\sqrt{2\pi c}}{\omega_0} \sqrt{-\eta \beta^2 E} \frac{\sigma_E}{E} \quad (2.3.10) \]

where \( \sigma_E/E \) is the maximum energy spread.

### 2.4 Basic Beam Observation and Instruments

Measurements of beam properties are important in improving the performance of an accelerator. In this section we introduce some basic beam diagnosis tools \[6, 17\].

#### 2.4.1 Beam Position Monitor

BPM is one of the most common beam diagnosis tools used for observing transverse motion. Two types of sensors predominate today, buttons and strip lines, as in Fig. 2.5 and Fig. 2.6.

Buttons are mounted flush with or slightly recessed from the metal beam tube. Maximum coupling occurs when the button diameter is equal to the bunch length. Their relatively low beam impedance recommends them for rings having short, high
intensity bunches, such as electron synchrotron radiation light source. The buttons in Fig. 2.5 are rotated to avoid synchrotron radiation. The strip lines in Fig. 2.6 have greater sensitivity but higher beam impedance.

Beam position is usually determined by the “difference over sum” method. Signals
from the buttons in Fig. 2.5 would be processed [17] as

\[ x = \frac{(A + D) - (B + C)}{A + B + C + D} \frac{1}{S_x} \]  
\[ y = \frac{(A + B) - (C + D)}{A + B + C + D} \frac{1}{S_y} \]  

(2.4.1)  
(2.4.2)

\( S_{x,y} \) are geometry dependent factors.
Chapter 3

Orbit Response Matrix and Lattice Modeling

In a circular accelerator, the closed orbit is the one who goes back to itself after one turn. For a ring with dipole errors the closed orbit may distort, and not go through the exact center of every magnets. The distortions may be different along the ring, and depend on both the magnitude and location of the dipole errors. This mechanism is also used by accelerator feedback system. It uses many orbit correctors distributed along the ring to steer the beam in a stable orbit, and close to the center of magnets as much as possible. Understanding the Closed Orbit Distortion (COD) may help us to diagnose the beam and magnets, design the feedback system, and steer the beam better.

In this chapter, first I introduce the theory of COD. The observed COD at \( s \) is in fact the product of the orbit kick at \( s_0 \) and Green’s function from \( s_0 \) to \( s \). Then we will discuss an important data set called Orbit Response Matrix (ORM). The ORM is a large set of data describing how the COD changes due to an increase of orbit corrector kick, and it can be used to model the lattice of a storage ring.
[31, 30, 29, 14, 36]. After that, I will explain how YAOC uses the ORM data to model an accelerator lattice. The Scaled Levenberg-Marquardt algorithm is also introduced, but the detailed discussions are left in Appendix B.

3.1 Dipole Error and Closed Orbit

The closed orbit of a perfect optics goes through the center of all magnets, while in practice, either from magnet errors, installation errors or even the operation requirement\(^{a}\), the particle beam is not exactly at the center. In order to maintain a proper beam orbit, or design the orbit feedback system, it is essential to understand the COD.

Suppose there is a dipole field error which gives a small angle kick \(\theta\) at \(s_0\). As

\(^{a}\)During injection, it is important to “kick” the beam from transport line into the ring, while keep the stored particles from hitting any chamber wall. The septum placed in the vacuum camber makes the geometry there not as same as other location, the optimization beam position is usually slightly away from the original center and the septum.
shown in Fig. 3.1, the closed orbit deviates from the ideal orbit.

The closed orbit condition at the exit of the dipole field error is

\[
M \begin{pmatrix} y_0 \\ y'_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y'_0 - \theta \end{pmatrix}
\] (3.1.1)

where \( M \) is the one turn transfer map from Eq. (2.2.13) for an otherwise ideal accelerator. By solving this equation, we can get the resulting COD at \( s_0 \)

\[
y_0 = \frac{\beta_0}{2} \theta \cot \pi \nu \quad y'_0 = \frac{1}{2} \theta (1 - \alpha_0 \cot \pi \nu)
\] (3.1.2)

where \( \beta_0 \) and \( \alpha_0 \) are Courant-Snyder parameters at \( s_0 \).

The resulting closed orbit at other location \( s \) can be obtained from the propagation of betatron oscillations

\[
\begin{pmatrix} y_1 \\ y'_1 \end{pmatrix} = M(s_1|s_0) \begin{pmatrix} y_0 \\ y'_0 \end{pmatrix}
\]

where \( M(s_1|s_0) \) is the transfer matrix from Eq. (2.2.25). The closed orbit at \( s \) is

\[
y_{co}(s) = \sqrt{\frac{\beta(s) \beta(s_0)}{2 \sin \pi \nu}} \cos (\pi \nu - |\phi(s) - \phi(s_0)|) \theta(s_0) \equiv G(s, s_0) \theta
\] (3.1.3)

where \( G(s, s_0) \) is called Green’s function from \( s_0 \) to \( s \).

For dipole errors distributed around the accelerator, the COD can be obtained by a linear superposition of dipole kicks because of the linearity of Hill’s equation

\[
y_{co}(s) = \sqrt{\frac{\beta(s)}{2 \sin \pi \nu}} \sum_{j=1}^{n} \sqrt{\beta(s_j)} \cos (\pi \nu - |\phi(s) - \phi(s_j)|) \theta_j = \sum_{j=1}^{n} G(s, s_j) \theta(s_j)
\] (3.1.4)

The continuous version is [20]

\[
y_{co}(s) = \int_{s}^{s+C} G(s, t) \frac{\Delta B(t)}{B \rho} \, dt
\] (3.1.5)

with the form of Green’s function [20].

Green’s function depends on the betatron functions and the phase advance between the corrector at \( s_0 \) and the BPM at \( s \), while the lattice functions \( \beta(s_0) \) and \( \beta(s) \)
come from the gradient of the focusing and defocusing components and are affected by the quadrupole errors. The Green’s function and the COD are well known the simulation codes such as MAD [22] when the lattice is given. We can also reverse the process by measuring all the COD along the ring, and find out the gradient of each focusing and defocusing components. It is also possible to fit for BPM errors, since they are only a linear transformation from real data to observed ones. This reverse process is called lattice modeling, i.e. reconstruct the parameters of a ring lattice from a set of experiment data. The procedure and algorithm of the reconstruction will be introduced in the following sections.

3.2 Orbit Response Matrix

The orbit response to a small kick is the product of the orbit kick strength and the Green’s function between the kick source and the orbit observation point. In accelerator operation, the kick is usually provided by an orbit corrector, and the closed orbit is recorded by a BPM. The Green’s function between these two points are determined by the betatron amplitude functions and the phase advance. The Green’s function describes the dynamics of COD in this segment of the ring.

As shown in Fig. 3.2, an increase of orbit kick at \( s_0 \) will bring an orbit distortion at \( s \). The ratio of orbit distortion and kick is \( G(s_0|s) = \Delta x/\Delta \theta \). Because of the linearity of Hill’s equation, this quantity is in fact the first derivative of COD with respect to a specific orbit corrector, and is independent of the closed orbit before increasing the kick by \( \Delta \theta \). The closed orbit distortion at one BPM location is the sum of all these dipole error effects.

The ORM \( G \), as the name says, is a matrix whose elements are the orbit change due to an increasing of the orbit kick. It is also called Corrector-to-BPM response
3.2. ORBIT RESPONSE MATRIX

Figure 3.2: Horizontal COD and Green’s function. (Fermilab Booster lattice)

matrix. The elements of ORM is in fact the Green’s function we have talked before.

\[
G_{ij} = \frac{\partial y(s_i)}{\partial \theta(s_j)} = \frac{\sqrt{\beta(s_i)\beta(s_j)}}{2 \sin \pi \nu} \cos(\pi \nu - |\phi(s_i) - \phi(s_j)|) \quad (3.2.1)
\]

where \( i = 1, 2, \ldots, m \) is the BPM index and \( j = 1, 2, \ldots, n \) is the orbit corrector index. Each column represents the COD due to one particular orbit corrector.

In matrix form, the data set can be written as

\[
\begin{pmatrix}
\Delta y_{11} & \Delta y_{12} & \cdots & \Delta y_{1n} \\
\Delta y_{21} & \Delta y_{22} & \cdots & \Delta y_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\Delta y_{m1} & \Delta y_{m2} & \cdots & \Delta y_{mn}
\end{pmatrix}_{m \times n} =
\begin{pmatrix}
G_{11} & G_{12} & \cdots & G_{1n} \\
G_{21} & G_{22} & \cdots & G_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
G_{m1} & G_{m2} & \cdots & G_{mn}
\end{pmatrix}_{m \times n} 
\begin{pmatrix}
\Delta \theta_1 \\
\Delta \theta_2 \\
\vdots \\
\Delta \theta_n
\end{pmatrix}_{n \times 1}
\]

or

\[
Y_{m \times n} = G_{m \times n} \Theta_{n \times n} \quad (3.2.3)
\]

where the element \( \Delta y_{ij} \) is the COD change at \( i^{th} \) BPM due to a small increase of kick \( \Delta \theta_j \) at \( j^{th} \) orbit corrector. \( G \) is the ORM as defined before, where \( G_{ij} = \Delta y_{ij}/\Delta \theta_j \).
ORM is obtained from experiment data as following

$$G = Y \Theta^{-1}$$  \hspace{1cm} (3.2.4)

The elements of $G$ are precisely determined in the experiment by direct measuring COD change at BPMs resulting from exciting each orbit corrector individually. Each time we excite (increase or decrease) the strength of one orbit corrector, and record the COD change at every BPM as a column vector in $Y$. The small change of correctors form a diagonal matrix $\Theta$. This procedure can be automated by the control system or a middle layer package such as AT [34].

One can also calculate the expected $G$ as long as given the design gradients of all quads with Eq. (3.2.1), or by using a simulation program such as MAD [22] or Elegant [1]. The later method is used in the benchmarks of YAO C.

Since the kick and observation can be both directions, the combinations of them have four sets of data,

$$\begin{cases} 
G^{XX} = X \Theta^{-1}_x \\
G^{ZX} = Z \Theta^{-1}_x \\
G^{XZ} = X \Theta^{-1}_z \\
G^{ZZ} = Z \Theta^{-1}_z 
\end{cases} \hspace{1cm} (3.2.5)$$

The theory of dipole error and closed orbit deviation can be used for orbit feedback and control system. Since a single kick will produce sine wave like COD, a set of distributed orbit correctors are used to balance each other, and achieve the desired orbit. The orbit feedback system can be implemented following Eq. (3.2.3). The goal is to find the vector $\Theta$, which makes a designed orbit at every BPM. The closed orbit at the $i^{th}$ BPM now is the sum of the $i^{th}$ row of $Y$ in Eq. (3.2.3). Usually the Singular Value Decomposition (SVD) is used to inverse $G$ and to solve for $\Theta$, where the ORM

\(^b{Only the closed orbit changes are recorded.}\)
3.3. **LATTICE MODELING WITH ORBIT RESPONSE MATRIX**

$G$ is derived from measurements. Another orbit correction method employs harmonic correction method.

### 3.3 Lattice Modeling with Orbit Response Matrix

Analysis of the ORM has been used to model the lattice and calibrate the optics of storage rings [31, 30, 29, 36]. The lattice of an operating machine is reconstructed from ORM data, and the magnet errors are corrected after comparing with optimized values.

Once the lattice is known, we can calculate the ORM. Conversely, given the measured response matrix from the real accelerator, one can model the lattice, such as getting the real gradients in each of the quads, BPM gain factors, orbit corrector scaling factors and their tilt angles. A computer code called LOCO (Linear Optics from Closed Orbits) has been developed to analyze the measured response matrix and determine the linear optics [31]. Other similar code was also developed and applied on the PSR (Proton Storage Ring) and the Fermilab Booster [14]. All these codes may encounter convergence difficulties problems in minimization iterations. Improvement on the numerical algorithm and more model benchmarks on various lattice are needed.

A new code based on the Scaled Levenberg-Marquardt algorithm in Ref. [26] is developed to overcome the convergence difficulties. Better than Newton method used in LOCO [29], this new algorithm is more robust and efficient. Systematic benchmarks have been carried out for the Fermilab Booster, and YAOOC have showed remarkable convergence in simulations. The benchmarks and testings on other storage ring, such as the design lattice of Taiwan Photon Source (TPS), Brookhaven National Lab VUV ring are also quite successful.
3.3.1 Motivation for a New Code

The magnet and diagnosis device parameters of a running storage ring can be expressed in vector form by

\[ x_{\text{actual}} = x_{\text{designed}} + x_{\text{err}} \quad \text{or} \quad x_{\text{actual}} = x_{\text{designed}} \ast (1.0 + b_{\text{err}}) \]

where the \( x_{\text{err}} \) is the absolute error vector, and \( b_{\text{err}} \) is relative error vector to designed parameters. For example, the focusing strength for main combined function dipoles of the Fermilab Booster is \( K_1 = 0.054 \) m\(^{-2}\). If two dipoles have an actual parameters \( x_{\text{actual}} = (0.059, 0.052)^T \), then the error vector is \( x_{\text{err}} = (0.005, -0.002)^T \), and the relative error vector \( b_{\text{err}} = (9.2\%, -3.7\%)^T \). For tilt angle of magnets, since the design value is 0 rad, instead of relative error vector they are only expressed in the form of absolute error vector. The goal of ORM analysis is to find the \( x_{\text{err}} \) or \( b_{\text{err}} \), and correct them. It is not a trivial problem especially when there are hundreds of parameters in \( x_{\text{err}} \) need to fit, i.e. it is a multidimensional optimization problem.

Some of the most important parameters we want to know about a lattice is the focusing and defocusing properties, i.e. the quadrupole gradient. The BPM errors can also be important. They may have horizontal or vertical gain factor errors, or may be misaligned.

These lattice parameters are not totally independent, and some of them may couple with each other. Decrease of one the errors may be compensated partly/fully by increasing/decreasing another error. It is sometimes hard to distinguish one from another. For example, Xiaobiao Huang has done successful modeling with ORM on the PSR lattice, but the same code sometimes failed on the Fermilab Booster due to the coupling of nearby quadrupole components. As shown in Fig. 3.3, the error lattice used for benchmark has a single error on the 9th upstream focusing dipole, about 4% of the nominal \( K_1 \). But instead of this single error lattice we used for benchmark, the fitted result shows an equivalent(or very close) lattice, which is not the one we
3.3. LATTICE MODELING WITH ORBIT RESPONSE MATRIX

Figure 3.3: ORM Simulation by X. Huang[14]

should get. After comparing the fitted result with the single error benchmark lattice, we can find that the single error can be partly compensated by the nearby magnet errors. This indicated by nonzero errors close to the 9th dipole in Fig. 3.3. When looking at the ORM data and the merit function $\chi^2$ from these two lattices, they are very close [14].

