ABSTRACT

DEWITT, MARTIN A. The Spectrum and Decays of Scalar Mesons in the Light-Front Quark Model. (Under the direction of Professor Chueng-Ryong Ji).

We use the light-front quark model (LFQM) to investigate the structure of the scalar mesons, mainly focusing on the three heavy isoscalar states $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$. We compute the spectrum of scalar mesons by diagonalizing a relativized, QCD-inspired model Hamiltonian written in a basis of 25 simple harmonic oscillator states. The masses are then used to perform a mixing analysis which assumes that the heavy isoscalars are mixtures of $n\bar{n} = \left(\frac{u\bar{u}+d\bar{d}}{\sqrt{2}}\right)$, $s\bar{s}$, and gg. The resulting quark–glue content is used along with the meson wave–functions determined from the spectrum to compute the decay rates for $f_0 \to \pi\pi$, $f_0 \to K\bar{K}$, and $f_0 \to \eta\eta$. We find that when the glueball contributions to the decays are ignored, the results are in poor agreement with the available data. However, when we estimate the effect of including the glueball contributions in the decays, a solution can be found which matches the data quite well. In this solution, the $f_0(1710)$ is mostly glueball (78%) while the $f_0(1500)$ and $f_0(1370)$ are dominantly mixtures of $n\bar{n}$ and $s\bar{s}$. Additionally, in this solution the glueball contributions to $K\bar{K}$ and $\eta\eta$ decays are significant while the contributions to $\pi\pi$ decay are negligible. This finding is in agreement with Chanowitz [1] who uses chiral perturbation theory arguments to show that the amplitude for a glueball to decay to a $q\bar{q}$ pair is proportional to the quark mass. This results in a suppression of the $\pi\pi$ decay channel compared to $K\bar{K}$ and $\eta\eta$.

The Spectrum and Decays of Scalar Mesons in the Light-Front Quark Model

by
Martin A. DeWitt

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fullfillment of the
requirements for the Degree of
Doctor of Philosophy

Physics

Raleigh, North Carolina

2008

Approved By:

Dr. Thomas Schaefer	Dr. Dean Lee
Dr. Chueng Ji Chair of Advisory Committee	Dr. Orlando Hankins

BIOGRAPHY

Martin DeWitt was born and raised in Winston-Salem, North Carolina. He obtained his B.S. in Aerospace Engineering from North Carolina State University in 1994 and an M.Ed. from the University of North Carolina at Greensboro in 1996. In 2008 he completed his work for a Ph.D. in Physics at North Carolina State University. Martin and his wife, Michelle, currently live in High Point, North Carolina where Martin is an instructor in the Department of Chemistry and Physical Sciences at High Point University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The assignment of the scalar($J^{PC} = 0^{++}$) $q\bar{q}$ states has long been an enigma in hadron spectroscopy. Unlike the elegant vector and tensor multiplets, it is still controversial which are the members of the expected $L = S = 1$ $q\bar{q}$ multiplet since there are now too many 0^{++} mesons observed in the region below 2 GeV for them all to be explained naturally within a $q\bar{q}$ picture [2]. For example, 2 isovector($IJ^{PC} = 1\ 0^{++}$)[$a_0(980)$, $a_0(1450)$] and 5 isoscalar($0\ 0^{++}$)[$f_0(600)$(or σ), $f_0(980)$, $f_0(1370)$, $f_0(1500)$, $f_0(1710)$] states have been reported by the Particle Data Group [2]. This has led to the suggestion that not all of them are $q\bar{q}$ states. The main reason for this situation is that around the relevant mass region there exist other structures such as $K\bar{K}$ molecules [3, 4, 5, 6, 7], glueballs [8, 9], and four-quark($qq\bar{q}\bar{q}$) systems [10, 11, 12, 13].

Interpreting the structure of each of the known scalars has proven to be a fairly controversial endeavor. Take, for example, the light scalars (*i.e.* those below 1 GeV). Due to some of the difficulties associated with $f_0(980)$ and $a_0(980)$—*e.g.* the strong couplings to $K\bar{K}$ in spite of their masses being at the $K\bar{K}$ threshold, and the large discrepancies of $\pi\pi$ [5, 6],$\gamma\gamma$ [7] and ϕ-radiative decay [14] widths between non-relativistic (NR) quark model predictions and experimental data—Weinstein and Isgur [3, 4] proposed the isospin $I = 0$ $f_0(980)$ and the $I = 1$ $a_0(980)$ within the NR potential model as the "$K\bar{K}$ molecules". However, a more conventional interpretation for these states has been given by Törnqvist and Roos [15, 16] who analyzed the data on the $f_0(980), f_0(1370), a_0(980)$ and $a_0(1450)$ as unitarized remnants of $q\bar{q}$ 1^3P_0 states with six parameters and theoretical constraints including flavor symmetry, the OZI rule, the equal-spacing rule for the bare $q\bar{q}$ states, unitarity, and analyticity. In this work, the authors concluded that the $f_0(980)$ and the

$f_0(1370)$ are two manifestations of the same $s\bar{s}$, while the $a_0(980)$ and the $a_0(1450)$ are two manifestaions of the same $u\bar{d}$ state.

The interpretation of the structures of some of the heavier scalars has been somewhat less controversial. For example, $f_0(1370)$ and $a_0(1450)$ are most commonly interpreted as $q\bar{q}$ states, even though the flavor assignments for each are still unclear. There has been some agreement on identifying $f_0(1500)$ as a glueball, possibly mixed with $n\bar{n} = (u\bar{u} + d\bar{d})/\sqrt{2}$ and $s\bar{s}$. This interpretation has followed both from lattice QCD which, in the quenched approximation, predicts that the lightest glueball has $J^{PC} = 0^{++}$ and a mass of 1.55–1.74 GeV [17, 18], as well as from the fact that $f_0(1500)$ decays strongly into $\pi\pi$ but not into $K\bar{K}$.

However, another possible interpretation for the glueball has been put forward by Chanowitz [1] who finds that the coupling of a spin zero glueball (G) to light $q\bar{q}$ pairs is chirally suppressed by a factor of m_q/m_G. This is because for $m_q = 0$, chiral symmetry requires the quark and antiquark to have equal chirality, hence unequal helicity, implying non–vanishing net angular momentum, so that the amplitude must vanish for $J = 0$ in the chiral limit. Since it is proportional to the quark mass, this suppression will be stronger for $u\bar{u}$ and $d\bar{d}$ than for $s\bar{s}$. So the glueball could have a significant contribution to $K\bar{K}$ decays, while its contribution to $\pi\pi$ would be small. The $f_0(1710)$ decays more strongly to $K\bar{K}$ than to $\pi\pi$, and this has been taken as evidence that is is mainly an $s\bar{s}$ state. However, if Chanowitz is correct one could interpret the $f_0(1710)$ as being the scalar glueball with its comparatively large $K\bar{K}$ width explained by chiral suppression. Chanowitz argues that this interpretation is also more consistent with the fact that the $f_0(1710)$ is very prominent in radiative Ψ decay. It is also interesting to note that Sexton, Vaccarino, and Weingarten have published lattice results [19] which are consistent with the predictions of Chanowitz's chiral suppression. They computed the partial decay widths for the scalar glueball to pairs of pseudoscalar mesons in the quenched approximation. They found that the coupling increased with increasing pseudoscalar meson mass, so that the decay width to $K\bar{K}$ was larger than the decay width to $\pi\pi$.

Close and Törnqvist [20] have proposed a scheme which sorts the light scalars into two distinct nonets: one nonet above 1 GeV and another below 1 GeV, with different physics operating in each. The nonet above the 1 GeV threshold is comprised of the $q\bar{q}$ states mixed with the scalar glueball. The glueball's presence is inferred from the overpopulation of isoscalars in this mass region. The nonet below 1 GeV is made up of $qq\bar{q}\bar{q}$ and meson-

meson molecules. As such, $f_0(980)$ and $a_0(980)$ can be thought of as superpositions of four-quark states and $K\bar{K}$ molecules. The authors of Ref. [20] demonstrate that such a scheme involving two scalar nonets can be described using two coupled linear sigma models.

In this thesis, we investigate Close and Törnqvist's proposal that the isoscalars $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ are mixtures of $n\bar{n} = (u\bar{u} + d\bar{d})/\sqrt{2}$, $s\bar{s}$, and gg using a nonperturbative model based on light–front dynamics (LFD). A major advantage of LFD is that the nontrivial (nonperturbative) QCD vacuum effects may be accumulated to the zero-modes of the LF Fock-components leaving the rest of the vacuum trivial. Using this advantage, one can give completely model-independent kinematic constraints to the LF helicity amplitudes as evidenced in the recent literature [21]. These model-independent constraints provide useful consequences in the phenomenology based on nonperturbative models and PQCD. The particular nonperturbative model that we apply in this work is the light–front quark model (LFQM).

This model has been used very successfully in both the pseudoscalar and vector meson systems [22, 23, 24, 25, 26, 27]. With only a few parameters, it has been used to predict pseudoscalar and vector meson spectra, starting with the up/down quarks and going all the way to the bottom quark sector. Additionally, once the trial LF wave functions were fixed from the spectroscopy, a variety of wave function–related observables such as form factors, charge radii, decay constants, radiative meson decays, two–photon decays of pseudoscalar mesons, *etc.* were computed. All of these were in very good agreement with the data. Given such remarkable success, we now extend the LFQM to investigate the scalar mesons.

The work is organized into 6 chapters. In Chapter 2 we begin with a basic introduction to the light–front formalism and present the essential features of the quark model used here.

In Chapter 3 we examine some fundamental issues in light-front dynamics that have plagued calculations in the past. Spurious divergences appear in light–front calculations that do not appear when the same calculations are done in the equal–time formulation. These divergences appear as the result of a failing of the naive method of performing the loop integrations in which one simply picks up the poles and assumes that the contribution from the arc at infinity is zero. We show that in certain cases, the integrands do not fall off sufficiently fast to guarantee that these arc contributions will vanish. As such, the arc conributions must be included in order to obtain the correct result. We fully investigate

these treacherous points so that they are understood and can be handled properly should they arise in the investigation of the scalar mesons.

In Chapter 4, we present a preliminary investigation of the scalar mesons. We construct a light–front scalar meson wave function for which the radial part is taken to be a single ground-state simple harmonic oscillator wave function. The wave function parameters are taken from previous pseudoscalar and vector meson investigations. These simple wave functions are used to compute absolute widths for the radiative decay processes $f_0 \to \gamma\gamma$, $f_0 \to \phi\gamma$, and $f_0 \to \rho\gamma$ which incorporated the effects of glueball-$q\overline{q}$ mixing. The mixed physical states are assumed to be $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ for which the flavor–glue content is taken from the mixing calculations of other works. Unfortunately, there is no reliable data for these radiative processes, so little can be concluded about the structure of these isoscalars. We also compute decay widths for the processes $\phi \to (0^{++})\gamma$ and $(0^{++}) \to \gamma\gamma$ involving the light scalars $f_0(980)$ and $a_0(980)$. We assume these to be $q\bar{q}$ states to see whether this structure alone can explain the data. The results we obtain are not consistent with well-established data, further supporting the idea that $f_0(980)$ and $a_0(980)$ are not conventional $q\bar{q}$ states.

In Chapter 5, we improve upon the preliminary work presented in Chapter 4 in a number of ways. First, we compute the meson spectrum using the relativized, QCD-inspired model Hamiltionan of Godfrey and Isgur [28]. Here, the Hamiltonian is written in a basis of 25 simple harmonic oscillator states. Second, we perform our own mixing calculation using the energies taken from the quark model spectrum. Unlike many other scalar meson mixing analyses that have been done, we do not assume glueball dominance. That is, we do not assume that the $q\bar{q}$–$q\bar{q}$ mixing is negligible compared to $q\bar{q}$–gg mixing, but leave the mixing scheme as general as possible. Third, we apply the improved wave functions from the spectrum calculation to the hadronic decays $f_0 \to \pi\pi$, $f_0 \to K\bar{K}$, and $f_0 \to \eta\eta$ for which there is more well–established data. Here, we find that the hadronic decay data can be explained for a scenario in which the $f_0(1710)$ is composed mostly of the scalar glueball, while the $f_0(1370)$ and $f_0(1500)$ are dominantly $n\bar{n}$–$s\bar{s}$ mixtures.

In Chapter 6 the results of our investigation are summarized and discussed, and a brief discussion of future work related to the structure of scalar mesons is presented.

Chapter 2

The Light-Front Quark Model

While in principle, one can parameterize space-time in a number of different ways, Dirac [29] showed that there are only three fundamentally different space-time parameterizations: the instant form, the front form, and the point form. These parameterizations are fundamental in the sense that one cannot be mapped to another by means of a Lorentz transformation [30]. Each form is characterized by a unique type of quantization hypersurface. For example, in the instant form, a system is initially defined at all points in space, (x, y, z), at some initial time, $t = t_0$. The system is then evolved forward in time by some increment Δt, such that a new description of the system is obtained at all points in space at the new time, $t = t_0 + \Delta t$. The quantization hypersurface is the surface along which $t = constant$. In this way, the instant form quantization hypersurfaces are much like the frames of a movie. In contrast, the front form uses light-cone time, $\tau = t + \frac{z}{c}$. The front form quantization hypersurface is the surface along which $\tau = constant$. This happens to be a plane tangent to the light-cone. A system is initially defined at $\tau = 0$ and is then evolved forward in in light-cone time. For the point form, the time is given by $\theta = \sqrt{t^2 - \frac{x^2+y^2+z^2}{c^2}}$. The quantization hypersurface for the point form is the surface along which $\theta = constant$, and this has the shape of a hyperboloid.

Each of the above parameterizations should yield the same physics. Therefore, the choice of which to use is somewhat arbitrary. While for most problems, the instant form which uses ordinary time seems most appropriate, there may be other problems for which this is not necessarily the case. As a simple example, consider an Earth observer who, at time $t = 0$, views light from a star that is a distance z away. The observer may ask, "Does the star exist at $t = 0$?" The answer to this question is not known since it took the light

some time to reach the Earth observer. It is clear, however, that the star must have existed at time $t = -\frac{z}{c}$. Therefore, the star does exist at the light-cone time $\tau = (-\frac{z}{c}) + \frac{z}{c} = 0$. In fact, one can consider a large group of stars, each different distances from the observer. Some may exist at $t = 0$ and some may not. However, all of these stars would exist at light-cone time $\tau = 0$. For this situation, the light-cone time, τ is more well-suited to describe the system than the instant time t [31].

It turns out that a quantum field theory defined at equal light-cone time, τ, has some advantages over a quantum field theory defined at equal time, t. This is especially true when one is attempting to solve bound-state problems. We will begin with a detailed discussion of the differences between instant form and the front form. From this point forward, the front form will be referred to as the light-front form.

2.1 Light-Front Coordinates

The transformation from instant coordinates to light-front coordinates,

$$(\tilde{x}^0, \tilde{x}^1, \tilde{x}^2, \tilde{x}^3) \to (x^+, x^-, x^1, x^2)\,, \tag{2.1}$$

is given by

$$\begin{aligned} x^+ &= \tilde{x}^0 + \tilde{x}^3 \\ x^- &= \tilde{x}^0 - \tilde{x}^3 \\ x^1 &= \tilde{x}^1 \\ x^2 &= \tilde{x}^2\,, \end{aligned} \tag{2.2}$$

where we have adopted the Lepage-Brodsky convention [32]. The transformation can be written in a more compact form

$$x^\mu = C^\mu_\nu \tilde{x}^\nu \tag{2.3}$$

where the transformation matrix, C^{μ}_{ν}, is given explicitly as

$$C^{\mu}_{\nu} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{2.4}$$

This transformation is composed of a rotation followed by a scaling. We can use the fact that the scalar product of two four-vectors should be invariant uder this transformation to derive the light-front metric tensor. The scalar product of two four-vectors, a and b, is

$$ab = (a^T)^{\mu} g_{\mu\nu} b^{\nu} = (\tilde{a}^T)^{\alpha} \tilde{g}_{\alpha\beta} \tilde{b}^{\beta} . \tag{2.5}$$

The instant four-vectors can be written as

$$\tilde{a}^{\alpha} = (C^{-1})^{\alpha}_{\mu} a^{\mu} \rightarrow (\tilde{a}^T)^{\alpha} = (a^T)^{\mu} ([C^{-1}]^T)^{\alpha}_{\mu} \tag{2.6}$$

$$\tilde{b}^{\beta} = (C^{-1})^{\beta}_{\nu} b^{\nu} . \tag{2.7}$$

Therefore, the scalar product can be written as

$$ab = (a^T)^{\mu} g_{\mu\nu} b^{\nu} = (a^T)^{\mu} \, ([C^{-1}]^T)^{\alpha}_{\mu} \, \tilde{g}_{\alpha\beta} \, (C^{-1})^{\beta}_{\nu} \, b^{\nu} \tag{2.8}$$

from which we can see that

$$g_{\mu\nu} = ([C^{-1}]^T)^{\alpha}_{\mu} \, \tilde{g}_{\alpha\beta} \, (C^{-1})^{\beta}_{\nu} . \tag{2.9}$$

Using the following form for the instant metric tensor,

$$\tilde{g}_{\mu\nu} = \tilde{g}^{\mu\nu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} , \tag{2.10}$$

we can easily show that the light-front metric tensor is given by

$$g_{\mu\nu} = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \qquad g^{\mu\nu} = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{2.11}$$

2.2 Light-Front Lorentz Transformations

Any transformation which leaves the metric tensor invariant is a Lorentz transformation. This follows from the invariance of the length of a four-vector, x^μ.

$$\begin{aligned} x'^\mu x'_\mu &= x^\alpha x_\alpha \\ x'^\mu g_{\mu\nu} x'^\nu &= x^\alpha g_{\alpha\beta} x^\beta \\ \Lambda^\mu{}_\alpha x^\alpha g_{\mu\nu} \Lambda^\nu{}_\beta x^\beta &= x^\alpha g_{\alpha\beta} x^\beta. \end{aligned} \tag{2.12}$$

This now gives that

$$g_{\alpha\beta} = \Lambda^\mu{}_\alpha g_{\mu\nu} \Lambda^\nu{}_\beta. \tag{2.13}$$

Any transformation satisfying this condition is a Lorentz transformation.

If we use the matrix notation G for the metric tensor and Λ for the Lorentz transformation then this condition is written as

$$G = \Lambda^T G \Lambda. \tag{2.14}$$

The Lorentz transformation can be written in terms of a generator, λ, as

$$\Lambda = e^{\omega\lambda}. \tag{2.15}$$

If we assume for the moment that ω is infintesimally small, we can write the infinitesimal Lorentz transformation. Keeping the first-order term only

$$\Lambda = 1 + \omega\lambda. \tag{2.16}$$

This transformation should leave the metric invariant. . .

$$\begin{aligned}(1+\omega\lambda^T)G(1+\omega\lambda) &= G \\ \lambda^T G + G\lambda &= 0 \\ \lambda = -G^{-1}\lambda^T G.& \end{aligned} \tag{2.17}$$

From this, we can find the structure of the generators of the Lorentz transformation.

2.2.1 Light-Front Generators

In its most general form, the generator of light-front Lorentz transformations can be written as

$$\lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} & \lambda_{14} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} & \lambda_{24} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} & \lambda_{34} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & \lambda_{44} \end{pmatrix}. \tag{2.18}$$

From the condition in Eq. 2.17, and the light-front metric tensors in Eq. 2.11 we find

$$\begin{aligned}\begin{pmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} & \lambda_{14} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} & \lambda_{24} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} & \lambda_{34} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & \lambda_{44} \end{pmatrix} &= -\begin{pmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} \lambda_{11} & \lambda_{21} & \lambda_{31} & \lambda_{41} \\ \lambda_{12} & \lambda_{22} & \lambda_{32} & \lambda_{42} \\ \lambda_{13} & \lambda_{23} & \lambda_{33} & \lambda_{43} \\ \lambda_{14} & \lambda_{24} & \lambda_{34} & \lambda_{44} \end{pmatrix} \\ &\quad \times \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\ &= \begin{pmatrix} -\lambda_{22} & -\lambda_{12} & 2\lambda_{32} & 2\lambda_{42} \\ -\lambda_{21} & -\lambda_{11} & 2\lambda_{31} & 2\lambda_{41} \\ \frac{1}{2}\lambda_{23} & \frac{1}{2}\lambda_{13} & -\lambda_{33} & -\lambda_{43} \\ \frac{1}{2}\lambda_{24} & \frac{1}{2}\lambda_{14} & -\lambda_{34} & -\lambda_{44} \end{pmatrix}\end{aligned} \tag{2.19}$$

So then, the generator of the most general Lorentz transformation, λ, is of the

form

$$\lambda = \begin{pmatrix} A & 0 & 2C & 2D \\ 0 & -A & 2E & 2F \\ E & C & 0 & B \\ F & D & -B & 0 \end{pmatrix}. \tag{2.20}$$

This is what I want to put

$$B^1_{LF} \equiv -i \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad B^2_{LF} \equiv -i \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$B^3_{LF} \equiv -i \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.21}$$

The six LF generators are then given as:

$$S^1_{LF} \equiv -i \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad S^2_{LF} \equiv -i \begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$S^3_{LF} \equiv -i \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \tag{2.22}$$

These are the generators of the light-front boosts and rotations written in the LF basis. They can be written in the instant form basis by using the basis transformation in equation

(**??**).

$$
\begin{aligned}
C^{-1}B^1_{LF}C &= -i\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} = K^1 - J^2 \\
C^{-1}B^2_{LF}C &= -i\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} = K^2 + J^1 \\
C^{-1}B^3_{LF}C &= -i\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = K^3
\end{aligned} \tag{2.23}
$$