The coupling is indeed caused by local minimums near real solution(global minimum) in parameter space or by the barriers which prevent further iterations for minimizing the merit function. When this situation is met in the iterative searches, a good algorithm can adaptively overcome the barrier, or adjust its strategy to make
sure in certain range, there is no point better than this local minimum. As we will see in the benchmark of YAOC, this coupling can be resolved, and even some more complicated lattices with random errors can be fitted. An example of benchmarks is shown in Fig. 3.4 where the errors used in benchmark lattice are precisely reconstructed. More discussions on benchmarks will be discussed in Chap. 4

3.3.2 Least-Square Modeling with ORM

Given a set of observations, such as ORM, one often wants to condense and summarize the data by fitting it to a model that depends on some adjustable parameters.
Sometime the model is simply a convenient class of functions, such as polynomial or Gaussian, and the fit supplies the appropriate coefficients. Other times the model’s parameters come from some underlying theory that the data are supposed to satisfy; examples are the storage ring lattice modeling in this dissertation. A proper set of quadrupole strength, BPM gain factors and their tilt angles should give the same ORM data as from experiments.

Although the application models are different, the basic approach is same: A “figure-of-merit” function is set up to measure the agreement of the ORM data from the experiment with the data from our model with a particular chosen parameter vector $x$. This function is conventionally arranged so that small values represent close agreement. The parameters of the model are then adjusted to achieve a minimum in the merit function, yielding best-fit parameters [28].

In our lattice modeling, the merit function is defined by the agreement of the experiment ORM data and some other lattice properties with simulations.

$$\chi^2(x) = \sum_{ij} \left( \frac{G_{ij}^{\{X,Z\},\text{meas}} - G_{ij}^{\{X,Z\},\text{model}}(x)}{\sigma_{G_{ij}}} \right)^2 + \sum_{x,z} \left( \frac{\nu_{x,z}^{\text{meas}} - \nu_{x,z}^{\text{model}}(x)}{\sigma_{\nu_{x,z}}} \right)^2 + \sum_k \left( \frac{\phi_k^{\{X,Z\},\text{meas}} - \phi_k^{\{X,Z\},\text{model}}(x)}{\sigma_{\phi_k}} \right)^2$$

$$\equiv \sum_{i=1}^m \left( \frac{f_i^{\text{meas}} - f_i^{\text{model}(x)}}{\sigma_i} \right)^2$$

(3.3.1)

where all $\sigma(\cdot)$ are the uncertainty of measurement, $G_{ij}^{\{X,Z\}}$ are the ORM elements from horizontal and vertical closed orbit shift due to a orbit kick, $\nu_{x,z}$ are tunes of betatron motion, and $\phi_k$ is phase advance between BPMs. For a lattice with $M$ BPMs and $N$ orbit correctors, the ORM can have as much as $2M \times 2N$ elements if kick and observation can work at both horizontal and vertical planes.

As long as the data are measurable or obtainable from other method, we can include them in $\chi^2(x)$. For example, the Independent Component Analysis (ICA)
method can give the phase information at every BPM, with \( \sigma_\phi = 0.005 \), and \( \sigma_{\nu_x, z} = 0.001 \) \cite{14, 15}. Although they have less terms in \( \chi^2 \) compared with ORM terms, they do help the convergence of fitting when close to the minimum of \( \chi^2 \).

The variance associated with the estimate of lattice parameters are

\[
\sigma^2(x_j) = \sum_{i=1}^{m} \sigma_i^2 \left( \frac{\partial x_j}{\partial f_i} \right)^2
\]

(3.3.2)

where \( x_j \) is the parameter of lattice model, \( f_i \) is each term in \( \chi^2 \), and \( \sigma_i \) is the corresponding uncertainty of measurement in \( f_i \).

### 3.3.3 Algorithm of Fitting

Once we have a correct model for accelerators, the parameters of the model are adjusted to achieve a minimum in the merit function, yielding the best-fit-parameters. The adjustment process is thus a problem in minimization in multi-dimensions.

An minimum point can be either global or local in a finite neighborhood. Finding a global extreme is, in general a very difficult problem. Two standard heuristics are widely used: (i) find local extreme starting from widely varying starting values of the independent variables, then pick the most extreme of these; or (ii) perturb a local extreme by taking a finite amplitude step away from it, then see if your routine returns you a better point, or always to the same one.

The choice of method can be based on the following considerations \cite{28}:

- **Function and its derivative.** We may choose between methods that need only evaluations of the function to be minimized and methods that also require evaluations of the derivative of that function. In our multidimensional case as Eq. (3.3.1), the derivative of \( \chi^2 \) is connected to the gradient of \( f \). Algorithms using the derivative are somewhat more powerful than those using only the function, but needs additional calculations of derivatives.
• **Scaling complexity.** Storage scaling with problem dimension (dimension of parameters \( \mathbf{x} \)). Is it \( n^2 \) or \( n \)? For a ORM analysis, limited by the number of BPMs and orbit correctors, this is not a major concern.

• **Convergence and its speed.** As we will see, although some algorithm guarantees the convergence, it takes too long to be practical in practice; e.g. when searching for the minimum of Rosenbrock function \( f(x, y) = (1-x)^2 + 100(y - x^2)^2 \) with steepest-descent method, the long valley shape near the minimum makes the iteration taking long time to converge, although it’s guaranteed.

The local minimum points may be surrounded by some barriers, which could prevent some approach method from passing over them to obtain a better point in the parameter space. In this situation, a proper search method is needed to adaptively extend a search in wider region. The barrier can be overcome as long as the search region is large enough.

In scientific computing, a great number of algorithms for minimization of a function use iteration to approach the minimum\(^6\). If we start at a point \( \mathbf{x}_0 \), in N-dimensional space, and proceed from there in some vector direction \( \mathbf{p} \), then any function of N variables \( f(\mathbf{p}) \) can be minimized along the line \( \mathbf{p} \) by a one-dimensional methods. One can come up various multidimensional minimization methods that consist of sequences of such linear minimizations. Different methods will differ by how, at each stage, they choose the next direction \( \mathbf{p} \) to try.

Gauss-Newton method and Steepest-descent (or named gradient) method are two of the most common methods of searching the extrema of a function. They are the basis of more complicated algorithms. The detailed discussions of these algorithms are in Appendix B.2 and B.1, and a robust and efficient algorithm called Levenberg-

\(^6\)Simulated annealing method is a new algorithm very different from the traditional Taylor-series-expansion based numerical methods.
Marquardt method is used in our analysis.

The Levenberg-Marquardt (L-M) algorithm is one of the most widely used optimization algorithm. It outperforms simple gradient descent (steepest descent) and other conjugate gradient methods in a wide variety of problems. It is first shown to be a blend of vanilla gradient descent and Gauss-Newton iteration [23]. Another perspective on the algorithm is provided by considering it as a trust-region searching method.

The merit function \( \chi^2 \) is the norm of a general vector \( f : R^n \rightarrow R^m \)

\[
\chi^2 \equiv \| f \| \quad (3.3.3)
\]

Here \( f_k \) is each term in Eq. (3.3.1), e.g. \( f_k = (G_{ij}^{\text{meas}} - G_{ij}^{\text{model}})/\sigma_{ij}, \ k = 1, 2, ..., m, \)

where \( m \) is the total number of terms in \( \chi^2 \). The derivative of \( \chi^2 \) can be written using the Jacobian matrix \( J \) of \( f \) w.r.t \( x \)

\[
J_{ij} = \frac{\partial f_i}{\partial x_j} \quad 1 \leq i \leq m, \ 1 \leq j \leq n. \quad (3.3.4)
\]

Considering the linear approximation,

\[
f(x_0 + p) = f(x_0) + Jp \quad (3.3.5)
\]

we can get \( \nabla \chi^2 = 2J^T(Jp + f(x_0)) \). After solving for the minimum by setting \( \nabla \chi^2 = 0 \), we obtain

\[
(J^TJ)^{-1}x_{\min} = -J^Tf(x_0) \quad (3.3.6)
\]

or \( x_{\min} = -(J^TJ)^{-1}J^Tf(x_0) \). In this way, an iteration step from \( x_0 \) to \( x_{\min} \) is done, and similarly a sequence \( \{x_i\} \) would be constructed to minimize \( \chi^2 \). This is one implementation of Gauss-Newton method without using the second derivatives of \( \chi^2 \) or \( f \) to \( x \). More discussion involving second order derivatives are in Appendix. B.2.

The key hallmark of L-M algorithm is to replace Eq. (3.3.6) by a “damped version”

\[
(J^TJ + \lambda I)x_{\min} = -J^Tf(x_0) \quad (3.3.7)
\]
or by a “scaled version”

\[(J^TJ + \lambda D^TD)x_{\text{min}} = -J^Tf(x_0)\]  

(3.3.8)

where \(\lambda\) is a non-negative damping factor adjusted at each iteration, and \(D\) is a scaling matrix chosen as a diagonal matrix where \(D_{\text{ii}} = \partial\chi^2/\partial x_i\). The latter one is called the Scaled L-M algorithm and is used in YAOC. \(\lambda\) also indicates the interpolation between Gauss-Newton and the gradient descent algorithms. During the iteration, if reduction of \(\chi^2\) is rapid a smaller value \(\lambda\) can be used bringing the algorithm closer to the Gauss-Newton algorithm. If an iteration gives insufficient reduction in the residual \(\lambda\) can be increased giving a step closer to the gradient descent direction, since \((J^TJ + \lambda D^TD)\) is nearly diagonal. Damping also handles situations where the Jacobian is rank deficient with singular \(J^TJ\).

The more detailed discussion of L-M algorithm is in Appendix B. The general structure is described in Algorithm 1, and refined in Algorithm 3 on page 121. In this algorithm, it iteratively searches for a sequence of \(\{x_i\}\) which brings \(\chi^2\) from an initial value to the minimum, here \(\lambda\) is the “damping factor” or “interpolation factor” controlling the solution of increment vector \(p\).

A more carefully designed algorithm would use a generalized trust region searching strategy to keep each step under control. It has more consideration than Algorithm 1, and may improve the convergence. Before going to Appendix, there are some highlights of scaled L-M algorithm listed in the following:

- **Choice of damping parameter** in Line 9 and Line 12 of Algorithm 1.

Various heuristic arguments have been put forward for the best choice for the damping parameter \(\lambda\). Theoretical arguments exist showing why some of these choices guaranteed local convergence of the algorithm; however these choices can make the global convergence of the algorithm suffer from the undesirable
Algorithm 1 Levenberg-Marquardt Method

1: // The stop condition of loop can depend on $\chi^2$ or $p$
2: Compute $\chi^2(x)$
3: Initialize $\lambda$. // or a modest value.
4: repeat
5: Solve for the increment $p$, i.e. $x_{min} = x + p$. in Eq. (3.3.7)
6: evaluate $\chi^2(x + p)$
7: if $\chi^2(x + p) \geq \chi^2(x)$ then
8: // $x + p$ is not a better point
9: Update (increase) $\lambda$, e.g. $\lambda = 10\lambda$.
10: else // $\chi^2$ is decreasing.
11: // A successful move, accept the new point
12: Update (decrease) $\lambda$, e.g. $\lambda = \lambda/10$, and $x = x + p$
13: end if
14: until converged (depends $\chi^2$ or $p$) or timeout

properties of steepest-descent, in particular very slow convergence close to the optimum.

The absolute values of any choice depends on how well-scaled the initial problem is. Marquardt recommended starting with a value $\lambda$ and a factor $\nu > 1$. Initially setting $\lambda = \lambda_0$ and computing the residual sum of squares $\chi^2$ after one step from the starting point with the damping factor of $\lambda = \lambda_0$ and secondly with $\lambda/\nu$. If both of these are worse than the initial point then the damping is increased by successive multiplication by $\nu$ until a better point is found with a new damping factor of $\lambda \nu^k$ for some $k$.

If use of the damping factor $\lambda/\nu$ results in a reduction in squared residual then
this is taken as the new value of $\lambda$ (and the new optimum location is taken as that obtained with this damping factor) and the process continues; if using $\lambda/\nu$ resulted in a worse residual, but using $\lambda$ resulted in a better residual then $\lambda$ is left unchanged and the new optimum is taken as the value obtained with $\lambda$ as damping factor.

- **Damping term.** A more general damping term would be $\lambda D^T D$ instead of $\lambda I$. This prevents a sudden change of one parameter $x_i$ by scaling it with $\partial \chi^2 / \partial x_i$, i.e. the diagonal scaling matrix $D^d$.

- **Trust region.** In order to be accepted as a proposed new increment $p$ must satisfy the condition $\|Dp\| < \Delta$, where $D$ is a diagonal scaling matrix and $\Delta$ is the size of the trust region. The components of $D$ are computed internally, using the column norms of the Jacobian to estimate the sensitivity of the residual to each component of $p$. This improves the behavior of the algorithm for badly scaled functions. The sudden change of one of the parameters may make the iteration unstable [26].

- **Judge the prediction.** The proposed step is now tested by evaluating the function at the resulting point, $x + p$. If the step reduces the $\chi^2$ sufficiently, and follows the predicted behavior of the function within the trust region, then it is accepted and the size of the trust region is increased. If the proposed step fails to improve the solution, or differs significantly from the expected behavior within the trust region, then the size of the trust region is decreased and another trial step is computed.

In summary, The Scaled Levenberg-Marquardt algorithm is not only an inter-

---


dThere are other choices of scaling matrix, e.g. [11, Fletcher 1971], and not necessary to be diagonal.
polation of Gauss-Newton and gradient descent algorithms as a vanilla version of L-M algorithm in Ref. [28, 23, 14]. It uses more parameters to control the iteration, and make the linear approximation based on Taylor series expansion valid in a certain trusted region. This region is adjusted according to how close the linear approximation to the realized value. A diagonal scaling matrix is also adapted to regulate the increment vector. The implementation used in YAOC is from GSL [13] and MINPACK [16].
Chapter 4

Benchmark and Application of YAOC

The benchmarks and applications of YAOC have been done on the Fermilab Booster, the Taiwan Photon Source (TPS) design lattice and the VUV ring of National Synchrotron Light Source (NSLS) at Brookhaven National Lab (BNL) [36]. Thousands of random error benchmarks show that YAOC is reliable code for ORM analysis, and can reconstruct the quadrupole and BPM errors in high accuracy.

4.1 Introduction

The real experiment data usually have noise and other uncertainties, which may make the results look different from simulations. Especially for a modeling problems, large enough noise may kill the whole modeling. For this reason, benchmarking the code with simulation data becomes very essential, and only with the success on the man-made data can assure us that the result of real data analysis is reliable. Since both the magnets and diagnostic instruments can affect the data we are getting, without
benchmark we can not really tell if we have corrected the lattice, or we have only changed the diagnostic system which make our measurement looks like an corrected one. For example, if the lattice has some gradient errors on quadrupoles, while our model only fits for the BPM errors. In this case, we would still follow the iteration of minimizing $\chi^2$ by changing the BPM gain factors and tilt angles. We may reduce $\chi^2$ to a very low level if not exactly zero. The data would show that two set of ORM data are very similar, since the $\chi^2$ is very small. But we know the lattice we got is not the one with gradient errors, and we will never find these errors by fitting for BPM errors. It is even worse that since we are changing the BPM to get a better set of data instead of lattice, we are not really improving the field guiding the stored beam and improve its qualities. The other measurement based on BPM will be misleading after this BPM correction. One example would be measuring the dispersion by observing the closed orbit distortion due to RF frequency change.

$$D = \frac{dx_{co}}{d(\Delta p/p_0)} = -\eta f_0 \frac{dx_{co}}{df_0}$$  \hspace{1cm} (4.1.1)

Of course, the $dx_{co}$ comes from a mis-calibrated BPM, but after the “ORM analysis” we thought it is right. By the spurious effects of the BPMs, we have a “accurate” measured dispersion. But this dispersion can’t be used for beam dynamics analysis, since it’s not the dispersion seen by a beam, but seen by us through the mis-calibrated BPM.