And for the rotations...

$$
\begin{aligned}
C^{-1}S^1_{LF}C &= -i\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = K^1 + J^2 \\
C^{-1}S^2_{LF}C &= -i\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = K^2 - J^1 \\
C^{-1}S^3_{LF}C &= -i\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = J^3
\end{aligned} \tag{2.24}
$$

2.2.2 Light-Front Lorentz Transformations

The Lorentz transformations are derived from the generators as follows

$$\Lambda = e^{\omega\lambda} = 1 + \omega\lambda + \frac{1}{2!}(\omega\lambda)^2 + \frac{1}{3!}(\omega\lambda)^3 + ... \tag{2.25}$$

Using the notation that the ω's associated with boosts are denoted by η's ($\omega(B^i) \equiv \eta_i$) and those associated with rotations are denoted by θ's ($\omega(S^i) \equiv \theta_i$), the Lorentz transformations for pure boosts and rotations are determined to be:

$$\Lambda^{B^1}_{LF} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \eta_1^2 & 1 & 2\eta_1 & 0 \\ \eta_1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \Lambda^{B^2}_{LF} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \eta_2^2 & 1 & 0 & 2\eta_2 \\ 0 & 0 & 1 & 0 \\ \eta_2 & 0 & 0 & 1 \end{pmatrix}$$

$$\Lambda^{B^3}_{LF} = \begin{pmatrix} e^{\eta_3} & 0 & 0 & 0 \\ 0 & e^{-\eta_3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.26}$$

$$\Lambda^{S^1}_{LF} = \begin{pmatrix} 1 & \theta_1^2 & 2\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \theta_1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \Lambda^{S^2}_{LF} = \begin{pmatrix} 1 & \theta_2^2 & 0 & 2\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \theta_2 & 0 & 1 \end{pmatrix}$$

$$\Lambda^{S^3}_{LF} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta_3) & \sin(\theta_3) \\ 0 & 0 & -\sin(\theta_3) & \cos(\theta_3) \end{pmatrix} \tag{2.27}$$

The Lorentz transformations can also be transformed into the instant basis as

follows:

$$
\begin{aligned}
C^{-1}\Lambda^{B^1}_{LF}C &= \begin{pmatrix} 1+\frac{1}{2}\eta_1^2 & \eta_1 & 0 & \frac{1}{2}\eta_1^2 \\ \eta_1 & 1 & 0 & \eta_1 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2}\eta_1^2 & -\eta_1 & 0 & 1-\frac{1}{2}\eta_1^2 \end{pmatrix} \\
C^{-1}\Lambda^{B^2}_{LF}C &= \begin{pmatrix} 1+\frac{1}{2}\eta_2^2 & 0 & \eta_2 & \frac{1}{2}\eta_2^2 \\ 0 & 1 & 0 & 0 \\ \eta_2 & 0 & 1 & \eta_2 \\ -\frac{1}{2}\eta_2^2 & 0 & -\eta_2 & 1-\frac{1}{2}\eta_2^2 \end{pmatrix} \\
C^{-1}\Lambda^{B^3}_{LF}C &= \begin{pmatrix} \cosh(\eta_3) & 0 & 0 & \sinh(\eta_3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sinh(\eta_3) & 0 & 0 & \cosh(\eta_3) \end{pmatrix}
\end{aligned}
\tag{2.28}
$$

And for the rotations...

$$
\begin{aligned}
C^{-1}\Lambda^{S^1}_{LF}C &= \begin{pmatrix} 1+\frac{1}{2}\theta_1^2 & \theta_1 & 0 & -\frac{1}{2}\theta_1^2 \\ \theta_1 & 1 & 0 & -\theta_1 \\ 0 & 0 & 1 & 0 \\ \frac{1}{2}\theta_1^2 & \theta_1 & 0 & 1-\frac{1}{2}\theta_1^2 \end{pmatrix} \\
C^{-1}\Lambda^{S^2}_{LF}C &= \begin{pmatrix} 1+\frac{1}{2}\theta_2^2 & 0 & \theta_2 & -\frac{1}{2}\theta_2^2 \\ 0 & 1 & 0 & 0 \\ \theta_2 & 0 & 1 & -\theta_2 \\ \frac{1}{2}\theta_2^2 & 0 & \theta_2 & 1-\frac{1}{2}\theta_2^2 \end{pmatrix} \\
C^{-1}\Lambda^{S^3}_{LF}C &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_3) & \sin(\theta_3) & 0 \\ 0 & -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}
\tag{2.29}
$$

The action of the Lorentz transformations on a light-front four-vector are

$$\begin{aligned}
\Lambda^{B^1}_{LF}\begin{pmatrix} x^+ \\ x^- \\ x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} x^+ \\ \eta_1^2 x^+ + x^- + 2\eta_1 x^1 \\ \eta_1 x^+ + x^1 \\ x^2 \end{pmatrix} \\
\Lambda^{B^2}_{LF}\begin{pmatrix} x^+ \\ x^- \\ x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} x^+ \\ \eta_2^2 x^+ + x^- + 2\eta_2 x^2 \\ x^1 \\ \eta_2 x^+ + x^2 \end{pmatrix} \\
\Lambda^{B^3}_{LF}\begin{pmatrix} x^+ \\ x^- \\ x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} e^{\eta_3} x^+ \\ e^{-\eta_3} x^- \\ x^1 \\ x^2 \end{pmatrix}
\end{aligned} \tag{2.30}$$

And for the rotations...

$$\begin{aligned}
\Lambda^{S^1}_{LF}\begin{pmatrix} x^+ \\ x^- \\ x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} x^+ + \theta_1^2 x^- + 2\theta_1 x^1 \\ x^- \\ \theta_1 x^- + x^1 \\ x^2 \end{pmatrix} \\
\Lambda^{S^2}_{LF}\begin{pmatrix} x^+ \\ x^- \\ x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} x^+ + \theta_2^2 x^- + 2\theta_2 x^2 \\ x^- \\ x^1 \\ \theta_2 x^- + x^2 \end{pmatrix} \\
\Lambda^{S^3}_{LF}\begin{pmatrix} x^+ \\ x^- \\ x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} x^+ \\ x^- \\ \cos(\theta_3) x^1 + \sin(\theta_3) x^2 \\ -\sin(\theta_3) x^1 + \cos(\theta_3) x^2 \end{pmatrix}
\end{aligned} \tag{2.31}$$

2.3 The Model

In the instant form of dynamics, the energy–momentum dispersion relation for a massive particle takes the irrational form

$$p^0 = \sqrt{\vec{p}^{\,2} + m^2}. \tag{2.32}$$

Converting this into the light–front, one obtains a rational energy–momentum dispersion relation given by

$$p^- = \frac{\vec{p}_\perp^{\,2} + m^2}{p^+}. \tag{2.33}$$

In the LFQM, the Fock state expansion is truncated such that only the lowest Fock state is retained. The spin-orbit part of this state is treated as a free state so that the angular momentum, parity, and charge conjugation can be assigned by the Melosh transformation [33]. The radial part is assumed to be a sum of simple harmonic oscillator basis states. The radial wave functions are then fixed by diagonalizing the model Hamiltonian, the explicit form of which is given in Chapter 5.

A meson will be composed of a quark and an antiquark with four–momenta $p_q = \left(p_q^+, p_q^-, \vec{p}_{q\perp}\right)$ and $p_{\bar{q}} = \left(p_{\bar{q}}^+, p_{\bar{q}}^-, \vec{p}_{\bar{q}\perp}\right)$ respectively. The total momentum of the meson is then $P = p_q + p_{\bar{q}}$. It is particularly convenient to describe the system in terms of the Lorentz invariant variables:

$$\begin{aligned} x &\equiv \frac{p_q^+}{P^+} \\ \vec{k}_\perp &\equiv (1-x)\vec{p}_{q\perp} - x\vec{p}_{\bar{q}\perp}. \end{aligned} \tag{2.34}$$

One can easily verify these are boost invariant by applying any of the boosts in Eq. 2.2.2 to the quark and antiquark four-momenta.

The model wave function is written as

$$\Psi^{J\ J_3}_{\lambda_q\ \lambda_{\bar{q}}}(x, \vec{k}_\perp) = \sqrt{\frac{\partial k_z}{\partial x}}\phi(x, \vec{k}_\perp)\mathfrak{R}^{J\ J_3}_{\lambda_q\ \lambda_{\bar{q}}}(x, \vec{k}_\perp). \tag{2.35}$$

where $\phi(x, \vec{k}_\perp)$ is the radial wave function and $\mathfrak{R}^{J\ J_3}_{\lambda_q\ \lambda_{\bar{q}}}(x, \vec{k}_\perp)$ is the spin–orbit wave function. The term under the square–root is the Jacobian of the variable transformation from ordinary equal–time coordinates to light–front coordinates. It is absorbed into the wave function

purely as a matter of convention.

The spin–orbit wave function is uniquely determined by the Melosh transformation. This is the transformation from the ordinary equal–time static spin–orbit wave function to the light–front spin–orbit wave function, and is given by

$$\mathfrak{R}_M(x,\vec{k}_\perp,m) = \frac{m + xM_0 - i\vec{\sigma}\cdot(\hat{n}\times\hat{k})}{\sqrt{(m+xM_0)^2+\vec{k}_\perp^2}} \ . \tag{2.36}$$

Applying the Melosh transformation to obtain the spin–orbit wave function yields

$$\begin{aligned}\mathfrak{R}^{J\ J_3}_{\lambda_q\ \lambda_{\bar{q}}}(x,\vec{k}_\perp) &= \sum_{\lambda'_q\ \lambda'_{\bar{q}}} \langle\lambda_q|\mathfrak{R}^\dagger_M(x,\vec{k}_\perp,m_q)|\lambda'_q\rangle \\ &\quad\times\langle\lambda_{\bar{q}}|\mathfrak{R}^\dagger_M(1-x,-\vec{k}_\perp,m_{\bar{q}})|\lambda'_{\bar{q}}\rangle\langle\frac{1}{2}\lambda'_q\frac{1}{2}\lambda'_{\bar{q}}|JJ_3\rangle \ .\end{aligned} \tag{2.37}$$

While there are a number of schemes used to treat the meson mass, M_0, we choose the invariant meson mass scheme [34, 35, 36]. The invariant meson mass is defined as

$$M_0^2 = \frac{\vec{k}_\perp^2 + m_q^2}{x} + \frac{\vec{k}_\perp^2 + m_{\bar{q}}^2}{1-x}. \tag{2.38}$$

In terms of the invariant meson mass, the z-component of the momentum is

$$k_z = \left(x - \frac{1}{2}\right)M_0 + \frac{(m_{\bar{q}}^2 - m_q^2)}{2M_0}, \tag{2.39}$$

which yields the Jacobian of the variable transformation from $(k_z,\vec{k}_\perp) \to (x,\vec{k}_\perp)$,

$$\frac{\partial k_z}{\partial x} = \frac{M_0}{4x(1-x)}\left[1 - \frac{(m_{\bar{q}}^2 - m_q^2)^2}{M_0^4}\right] \ . \tag{2.40}$$

when differentiated with respect to x.

Chapter 3

Treacherous Points in the LF Formalism

It is well known that spurious divergences appear in light-front calculations that do not appear when these same computations are done in the standard equal-time formulation. These divergences have been regulated with various methods including the principal-value prescription, the Mandelstam-Liebbrandt prescription, cutoffs, smearing, and BPHZ-like differentiation and reintegration. In Ref. [37], an end-point singularity was seen in the (1+1)-dimensional calculation of the pseudoscalar (and scalar) elastic form factor. The form factor was computed by first writing down the one-loop, triangle Feynman diagram and then integrating over the light-front energy by the method of residues. When the plus component of the current was used, the form factor was finite and equal to the manifestly covariant result. However, when the minus component of the current was used, an end-point singularity appeared in the result. Once this singular piece was removed by hand, the remaining amplitude was equal to the manifestly covariant result. In a subsequent paper [38] it was shown that this end-point singularity could be removed by smearing the photon vertex, which introduces a cutoff parameter, Λ. The singularity was removed for finite values of lambda, as well as in the limit $\Lambda \to \infty$.

However, further investigation has shown that when such light-front calculations are handled properly, spurious divergences do not appear in the final result and there is no need for any regularization procedure. Thus far, such divergences have appeared because of the application of the naive method of performing the loop integrations in which one simply

picks up the residues, and ignores the contribution along the arc used to close the contour at infinity. In the ordinary equal-time (or instant form) Hamiltonian formalism, this procedure is safe unless one utilizes a common trick to reduce the number of denominators because the structure of the propagators is such that the integrand will always vanish sufficiently fast as the energy (k^0) goes to infinity. Consequently, the contribution along the arc at infinity vanishes and the full result is simply given by the sum of the residues. However, in the light front there are cases in which the integrand does not vanish sufficiently fast as the light-front energy (k^-) goes to infinity. Then, the arc contributions are nonzero and must be included in order to obtain the correct result. Moreover, we show in this work that the point contributions occur when the integration contour in k^- plane crosses a moving singularity. In a paper devoted to the discussion of the equivalence of light-front (LF) and covariant QED, Misra and Warawdekar [39] also spotted the pole at infinity where we use arc and point contributions. Their treatment, based on the method of Ref. Ligterink1995, completely restores equivalence at the one-loop level.We also notice similar considerations of finding counterterms in LF Hamiltonians. In the classic paper [40] where the discretized light cone quantization (DLCQ) was introduced, the self-induced inertias were found when the normal ordered Hamiltonian was considered. Later, Burkardt [41] constrained the finite part of noncovariant counterterms in effective LF Hamiltonians.

In this work, in order not to obscure the main issue, we will investigate the (1+1) dimensional calculations of both a simple vector two-point function (Section 3.1) and the pseudoscalar charge form factor (Section 3.2). Extensions to the (3+1) dimensional calculations are straightforward. In Section 3.1.1 we will show how neglecting the arc contributions at infinity leads to the appearance of an end-point singularity in the minus component of the vector two-point function. We will also show that computing these arc contributions is not an easy task. In fact, it may not be possible in general to explicitly compute these contributions. Therefore, in Section 3.1.2, we will introduce an alternative method of evaluating the integral which includes the arc contributions, but does not require that they be computed explicitly. This alternative method removes the end-point singularity and restores the equivalence between the manifestly covariant and light-front results. In Section 3.1.3 we will show how there are similarities between this alternative method and the tensor method which involves reducing an general integral to a sum of scalar n-point functions. In Section 3.2, we will show how our alternative method can be applied to the calculation of the pseudoscalar charge form factor, restoring the equivalence between the form factor obtained

from the minus component of the current and that obtained in a manifestly covariant way. Finally, we will finish with some concluding remarks in Section 3.3.

3.1 The Vector Two-Point Function

3.1.1 Origin of the End-Point Singularity

To show how the problem of the false end-point singularity arises and to discuss how it should be resolved, we will use the simple example of the vector two-point function in (1+1) dimensions,

$$V^{\mu}_{k_1(k-p)_2} = \int d^2k \frac{k^{\mu}}{[k^2 - m_1^2 + i\epsilon][(k-p)^2 - m_2^2 + i\epsilon]}. \tag{3.1}$$

This two-point function could arise, for example, in the calculation of the fermion self-energy (See Fig. 3.1). The result for this integration takes the form $V^{\mu}_{k(k-p)} = Ip^{\mu}$. In terms of the Feynman parameter, x, the manifestly covariant result for I is given by

$$I_{cov} = -i\pi \int_0^1 dx \frac{x}{x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2}. \tag{3.2}$$

To evaluate this same integral in the light-front, it is necessary to rewrite it in terms of the light-front energy ($k^- = k^0 - k^1$) and momentum ($k^+ = k^0 + k^1$), and then perform the integration in the complex k^- plane. Here, $k^2 = k^+k^-$ and so

$$V^{\mu}_{k(k-p)} = \frac{1}{2} \int dk^+ dk^- \frac{k^{\mu}}{k^+(k^+ - p^+)(k^- - k_1^-)(k^- - k_2^-)}, \tag{3.3}$$

where $k_1^- = \frac{m_1^2 - i\epsilon}{k^+}$ and $k_2^- = p^- + \frac{m_2^2 - i\epsilon}{k^+ - p^+}$. Let us first examine the plus component ($\mu = +$). Since the positions of the two poles depend on the value of k^+ we must first break the range of the k^+ integration into different regions. For $k^+ < 0$, both k_1^- and k_2^- are located in

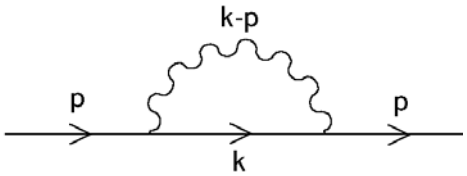

Figure 3.1: Fermion self-energy.

the upper half-plane. For $0 \leq k^+ \leq p^+$, k_1^- shifts to the lower half-plane. And finally for $k^+ > p^+$, both poles are located in the lower half-plane. For the first region ($k^+ < 0$) and the third region ($k^+ > p^+$) we can close the contour in the lower and upper half-planes respectively such that the result of these integrations is zero. This leaves only the second region ($0 \leq k^+ \leq p^+$) where, if we pick up the pole k_1^-, we find the light-front result to be,

$$I_{LF+} = -i\pi \int_0^1 dx \frac{x}{x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2}, \tag{3.4}$$

where we have used the definition, $k^+ = xp^+$. This is obviously identical to the manifestly covariant result of Eq.(3.2).

Now, for the minus component ($\mu = -$), the integration is a bit more tricky. If we naively follow the same procedure as for the plus component, we would say that the integration in the first and third regions is zero. In the second region, if we pick up the k_1^- pole and take the result to be just the residue we would obtain (again with $k^+ = xp^+$)

$$\begin{aligned} I_{LF-}^{singular} &= -\frac{i\pi}{p^2} \int_0^1 dx \frac{m_1^2}{x[x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2]} \\ &= -\frac{i\pi}{p^2} \int_0^1 dx \frac{(1-x)p^2 + m_1^2 - m_2^2}{x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2} + \frac{i\pi}{p^2} \int_0^1 \frac{dx}{x}, \end{aligned} \tag{3.5}$$

which is clearly not the same as I_{cov} and I_{LF+}. In fact, this result involves an end-point singularity at $x = 0$. In the second line above we have explicitly separated out this singular part from the rest of the integral. Had we chosen to pick up the k_2^- pole instead of the k_1^- pole, there would be an overall factor of (1-x) in the denominator and the end-point singularity would occur at x=1. However, as we mentioned before, we have not yet accounted for the contributions along the arcs at infinity. To see that these contributions should exist, examine Eq.(3.3) and note that for the minus component there are equal powers of k^- in the numerator and denominator. This means that as $k^- \to \infty$, the integrand does not fall off to zero. Consequently, there will be nonzero arc contributions in all three regions of the k^+ integration. This situation did not occur for the plus component since, in that case, there was only one factor of k^- in the numerator.

Unfortunately, it turns out that explicitly evaluating the arc contributions is, in general, not straightforward. The way in which one would usually proceed would be to

substitute $k^- = Re^{i\phi}$ into the minus component of Eq.(3.3),

$$V^-_{arc} = \lim_{R\to\infty}\left[\frac{1}{2}\int \frac{dk^+}{k^+(k^+-p^+)}\int d\phi(iRe^{i\phi})\frac{Re^{i\phi}}{(Re^{i\phi}-k_1^-)(Re^{i\phi}-k_2^-)}\right], \tag{3.6}$$

where ϕ runs from $0 \to \pi$ if the contour is closed in the upper half-plane and from $0 \to -\pi$ if it is closed in the lower half-plane. The ϕ integration is not easy to evaluate. The only straightforward way to compute this would be to neglect k_1^- and k_2^- as $R \to \infty$. The ϕ integration then becomes trivial. However, there are specific points ($k^+ = 0$ and $k^+ = p^+$) at which k_1^- and k_2^- themselves tend to infinity, and it is therefore not certain that neglecting them is safe. To see that this is indeed the case, we proceed in this straightforward way. The arc contributions for each of the three regions of k^+ are then found to be:

$$k^+ < 0 \quad : \quad I^{arc1}_{LF^-} = \frac{i\pi}{2p^2}\int_{-\infty}^{0}\frac{dx}{x(1-x)} \tag{3.7}$$

$$0 \le k^+ \le p^+ \quad : \quad I^{arc2}_{LF^-} = \frac{i\pi}{2p^2}\int_{0}^{1}\frac{dx}{x(1-x)} \tag{3.8}$$

$$k^+ > p^+ \quad : \quad I^{arc3}_{LF^-} = \frac{-i\pi}{2p^2}\int_{1}^{\infty}\frac{dx}{x(1-x)}. \tag{3.9}$$

Since $\int_{-\infty}^{\infty} + \int_{arc} = \Sigma(Residues)$, these arc contributions must be subtracted from the sum of the residues to give the final result. Conveniently, if we let $x \to (1-x)$ in Eq.(3.9) and add it to Eq.(3.7), these two terms cancel each other exactly. So the only remaining contribution is that of Eq.(3.8). This remaining contribution can be rewritten in the following way,

$$\begin{aligned} I^{arc2}_{LF^-} &= \frac{i\pi}{2p^2}\int_0^1 \frac{dx}{x(1-x)} = \frac{i\pi}{2p^2}\int_0^1 dx\left(\frac{1}{x}+\frac{1}{1-x}\right) \\ &= \frac{i\pi}{2p^2}\int_0^1 dx\left(\frac{1}{x}+\frac{1}{x}\right) = \frac{i\pi}{p^2}\int_0^1\frac{dx}{x}. \end{aligned} \tag{3.10}$$

Note that this arc contribution is exactly equal to the singular part of Eq.(3.5). Subtracting the arc contribution of Eq.(3.10) from the residue given by Eq(3.5) gives,

$$I^{arc\ method}_{LF^-} = -\frac{i\pi}{p^2}\int_0^1 dx\frac{(1-x)p^2+m_1^2-m_2^2}{x(1-x)p^2+x(m_1^2-m_2^2)-m_1^2}. \tag{3.11}$$

Evaluating this numerically we find that it is not equal to I_{cov}. So this straightforward method of computing the arc contributions does not restore the equivalence between the light-front and manifestly covariant calculations. It evidently misses contributions from the

two points $k^+ = 0$ and $k^+ = p^+$ which are needed to restore this equivalence. To include the contributions from these points, we have devised an alternative method of evaluating this integral which, by coincidence, turns out not to require explicit calculation of the arc contributions.