In multi-dimension optimization, it is very hard to find a global minimum of $\chi^2$, especially when there are uncertainties in our measurement data. But since most of the accelerator lattices belong to only a few known categories, e.g. FODO cell, DBA (Double Bend Achromat) cell etc and TBA (Triple Bend Achromat), Once extensive benchmark were done on one of these categories, we would get better understanding of the other accelerators in the same category. We then are more confident on the following questions: how unique is the final output, is it really a cosmetic effect from
4.2. BENCHMARKS ON FERMILAB BOOSTER

the diagnostic instruments like BPM, or it is from the true lattice magnets, what is the main concern of lattice fitting. After the benchmarks, the analysis on the real experiment data would be more reliable.

The scheme of benchmark is illustrated in Fig. 4.1. First a random lattice, we call it benchmark lattice, is generated by adding small random errors to the lattice elements, such as quadrupole strength, gain factor of BPMs, scaling factor of correctors and their tilt angles. The ORM data of this benchmark lattice are then obtained from the theories in Chap. 3. This simulated ORM is used as our experiment data from a “uncalibrated/unoptimized lattice” whose parameters we want to understand. The ORM data are input into YAOC to fit for those small random errors. These output errors from YAOC are then compared with the random errors we have generated for the benchmark lattice to check if they are successfully predicted or not.

![Scheme of benchmark](image)
4.2 Benchmarks on Fermilab Booster

Fermilab Booster is a common proton source of the major high energy experiments at Fermi National Accelerator Laboratory. It has a circumference of 474 meters. The beam is injected into the Booster from the 400 MeV transport line which carries the beam output from the linac accelerator. The Booster can accelerate 84 bunches of proton beams from 400 MeV to 8 GeV in less than 67 ms for the Main Injector accelerator. Booster also provides beam for the MiniBooNE experiment and the NuMI facility and MINOS experiment.

The Fermilab Booster has 24 identical cells each composed of 4 main magnets. Those 4 magnets in each cell are indicated by FU (focusing, upstream), DU (defocusing, upstream), DD (defocusing, downstream) and FD (focusing, downstream). Between two nearby defocusing magnets there is a long straight section, and a short straight section between two focusing magnets. These 96 combined function magnets serve as both dipoles and quadrupoles. The focusing/defocusing strength $K_1$ for these combined function magnets is around $+0.0577m^{-2}$/$-0.0542 m^{-2}$ respectively. Each cell has 2 BPMs placed in short and long straight sections, and 48 BPMs in total. The BPMs are placed at locations with maximum $\beta$ function to increase the sensitivity. Some of the parameters are given in Table 4.1. Fig. 4.2 shows the twiss parameters in one cell.
4.2. BENCHMARKS ON FERMILAB BOOSTER

Figure 4.3: Lattice functions of the Fermilab Booster.

4.2.1 Gradient Error

The ORM elements are proportional to the square root of betatron amplitude functions $\beta(s)$ at the BPM and $\beta(s_0)$ at the orbit corrector, while the $\beta$-function depends on the distribution of quadrupole strengths. This relationship makes it possible to analyze the gradient errors by looking at the ORM data.
### Table 4.1: Fermilab Booster Parameters (December 5, 2002)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circumference (m)</td>
<td>474.2</td>
</tr>
<tr>
<td>Average machine radius (m)</td>
<td>75.47</td>
</tr>
<tr>
<td>Injection kinetic energy (MeV)</td>
<td>400</td>
</tr>
<tr>
<td>Extraction kinetic energy (GeV)</td>
<td>8</td>
</tr>
<tr>
<td>Harmonic number</td>
<td>84</td>
</tr>
<tr>
<td>Protons per bunch</td>
<td>$6 \times 10^{10}$</td>
</tr>
<tr>
<td>Lattice</td>
<td>FOFODODO</td>
</tr>
<tr>
<td>Super-periodicity</td>
<td>24</td>
</tr>
<tr>
<td>Cell length (m)</td>
<td>19.758</td>
</tr>
<tr>
<td>Length of combined function magnet (m)</td>
<td>2.889612</td>
</tr>
<tr>
<td>Magnet per cell</td>
<td>4</td>
</tr>
<tr>
<td>Magnet total</td>
<td>96</td>
</tr>
<tr>
<td>Number of straight sections</td>
<td>24 Long, 24 Short, 48 Mini</td>
</tr>
<tr>
<td>Length of each straight section (m)</td>
<td>6(Long), 1.2(Short), .5 (Mini)</td>
</tr>
<tr>
<td>Max/Min $\beta_x$ (m)</td>
<td>33.67 (Short)/6.12 (Long)</td>
</tr>
<tr>
<td>Max/Min $\beta_y$ (m)</td>
<td>20.46 (Long)/5.27 (Short)</td>
</tr>
<tr>
<td>Max/Min $D_x$ (m)</td>
<td>3.19 (Long)/1.84 (Short)</td>
</tr>
<tr>
<td>Phase advance per cell</td>
<td>$\phi_x = 100.5$, $\phi_y = 102$</td>
</tr>
<tr>
<td>Horizontal, vertical tune</td>
<td>$\nu_x = 6.7$, $\nu_y = 6.8$</td>
</tr>
<tr>
<td>Natural chromaticity $\xi_x$, $\xi_y$</td>
<td>-9.2, -7.0</td>
</tr>
<tr>
<td>Transition $\gamma_t$</td>
<td>5.45</td>
</tr>
</tbody>
</table>

Including the gradient error, Hill’s equation for the perturbed betatron motion about a closed orbit is

$$\frac{d^2y}{ds^2} + [K(s) + k(s)]y = 0 \quad (4.2.1)$$

where $K(s)$ is the focusing function of the ideal machine discussed before, and $k(s)$ is a small perturbation, which can come from the quadrupole or misaligned sextupole.
Let $M_0$ be the one turn transfer matrix of the ideal machine, and $\alpha(s), \beta(s), \gamma(s)$ be betatron amplitude functions of the unperturbed machine. The transfer matrix of this infinitesimal localized perturbing quadrupole error is

$$ m(s_1) = \begin{pmatrix} 1 & 0 \\ -k(s_1)ds_1 & 1 \end{pmatrix} \quad (4.2.2) $$

The one turn transfer matrix (OTM) $M(s_1) = M_0(s_1)m(s_1)$ becomes

$$ M(s_1) = \begin{pmatrix} \cos \Phi_0 + \alpha_1 \sin \Phi_0 - \beta_1 k(s_1)ds_1 \sin \Phi_0 & \beta_1 \sin \Phi_0 \\ -\gamma_1 \sin \Phi_0 - [\cos \Phi_0 + \alpha_1 \sin \Phi_0] k(s_1)ds_1 \cos \Phi_0 & \cos \Phi_0 - \alpha_1 \sin \Phi_0 \end{pmatrix} \quad (4.2.3) $$

where $\alpha_1 = \alpha(s_1), \beta_1 = \beta(s_1), \gamma_1 = \gamma(s_1)$. $\Phi_0 = 2\pi \nu_0$ is the unperturbed betatron phase advance in one complete revolution. The phase advance of the perturbed machine can be obtained from the trace of $M(s_1)$, i.e.

$$ \cos \Phi - \cos \Phi_0 = -\frac{1}{2} \beta(s_1)k(s_1)ds_1 \sin \Phi_0, \quad \text{or} \quad \Delta \Phi \approx \frac{1}{2} \beta(s_1)k(s_1)ds_1 \quad (4.2.4) $$

Thus the betatron tune shift is

$$ \Delta \nu = \frac{1}{4\pi} \beta(s_1)k(s_1)ds_1 \quad (4.2.5) $$

The existence of linear perturbation terms causes not only the tune shift but also betatron functions to vary around the ring or along a beam line. This variation can be derived by observing the perturbation of the one turn transfer matrix at a particular location. The one turn transfer matrix at $s_2$ is

$$ M(s_2) = M(s_2 + C|s_1)m(s_1)M(s_1|s_2) \quad (4.2.6) $$

Using Eq. (2.2.25), the change of off-diagonal matrix element is

$$ \Delta[M(s_2)_{12}] = -k_1ds_1\beta_1\beta_2 \sin[\nu_0(\phi_1 - \phi_2)] \sin[\nu_0(2\pi + \phi_2 - \phi_1)] \quad (4.2.7) $$
where $\beta_1 = \beta(s_1)$, $\beta_2 = \beta(s_2)$, $\phi_1 = \psi(s_1)/\nu_0$ and $\phi_2 = \psi(s_2)/\nu_0$. Since $M = \beta \sin \Phi$, we have

$$
(\Delta \beta_2) \sin \Phi_0 = \Delta [M(s_2)]_{12} - \beta_2 \cos \Phi_0 \Delta \Phi
$$

$$
= -\frac{1}{2} k_1 d s_1 \beta_1 \beta_2 \cos [2\nu_0 (\pi - \phi_1 + \phi_2)]
$$

or

$$
\frac{\Delta \beta_2}{\beta_2} = -\frac{1}{2 \sin \Phi_0} k_1 d s_1 \beta_1 \cos [2\nu_0 (\pi - \phi_1 + \phi_2)]
$$

Since the ORM element $G_{ij} \propto \sqrt{\beta_i \beta_j}$, and a gradient error at $s_k$ would change $\beta(s)$ at both BPM and orbit corrector, a lattice model with gradient errors can be fitted by using ORM data.

The quadrupole gradient errors used in our benchmark are listed in Table 4.2. There are 96 parameters in total, and in order for easy description of the error size, they are chosen to follow a uniform or Gaussian distribution.

<table>
<thead>
<tr>
<th>Quad.</th>
<th>Error</th>
<th>Design</th>
<th>Error</th>
<th>Num. of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1(F)$</td>
<td>$+0.0577$ m$^{-2}$</td>
<td>$\pm 5% \sim \pm 20%$</td>
<td>$2 \times 24$</td>
<td></td>
</tr>
<tr>
<td>$K_1(D)$</td>
<td>$-0.0542$ m$^{-2}$</td>
<td>$\pm 5% \sim \pm 20%$</td>
<td>$2 \times 24$</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>96</td>
</tr>
</tbody>
</table>

The full ORM has 9216 elements and betatron tunes are also used in merit function $\chi^2$. One example of fitting the gradient errors is shown in Fig. 4.4. In the benchmarks, no BPM noise was considered. The fitted lattice agrees with what was used to generate the pseudo-experiment ORM, i.e. the benchmark lattice. The difference between them is less than $10^{-7}$ which is close to the precisions we used in input/output.
Consideration of propagation of errors shows that the variance $\sigma_{x_i}$ in the value of any function will be

$$\sigma_{x_i}^2 = \sum_{j=1}^{m} \sigma_j^2 \left( \frac{\partial x_i}{\partial f_j} \right)^2$$  \hspace{1cm} (4.2.10)

where $m$ is the total number of experiment data point or the number of ORM terms in $\chi^2$. $x$ is the parameters in our model, like the gradient error here and $\sigma_j$ is uncertainty of $f_j$. The variance(confidence of fitting) of each gradient error is also shown in Fig. 4.4. Assuming same $\sigma_j$, for ORM element the $\partial f/\partial y_i$ depends on the $\beta$ at BPM and orbit corrector, and also the $\beta$ at gradient error, and the phase advance between this error and BPM, orbit corrector.

Figure 4.4: One benchmark example of fitting gradient error.
4.2.2 Gradient and Tilt

Besides the gradient error, the quadrupole may have nonzero tilt angle. The effect of the tilt angle of an element is that the reference system is rotated by $\theta$ at element entrance, and $-\theta$ at element exit. The transfer matrix $M$ becomes $\bar{M} = RMR^{-1}$ where $R$ is a linear transfer matrix (rotation matrix about $s$-axis)

$$
R = \begin{pmatrix}
\cos \theta & 0 & \sin \theta & 0 & 0 \\
0 & \cos \theta & 0 & \sin \theta & 0 \\
-\sin \theta & 0 & \cos \theta & 0 & 0 \\
0 & -\sin \theta & 0 & \cos \theta & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

The tilt of a quadrupole would produce a coupling between horizontal and vertical directions. For this kind of lattice, the ORM $G_{XY}$ and $G_{YX}$ are no longer zero matrices.

The benchmark lattice with both gradient error and rotation error is list in Table 4.3. There are 192 parameters in total, each quadrupole has strength and tilt angle errors.

<table>
<thead>
<tr>
<th>Quad.</th>
<th>Error</th>
<th>Design</th>
<th>Error</th>
<th>Num. of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1(F)$</td>
<td>$+0.0577 \ m^{-2}$</td>
<td>$\pm 5% \sim \pm 20%$</td>
<td>$2 \times 24$</td>
<td></td>
</tr>
<tr>
<td>$K_1(D)$</td>
<td>$-0.0542 \ m^{-2}$</td>
<td>$\pm 5% \sim \pm 20%$</td>
<td>$2 \times 24$</td>
<td></td>
</tr>
<tr>
<td>$\theta_qf$</td>
<td>0.0 rad</td>
<td>0.02 rad</td>
<td>$2 \times 24$</td>
<td></td>
</tr>
<tr>
<td>$\theta_qd$</td>
<td>0.0 rad</td>
<td>0.02 rad</td>
<td>$2 \times 24$</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>192</td>
<td></td>
</tr>
</tbody>
</table>
As we have seen in Fig. 3.3, the coupling between two nearby quadrupoles may create local minimum in fitting. Their strength can compensate with each other while still keep $\chi^2$ at a reasonable low level. For example, the QFU and QDU are so close that, increasing the strength of one magnet can be balanced by changing the strength of another magnet, the total effect on $\beta(s_{BPM})$ is

$$\frac{\Delta \beta_{BPM}}{\beta_{BPM}} = -\frac{1}{2} k_1 d s_1 \beta_1 \cos[2\nu_0 (\pi + \phi_{BPM} - \phi_1)]$$

$$- \frac{1}{2} k_2 d s_2 \beta_2 \cos[2\nu_0 (\pi + \phi_{BPM} - \phi_2)]$$

(4.2.12)

where the right hand side are two terms from gradient error of two nearby magnets, as indicated by Eq. (4.2.9). The subscript 1 and 2 are indices of two magnets, and $k$, $\beta$, $\phi$ are same as Eq. (4.2.9). Since the magnets are very close, the phase and beta function have $\phi_1 \approx \phi_2$, and $\beta_1 \approx \beta_2$. In parameter space, we can find some $(k_1, k_2)$ pair such that, they are different from $(0, 0)$, but they can keep $\Delta \beta_{BPM}$ at the BPMs very small, if not exactly zero.

The compensation of nearby magnets may produce a lattice having similar ORM as the benchmark lattice. This makes the fitting more difficult. Every one of these similar lattice is a pair of local minimum in parameter space.

On the other hand, the compensation may not be complete when considering that the phase advance $\phi$ are not exactly same. The derivative of ORM element $G_{ij}$ w.r.t $\beta$ may also not be the same, therefore it is still possible to distinguish the compensation effects. A more robust algorithm may solve this problem by searching adaptively in a trust region, according to $\chi^2$ and its derivative to gradient error. In our benchmarks, YAOQ can successfully fit the single error model, such as the one in Fig. 3.3.

A more complicated benchmark lattice is shown in Fig. 4.5. $x$-axis is the cell index or index of magnets in his family(four families in total, QFU, QDU, QDD and QFD). It shows that YAOQ can successfully breaks the coupling between the nearby magnets shown in Ref. [14]. This benchmark lattice could not be fitted in Ref. [14].
but was solved by YAOC. Results of other parameters are not shown here, but they have same level of agreement.

**Gradient, Quadrupole Tilt and BPM error Modeling**

The validity of ORM analysis on quadrupole errors has been proven by a serial benchmarks. But all the data we used before assumes that the BPMs have no error, nor random noise. But in practice, the diagnostic instruments such as BPM can also have errors, and of course these errors can make the ORM different from the design lattice. In YAOC, we can include the BPM errors in our model and fit them, therefore the calibration of BPMs can be done based on our analysis.
The error of BPMs are mainly gain factors and rolls, i.e. the horizontal and vertical gain factor and roll angle. The measured closed orbit distortion is a transform of actual ones due to the BPM errors:

\[
\begin{pmatrix}
\Delta x \\ \Delta z
\end{pmatrix}_{\text{meas}} = \begin{pmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{pmatrix}
\begin{pmatrix}
b_x & 0 \\
0 & b_y
\end{pmatrix}
\begin{pmatrix}
\Delta x \\ \Delta y
\end{pmatrix}_{\text{act}}
\] (4.2.13)

where the first matrix in right hand side represents the rotation \( \theta \), the second matrix comes from the gain factors of the BPM, \( b_x \) and \( b_y \) are horizontal and vertical signal gain factors. The expanded equations are

\[
\Delta x = b_x \cos \theta \Delta x_0 - b_y \sin \theta \Delta y_0
\]
\[
\Delta y = b_x \sin \theta \Delta x_0 + b_y \cos \theta \Delta y_0
\] (4.2.14)

or

\[
M^X_{ij} = b_x \cos \theta G^X_{ij} - b_y \sin \theta G^Y_{ij}
\]
\[
M^Y_{ij} = b_y \sin \theta G^X_{ij} - b_x \cos \theta G^Y_{ij}
\] (4.2.15)

where \( M^X_{ij} \) and \( M^Y_{ij} \) are the new ORM elements from horizontal and vertical COD. \( G_{ij} \) are Green’s Function between orbit corrector and BPM, and it is a pure dynamics quantity, not “contaminated” by BPM errors.

The errors considered in the benchmark lattices are listed in Table. 4.4. In this case, all quadrupole and BPM errors are fitted.

Fig. 4.6 is an output example from YAOC. The errors of BPM rolls and gains are uniform distribution around 0.2 rad and 0.2. To test the reliability of YAOC, some of the benchmark lattice even have a BPM gain factor as large as 1.6, and it is also fitted correctly.

As \( M_{ij} \) is a well defined linear function of BPM errors, in the iteration of minimizing \( \chi^2 \), the one turn matrix is not needed to be recalculated when calculate the Jacobian matrix of the ORM data. Instead of finite difference method the derivatives can be pre-calculated from Eq. (4.2.15) and hard-coded into program. In this way we
Table 4.4: Quadrupole and BPM Error

<table>
<thead>
<tr>
<th>Error</th>
<th>Design</th>
<th>Error</th>
<th>Num. of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad.</td>
<td>$K_1(F)$</td>
<td>+0.0577 m$^{-2}$</td>
<td>5% – 20%</td>
</tr>
<tr>
<td></td>
<td>$K_1(D)$</td>
<td>−0.0542 m$^{-2}$</td>
<td>5% – 20%</td>
</tr>
<tr>
<td></td>
<td>$\theta_{qf}$</td>
<td>0.0</td>
<td>±0.02 rad</td>
</tr>
<tr>
<td></td>
<td>$\theta_{qd}$</td>
<td>0.0</td>
<td>±0.02 rad</td>
</tr>
<tr>
<td>BPM</td>
<td>$b_x$</td>
<td>1.0</td>
<td>±0.3</td>
</tr>
<tr>
<td></td>
<td>$b_z$</td>
<td>1.0</td>
<td>±0.3</td>
</tr>
<tr>
<td></td>
<td>$\theta_{bpm}$</td>
<td>0.0</td>
<td>±0.1 rad</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td>336</td>
</tr>
</tbody>
</table>

Figure 4.6: Benchmark of BPM errors. Quadrupole errors are also considered in the model.

can speed up the fitting by about a factor of 2, depending on how many BPMs and other errors. The BPM noise effect will be discussed in Section 4.3.1.
4.2.3 Statistics

The performance of modeling problems highly depends on the numerical algorithm and how far the initial point is away from the solution in parameter space. But as we know the global minimum is very hard to get for multi-dimension fitting problems, we can not get any guarantee from one or two successes in the benchmarks. In order to understand more properties of ORM analysis, and show that YAOC works not only on a particular set of lattice errors but also many other cases, we have studied thousands of random error sets. Since most of the accelerator lattices are built upon similar blocks, such as FODO cell, Double Achromat Bend (DBA), and Triple Achromat Bend (TBA), the the statistical result on one accelerator lattice may help us to understand the application of ORM on other accelerators built upon the same blocks.

In our study, the benchmark lattice, same as before, are generated by adding an error term to the design values of each parameter. The random errors follow a certain distribution, uniform or Gaussian. The error size is characterized by the range of uniform distribution or $\sigma$ of Gaussian distribution. In each set of errors, an ORM is generated and input into YAOC. After the modeling and analysis, the output is then compared with the known benchmark lattice.

A study of Gaussian errors is shown in Fig. 4.7. About 1000 benchmark lattices with random errors are analyzed with ORM. The errors have $\sigma = 2\%K_1$, and all of them are predicted in a very accurate way. The differences between the benchmark error and our prediction, $|K_1^{\text{fit}} - K_1^{\text{actual}}|$, are all below $10^{-4}$, mostly below $10^{-7}$. Since the nominal $K_1$ is $\pm 0.05$, and the errors are $2\%$ of it. These results in Fig. 4.7 are

---

*aHere we do not really care about whether the real errors should be a Gaussian distribution or not, we will only focused on the randomness, and the error size, which describes how far it is away from the design lattice*
very reliable. The first bin in Fig. 4.7 around $|K_1^{\text{fit}} - K_1^{\text{actual}}| > 10^{-10}$ are results that are in fact $|K_1^{\text{fit}} - K_1^{\text{actual}}| \approx 0$. We are limited by the input/output precision, i.e. $10^{-10}$, to represent them.

Fig. 4.8 shows the statistical result of a model with gradient errors, 224 random cases are analyzed there. The errors are uniformly distributed and within the range of $[-0.008 \text{ m}^{-2}, 0.008 \text{ m}^{-2}]$, i.e. $\pm 16\%$ of the nominal body quadrupole component $q_{sf}$ or $q_{sd}$ which are around $\pm 0.05$. From the difference between fitted gradient and

\footnote{For such a large error, it took more time to generate a benchmark lattice which is stable. Among all the random lattice, I got 798 unstable ones, and only 224 stable lattice for benchmark.}
4.2. BENCHMARKS ON FERMILAB BOOSTER

Figure 4.8: Statistical result of gradient fit.

benchmark gradient, we can see in most cases, the fitted result are very close to benchmark lattice. The differences between them, \(|K_1^{\text{fit}} - K_1^{\text{actual}}|\), are mostly less than \(10^{-7}\), and mostly around \(10^{-8}\). Our input/output precision is around \(10^{-10}\). The disagreement of our fitting is well controlled less than 0.0125\% of the gradient strength we want to model. Some of the lattice errors are not correctly reconstructed, and have a disagreement greater than \(10^{-4}\). These not so good result come from that the errors here are too large, and the initial design lattice is two far away from the benchmark lattice. In this case, it is very hard to reach the global minimum.

A model including quadrupole tilt angle errors is also studied statistically. The
errors in benchmark lattice is shown in Fig. 4.9 and Fig. 4.10. The gradient errors have a uniform distribution within ±10% \( \text{qsf} \), and the tilt angle error of quadrupoles are within 0.01 rad. The agreement of our fitting, i.e. the difference between fitted and actual value is always better than \( 10^{-4} \). The histogram shows that the differences are mostly less than \( 5 \times 10^{-6} \ m^{-2} \) and \( 10^{-7} \ \text{rad} \). Both quadrupole strength and tilt angle are precisely reconstructed.

In summary, YAOCC has succeeded in the modeling of quadrupole gradient error, quadrupole tilt angle error, BPM gain factor and tilt angle error. The statistical study of nearly two thousand random benchmark lattice has proven the success of YAOCC on Fermilab booster. The error size used in our benchmarks a listed in Table 4.5.

In our simulation on thousands of different random error sets based on Fermilab
Booster lattice, we reconstructed over 90% of them. The accuracy goes better than $10^{-7} \, m^{-2}$ which is only about $10^{-5}$ of the body value $0.05 \, m^{-2}$. The uncertainty of the reconstructed gradient and BPM errors are around $0.01\%$. The rest cases fail because these random seeds produce unstable lattice, the ORM does not exist.
### Table 4.5: Error size in statistical studies

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadrupole</td>
<td>gradient $K_1$</td>
<td>$\pm 0.05 , m^{-2}$</td>
<td>$0.008 , m^{-2}$</td>
<td>$&gt; 1000$</td>
</tr>
<tr>
<td></td>
<td>tilt $\theta_q$</td>
<td>0.0</td>
<td>$\pm 0.05$</td>
<td>$&gt; 200$</td>
</tr>
</tbody>
</table>

#### 4.3 The Effects of BPMs

In the earlier benchmarks, no BPM noise is considered, and this is not true for analyzing real experiment data. Although we can do multiple measurement and average out the noise effect, but a simulation on BPM noise would guide us in the real measurement and data analysis. It can help us to increase the accuracy and reduce the unnecessary number of measurements.

The ORM data is determined by the betatron functions at BPM-orbit-corrector pair, and the phase advance between them. A proper set of BPMs can have better constraint on the lattice parameters, and brings uncoupled, unique solution to the lattice. Since the ORM element has a cosine term in it, an intuitive layout of BPMs would be placing several BPMs in each period of betatron oscillation. Both plus and minus sign of $\cos(\pi \nu - |\phi_1 - \phi_2|)$ should be covered. A long section without BPM in it will certainly provide the possibility that the quadrupoles in this section compensate for each other, and the result of ORM analysis may be not unique.

#### 4.3.1 BPM Noise

The BPM may have a certain degree of noise and resolution, which may make the ORM from model not exact. From central limit theory, the uncertainty of our measurement decrease as we increase the number of measurement, and the it scales as
The size of noise was studied in our simulation, and the study can help us to decide the number of measurement \( n \) in the real experiment. We add a random noise into the “measured ORM”, then analyze it with YAOC. In this way, our \( \chi^2 \) can not be reduced to a very small number (close to zero) due to the noise.

One way to add the noise is

\[
\left( \frac{\Delta G_{ij}}{G_{ij}} \right)^2 = \left( \frac{\Delta y_i}{y_i} \right)^2 + \left( \frac{\Delta \theta_j}{\theta_j} \right)^2
\]

(4.3.1)

Since in YAOC, the ORM element \( G_{ij} \) is derived from closed orbit condition instead of turn by turn tracking as in MAD, we do not have explicit form of \( \Delta y_i \) or \( \Delta \theta_j \), but only the ratio of them, i.e. \( \Delta y_i / \Delta \theta_j \). From Eq. (4.3.1), the relative noise on \( G_{ij} \) is about the same order as BPM noise and orbit corrector. This means that assuming 10 \( \mu \)m noise of BPM, 1 mm orbit change will be equivalent to \( 10^{-2} \) relative random change on \( G_{ij} \). The random change on \( G_{ij} \) may be doubled if the orbit corrector has similar noise level as BPM.

We use RMS difference, \( rms(\cdot) \), to characterize the agreement of two sets of data as following

\[
rms(x) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x'_i)^2}{n}}
\]

(4.3.2)

where \( x \) and \( x' \) are two sets of data we want to compare, each has \( n \) terms. In this dissertation I am always comparing my fitted result with the errors I prepared for benchmark lattices. The \( rms(\cdot) \) in Eq. (4.3.2) can describe both the initial noise level on \( G \), and the agreement of fitting on gradient error \( K_1, \theta_q, \) BPM error \( b_x, b_y \) and \( \theta_{bpm} \).

Figure 4.11 and 4.12 show the dependence of fitted result on BPM noise. The model has 336 parameters, including quadrupole and BPM errors we talked before.

\(^c\)As introduced before, it is the data from simulation, but used as an experiment data. Since we know precisely how this benchmark lattice generates the “measured ORM”, it is easy for us to check if our code really get the lattice which produced the “measured ORM”.
and the benchmark lattice is a single error lattice same as Fig. 3.3. If in experiment, the observed COD is 1 mm, Fig. 4.11 tells us how the fitted result becoming worse when the relative BPM noise level increases from $10^{-4}$ (or 0.1 µm) to $10^{-2}$ (or 10 µm). For this case, the nominal quadrupole strength $K_1$ is about ±0.05 m$^{-2}$, while the RMS of our fitting is $2 \times 10^{-4}$ at 10 µm BPM noise level, and the quadrupole tilt angle is about 1 mrad.

For Fermilab Booster single error lattice, $10^{-2}$ relative BPM noise, about 10 µm noise on 1 mm orbit shift, can still give an acceptable result as shown in Fig. 4.11 and Fig. 4.12. The rms result of tilt angles are about 1 mrad for quadrupole, and
4.3. THE EFFECTS OF BPMS

0.1 mrad for BPMs. For quadrupole strength and BPM gain factors, the rms results are 0.4% $* K_1$ m$^{-2}$ and 0.3%. One of the example where BPM noise is 10 $\mu m$ is shown in Fig. 4.13. The agreement of fitting is better than 0.1% of nominal quadrupole strength.

4.3.2 Consecutively Disabled BPMs

ORM analysis relies on how many BPMs and their distribution in the ring. The more BPMs, the more data in $\chi^2$, the more determined of the system. As we will find in the simulations, if a consecutive set of BPMs are disabled, the fit result within
that region will not be as good as the full functioning case. It means that in the region where a lot of BPMs are missed, $\Delta K_1$ has more freedom to vary and balance each other, still keep similar magnitude of $\chi^2$. This missing BPM lattice can cause the ambiguity, where two or more different lattices (with different gradient or BPM errors) give similar $\chi^2$. In this case, no unique solution is guaranteed. The similar situation is the coupling between the neighbor magnets, especially when they are very close to each other, but no BPM placed between them. Therefore increasing one focusing strength can be partly compensated by changing the other [14], and in this case it produces a local minimum. The more local minimum close to iteration path,
the more difficult for getting convergence in least squares fitting. The coupling may be resolved by using a more robust algorithm, but it becomes impossible when not enough BPMs or constraint points are available. By the constraint points we mean the proper layout of the BPMs.