3.1.2 An Alternative Method

The inablilty of the usual method of evaluating light-front integrals (*i.e.* first integrating in the complex k^- plane and then integrating over k^+) to account for the contributions at single points is not necessarily restricted to cases in which arc contributions exist. For example, consider the simple integral,

$$S = \int d^2k \frac{1}{(k^2 - m^2 + i\epsilon)^2}. \tag{3.12}$$

This integral is obviously finite and, when evaluated in a manifestly covariant way, immediately yields $\frac{i\pi}{m^2}$. However, when we write this integral in light-front coordinates,

$$\begin{aligned} S &= \frac{1}{2}\int dk^+ dk^- \frac{1}{(k^+k^- - m^2 + i\epsilon)^2} \\ &= \frac{1}{2}\int dk^+ dk^- \frac{1}{k^+k^+[k^- - (\frac{m^2 - i\epsilon}{k^+})]^2}, \end{aligned} \tag{3.13}$$

we see that there is a double pole at $\frac{m^2 - i\epsilon}{k^+}$. Therefore, when integrating by the usual method, we can close the contour in the opposite half-plane and we obtain zero. Moreover, the integrand falls off to zero as $k^- \to \infty$, so there are no arc contributions. This method of evaluating the light-front integral is clearly unable to reproduce the simple result obtained in the manifestly covariant calculation. To see where the problem arises one must appeal to a different method of evaluation. If, for example, we evaluate the k^- integral on a finite

interval and then take the limit as the interval goes to infinity, we obtain

$$\begin{aligned} S &= \frac{1}{2}\lim_{R\to\infty}\int_{-\infty}^{\infty}dk^+\int_{-R}^{R}dk^-\frac{1}{(k^+k^- - m^2+i\epsilon)^2} \\ &= \frac{1}{2}\lim_{R\to\infty}\int_{-\infty}^{\infty}dk^+\left(\frac{-1}{k^+}\right)\left[\frac{1}{Rk^+ - m^2+i\epsilon}+\frac{1}{Rk^+ + m^2 - i\epsilon}\right] \\ &= -\lim_{R\to\infty}\frac{1}{R}\int_{-\infty}^{\infty}dk^+\frac{1}{[k^+ - (\frac{m^2-i\epsilon}{R})][k^+ - (\frac{-m^2+i\epsilon}{R})]} \\ &= -\lim_{R\to\infty}\frac{1}{R}\left[\frac{-2\pi i}{\frac{m^2-i\epsilon}{R} - \frac{-m^2+i\epsilon}{R}}\right] \\ &= \frac{i\pi}{m^2}. \end{aligned} \tag{3.14}$$

If one examines the integrand in the second line, it is clear that as $R\to\infty$ the function is essentially zero everywhere except near $k^+ = 0$. Therefore the entire value of this integral comes from the region where $k^-\to\infty$ as $k^+\to 0$. The usual way of evaluating the light-front integral fails to see the contribution at this point and therefore fails to reproduce the manifestly covariant result.

This is the same problem we are faced with when trying to evaluate the vector two-point function. As $k^-\to\infty$, there are contributions that come from the points where $k^+\to 0$ and where $k^+\to p^+$. The usual method of evaluating the integral fails to see these points. In order that these contributions be included, we propose an alternative solution method. The method involves first determining the form of the singular pieces of the integral. Next, these pieces are subtracted from the original integral which renders it regular. Finally, we compute these singular pieces explicitly using light-front cylindrical coordinates and add the result back to the now regular integral.

To illustrate in detail how to compute I_{LF-} using this alternative method, we will now return to the minus component of the vector two-point function,

$$V^-_{k_1(k-p)_2} = \frac{1}{2}\int dk^+dk^-\frac{k^-}{[k^+k^- - m_1^2+i\epsilon][(k^+-p^+)(k^- - p^-) - m_2^2+i\epsilon]}. \tag{3.15}$$

Note that as $k^-\to\infty$ there are linear divergences as $k^+\to 0$ and as $k^+\to p^+$. Let us

isolate the singular pieces in each of these two cases. First, as $k^- \to \infty$ and $k^+ \to 0$,

$$\begin{aligned} V^-_{k_1(k-p)_2} &\to \frac{1}{2}\int dk^+dk^- \frac{k^-}{[k^+k^- - m_1^2 + i\epsilon][-p^+k^-]} \\ &= -\frac{1}{2p^2}\int dk^+dk^- \frac{p^-}{[k^+k^- - m_1^2 + i\epsilon]} \equiv V_1^-. \end{aligned} \tag{3.16}$$

Secondly, in the case where $k^- \to \infty$ and $k^+ \to p^+$,

$$\begin{aligned} V^-_{k_1(k-p)_2} &\to \frac{1}{2}\int dk^+dk^- \frac{k^-}{[p^+k^-][(k^+ - p^+)(k^- - p^-) - m_2^2 + i\epsilon]} \\ &= \frac{1}{2p^2}\int dk^+dk^- \frac{p^-}{[(k^+ - p^+)(k^- - p^-) - m_2^2 + i\epsilon]} \equiv V_2^-. \end{aligned} \tag{3.17}$$

Now, if we define $D_1 = [k^+k^- - m_1^2 + i\epsilon]$ and $D_2 = [(k^+ - p^+)(k^- - p^-) - m_2^2 + i\epsilon]$, then subtracting the two singular pieces from the vector two-point function and subsequently adding them back gives,

$$\begin{aligned} V^-_{k_1(k-p)_2} &= \frac{1}{2}\int dk^+dk^- \frac{k^-}{D_1D_2} - V_1^- - V_2^- + V_1^- + V_2^- \\ &= \frac{p^-}{2p^2}\int dk^+dk^- \left[\frac{p^+k^-}{D_1D_2} + \frac{1}{D_1} - \frac{1}{D_2}\right] + V_1^- + V_2^- \\ &= \frac{p^-}{2p^2}\int dk^+dk^- \left[\frac{p^+k^- + D_2 - D_1}{D_1D_2}\right] + V_1^- + V_2^- \\ &= \frac{p^-}{2p^2}\int dk^+dk^- \left[\frac{-k^+p^- + p^2 + m_1^2 - m_2^2}{D_1D_2}\right] + V_1^- + V_2^-. \end{aligned} \tag{3.18}$$

Notice that the term in square brackets has now been turned into an expression that, when integrated, no longer involves any arc contributions. The term with k^- has been cancelled and replaced with a k^+ term that falls off as $k^- \to \infty$. Performing the pole integration for this gives,

$$V^-_{k_1(k-p)_2} = \frac{-i\pi p^-}{p^2}\int_0^1 dx \frac{(1-x)p^2 + m_1^2 - m_2^2}{x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2} + V_1^- + V_2^-. \tag{3.19}$$

Now, to evaluate the two singular pieces, V_1^- and V_2^-, the integrals are written in light-front cylindrical coordinates: $k^+ = R\cos\phi$ and $k^- = R\sin\phi$. In this way V_1^-, for

example, becomes

$$\begin{aligned} V_1^- &= -\frac{p^-}{2p^2}\int dk^+dk^-\frac{1}{k^+k^- - m_1^2} \\ &= -\frac{p^-}{2p^2}\lim_{R\to\infty}\int_0^R rdr\int_0^{2\pi}d\phi\frac{1}{r^2\cos\phi\sin\phi - m_1^2} \\ &\approx -\frac{p^-}{2p^2}\bigg[-\pi(\pi+4i\ln(R))+2\pi i\ln(m_1^2)+O\bigg(\frac{1}{R}\bigg)\bigg]. \end{aligned} \tag{3.20}$$

A similar expression can be found for V_2^- which differs by an overall minus sign, and in which m_1 is replaced with m_2. Adding V_1^- and V_2^-, the constants and R-dependent terms cancel giving

$$V_1^- + V_2^- = -\frac{i\pi p^-}{p^2}\ln\bigg(\frac{m_1^2}{m_2^2}\bigg). \tag{3.21}$$

So substituting Eq.(3.21) into Eq.(3.19) and dividing by p^- we obtain

$$I_{LF-}^{alt\,method} = -\frac{i\pi}{p^2}\int_0^1 dx\frac{(1-x)p^2+m_1^2-m_2^2}{[x(1-x)p^2+x(m_1^2-m_2^2)-m_1^2]}+\frac{i\pi}{p^2}\ln\bigg(\frac{m_1^2}{m_2^2}\bigg). \tag{3.22}$$

When this result is integrated numerically, it is shown to be identical to the manifestly covariant result of Eq.(3.2). Note that the first term of this result is identical to Eq.(3.11) which was found using the straightforward method of evaluating the arc contributions. The second term in the above result gives the contributions coming from the points $k^+ = 0$ and $k^+ = p^+$. Therefore, our alternative method determines all of the contributions that were missed by the usual method of evaluating light-front integrals, and which are needed to demonstrate the equivalence between the light-front and manifestly covariant results.

Although the straightforward method of explicitly evaluating the arc contributions fails in general, there is a special case in which it can be shown to work. If we examine the case where the propagator masses are equal ($m_1 = m_2 \equiv m$), then the result for I_{LF-} found by computing the arc contributions explicitly, Eq.(3.11), reduces to

$$I_{LF-}^{arc\,method} = -\frac{i\pi}{2}\int_0^1 dx\frac{1}{[x(1-x)p^2-m^2]}. \tag{3.23}$$

Likewise, the result found by our alternative method, Eq.(3.22), reduces to

$$I_{LF-}^{alt\,method} = -i\pi\int_0^1 dx\frac{1-x}{[x(1-x)p^2-m^2]}. \tag{3.24}$$

Using the fact that if λ is a function of x that is invariant under $x \to (1-x)$, then $\int_0^1 dx \frac{1-x}{\lambda} = \frac{1}{2}\int_0^1 dx \frac{1}{\lambda}$, one can see that Eq.(3.23) is identical to Eq.(3.24). In this special case, the contributions coming from the two points $k^+ = 0$ and $k^+ = p^+$ cancel each other out exactly, and it is therefore safe to ignore them when computing the arc contributions explicitly.

3.1.3 Similarities with the Tensor Method

We now wish to show how the method introduced in Sec. 3.1.2 is very similar to the tensor method which reduces a given loop integral to a sum of scalar n-point functions. The tensor method is applied to the vector two-point function in the following way.

$$\int d^2k \frac{k^\mu}{[k^2 - m_1^2 + i\epsilon][(k-p)^2 - m_2^2 + i\epsilon]} = Ip^\mu. \tag{3.25}$$

Contracting both sides with p_μ and solving for I yields,

$$I = \frac{1}{p^2}\int d^2k \frac{k\cdot p}{[k^2 - m_1^2 + i\epsilon][(k-p)^2 - m_2^2 + i\epsilon]}. \tag{3.26}$$

Defining the denominators $D_1 = [k^2 - m_1^2 + i\epsilon]$ and $D_2 = [(k-p)^2 - m_2^2 + i\epsilon]$, we can note that $(k\cdot p) = (D_1 - D_2 + p^2 + m_1^2 - m_2^2)/2$. Substituting this into Eq.(3.26) yields,

$$I = \frac{1}{2p^2}\int d^2k \left[\frac{1}{D_2} - \frac{1}{D_1} + \frac{p^2 + m_1^2 - m_2^2}{D_1 D_2}\right]. \tag{3.27}$$

The expression for I has now been reduced from a vector two-point function to a sum of scalar one-point and two-point functions. It can be evaluated either in a manifestly covariant way or in the light-front. The manifestly covariant result is given by,

$$I_{cov} = -\frac{i\pi}{2p^2}\left[\int_0^1 dx \frac{p^2 + m_1^2 - m_2^2}{x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2} + \ln\left(\frac{m_1^2}{m_2^2}\right)\right], \tag{3.28}$$

which can numerically be shown to be equivalent to Eq.(3.2).

To evaluate I in the light-front, we begin by comparing Eqs.(3.27) and (3.18). In Eq.(3.18) we artificially separated out the singular parts of the integrand. But note how in Eq.(3.27) the tensor method naturally separates out the singular terms. The two singular pieces in Eq.(3.18), V_1^- and V_2^-, correspond to the two scalar one-point functions

in Eq.(3.27) in the following way,

$$\begin{aligned}\int d^2k \frac{1}{D_1} &= -\frac{p^2}{p^-}V_1^- \\ \int d^2k \frac{1}{D_2} &= \frac{p^2}{p^-}V_2^-. \end{aligned} \tag{3.29}$$

Taking the difference between these two scalar one-point functions gives

$$\begin{aligned}\int d^2k \left[\frac{1}{D_2} - \frac{1}{D_1}\right] &= \frac{p^2}{p^-}(V_1^- + V_2^-) \\ &= -i\pi \ln\left(\frac{m_1^2}{m_2^2}\right). \end{aligned} \tag{3.30}$$

As for the scalar two-point function in Eq.(3.27), since there are no possible arc contributions the pole integration is straightforward and gives

$$\begin{aligned}\int d^2k \frac{1}{D_1 D_2} &= \frac{1}{2}\int \frac{dk^+dk^-}{k^+(k^+ - p^+)[k^- - k_1^-][k^- - k_2^-]} \\ &= -i\pi \int_0^1 dx \frac{1}{x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2}. \end{aligned} \tag{3.31}$$

Substituting the light-front results (3.30) and (3.31) into Eq.(3.27) gives

$$I_{LF} = -\frac{i\pi}{2p^2}\left[\int_0^1 dx \frac{p^2 + m_1^2 - m_2^2}{x(1-x)p^2 + x(m_1^2 - m_2^2) - m_1^2} + \ln\left(\frac{m_1^2}{m_2^2}\right)\right], \tag{3.32}$$

which turns out to be identical to the manifestly covariant result in Eq.(3.28).

The tensor method provides a natural way in which to separate the singular pieces from the integral being considered. Also, it seems to guarantee that, at least in (1+1) dimensions, any integral that is finite when computed in a manifestly covariant way will be free of spurious divergences when computed in the light-front. This can be seen from the fact that the tensor method reduces an integral to a sum of scalar n-point functions. In (1+1) dimensions, power counting tells us that scalar two-point functions and higher are all free of potential spurious divergences. This leaves just the scalar one-point functions to be concerned with. Even when evaluated in a manifestly covariant way, the scalar one-point functions are divergent. However, if the full manifestly covariant result is finite, then this means that the scalar one-point functions must come in pairs such that the divergent parts cancel each other out. Since, as we have shown in Sec. 3.1.2, the scalar one-point functions

have a similar structure in the light-front, it is reasonable to conclude that the divergent parts of these will always cancel as well. Therefore, in general, no spurious divergences of the type that we have seen so far should appear in a light-front calculation.

3.2 Pseudoscalar Charge Form Factor

We now intend to apply the lessons we have learned from investigating the vector two-point function to the calculation of the pseudoscalar charge form factor. The current corresponding to the diagram of Fig. 3.2 is given by

$$\langle p'|J^\mu|p\rangle = \frac{4N}{(2\pi)^2}\int d^2k \frac{k^\mu(-k^2+2m_1m_2-m_2^2+p\cdot p')+p^\mu(k^2-k\cdot p'-m_1m_2)+p'^\mu(k^2-k\cdot p-m_1m_2)}{[k^2-m_1^2+i\epsilon][(k-p)^2-m_2^2+i\epsilon][(k-p')^2-m_2^2+i\epsilon]} \tag{3.33}$$

where N is a normalization constant. If we define the form factor by $\langle p'|J^\mu|p\rangle = i(p^\mu + p'^\mu)F(q^2)$, then in terms of the Feynman parameters x and y the manifestly covariant result is given by

$$\begin{aligned} F_{cov}(q^2) &= \frac{N}{\pi}\int_0^1 dx \int_0^{1-x} dy \\ &\quad \times \frac{[y+2x(1-x-y)](m_2^2-m_1^2)+(1-x)(m_1^2+m_1m_2)-x[(x+y-1)^2M^2-xyq^2+m_1m_2]}{[(x+y)(x+y-1)M^2-xyq^2+(1-x-y)m_1^2+(x+y)m_2^2]^2}, \end{aligned} \tag{3.34}$$

where we have used $p^2 = p'^2 = M^2$. We can now examine the plus and minus components of the current separately. Just as in the case of the vector two-point function, the plus component of the current contains no arc contributions and easily gives the same result as

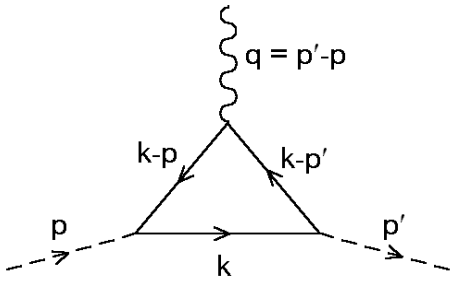

Figure 3.2: Pseudoscalar charge form factor ($p^2 = p'^2 = M^2$).

Eq.(3.34). However, the minus component involves an integrand that does not vanish as $k^- \to \infty$. This minus component of the cuurent can be written as

$$\begin{aligned}\langle p'|J^-|p\rangle &= \frac{N}{2\pi^2}\int dk^+dk^- \\ &\times \frac{-k^+k^-k^- - p^-p'^-k^+ + (2m_1m_2 - m_2^2)k^- + (k^+k^- - m_1m_2)(p^- + p'^-)}{[k^+k^- - m_1^2 + i\epsilon][(k^+ - p^+)(k^- - p^-) - m_2^2 + i\epsilon][(k^+ - p'^+)(k^- - p'^-) - m_2^2 + i\epsilon]}.\end{aligned} \tag{3.35}$$

If we evaluate this naively by taking simply the sum of the residues and neglecting the arc contributions, we obtain

$$\begin{aligned}F_{LF^-}^{residues}(q^2) = \frac{N}{\pi}\left(\frac{1+\alpha}{2+\alpha}\right)\Bigg[&\int_0^1 dx \frac{(1-x)^2M^2 + (1+\alpha)(m_1-m_2)^2(\frac{m_1^2}{M^2}) + (2+\alpha)(1-x)(m_1m_2 - m_1^2)}{[x(1-x)M^2 + (1-x)(m_1^2 - m_2^2) - m_1^2][(1-x)(\alpha+x)M^2 + (1-x)(1+\alpha)(m_1^2 - m_2^2) - (1+\alpha)^2m_1^2]} \\ &- \int_0^\alpha dx \frac{x(1+x)(\alpha-x)M^2 - (1+\alpha)^2(m_1-m_2)^2(\frac{m_2^2}{M^2}) - (1+\alpha)(\alpha-x)[x(m_1^2-m_2^2) + \alpha(m_2^2 - m_1m_2)] + (1+\alpha)^2(1+\alpha-x)m_2^2}{\alpha[(1+x)(\alpha-x)M^2 + (1+x)(1+\alpha)(m_1^2-m_2^2) - (1+\alpha)^2m_1^2][x(\alpha-x)M^2 + (1+\alpha)m_2^2]} \\ &+ \frac{1}{\alpha M^2}\int_0^\alpha dx \frac{1}{\alpha - x}\Bigg],\end{aligned} \tag{3.36}$$

which contains an end-point singularity at $x = \alpha$. We have explicitly separated the end-point singularity from the rest of the integral. We now know that this end-point singularity arises because we have neglected to include the contributions along the arcs at infinity. Looking at Eq.(3.35), power counting in k^- tells us that the first term in the numerator is the only one that will have this arc contribution. Writing this "bad" term by itself,

$$\begin{aligned}\langle J^-\rangle_{bad} &= \frac{N}{2\pi^2}\int dk^+dk^- \\ &\times \frac{-k^+k^-k^-}{[k^+k^- - m_1^2 + i\epsilon][(k^+ - p^+)(k^- - p^-) - m_2^2 + i\epsilon][(k^+ - p'^+)(k^- - p'^-) - m_2^2 + i\epsilon]}.\end{aligned} \tag{3.37}$$

To be safe, we cannot simply calculate the arc contributions in a straightforward way. Instead, we will use our alternative method. Note that as $k^- \to \infty$, there are potential linear divergences as $k^+ \to 0$, $k^+ \to p^+$, and $k^+ \to p'^+$. We will isolate these one at a time, beginning with the case where $k^+ \to 0$ while $k^- \to \infty$,

$$\begin{aligned}\langle J^-\rangle_{bad} &\to \frac{N}{2\pi^2}\int dk^+dk^- \frac{-k^+k^-k^-}{[k^+k^- - m_1^2 + i\epsilon][-p^+k^-][-p'^+k^-]} \\ &= 0.\end{aligned} \tag{3.38}$$