![Figure 4.14: Fitted gradient error, when the first 5 BPMs are disabled.](image)

When 5 BPMs in a row are disabled, the fitted gradient errors become worse. As shown in Fig. 4.14 and Fig. 4.15, where the agreement of fitting strongly depends on the location of disabled BPMs. When the first 5 BPMs are disabled, the fitted result of gradient near these BPMs are not quite right as other regions, similar conclusion is obtained from the situation where disabling other 5 BPMs half circumference away.
We connect this phenomena to Nyquest theorem, where your sampling frequency need at least twice of the signal to correctly reconstruct it. Here the betatron oscillation is also a sinusoidal form, and this oscillation comes from the gradient of accelerator lattice. The horizontal betatron oscillation tune of the Booster is about 6.7, which is equivalent to 100° per FODO cell. The disabled consecutive 5 BPMs spans over $2\frac{1}{2}$ cells, and can make the betatron oscillation free of observation (constraint). Due to this freedom, the experiment data can not predict the betatron oscillation function in this region, therefore the gradient of the quadrupoles in this region can freely combine with each other, but make the data outside of these cells consistent with experiment.
4.3. **THE EFFECTS OF BPMS**

In the benchmarks, we have been using 48 BPMs and 48 virtual correctors at same location as BPM to benchmark YAOC. The benchmark lattice scales from 96 parameters to 336 parameters including quadrupole and BPM errors.

The number of BPMs or the number of ORM terms used in YAOC may also affect the final result. Depending the size of parameter space, more parameters need more data to fit. For the Fermilab Booster, a model with only quadrupole errors, but no BPM errors present, 24 to 32 BPMs may be enough. A more general conclusion is that a FODO cell type lattice, need $3 \sim 4$ BPMs for each betatron oscillation period.

Fig. 4.16 uses only 16 BPMs, and 16 correctors to fit for quadrupole errors. Apparently $1/9$ of the normal number of ORM terms is not enough. For this same benchmark lattice, 24 BPMs and 24 correctors are enough, and the result is shown in Fig. 4.17, although the tilt angles of quadrupoles did not agree as well as the gradient.

For the models with maximum number of parameters we have tried, where both...
CHAPTER 4. BENCHMARK AND APPLICATION OF YAOC

Figure 4.17: Fitting quadrupole errors with 24 BPMs

quadrupole and BPM errors are considered, 24 BPMs are reasonably enough for gradient error. The main concern is still the tilt angle of quadrupoles, which has larger RMS difference than any other parameters. Considering the quadrupole tilt angles in this benchmark lattice are a few tens mrad, this RMS difference is not negligible. The reason for this not so good benchmark is that there is a degeneracy in ORM analysis, and this degeneracy unfortunately is intrinsic, can’t be changed by an algorithm or lay out of BPMs. Because you can’t tell any difference if all the magnets has the same tilt angle. i.e. if we rotate all the magnets by the same angle $\theta$, all the ORM will not change. Although it’s probability is very small, it sometimes do affect our fitting during the iterations. When it presents, it’s very hard to get the right tilt angle for quadrupoles or BPMs.

As shown in Fig. 4.18, in order to get an agreement better than 1% relatively, YAOC needs more than 80% of the BPMs in Fermilab Booster to model the quadrupole and BPM errors. Less is needed when only gradient errors are concerned.
4.3. Conclusion on BPM effects

From the benchmarks on Fermilab booster, we studied the effects of BPM noise and BPM numbers. $10^{-2}$ relative change on ORM elements are an acceptable noise level for single error lattice, and this corresponds to about $10 \mu m$ noise level at each BPM when the closed orbit shift is around $1 \ mm$. If corrector noise is considered, this number may be discounted. i.e. we need $2 \ mm$ COD if both BPMs and correctors have relatively $1\%$ noise level. For larger noise, we have to increase the kick strength to increase the signal noise ratio on the BPMs.

For FODO cell lattice, at least $4$ BPMs per betatron oscillation period is neces-
sary for gradient fitting, this number may increase if more errors enter in, such as quadrupole tilt angle and BPM errors. In this case, YAOC needs at least 80% of the BPMs to fit for all these errors. The relative low agreement of tilt angle when less BPMs are used may comes from the intrinsic degeneracy of ORM analysis that rotating all the magnets by an angle $\theta$ will not change any ORM element at all.

Consecutive non-functioning BPMs in ORM analysis would bring unresolvable coupling between the quadrupoles in that BPM-missing region, especially for a lattice where the quadrupoles are very close to each other.

### 4.4 Taiwan Photon Source

Taiwan Photon Source (TPS) will be a medium energy high performance and low emittance storage ring at National Synchrotron Radiation Research Center (NSRRC). One of the conceptual design is 24-cell double bend achromat (DBA) structure with circumference 518.4 m. The main lattice parameters of this design is in Table. 4.6 [18, 19].

The lattice functions of one of the six super-periods are shown in Fig. 4.19. By allowing slightly positive dispersion in the long straights, the natural emittance of 1.7 nm-rad can be achieved. With a 6-fold symmetry configuration, the ring provides 6 long straights for injection, long insertion devices, and superconducting RF modules. 168 BPMs in total are distributed along the ring.

In the ORM analysis point of view, TPS DBA lattice is quite different from Fermilab Booster that its model is larger with twice parameters, and the lattice block is DBA instead of FODO cell. The design lattice has 48 quadrupoles, divided into two categories, QL and QS, with focusing strength $\pm0.8\sim1.2$ and $\pm1.2\sim1.6$ respectively. Due to a large number of BPMs, the full set of ORM elements is around 112,898 terms. The possible errors used in the benchmark are listed in Table 4.7,
over 600 parameters including quadrupole and BPM errors are considered. In benchmarks, each quadrupole has a relative random error about 0.1% and a tilt angle error around 10 mrad; the BPM gain factor errors and tilt angles are exaggerated to 20% and 100 mrad to test the potential of YAOC. Each element has independent errors regardless of being powered by same power supply or not.
Figure 4.19: Lattice function of NSRRC TPS Storage Ring design lattice
### Table 4.7: Model Parameters

<table>
<thead>
<tr>
<th></th>
<th>Error</th>
<th>Design</th>
<th>Error</th>
<th>Num. of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad.</td>
<td>$K_1(L) [m^{-2}]$</td>
<td>$\pm 0.8 \sim 1.2$</td>
<td>$\pm 1.2 \sim 1.6$</td>
<td>$\pm 0.1 %$</td>
</tr>
<tr>
<td></td>
<td>$\theta_{ql}$</td>
<td>0.0 mrad</td>
<td>$\pm 10$ mrad</td>
<td>3 x 6</td>
</tr>
<tr>
<td></td>
<td>$\theta_{qs}$</td>
<td>0.0 mrad</td>
<td>$\pm 10$ mrad</td>
<td>5 x 6</td>
</tr>
<tr>
<td>BPM</td>
<td>$b_x$</td>
<td>1.0</td>
<td>$\pm 25%$</td>
<td>8 x 6</td>
</tr>
<tr>
<td></td>
<td>$b_z$</td>
<td>1.0</td>
<td>$\pm 25%$</td>
<td>28 x 6</td>
</tr>
<tr>
<td></td>
<td>$\theta_{bpm}$</td>
<td>0.0</td>
<td>$\pm 0.10$ rad</td>
<td>28 x 6</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td>600</td>
</tr>
</tbody>
</table>
Figure 4.20: Quadrupole errors
Figure 4.21: BPM errors
One set of quadrupole errors are fitted as shown in Fig. 4.20. The benchmark lattice errors for quadrupoles are focusing strength $\Delta K_1$ and tilt angle $\Delta \theta$. The agreement of $\Delta K_1$ is better than a few percent of the errors, and $\Delta \theta$ better than 0.06 mrad. The BPM errors are also fitted at the same time, shown as in Fig. 4.21. They have much better agreement than quadrupole strength, and most of the numbers are limited by input/output precision.

![NSRRC Cell 2/6](image.png)

Figure 4.22: Vertical $\beta$-function before and after fit

The $\beta$-function depends on the distribution of gradient along the ring, after the gradient is fitted, the distortion of $\beta$-function should also be successfully predicted by YAOC. Fig. 4.22 shows the vertical betatron function distortion between the design
4.4. TAIWAN PHOTON SOURCE

lattice and the benchmark lattice. The benchmark lattice is prepared by adding ±1~2% gradient errors to the nominal values. This will bring 30~40% of vertical $\beta$-function distortion, shown as red and blue lines in Fig. 4.22. After fitting for the errors in benchmark lattice, YAOC successfully predict the betatron function. Therefore the symmetry of lattice (betatron function) can be restored after correcting those errors.

600 parameters are a challenge for multi-dimensional optimization, especially when the merit function is not a simple linear function of these parameters. A few of the benchmark lattices of TPS are not fully fitted, although they are very close. In Fig. 4.23, the BPM errors are fitted in a acceptable level, but not for quadrupole errors. The tilt angle errors of BPM was fitted in a good precision, while the gain factors have a small shift. As we have done in the benchmarks on Fermilab Booster, the fitting of BPM errors are quite stable. The reason is that the ORM elements are a well defined linear function of BPM errors, and the dependence is simply a rotation and a scaling on two transverse axes. The tilt angles of quadrupoles are fitted in an acceptable way, but the fitting for quadrupole strength is not good enough. This shows that global minimization is not guaranteed, and more techniques are need to make such a big model with 600 parameters converged.

In summary, YAOC has been benchmarked on a DBA lattice, one of the Taiwan Photon Source (TPS) design lattices. Different from a FODO cell lattice as in Fermilab Booster, it has 600 parameters to fit. In several benchmarks listed in Table. 4.7, we have successfully fitted the quadrupole strength, quadrupole tilt angle, BPM horizontal/vertical gain factor and BPM tilt angle but failed on one of them. The betatron function distortion caused by gradient errors is successfully predicted, and the correction of the errors has restored the 6-fold symmetry of lattice.
Figure 4.23: One unsuccessful example on quadrupole errors, BPM gain factors look like shifted.
4.5 NSLS VUV Ring

The VUV Ring at the National Synchrotron Light Source was one of the first of the 2nd generation light sources to operate in the world. The final lattice design was completed in 1978 and the first stored beam was achieved in December of 1981 at 600 MeV and the first photons were delivered to beamlines in May 1982. The routine operation is 1 ampere at 800 MeV.

Figure 4.24: Lattice functions

The basic lattice design for the VUV ring was a four superperiod Green-Chasman
or DBA (Double Bend Achromatic) lattice with circumference of 51 meters. The lattice functions are shown in Fig. 4.24. The achromatic correction results from a single family of two focusing quadrupoles in the center of the dispersion straight section, separated by a single focusing sextupole. Tune adjustment and matching of the optic functions in the dispersion-free straight sections is provided by a quadrupole doublet.

Table 4.8: Quadrupole Gradient Error

<table>
<thead>
<tr>
<th>Quad.</th>
<th>Design</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1(F)$</td>
<td>$+1.81 , m^{-2}$</td>
<td>$\pm5% - 10%$</td>
</tr>
<tr>
<td>$K_1(D)$</td>
<td>$-1.12 , m^{-2}$</td>
<td>$\pm5% - 10%$</td>
</tr>
</tbody>
</table>

The parameters used in benchmark is listed in Table 4.8. Since it is a small DBA cell lattice, the convergence and agreement should be better than the large rings discussed before. The result of each benchmark is not shown here, but has been confirmed that it has similar agreement as for the Fermilab Booster and the TPS DBA design lattice.

The RMS difference of betatron function and gradient between benchmark lattice and the lattice we get in each iteration are shown in Fig. 4.25. The gradient error at the beginning is 0.15, about 10% of the nominal quadrupole strength. As we minimizing the $\chi^2$ of ORM, interestingly, at one point, the RMS difference of gradient error increased a little, i.e. instead of approaching the benchmark lattice, the error gets away from it before converging toward the true solution. This is an indirect proof of local minimum, where the $\chi^2$ is smaller, but the lattice is not closer to the expected one. From the initial point to the 9th iteration, $\chi^2$ decreased a factor of $10^{-3}$. $rms(K_1)$ has same order of decreasing. The betatron function is closely
correlated to the gradient. As we have expected, it shows that $\beta$-function is a good measurement of gradient error fitting.

One application of YAOC and ORM analysis is determining the source of vertical orbit oscillations. The perturbation at two BPMs are shown in Fig. 4.27 as blue lines, about $0.2$ mm at PUE02\textsuperscript{d} and $0.02$ mm at PUE17. The oscillation may come from one location that the beam is given a perturbation, i.e. a kick angle $\theta$. One way to find the location of kick is ORM analysis with YAOC. A set of points along the

\textsuperscript{d}PUE (Pick-up Electrode) is another name of BPM. PUE02 is a device name in VUV ring
ring is chosen as the candidate location of the perturbation, and the agreement is described by a merit function $\chi^2$, shown in Fig. 4.26. The $\chi^2$ comes from the ORM data (or vertical orbit shift) produced by each candidate perturbation. The minimum point of $\chi^2$ indicates that the source of perturbation is around $s = 0$ m, which is the injection point. A further discussion with engineers concludes that a device called BUISH is the most possible source of vertical orbit oscillations. After turning it off, the vertical orbit is well controlled under 0.02 mm level. This is shown as the green lines in Fig. 4.27.
Figure 4.27: Vertical orbit oscillation at two BPMs
Chapter 5

NSLS VUV Ring Lifetime

The beam lifetime for an electron storage ring is usually limited by gas scattering or Touschek scattering. Understanding and increasing beam lifetime is important as it provides more integrated flux for the users as well as reduces radiation exposure. Lifetime in the VUV ring has never been fully understood, measured lifetime is shorter than earlier calculations.

A series of studies was recently done on VUV ring to understand the dominating effect. Both beam measurement and simulation indicate that it is the 1.5 m dispersion and small physical aperture limiting the survival of scattered particles in Touschek effect [37]. The scattering on residual gas are also not negligible at high current.