So it turns out that the potentially singular term vanishes in this case. Now, for the case in which $k^+ \to p^+$ as $k^- \to \infty$,

$$\begin{aligned}\langle J^-\rangle_{bad} &\to \frac{N}{2\pi^2}\int dk^+dk^- \frac{-p^+k^-k^-}{[p^+k^-][(k^+-p^+)(k^--p^-)-m_2^2+i\epsilon][(p^+-p'^+)k^-]} \\ &= \frac{N}{2\pi^2}\int dk^+dk^- \frac{1}{q^+[(k^+-p^+)(k^--p^-)-m_2^2+i\epsilon]} \equiv \langle J^-\rangle_2.\end{aligned} \tag{3.39}$$

And finally, for the case in which $k^+ \to p'^+$ as $k^- \to \infty$,

$$\begin{aligned}\langle J^-\rangle_{bad} &\to \frac{N}{2\pi^2}\int dk^+dk^- \frac{-p'^+k^-k^-}{[p'^+k^-][(p'^+-p^+)k^-][(k^+-p'^+)(k^--p'^-)-m_2^2+i\epsilon]} \\ &= \frac{N}{2\pi^2}\int dk^+dk^- \frac{-1}{q^+[(k^+-p'^+)(k^--p'^-)-m_2^2+i\epsilon]} \equiv \langle J^-\rangle_3.\end{aligned} \tag{3.40}$$

If we define $D_1 = [k^+k^- - m_1^2 + i\epsilon]$, $D_2 = [(k^+ - p^+)(k^- - p^-) - m_2^2 + i\epsilon]$, and $D_3 = [(k^+ - p'^+)(k^- - p'^-) - m_2^2 + i\epsilon]$, then subtracting Eqs.(3.39) and (3.40) from Eq.(3.37) and subsequently adding them back gives

$$\begin{aligned}\langle J^-\rangle_{bad} &= \langle J^-\rangle_{bad} - \langle J^-\rangle_2 - \langle J^-\rangle_3 + \langle J^-\rangle_2 + \langle J^-\rangle_3 \\ &= \frac{N}{2\pi^2}\int dk^+dk^- \left[\frac{-k^+k^-k^-}{D_1D_2D_3} - \frac{1}{q^+D_2} + \frac{1}{q^+D_3}\right] + \langle J^-\rangle_2 + \langle J^-\rangle_3 \\ &= \frac{N}{2\pi^2}\int dk^+dk^- \left[\frac{-q^-k^+k^+k^- + m_1^2k^+q^- - m_1^2k^-q^+}{q^+D_1D_2D_3}\right] + \langle J^-\rangle_2 + \langle J^-\rangle_3.\end{aligned} \tag{3.41}$$

Note that the first term of the last line is now free of any arc contributions. Also note that $\langle J^-\rangle_2$ and $\langle J^-\rangle_3$ both have the form of Eq.(3.20), and they differ only by an overall sign. Since these two terms have the same mass term, $\ln(m_2^2)$, they end up completely cancelling each other out when added together. Therefore, substituting Eq.(3.41) back into Eq.(3.35) gives

$$\begin{aligned}\langle p'|J^-|p\rangle &= \frac{N}{2\pi^2}\int dk^+dk^- \\ &\quad \times \frac{[-k^+k^+(\frac{q^-}{q^+}) - (m_1-m_2)^2 + k^+(p^-+p'^-)]k^- + [m_1^2(\frac{q^-}{q^+}) - p^-p'^-]k^+ - m_1m_2(p^-+p'^-)}{[k^+k^- - m_1^2+i\epsilon][(k^+-p^+)(k^--p^-)-m_2^2+i\epsilon][(k^+-p'^+)(k^--p'^-)-m_2^2+i\epsilon]}.\end{aligned} \tag{3.42}$$

The pole integration can now be carried out without any complications, giving for the form factor

$$F_{LF-}(q^2)=\frac{N}{\pi}\left(\frac{1+\alpha}{2+\alpha}\right)\left[\int_0^1 dx\frac{(1-x)^2M^2+(1+\alpha)(m_1-m_2)^2(\frac{m_1^2}{M^2})+(2+\alpha)(1-x)(m_1m_2-m_1^2)}{[x(1-x)M^2+(1-x)(m_1^2-m_2^2)-m_1^2][(1-x)(\alpha+x)M^2+(1-x)(1+\alpha)(m_1^2-m_2^2)-(1+\alpha)^2m_1^2]}\right.$$
$$\left.-\int_0^\alpha dx\frac{x(1+x)(\alpha-x)M^2-(1+\alpha)^2(m_1-m_2)^2(\frac{m_2^2}{M^2})-(1+\alpha)(\alpha-x)[x(m_1^2-m_2^2)+\alpha(m_2^2-m_1m_2)]+(1+\alpha)^2(1+\alpha-x)m_2^2}{\alpha[(1+x)(\alpha-x)M^2+(1+x)(1+\alpha)(m_1^2-m_2^2)-(1+\alpha)^2m_1^2][x(\alpha-x)M^2+(1+\alpha)m_2^2]}\right]. \tag{3.43}$$

When evaluated numerically, this light-front result for the form factor is identical to the manifestly covariant result of Eq.(3.34).

In retrospect, we can now see that this form factor calculation could have been done by explicitly computing the arc contributions in the straightforward way described in Sec. 3.1.1. This is because the contributions coming from the two points $k^+ = p^+$ and $k^+ = p'^+$—Eqs.(3.39) and (3.40) respectively—cancel each other out exactly. To see that this is in fact the case, we have explicitly computed the arc contribution to the form factor in the straightforward way and obtained,

$$F^{arc}_{LF-}(q^2) = \frac{N}{\pi}\left(\frac{1+\alpha}{2+\alpha}\right)\left[\frac{1}{\alpha M^2}\int_0^\alpha dx\frac{1}{\alpha-x}\right]. \tag{3.44}$$

Note that this is equivalent to the singular part in the sum of the residues given by Eq.(3.36). When this arc contribution is subtracted from the sum of the residues, the part that is left over is identical to the form factor determined by our alternative method, Eq.(3.43).

3.3 Summary and Concluding Remarks

In summary, we have demonstrated that the spurious divergences that have so far been seen in light-front calculations are a result of neglecting the arc contributions at infinity. Once these contributions are accounted for, the spurious divergences disappear and the equivalence between the light-front and manifestly covariant results is restored. However, computing the arc contributions explicitly is not necessarily an easy task. We have shown that, in general, a simple straightforward computation of these arc contributions can miss other important contributions coming from specific points in the range of integration. Therefore, we have developed an alternative method which ensures that all of these relevant contributions will be accounted for. In certain specific cases where the contributions coming

from single points happen to cancel each other out exactly, a straightforward calculation of the arc contributions is all that is needed to remove the end-point singularities and restore equivalence.

In summary, the final answer is obtained in two steps: First the result of the naive application of the residues theorem is corrected by the arc contributions, and second the naive arc contributions are corrected by the effects of the moving poles, i.e. the point contributions. The need for the inclusion of of self-induced inertias discussed in normal ordered LF Hamiltonians [40] corresponds to our first step of finding the need to include the arc contribution. Burkardts finding Δm^2_{kin} [41] as a finite contribution consistent with the parity requirement corresponds to our second step of finding a finite point contribution. The finite point contribution we find has a logarithmic dependence on the mass, similar to Burkardts Δm^2_{kin}.Thus, it is conceivable that our findings may point to some deficiency in the light-front perturbation theory (LFPT) rules.

It is interesting to note that if we use LFPT rules to compute the pseudoscalar charge form factor, the same end-point singularity that we have seen arise in other calculations is manifested. While it is possible to use some type of perscription to regularize this divergence, we have shown in this paper that such a divergence does not really exist in the first place. Therefore, the same deficiencies that plague other methods of calculation are also present in LFPT. We believe that there must be some way to modify LFPT so as to include the contributions from the arcs at infinity.

Due to the rational energy-momentum dispersion relation, light–front dynamics has distinguished features compared to the other forms of Hamiltonian dynamics. In particular, vacuum fluctuations are suppressed and there is a greater number of kinematic generators. Overall, these distinguished features can be regarded as advantageous in hadron phenomenology. However, in return, LFD harbors treacherous points as we presented just one example of them in this work. Thus, careful investigations of these points and judicious ways of handling them should be a priority in order for LFD to be distinctively useful compared to other forms of Hamiltonian dynamics.

Chapter 4

Preliminary Investigation of Scalar Mesons

4.1 Introduction

In this work, we attempt to shed light on the structures of the scalar mesons by investigating their radiative decays with the LFQM. Extending the LFQM to include scalars mainly involves the construction of a new $^3P_0$ light–front model wave function which we discuss in detail in Section 4.2. Here, this scalar wave function is used to study the radiative decays of the heavy scalars $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$, as well as the light scalars $a_0(980)$ and $f_0(980)$. In the case of the heavy scalars, we adopt the scheme of Close and Törnqvist in which $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ are considered to be mixtures of $n\bar{n}$, $s\bar{s}$, and gg. The flavor–glue content of each state is taken from mixing analyses done by Lee and Weingarten [42] and by Close and Kirk [43]. Taken together, these works provide mixing amplitudes for three distinct cases of the scalar glueball mass: (1) a heavy glueball ($M_{n\bar{n}} < M_{s\bar{s}} < M_{gg}$), (2) a medium weight glueball ($M_{n\bar{n}} \lesssim M_{gg} < M_{s\bar{s}}$), and (3) a light glueball ($M_{gg} < M_{n\bar{n}} < M_{s\bar{s}}$). The details of these three mixing scenarios are outlined in Section 4.3. In the case of the light scalars, $a_0(980)$ and $f_0(980)$ are assumed to be conventional $q\bar{q}$ states. The flavor content of $a_0(980)$ is then $(u\bar{u} - d\bar{d})/\sqrt{2}$, and that of $f_0(980)$ is some superposition of $n\bar{n}$ and $s\bar{s}$. Rather than attempting to determine the degree of mixing for $f_0(980)$, it suffices to examine the two ideally mixed cases: $f_0(980) = n\bar{n}$ and $f_0(980) = s\bar{s}$.

In Section 4.4, the general forms of the Q^2–dependent transition form factors for the processes $(1^{--}) \to (0^{++})\gamma^*$, $(0^{++}) \to (1^{--})\gamma^*$, and $(0^{++}) \to \gamma\gamma^*$ are derived. In the limit as $Q^2 \to 0$, these form factors yield the decay constants for the real photon processes which can then be used to compute the corresponding decay widths. In Section 4.5, we present our numerical results for the heavy scalars. This includes the form factors and decay widths for the specific processes $f_0 \to \phi\gamma$, $f_0 \to \rho\gamma$, and $f_0 \to \gamma\gamma$. Our results for the light scalars involved in the processes $\phi \to f_0(a_0)\gamma$ and $f_0(a_0) \to \gamma\gamma$ are also given. A summary of the work's salient points and a brief discussion is given in Section 4.6. In Section 6, the explicit form of the trace used in section 4.4 is presented.