5.1 The NSLS VUV Ring

The NSLS VUV Ring has a nominal energy of 800 MeV, 4 cell DBA lattice, and 51 m circumference. The maximum current for a regular operation is 1 A. Top-off injection is applied every 4-5 hours. Some machine parameters are listed in Table 5.1, and the lattice function is of one DBA cell is shown in Fig. 5.1,
### Table 5.1: VUV ring parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal operating energy $E$</td>
<td>0.800 [GeV]</td>
</tr>
<tr>
<td>Peak operating current</td>
<td>1.0 [A]</td>
</tr>
<tr>
<td>Circumference</td>
<td>51.017 [m]</td>
</tr>
<tr>
<td>Dipole field, $B(\rho)$</td>
<td>1.41 [T] (1.91 [m])</td>
</tr>
<tr>
<td>Momentum compaction $\alpha_c$</td>
<td>0.0235</td>
</tr>
<tr>
<td>RF frequency</td>
<td>52.887 [MHz]</td>
</tr>
<tr>
<td>RF peak voltage</td>
<td>80 [kV]</td>
</tr>
<tr>
<td>Tune, $\nu_x, \nu_y, \nu_s$</td>
<td>3.14, 1.26, 0.0018</td>
</tr>
<tr>
<td>Natural energy spread, $\sigma_E/E$</td>
<td>$4.86 \times 10^{-4}$</td>
</tr>
<tr>
<td>Typical bunch mode (RF buckets)</td>
<td>7(9)</td>
</tr>
<tr>
<td>Damped emittance, $\epsilon_x/\epsilon_y$</td>
<td>160/4 nm-rad</td>
</tr>
<tr>
<td>$\beta_x$, max, min, average</td>
<td>11.4, 1.0, 5.3 [m]</td>
</tr>
<tr>
<td>$\beta_y$, max, min, average</td>
<td>14.1, 2.8, 8.1 [m]</td>
</tr>
<tr>
<td>Dispersion $D_x$, max</td>
<td>1.5 [m]</td>
</tr>
</tbody>
</table>

The beam lifetime $\tau$ is defined as

$$\frac{1}{\tau} = \frac{1}{I} \frac{dI}{dt} = \frac{1}{N} \frac{dN}{dt} \quad (5.1.1)$$

where $I$ is the current of circulating beams, $N$ the number of particles. The lifetime in regular operations is complicated due to fairly large beam current and hence a variety of associate collective effects, including significant potential well distortion and microwave instability. Furthermore, significant beam loading creates bunch-length and bunch-shape spread between the 7 bunches used for regular operations. Finally, an actively powered 4th harmonic RF system is used to gain lifetime by stretching the bunch length. For these studies we went to simpler experimental conditions, with
5.1. THE NSLS VUV RING

Total beam lifetime usually consists of two parts, scattering from residual gas and scattering within the beam. The gas scattering lifetime is comprised of the elastic and inelastic scattering on electrons and nuclei, while the self-scattering within a beam is called Touschek effect.
5.2 Beam-gas interaction

Residual gases in a high-vacuum system are one of the most important concerns from the viewpoint of long lifetime for circulating beams. Typical residual gases in an ultra-high vacuum (UHV) system at low pressures and room temperature are \( \text{H}_2 \), \( \text{H}_2\text{O} \), \( \text{CO} \), \( \text{N}_2 \) and \( \text{CO}_2 \). Gasses are attached to the walls by adsorption or absorption. The outgassing may happen at the wall via thermal desorption, electron stimulated desorption, photodesorption and ion-induced desorption.

There are essentially four beam-gas interactions that may cause the beam loss [10]. The loss rate is characterized by the cross section of beam-gas interactions.

1. **Elastic scattering on nuclei.** After scattering, if the induced betatron motion amplitude exceeds the vacuum chamber aperture, the particle gets lost. The total cross section for this process is

\[
\sigma_{t1} = \frac{4r_e^2 Z^2 \pi}{\gamma^2} \frac{\langle \beta \rangle^2}{a^2} \tag{5.2.1}
\]

where \( r_e \) is the classical electron radius, \( \gamma \) is the normalized energy, \( Z \) is the atomic number for residual gas components, \( \langle \beta \rangle \) is the average betatron function and \( a \) is the half chamber aperture.

2. **The bremsstrahlung on nuclei.** It is an inelastic scattering process, and the energy loss makes particle’s energy deviation \( \delta E/E \) exceeds the RF acceptance of the ring, then gets lost. The total cross section for this process is

\[
\sigma_{t2} = \frac{4r_e^2 Z^2}{137} \frac{4}{3} \ln \frac{183}{Z^{1/3}} \left( \ln \frac{1}{\epsilon_{\text{RF}}} - \frac{5}{8} \right) \tag{5.2.2}
\]

where \( \epsilon_{\text{RF}} \) is the RF acceptance. It comes from the RF bucket height in Eq. (5.3.2).
3. *The elastic scattering on electrons.* In scattering, the particle transfers its energy to the electrons of residual gas, and its energy deviation may exceed the RF acceptance, therefore gets lost. The cross section is

\[ \sigma_{el} = \frac{2\pi r_e^2 Z}{\gamma} \frac{1}{\epsilon_{RF}} \]  

(5.2.3)

4. *Inelastic scattering on electrons.* The cross section for this interaction is

\[ \sigma_{in} = \frac{4r_e^2 Z}{137} \frac{1}{3} \left( \ln \frac{2.5\gamma}{\epsilon_{RF}} - 1.4 \right) \left( \ln \frac{1}{\epsilon_{RF}} - \frac{5}{8} \right) \]  

(5.2.4)

The residual gas pressure in electron storage ring strongly depends on the beam current. The equilibrium pressure \( P \) results from static pressure \( P_0 \) with no beam circulating, a dynamic effect proportional to the stored current. This dynamic part comes from the gas desorption process caused by synchrotron radiation. The photoelectrons are emitted over a large solid angle and hit the walls too, driving the molecules out. In the case of ultra vacuum systems \( 10^{-10} \) Torr the gas composition is approximately 90% \( \text{H}_2\text{O} \) and 10% \( \text{CO} \). Residual gas analyzer (RGA) has been used in VUV ring to detect minute impurities in the vacuum chamber.

For beam-gas interaction, the beam lifetime is

\[ \frac{1}{\tau_g} = \sigma_{tg} c n , \]  

(5.2.5)

where \( c \) the particle velocity, and \( n \) the density of the residual gas which is related to the gas pressure:

\[ n [m^{-3}] = 3.217 \times 10^{22} P [\text{Torr}] . \]  

(5.2.6)

The beam-gas scattering lifetime due to the above effects are summarized in Fig. 5.2. In this figure, we are assuming the momentum acceptance is dominated by RF instead of other dynamics. For energy like VUV ring, \( E = 800 \text{ MeV} \), aperture \( a = 21 \text{ mm} \), the bremsstrahlung becomes determinant. The total gas scattering
Figure 5.2: Gas scattering lifetime, \((Z = 7.0)\)

lifetime is around 15 hours at 5 nTorr, and 7 hours at 10 nTorr. The contribution from residual gas may increase if the RF acceptance is underestimated, i.e. limited by other dynamics as we will see in the following sections. The ion trapping may also contribute to the lifetime, due to a high beam current, but it’s not quantitatively studied yet.

Recently, a vacuum leak in one of the beamline of VUV ring proved that the effect of residual gas on beam lifetime is not negligible. As shown in Fig. 5.3, the leak happens when the beam current is about 530 mA, the pressure detected by the
vacuum pump close to the leak jumped from 2.5 nTorr to about 4 nTorr. The lifetime dropped $10 - 15\%$ at the same time, and nearly 30\% in 12 hours. The vacuum history data did not show any significant change on other 7 vacuum pumps distributed along the ring. Fig. 5.4 shows the lifetime and residual gas pressure of the second injection 12 hours after the vacuum leak happened.

Fig. 5.5 shows the average vacuum pressure dependence on beam current, and compares the pressure in the vacuum leak day and a normal operation day. It is doubled due to the vacuum leak. The corresponding lifetime change is shown in
Fig. 5.6. We can subtract the lifetime due to the vacuum leak by comparing it with normal operation data, and it is about 13 hours. Since this leak roughly doubled the vacuum pressure, we expect the contribution from gas lifetime in a normal operation day would be the same amount, about 13 hours, at 500-1000 mA. Lower current would have longer lifetime, but from the linear dependence in region 500-1000 mA, we would still get a number close to 20 hours. A more detailed and accurate calculation would include the gas composition data from residual gas analyzer (RGA).

Residual gas in the vacuum system can lead to other undesirable effects such as
emittance growth. The measured transverse beam size is larger than theoretical calculations. Another problem that occurs with intense negative beams is ion trapping. Ions produced by the beam are trapped in the potential well of the beam charge, so that the effective gas density seen by the beam is many times larger than that indicated by the measured pressure. Bunched beams can avoid this problem if the bunch intensity and spacing are such that the ions are accelerated into the walls before the next bunch comes around [25].
5.3 Touschek Lifetime

Touschek lifetime comes from the effect that scattering of two particles in their center of mass frame transfers transverse momentum into longitudinal momentum. Particle loss happens when the longitudinal momentum deviation exceeds the applicable momentum deviation [20, 9]. This momentum aperture may come from dynamic or physical aperture, or the RF bucket height, whichever is the smallest. Touschek
5.3. TOUSCHEK LIFETIME

The lifetime describing this process for a flat beam is \([4, 38]\):

\[
\frac{1}{\tau_t} = -\frac{1}{N_b} \frac{dN_b}{dt} = \frac{r_c^2 c N_b}{8 \pi \sigma_x \sigma_y (\Delta p/p)^3 \gamma^2} C(\zeta) \tag{5.3.1}
\]

where

\[
C(\zeta) \equiv \sqrt{\zeta} \left( -\frac{3}{2} e^{-\zeta} + \frac{\zeta}{2} \int_{\zeta}^{\infty} \frac{\ln u}{u} e^{-u} du + \frac{3 \zeta - \zeta \ln \zeta + 2}{2} \int_{\zeta}^{\infty} \frac{e^{-u}}{u} du \right)
\]

and \(\zeta = \left(\frac{\beta_x}{\gamma \sigma_x (\Delta p/p)}\right)^2\), which depends on betatron amplitude function \(\beta_x\), beam energy \(\gamma\), and momentum acceptance \(\Delta p/p\) at the point of scatter. The momentum acceptance can be defined by mechanical aperture, dynamic aperture and RF bucket height. We will discuss it later. Touschek lifetime comes from an average of Eq. (5.3.1) over the whole ring.

For an equilibrium state of electron storage ring, where the required energy gain for acceleration or compensation of synchrotron radiation losses is \(U_0\) per turn, the RF bucket height is \([20]\)

\[
\Delta E/E_{rf} = \left(\frac{2eV}{\pi \beta^2 Eh |\eta|}\right)^{\frac{1}{2}} Y(\phi_s) \tag{5.3.2}
\]

where \(E\) is beam energy, \(\beta \approx 1\) for electron ring, \(h\) is harmonic number, \(\eta\) is phase slip factor and \(\phi_s\) is phase of synchronous particle. The bucket height factor is \(Y(\phi_s) = |\cos \phi_s - \frac{\pi - 2\phi_s}{2} \sin \phi_s|^{1/2}\). If there are other effects that limit the momentum acceptance, the tolerable momentum aperture will be smaller than that in Eq. (5.3.2), i.e. \(\Delta p/p < \Delta E/E_{rf}\), and the Touschek lifetime will be reduced. This is indeed the case for NSLS VUV ring.

A general formula for an arbitrary ratio of horizontal to vertical betatron amplitudes, arbitrary energies, and derivative of the lattice functions, \(\beta'_{x,y}\) and \(D'_{x,y}\), was introduced by A. Piwinski in Ref. [27, DESY 98-179]. The lifetime is given as

\[
\frac{1}{\tau_{tp}} = \left\langle \frac{r_c^2 c N_p}{8 \pi \gamma^2 \sigma_s \sqrt{\sigma_x^2 \sigma_y^2 - \sigma_p^2 D_x^2 D_y^2}} F(\tau_m, B_1, B_2) \right\rangle \tag{5.3.3}
\]
where \( r_p \) classical particle radius, \( N_p \) number of particles per bunch, \( \gamma \) particle energy, \( \sigma_{x,z,s} \) 3D bunch dimension, \( D_{x,z} \) dispersion function.

\[
F(\tau_m, B_1, B_2) = \sqrt{\pi (B_1^2 - B_2^2) \tau_m} \\
\int_{\tau_m}^{\infty} \left[ (2 + \frac{1}{\tau})^2 \left( \frac{\tau/\tau_m}{1 + \tau} - 1 \right) + 1 - \frac{\sqrt{1 + \tau}}{\sqrt{\tau/\tau_m}} \right. \left. \right. \\
- \frac{1}{2\tau} \left( 4 + \frac{1}{\tau} \ln \frac{\tau/\tau_m}{1 + \tau} \right) e^{-B_1\tau} I_0(B_2\tau) \frac{\sqrt{\tau}}{\sqrt{1 + \tau}} d\tau
\]

(5.3.4)

and

\[
B_1 = \frac{\beta_x^2}{2 \beta^2 \gamma^2 \sigma_{x\beta}} (1 - \frac{\sigma_h^2 \hat{D}_x^2}{\sigma_{x\beta}^2}) + \frac{\beta_z^2}{2 \beta^2 \gamma^2 \sigma_{z\beta}^2} (1 - \frac{\sigma_h^2 \hat{D}_z^2}{\sigma_{z\beta}^2})
\]

\[
B_2^2 = B_1^2 - \frac{\beta_x^2 \beta_z^2 \sigma_h^2}{\beta^4 \gamma^4 \sigma_{x\beta}^4 \sigma_{z\beta}^4 \sigma_p^2} (\sigma_x^2 \sigma_z^2 - \sigma_p^2 D_x^2 D_z^2)
\]

(5.3.5)

An ELEGANT related program based on this general formula has been written by A. Xiao to facilitate the calculation [35, 1].

From the discussion above, Touschek lifetime depends on the following parameters:

- **Single bunch current.** Different from beam-gas scattering, the Touschek effect is scattering within bunch, it depends only on single bunch current.

- **Beam volume,** larger \( \sigma_x, \sigma_z \) and \( \sigma_\ell \) will reduce the probability of scattering, and the lifetime is inversely proportional to the bunch density.

- **Energy,** which is hidden in emittance \( \epsilon_x \), RF acceptance in \( C(\zeta) \), and bunch length \( \sigma_\ell \).

- **Momentum acceptance.** It is the smaller of physical aperture or RF bucket.

Based on these aspects, we did our experiments to understand Touschek lifetime in VUV ring.
5.3.1 Bunch Length Measurement

A photodiode used for bunch length measurement has a pulse response. The bunch-length is calibrated from the raw measurement data by:

\[ \sigma_{\text{obs}}^2 - \sigma_{\text{ls}}^2 = \sigma_\ell^2 \]  \hspace{1cm} (5.3.6)

where \( \sigma_{\text{obs}} \) is the observed bunch-length, \( \sigma_{\text{ls}} \) is the diode-scope system pulse response (assumed a constant), and \( \sigma_\ell \) is the beam bunchlength.

The calibration is done by measuring bunchlength at different RF power. Since the product of \( \nu_s \) and \( \sigma_\ell \) does not change with RF, the real bunchlength can be obtained. \( \eta = -0.0235, \sigma_E/E = 4.64 \times 10^{-4} \)

The bunch length for a synchrotron is described by

\[ \sigma_\ell = \frac{c|\eta|\sigma_E}{\omega_s \frac{E}{E}} = \sqrt{\frac{2\pi a \hbar c^2}{\omega_{\text{rf}}^2 \cos \phi_s eV_{\text{rf}}}} \sigma_E \]  \hspace{1cm} (5.3.7)

where \( \sigma_{\text{ls}} \) is the beam momentum spread.

Since

\[ \sigma_\ell \omega_s = c|\eta| \frac{\sigma_E}{E} \]

is only a property of lattice, i.e. a constant when changing the RF voltage. We can fit for \( \sigma_{\text{ls}} \) by measuring \( \omega_s \) and fitting the \( \sigma_{\text{obs}} \) using Eq. (5.3.6). The RF voltage and power are related by fixed shunt impedance, \( R_s \), therefore we can discuss the bunchlength and lifetime on RF power in stead of voltage. The result is shown in Fig.5.7 and has been also confirmed by streak camera measurements.

5.3.2 Momentum Acceptance

The closed orbit for a off-momentum particle depends on a product of its momentum deviation \( \delta \) and dispersion \( D_x(s) \). At the center of dispersive region, so after a Touschek scattering with energy change \( \delta \), the closed orbit of the reference particle
Figure 5.7: RMS bunch length. Single bunch operation at $I = 5\, mA$

will change from 0 to $D_x(s_0)\delta$, and execute a larger betatron oscillation along its new closed orbit. The physical coordinate of the betatron oscillation depends on the betatron amplitude and $\beta_x$, therefore the particle may get lost at large $\beta_x$ locations, e.g. the straight section of VUV ring. An estimation of transverse momentum acceptance can be done based on lattice functions in Fig. 5.1, with the normal beam pipe dimension of $\pm 40 \times \pm 20$ mm, and the septum aperture of $\pm 35 \times \pm 20$ mm. The physical coordinate of the scattered amplitude in the septum straight section with larger $\beta_x$ should be $\sqrt{\frac{\beta_x(s)}{\beta_x(s_0)}} D_x \delta$, where $\beta_x(s) \approx 12 \, m$, $\beta_x(s_0) \approx 2.8 \, m$, and $D_x \approx 1.5 \, m$. 
Therefore the maximum momentum deviation is limited by the septum transverse aperture to $\frac{\Delta p}{p}(s_0) \approx 1\%$.

To find the momentum aperture along the whole ring particle tracking was used. A module has been implemented in Elegant [1] to find the transverse momentum acceptance at each location along the ring, and the result for VUV ring is shown in Fig. 5.8. The straight line is the RF bucket height which dominates the momentum acceptance in the non-dispersive region, while in the dispersive region the transverse momentum acceptance is dominating [33].

Figure 5.8: Momentum aperture.
5.3.3 Effect of Physical Aperture

Since we believe the momentum acceptance is limited by the ±35 mm physical aperture near injection septum and the dispersion $D_x \approx 1.5$ m at the center of DBA cell. An experiment was done to study this effect. In this study the orbit was bumped locally in injection region, while kept unchanged at other locations. Fig. 5.9 shows that the lifetime is sensitive to beam steering in this region. A significant lifetime drop happens when the beam is moved close to or away from septum, where we have

Figure 5.9: Lifetime variation with closed orbit locally bumped near injection septum.
small physical aperture. Currently the beam near the septum is locally centered (-8 mm) away from the global reference orbit. This centers the closed orbit between the septum and inner chamber apertures. The sharper dependence at negative values comes from a nonlinearity of the BPM readings and a longer aperture of the inner wall versus the shorter longitudinal aperture of the septum.

A program provided by Elegant [1] was used to calculate Touschek lifetime for the VUV ring model by averaging the loss around the ring for the lattice shown in

![Graph](image-url)

**Figure 5.10:** Touschek lifetime simulation and experiment. Single bunch operation, ≈ 130 mA
Fig. 5.1 and momentum aperture in Fig. 5.8. The resulting lifetime times current as a function of RF power is shown in Fig. 5.10. A break point exists around 1.2 kW, and separates the lifetime into two regions dominated by different effects. Above 1.2 kW, increasing RF voltage only improve the overall momentum acceptance slightly, since almost half of the ring is dominated by a smaller transverse momentum acceptance. Below 1.2 kW, it is always RF dominated, and lifetime become more sensitive to RF bucket height. The simulation disagrees with the measurement indicating a possibly smaller physical aperture exists than the designed $\pm 35$ mm aperture. This is shown for several smaller values of aperture down to $\pm 25$ mm.

5.4 Conclusion

We have done both experiments and simulations to understand the lifetime of VUV ring. Gas scattering has relatively weak effect, while Touschek scattering dominates in VUV ring beam lifetime. The large peak dispersion and small physical aperture near injection septum accounts for the essential limitation on momentum acceptance, therefore higher RF voltage won’t improve the beam lifetime, but below 1.2 kW may bring down the lifetime quickly. The difference between simulation and experiments show the lifetime is smaller than can be explained by the known apertures in the ring. Further studies will be needed to identify its source.
Chapter 6

Conclusion

6.1 YAO and ORM analysis

A new orbit code called YAO is developed and benchmarked on various lattice including the Fermimab Booster, the Taiwan Photon Source DBA design lattice, and the VUV ring at National Synchrotron Light Source, BNL.

YAO has implemented the transfer matrix for drift space, quadrupole, sextupole, BPM, corrector and bending magnets up to third order. Users can also easily define their own elements with the VMatrix class. The operator overloading feature makes the manipulation (multiplication) of transfer matrix more in a nature way.

The Scaled Levenberg-Marquardt algorithm is used in YAO to optimize our lattice model and fit the magnet and diagnosis instrument errors, such as quadrupole strength, quadrupole tilt angles, BPM horizontal/vertical gain factors and BPM tilt angles. The convergence property is studied statistically on over a thousand of random cases, and most of the errors are successfully predicted by YAO, except for a few benchmarks on TPS lattice where fitting 600 parameters simultaneously is a challenge. YAO has also resolved the coupling between nearby magnets, which can not be
The BPM effects are studied to improve the efficiency of ORM analysis. It is suggested that a consecutively disabled BPM will bring down the quality of fitted error, and at least 4 BPMs per betatron oscillation period is required to fit gradient errors. More BPMs are needed if other errors such as quadrupole tilt angles and BPM errors are included. With 1% relative BPM noise level, at about 1 mm closed orbit shift, YAOC can still get an acceptable result for Fermilab Booster lattice.

YAOC can also help study the vertical orbit oscillations. By assuming the perturbation source at different locations, and calculate the likelihood of their possibilities, YAOC successfully predicted that the BUISH device near injection septum makes the vertical orbit oscillate with 0.2 mm amplitude. It is confirmed in the experiment.

### 6.2 VUV ring beam lifetime study

Both experiments and simulations were done to understand the lifetime of VUV ring. Gas scattering has relatively weak effect, but is not negligible at high current. Touschek scattering dominates in VUV ring beam lifetime. The large peak dispersion and small physical aperture near injection septum accounts for the essential limitation on momentum acceptance, this is proven by particle tracking and orbit bump experiment. For this reason, the higher RF voltage won’t improve the beam lifetime, but below 1.2 kW may bring down the lifetime quickly. The difference between simulation and experiments show the lifetime is smaller than can be explained by the known apertures in the ring.
Bibliography


Appendix A

Matrix Decomposition

A.1 LU Decomposition

LU decomposition expresses a matrix as the product of a lower and upper triangular matrix. It is used in numerical analysis to solve systems of linear equations or find the inverse of a matrix[12]. Let $A$ be a square matrix. An LU decomposition has the form

$$A = LU$$  \hspace{1cm} \text{(A.1.1)}

or

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$  \hspace{1cm} \text{(A.1.2)}

as an example for $3 \times 3$ matrix $A$. 

105
A.2 Cholesky Decomposition

If a matrix $A$ is Hermitian and positive definite, then $A$ can be decomposed into a lower triangular matrix and the transpose of the lower triangular matrix

$$M = LL^*$$

(A.2.1)

$A$ is Hermitian means that $a_{ij} = \overline{a_{ji}}$, where $\overline{x}$ is the complex conjugate of $x$. Positive definite means for all non-zero vectors $x \in \mathbb{C}^n$, $\overline{x}Az > 0$.

The Cholesky decomposition always exists and is unique. Furthermore, computing the Cholesky decomposition is more efficient and numerically more stable than computing the LU decomposition.

A.3 QR Decomposition

The QR decomposition of a matrix $A$ is a decomposition of the matrix into an orthogonal and a triangular matrix.

$$A = QR$$

(A.3.1)

where $Q$ is an orthogonal matrix, and $R$ is an upper triangular matrix. Analogously, we can define the QL, RQ, and LQ decompositions of $A$.

A.4 SVD

Singular value decomposition (SVD) is an important factorization of a rectangular real or complex matrix. Suppose $A$ is an $m \times n$ matrix. Then there exists a factorization of the form

$$A = U\Sigma V^*$$

(A.4.1)
where $U$ is an $m \times m$ unitary matrix, the matrix $\Sigma$ is $m \times n$ with nonnegative numbers on the diagonal and zeros off the diagonal, and $V^*$ denotes the conjugate transpose of $V$, an $n \times n$ unitary matrix. Such a factorization is called a SVD of $A$.

A common convention is to order the values $\Sigma_{ii}$ in non-increasing fashion. In this case, the diagonal matrix $\Sigma$ is uniquely determined by $M$ (though the matrices $U$ and $V$ are not). The rank of $M$ is the number of nonzero entries in $\Sigma$.

A non-negative real number $\sigma$ is a singular value for $A$ if and only if there exist unit-length vectors $u$ and $v$ such that

$$ Av = \sigma u \quad \text{and} \quad A^*u = \sigma v \quad (A.4.2) $$

The vectors $u$ and $v$ are called left-singular and right-singular vectors for $\sigma$, respectively.

The SVD can be used for computing the pseudo-inverse of a matrix,

$$ A^{-1} = V\Sigma^+U^* \quad (A.4.3) $$

where $\Sigma^+$ is the transpose of $\Sigma$ with every nonzero entry replaced by its reciprocal. The pseudo-inverse is one way to solve linear least squares problems.
Appendix B

Algorithms for Optimization

The term optimization, in numerical analysis refers to the study of problems in which to minimize or maximize a real function by systematically choosing the values of real variables from within an allowed set.

The nonlinear least squares problem is to find a local minimizer of

\[ \Phi(x) = \frac{1}{2} \sum_{i=1}^{m} \left( \frac{y_i - y_i(x)}{\sigma_i} \right)^2 = \frac{1}{2} \| f(x) \|^2 \]  

where \( x \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}^m \) is continuously differentiable, and \( \| \cdot \| \) refers to the vector norm. \( y_i \) and \( \sigma_i \) are constants independent of the choice of minimizer \( x \), and they are related to the experiment data in applications.

There are several algorithms for nonlinear least squares problems, the most well knowns are the Gauss-Newton method, Steepest-descent method and Levenberg-Marquardt algorithm.

B.1 Steepest descent method

Steepest descent or gradient descent is an optimization algorithm. To find a local minimum, it takes steps proportional to the negative of the gradient (or the approximate
gradient) of the function at the current point.

In a neighborhood of a point $x_0$, a function $\Phi(x)$ increases fastest if one goes from $x_0$ in the direction of the gradient of $\Phi$ at $x_0$, $\nabla \Phi(x_0)$. It follows that, if

$$x_1 = x_0 - \gamma \nabla \Phi(x_0)$$

for $\gamma > 0$ a small enough number, then $\Phi(x_0) > \Phi(x_1)$. Starting from a guess $x_0$ of the local minimum, one can construct the following iteration scheme

$$x_{n+1} = x_n - \gamma_n \nabla \Phi(x_n), \quad n > 0 \quad \text{(B.1.1)}$$

we have

$$\Phi(x_0) \geq \Phi(x_1) \geq \Phi(x_2) \geq \cdots,$$

so hopefully the sequence $\{x_n\}$ converges to the desired local minimum.

The step size factor $\gamma_n$ can be chosen in various ways

- From the fact that, at $x_{n+1}$, $\nabla \Phi(x_{n+1}) \perp \nabla \Phi(x_n)$, i.e. the function $\Phi(x)$ can not be reduced any more along the direction of $-\nabla \Phi(x_n)$.

- Conjugate gradient. It proceeds not down the new gradient, but in a direction that is somehow constructed to be conjugate to the old gradient, and, insofar as possible, to all previous directions traversed.

### B.2 Gauss-Newton method

In mathematics, the Gauss-Newton algorithm is used to solve nonlinear least squares problems. It is a modification of Newton’s method\(^a\) that does not use second derivatives, and is due to Carl Friedrich Gauss.

\(^a\)Also known as the Newton-Raphson method or the Newton-Fourier method. It is an efficient algorithm for finding approximations to the zeros or roots of a real valued function. It can also be used to find a minimum or maximum of such a function by finding a zero in the function’s first derivative.
B.2. GAUSS-NEWTON METHOD

For one dimension least squares problems, the minimizer $x^*$ can be found by applying Newton’s method to $\Phi(x)$. The Taylor expansion of $\Phi(x)$

$$\Phi(x + \delta) = \Phi(x) + \Phi'(x)\delta + \frac{1}{2}\Phi''(x)\delta^2 \quad (B.2.1)$$

is minimized when $\delta$ solves the linear equation $\partial \Phi(x + \delta) / \partial \delta = 0$:

$$\Phi'(x) + \Phi''(x)\delta = 0 \quad (B.2.2)$$

and $\Phi''(x)$ is positive\(^{b}\). Provided that $\Phi(x)$ is a twice-differentiable function and the initial guess $x_0$ is chosen close enough to $x^*$, the sequence $\{x_n\}$ defined by

$$x_{n+1} = x_n - \frac{\Phi'(x_n)}{\Phi''(x_n)}, \quad n \geq 0 \quad (B.2.3)$$

will converge towards $x^*$.

This iterative scheme can be generalized to multidimension problems by replacing the derivative with the gradient, $\nabla \Phi(x)$, and the reciprocal of the second derivative with the inverse of the Hessian matrix, $[H\Phi(x)]$.

$$[H\Phi(x)] = \begin{pmatrix}
\frac{\partial^2 \Phi}{\partial x_1^2} & \frac{\partial^2 \Phi}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \Phi}{\partial x_1 \partial x_n} \\
\frac{\partial^2 \Phi}{\partial x_2 \partial x_1} & \frac{\partial^2 \Phi}{\partial x_2^2} & \cdots & \frac{\partial^2 \Phi}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 \Phi}{\partial x_n \partial x_1} & \frac{\partial^2 \Phi}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \Phi}{\partial x_n^2}
\end{pmatrix} \quad (B.2.4)$$

One obtains the iterative scheme

$$x_{n+1} = x_n - [H\Phi(x_n)]^{-1}\nabla \Phi(x_n), \quad n \geq 0 \quad (B.2.5)$$

or $x_{n+1} = x_n + \delta$, where the increment is

$$\delta = -[H\Phi(x_n)]^{-1}\nabla \Phi(x_n), \quad n \geq 0 \quad (B.2.6)$$

\(^{b}\)Convex condition, i.e. $f(x + \frac{u}{2}) \leq \frac{f(x) + f(u)}{2}$. A twice differentiable function of one variable is convex on an interval if and only if its second derivative is non-negative there.
Usually Newton’s method is modified to include a small step size \( \gamma > 0 \) instead of \( \gamma = 1 \).

\[
x_{n+1} = x_n - \gamma [H \Phi(x_n)]^{-1} \nabla \Phi(x_n), \quad n \geq 0.
\] (B.2.7)

Eq. (B.2.5) and Eq. (B.2.6) can also come from the Taylor expansion of \( \Phi(x) \)

\[
\Phi(x + \delta) = \Phi(x) + \nabla \Phi(x) \cdot \delta + \frac{1}{2} \delta^T [H \Phi(x)] \cdot \delta
\] (B.2.8)

The minimizer \( \delta \) comes from

\[
\frac{\partial \Phi(x + \delta)}{\partial \delta} = 0
\] (B.2.9)

In some of the applications, such as least square problems, the expected minimum point is \( \Phi(x^*) = 0^c \), and they can solve the increment \( \delta \) directly from the first order approximation. For each \( f_i \) in \( \Phi(x) \)

\[
f_i^{n+1} = f_i^n + \sum_j \frac{\partial f_i}{\partial x_j} \delta_j
\]

and we are expecting the next step is the minimum \( f_i^{n+1} = 0 \), that is

\[-f = J \delta\]

where \( J \) is Jacobian Matrix.

\[
J = \begin{pmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\
\frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\
\frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \ldots & \frac{\partial f_m}{\partial x_n}
\end{pmatrix}
\] (B.2.10)

More commonly used forms are

\[
J^T J \delta = -J^T f
\] (B.2.11)

\(^c\)It is usually called \( \chi^2 \), but \( \chi^2 = 2\Phi(x) \), the factor 1/2 in \( \Phi(x) \) makes the second derivatives without a constant 2.
or
\[ x_{n+1} = x_n - ([Jf]^T[Jf])^{-1}[Jf]f(x_n) \] (B.2.12)

The iteration scheme Eq. B.2 is used by LOCO (Linear Optics from Closed Orbits) [31, 29], which is a MATLAB package for ORM analysis.

Since linear equation B.2 is usually an overdetermined problem (e.g., in [31], singular value decomposition (SVD) is used to invert \( \frac{\partial f}{\partial x_j} \) and solve for \( \delta \).

### B.3 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt, as we introduced in Sec. 3.3.3, is an interpolation between gradient descent method and Gauss-Newton method. The proof of convergence and the detailed scaling variation of this algorithm is introduced in the following sections.

#### B.3.1 Theoretical Basis of Levenberg-Marquardt Algorithm

Theoretical basis of algorithm

**Theorem B.3.1.** Let \( \lambda \geq 0 \) be arbitrary and let \( \delta_0 \) satisfy the equation
\[ (J^TJ + \lambda I)\delta_0 = -J^Tf \] (B.3.1)

then \( \delta_0 \) minimizes \( \Phi \) on the sphere whose radius \( \|\delta\| \) satisfies
\[ \|\delta\| = \|\delta_0\| \]

By method of Lagrange we can prove it [23]. The result gives us \( \delta \) which minimize \( \Phi \) under some constraints. It is clear that \( J^TJ + \lambda I \) is positive definite.

It is also known that, the angle between between increments from gradient descent and Gauss-Newton method is a decreasing function of \( \lambda \).
Theorem B.3.2. Let $\delta(\lambda)$ be the solution to B.3.1. Then $\|\delta(\lambda)\|^2$ is a continuous decreasing function of $\lambda$, such that $\lambda \to \infty$, $\|\delta(\lambda)\|^2 \to 0$.

Theorem B.3.3. Let $\gamma$ be the angle between $\delta_0$ and $\delta_g$. Then $\gamma$ is a continuous decreasing function of $\lambda$, such that $\lambda \to \infty$, $\gamma \to 0$. Since $\delta_g$ is independent of $\lambda$, it follows that $\delta_0$ rotates toward $\delta_g$ as $\lambda \to \infty$.

This three theorems give us the following conclusion.

1. $\lambda$ interpolates between Gauss-Newton and gradient descent methods.

2. When $\lambda = \infty$, it is gradient descent method, while $\lambda = 0$ is a Gauss-Newton method.

3. Sufficiently large $\lambda$ always makes $\Phi$ decrease.

4. At each iteration we desire to minimize $\Phi$ in the (approximately) maximum neighborhood over which the linearized function will give adequate representation of the nonlinear function.

### B.3.2 Classical Levenberg-Marquardt Algorithm

A vanilla implementation of L-M algorithm is presented here, and it follows the introduction in Ref.[28, 23].

The merit function $\chi^2$ is well approximated by a quadratic form:

$$\chi^2(a) \approx \gamma - d \cdot a + \frac{1}{2} a \cdot D \cdot a \quad (B.3.2)$$

where $d$ is an $n$-vector and $D$ is an $n \times n$ matrix. If the approximation is a good one, we know how to jump from the current trial parameters $a_{\text{cur}}$ to the minimizing ones $a_{\text{min}}$ in a single leap:

$$a_{\text{min}} = a_{\text{cur}} + D^{-1} \cdot \left[-\nabla \chi^2(a_{\text{cur}})\right] \quad (B.3.3)$$
On the other hand, B.3.2 might be a poor local approximation to the shape of the function that we are trying to minimize at \( a_{\text{cur}} \). In that case, about all we can do is take a step down the gradient, as in the steepest descent method

\[
a_{\text{min}} = a_{\text{cur}} + D^{-1} \cdot [-\nabla \chi^2(a_{\text{cur}})]
\] (B.3.4)

where the constant is small enough not to exhaust the downhill direction.

The gradient of \( \chi^2 \) with respect to the parameters \( a \), which will be zero at the \( \chi^2 \) minimum, has components

\[
\frac{\partial \chi^2}{\partial a_k} = -2 \sum_{i=1}^{N} \left[ \frac{y_i - y(x_i; a)}{\sigma_i^2} \right] \frac{\partial y(x_i; a)}{\partial a_k} \quad k = 1, 2, \ldots, M
\] (B.3.5)

\[
\frac{\partial \chi^2}{\partial a_k \partial a_\ell} = 2 \sum_{i=1}^{N} \left[ \frac{\partial y(x_i; a)}{\partial a_k} \frac{\partial y(x_i; a)}{\partial a_\ell} - \left[ y_i - y(x_i; a) \right] \frac{\partial^2 y(x_i; a)}{\partial a_\ell \partial a_k} \right] \quad k = 1, 2, \ldots, M
\] (B.3.6)

By defining

\[
\beta_k \equiv -\frac{1}{2} \frac{\partial \chi^2}{\partial a_k} \quad \alpha_{k\ell} \equiv \frac{1}{2} \frac{\partial^2 \chi^2}{\partial a_k \partial a_\ell}
\] (B.3.7)

making \([\alpha] = \frac{1}{2}D\) in equation B.3.3, in terms of which that equation can be rewritten as the set of linear equations

\[
\sum_{\ell=0}^{M-1} \alpha_{k\ell} \delta a_\ell = \beta_k
\] (B.3.8)

SVD is used to solve for the increments \( \delta a_\ell \), added to the current approximation, give the next approximation.

Equation B.3.4, the steepest descent formula, translated to

\[
\delta a_\ell = \text{constant} \times \beta_\ell
\] (B.3.9)
Levenberg and Marquardt has given a elegant method for varying smoothly be-
tween the extremes of the inverse-Hessian method B.3.3 and the steepest descent
method B.3.4. The latter method is used far from the minimum, switching continu-
ously to the former as the minimum is approached.

Marquardt suggests to use an factor $\lambda$ to control the iteration, and combine equa-
tion B.3.8 and B.3.9[28, 23]:

$$\alpha'_{jj} \equiv \alpha_{jj}(1 + \lambda) \quad (B.3.10)$$

$$\alpha'_{jk} \equiv \alpha_{jk} \quad (j \neq k) \quad (B.3.11)$$

and $\delta a_\ell$ for both cases is solved from

$$\sum_{\ell=0}^{M-1} \alpha'_{k\ell} \delta a_\ell = \beta_k \quad (B.3.12)$$

When $\lambda$ is very large, the matrix $\alpha'$ is forced into being diagonally dominant, so
equation B.3.12 goes over identical to B.3.3. On the other hand, as $\lambda \to 0$, equation
B.3.12 goes over to B.3.4

The meaning of $\lambda$, $\nu$ and the update of $\lambda$ are later published in SIAM News,
Vol. 26, Number 6, October 1993.

\begin{itemize}
  \item ...two particular features that were absent from Levenberg’s prior work. \textbf{First},
  unlike Levenberg, Marquardt did not insist on finding a local minimum of the
  cost function at each step. In this way he avoided the relatively slow convergence
  often encountered in steepest descent techniques as they work their way along
  a narrow zigzag path, crossing and recrossing the floor of the banana-shaped
  valley in the cost function surface.
  \item \textbf{Second}, and of equal importance, Marquardt implemented his method in For-
  tran and tested it ”on a large number of problems.” red His code contained a
\end{itemize}
Algorithm 2 Plain Levenberg-Marquardt

1: Compute $\chi^2(a)$
2: $\lambda = 0.001, \nu = 10$ //or a modest value.
3: Solve the linear equations Eq. (B.3.8) for $\delta a$
4: evaluate $\chi^2(a + \delta a)$
5: if $\chi^2(a + \delta a) \geq \chi^2(a)$ then
6: $\lambda \leftarrow \nu \lambda$ //or any other substantial factor
7: go back to 3
8: else
9: $\lambda \leftarrow \lambda / \nu$
10: $a \leftarrow a + \delta ay$
11: go back to 3
12: end if

particular feature, mentioned only in a long footnote in his 1963 paper, that treated cases in which the diagonal parameter had grown unreasonably large.

- "Many people initially programming the method have omitted the step described in the footnote in their computer software," Marquardt explains, "but it is red very critical. The algorithm is not as robust without it. ..."

On occasion in problems where the correlations among the parameter-estimates are extremely high ($r > 0.99$) it can happen that $\lambda$ will be increased to unreasonably large values. It has been found helpful for these instances to alter test iii The revised test is

$$b^{(r+1)} = b^{(r)} + K^{(r)} \delta^{(r)}, \quad K^{(r)} \leq 1 \quad (B.3.13)$$

Noting that the angle $\gamma^{(r)}$ is a decreasing function of $\lambda^{(r)}$, select a criterion
angle $\gamma_0 < \pi/2$ and take

$$K^{(r)} = 1 \text{ if } \gamma^{(r)} \geq \gamma_0$$

... A suitable choice for the criterion angle is $\gamma_0 = \pi/4$.

... In the presence of very high correlations, the positive definiteness of $A$ can break down due to rounding error alone.

### B.3.3 Scaled Levenberg-Marquardt Algorithm

This section is mostly from Refs. [26, 23].

We want to minimize our merit function $\chi^2$, and approach to the final minimum by a serial of increment of $x$ of our model.

In general, we could minimize $\sqrt{\chi^2} = \|f(x+p)\|$ as a function of $p$, then $x+p$ would be the desired solution.

As in plain version of Levenberg-Marquardt method, we linearize $f(a+p)$ and obtain the linear least squares problem $\sqrt{\chi^2} = \|f(a) + f'(x)p\|

This linearization is not valid for all values of $p$ as before. But within a hyper-ellipsoid, it should be true. So we consider the constrained linear least squares problem

$$\min\{\sqrt{\chi^2}(p) : \|Dp\| \leq \Delta\} \quad (B.3.14)$$

A more general one is

$$\min\{\|f + Jp\| : \|Dp\| \leq \Delta\} \quad (B.3.15)$$

The basis for the Levenberg-Marquardt method is the result that if $p^*$ is a solution to Eq. (B.3.15), then $p^* = p(\lambda)$ for some $\lambda \geq 0$ where[26, 23]

$$p(\lambda) = -(J^TJ + \lambda D^TD)^{-1}J^Tf \quad (B.3.16)$$
There are two possibilities:

1. \( \lambda = 0 \) and \( \|Dp(0)\| \leq \Delta \), in which case \( p(0) \) is the solution to Eq. (B.3.15) for which \( \|Dp\| \) is least, or

2. \( \lambda > 0 \) and \( \|Dp(0)\| = \Delta \), and then \( p(\lambda) \) is the unique solution to Eq. (B.3.15).

The above statement comes after three theorems in [23], but here we have scaling matrix \( D \).

In [26], J. J. More introduced a scaling matrix \( D \), this theorem should be revised for

\[
(J^TJ + \lambda DD^T) p(\lambda) = -J^Tf \tag{B.3.17}
\]

Fast but unreliable when \( \lambda = 0 \) and \( J \) is nearly rank deficient. \( J^TJ \) and \( DD^T \) lead to unnecessary underflows and overflows.

A better way to get \( p \) ([Osborne 1972]) is to solve the following equation with QR decomposition.

\[
\begin{pmatrix}
J \\
\sqrt{\lambda}D
\end{pmatrix}p \approx -
\begin{pmatrix}
f \\
0
\end{pmatrix} \tag{B.3.18}
\]

It’s not hard to follow the original proof by the method of Lagrange.

The choice of \( \Delta \) depends on the ratio between the actual reduction and the predicted reduction obtained by the correction.

\[
\rho(p) = \frac{\|f(x)\|^2 - \|f(x + p)\|^2}{\|f(x)\|^2 - \|f(x) + f'(x)p\|^2} \tag{B.3.19}
\]

It measures the agreement between the linear model and the (nonlinear) function.

The scheme for updating \( \Delta \) has the objective of keeping the value of B.3.19 at a reasonable level. Thus, if \( \rho(p) \) is close to unity (i.e. \( \rho(p) \geq 3/4 \)), we may want to increase \( \Delta \), but if \( \rho(p) \) is not close to unity (i.e. \( \rho(p) \leq 1/4 \)), then \( \Delta \) must be decreased. see [26] for more details.
• increase $\Delta$, multiply by a constant factor not less than one.

• decrease $\Delta$(Fletcher[1971])

The purpose of the matrix $D_k$ in the Levenberg-Marquardt algorithm is to take into account the scaling of the problem. In [26] they chose

$$D_k = \text{diag}(d_1^{(k)}, \ldots, d_n^{(k)}),$$  

(B.3.20)

where

$$d_i^{(0)} = \|\partial_i F(a_0)\|$$  

(B.3.21)

$$d_i^{(k)} = \max\{d_i^{(k-1)}, \|\partial_i F(a_k)\|\}, \quad k \geq 1$$  

(B.3.22)

This choice is over Fletcher’s[11], and please see [26] for more details.

An improved version of Levenberg-Marquardt method is widely used, and provided as numerical routine or library[13, 16]. it looked at this nonlinear fitting more carefully[26], therefore more robust and efficient,
Algorithm 3 Improved Levenberg-Marquardt

1: $\sigma \in (0, 1)$

2: if $\|D_k J_k f_k\| \leq (1 + \sigma) \Delta_k$ then

3: $\lambda_k \leftarrow 0$

4: $p_k = -J_k f_k$

5: else

6: Determine $\lambda_k > 0$ such that if

\[
\begin{pmatrix}
J_k \\
\lambda_k^{1/2} D_k
\end{pmatrix} \approx -
\begin{pmatrix}
f_k \\
0
\end{pmatrix}
\] (B.3.23)

then $(1 - \sigma) \Delta_k \leq \|D_k p_k\| \leq (1 + \sigma) \Delta_k$

7: end if

8: Compute the ratio $\rho_k$ of actual to predicted reduction.

9: if $\rho \leq 0.0001$ then

10: $x_{k+1} = x_k$ and $J_{k+1} = J_k$

11: else

12: $x_{k+1} = x_k + p_k$ and compute $J_{k+1}$

13: end if

14: if $\rho_k \leq 1/4$ then

15: Set $\Delta_{k+1} \in [\Delta_k/10, \Delta_k/2]$

16: else if $\rho_k \in (1/4, 3/4)$ and $\lambda_k = 0$, or $\rho_k \geq 3/4$ then

17: $\Delta_{k+1} = 2 \|D_k p_k\|.$

18: end if

19: Update $D_{k+1}$ by equation B.3.20 and B.3.21