4.2 The Model Wave Functions

One of the popular quark models in the light-front formalism is the invariant meson mass(IM) scheme[36, 35, 34] in which the invariant meson mass square M_0^2 is given by

$$M_0^2 = \sum_i^2 \frac{\mathbf{k}_{i\perp}^2 + m_i^2}{x_i}. \tag{4.1}$$

In our analysis, we will only consider the light-meson sector(u, d, and s quarks) with equal quark and anti-quark masses($m_q = m_{\bar{q}}$).

The light-front $q\bar{q}$ bound-state wave function of the scalar($^3P_0$) and vector($^3S_1$) mesons can be written in the following covariant form

$$\begin{aligned} \Psi_M(x_i, \mathbf{k}_{i\perp}, \lambda_i) &= \bar{u}_{\lambda_q}(p_q)\Gamma_M v_{\lambda_{\bar{q}}}(p_{\bar{q}})\phi_M(x_i, \mathbf{k}_{i\perp}) \\ &= \mathcal{R}^M_{\lambda_q\lambda_{\bar{q}}}(x_i, \mathbf{k}_{i\perp})\phi_M(x_i, \mathbf{k}_{i\perp}), \end{aligned} \tag{4.2}$$

where $\mathcal{R}^M_{\lambda_q\lambda_{\bar{q}}}$ is the spin-orbit wave function, which is obtained by the interaction independent Melosh transformation [33] from the ordinary equal-time static spin-orbit wave function, and $\phi_M(x_i, \mathbf{k}_{i\perp})$ is the radial wave function. The operators Γ_M for the scalar(S)

and vector(V) mesons are given by

$$\begin{aligned}
\Gamma_S &= \frac{(\not{P}_S + M_0^S)\left(\frac{K\cdot P_S}{M_0^S} - \not{K}\right)}{(M_0^S + m_q + m_{\bar{q}})\sqrt{2[(M_0^S)^2 - (m_q^2 - m_{\bar{q}}^2)]}} \\
\Gamma_V &= \frac{-(\not{P}_V + M_0^V)\ \not{\epsilon}(J_3)}{(M_0^V + m_q + m_{\bar{q}})\sqrt{2[(M_0^V)^2 - (m_q^2 - m_{\bar{q}}^2)]}},
\end{aligned} \tag{4.3}$$

where $P_{(S,V)} \equiv (p_q + p_{\bar{q}})$, $K \equiv (p_{\bar{q}} - p_q)/2$ is the relative four-momentum between the quark and antiquark, and ϵ is the polarization four-vector of the vector meson(with momentum P_V), which is given by

$$\begin{aligned}
\epsilon^\mu(\pm) &= [\epsilon^+, \epsilon^-, \epsilon_\perp] = \left[0, \frac{2}{P_V^+}\epsilon_\perp(\pm)\cdot\mathbf{P_{V\perp}}, \epsilon_\perp(\pm)\right], \\
\epsilon_\perp(\pm) &= \mp\frac{(1, \pm i)}{\sqrt{2}}, \\
\epsilon^\mu(0) &= \frac{1}{M_V}\left[P_V^+, \frac{\mathbf{P}_{V\perp}^2 - M_V^2}{P_V^+}, \mathbf{P}_{V\perp}\right].
\end{aligned} \tag{4.4}$$

The operator Γ_V was derived in Ref. [44]. We followed the same procedure, detailed in Ref. [44], in order to derive Γ_S. Note that in the case of the vector meson, the operator Γ_V has the expected form, $(\not{P} + M)\ \not{\epsilon}$. However, because the scalar ($^3P_0$) state possesses non-zero orbital angular momentum, the proper Melosh–transformed spin-orbit wave function is not simply given by $\Gamma_S = (\not{P} + M)$, as one might expect. The form is more complicated as shown in Eq.(4.3), and it depends explicitly upon the relative momentum between the meson's constituents. This same type of behavior was demonstrated for the axial–vector meson in Ref. [44]. Since the axial–vector ($^3P_1$) state also possesses non-zero orbital angular momentum, the spin-orbit wave function is not simply given by $(\not{P} + M)\ \not{\epsilon}\gamma^5$. The correct form contains an additional factor which explicitly depends on the relative momentum between the quark and anti-quark. It is interesting to note, however, that in the case where $m_q = m_{\bar{q}} = m$ (which we use throughout this work), the expressions in Eq. (4.3) reduce to

$$\begin{aligned}
\Gamma_S &= \frac{1}{2\sqrt{2}M_0^S}(\not{P}_S + M_0^S), \\
\Gamma_V &= \frac{-1}{\sqrt{2}M_0^V(M_0^V + 2m)}(\not{P}_V + M_0^V)\ \not{\epsilon}(J_3).
\end{aligned} \tag{4.5}$$

So, in the equal mass case, Γ_S does have the expected form. We confirmed the similar reduction of the axial-vector meson wave function in the equal mass case. These expressions can be further simplified to the form we will use in our analysis:

$$\begin{aligned}\Gamma_S &= \frac{1}{2\sqrt{2}}, \\ \Gamma_V &= \frac{-1}{\sqrt{2}M_0^V}\left[\not{\epsilon}(J_3) - \frac{\epsilon\cdot(p_q - p_{\bar{q}})}{M_0^V + 2m}\right].\end{aligned} \tag{4.6}$$

The spin-orbit wave functions satisfy the following relations

$$\begin{aligned}\sum_{\lambda_q\lambda_{\bar{q}}} \mathcal{R}^{S\dagger}_{\lambda_q\lambda_{\bar{q}}}\mathcal{R}^{S}_{\lambda_q\lambda_{\bar{q}}} &= \frac{1}{4}(M_0^2 - 4m^2) = |\mathbf{k}|^2, \\ \sum_{\lambda_q\lambda_{\bar{q}}} \mathcal{R}^{V\dagger}_{\lambda_q\lambda_{\bar{q}}}\mathcal{R}^{V}_{\lambda_q\lambda_{\bar{q}}} &= 1,\end{aligned} \tag{4.7}$$

where $\mathbf{k} = (\mathbf{k}_\perp, k_z)$ is the three momentum of the constituent quark and $k_z = (x - \frac{1}{2})M_0$. Note that the total wave function $\Psi_S(x, \mathbf{k}_\perp)$ for the scalar meson vanishes at $|\mathbf{k}| = 0$ in accordance with the property of P-wave function.

For the radial wave functions $\phi_M(x_i, \mathbf{k}_{i\perp})$, we shall use the following gaussian wave functions for the scalar and vector mesons

$$\begin{aligned}\phi_S(x, \mathbf{k}_\perp) &= \mathcal{N}\sqrt{\frac{2}{3\beta^2}}\sqrt{\frac{\partial k_z}{\partial x}}\exp(-\mathbf{k}^2/2\beta^2), \\ \phi_V(x, \mathbf{k}_\perp) &= \mathcal{N}\sqrt{\frac{\partial k_z}{\partial x}}\exp(-\mathbf{k}^2/2\beta^2),\end{aligned} \tag{4.8}$$

where $\mathcal{N} = 4\pi^{3/4}\beta^{-3/2}$ and $\partial k_z/\partial x$ is the Jacobian of the variable transformation $\{x, \mathbf{k}_\perp\} \to \mathbf{k} = (k_z, \mathbf{k}_\perp)$ defined by

$$\frac{\partial k_z}{\partial x} = \frac{M_0}{4x(1-x)}, \tag{4.9}$$

and the normalization factors are obtained from the following normalization of the total wave function,

$$\int_0^1 dx \int \frac{d^2\mathbf{k}_\perp}{16\pi^3}|\Psi_M(x_i, \mathbf{k}_\perp)|^2 = 1. \tag{4.10}$$

The wave functions depend on only two model parameters: the constituent quark mass, m, and the binding strength, β. For these parameters, we use the values determined in Ref. [22]

which are ($m_{u,d} \equiv m_n = 0.22$ GeV, $\beta_{u,d} \equiv \beta_n = 0.3659$ GeV) and ($m_s = 0.45$ GeV,$\beta_s = 0.4128$ GeV).

Since our analysis deals with ϕ decays, a value for the ω–ϕ mixing angle is necessary. For the mixing formula [22] we use

$$\begin{aligned} |\phi\rangle &= -\sin\delta_{\omega\text{-}\phi}|n\bar{n}\rangle - \cos\delta_{\omega\text{-}\phi}|s\bar{s}\rangle \\ |\omega\rangle &= \cos\delta_{\omega\text{-}\phi}|n\bar{n}\rangle - \sin\delta_{\omega\text{-}\phi}|s\bar{s}\rangle, \end{aligned} \tag{4.11}$$

where $\delta_{\omega\text{-}\phi} = (\theta_{SU(3)} - 35.3^\circ)$ is the mixing angle. For "ideal" mixing (*i.e.* $\phi = s\bar{s}$), $\theta_{SU(3)} = 35.3^\circ$. The mixing angle has been analyzed in various ways. For example, $\delta_{\omega\text{-}\phi} \approx -3.3^\circ$ is favored in Refs. [45, 46, 47], while the conventional Gell–Mann–Okubo mass formula [2] for the exact SU(3) limit predicts $\delta_{\omega\text{-}\phi} \approx 0.7^\circ$ (or $\theta_{SU(3)} \approx 36^\circ$) in the linear mass scheme and $\delta_{\omega\text{-}\phi} \approx 3.7^\circ$(or $\theta_{SU(3)} \approx 39^\circ$) in the quadratic mass scheme. Taking into account SU(3) symmetry breaking and using the parametrization for a quadratic mass matrix given by Scadron [48], we obtained $\delta_{\omega\text{-}\phi} = \pm 7.8^\circ$ (see Ref. [22] for more detail). In the present work, we use this larger variation (from -7.8° to $+7.8^\circ$) to show the sensitivity of our results to the value of the mixing angle. As we shall see later in Section 4.5, the decay widths for the processes we are investigating are in fact not very sensitive to the value of $\delta_{\omega\text{-}\phi}$.

4.3 Scalar Mixing Amplitudes

Glueball–$q\bar{q}$ mixing can be described using a mass mixing matrix. Written in the $|gg\rangle$, $|s\bar{s}\rangle$, $|n\bar{n}\rangle$ basis, this takes the form [42, 43]

$$M = \begin{pmatrix} M_{gg} & f & \sqrt{2}fr \\ f & M_{s\bar{s}} & 0 \\ \sqrt{2}fr & 0 & M_{n\bar{n}} \end{pmatrix}, \tag{4.12}$$

where $f = \langle gg|M|s\bar{s}\rangle$ and $r = \langle gg|M|n\bar{n}\rangle / \sqrt{2}\langle gg|M|s\bar{s}\rangle$. As described in the introduction, the mixing is assumed to be among the three isoscalar states $f_0(1370), f_0(1500)$ and

$f_0(1710)$. These mixed physical states can be written as

$$\begin{pmatrix} |f_0(1710)\rangle \\ |f_0(1500)\rangle \\ |f_0(1370)\rangle \end{pmatrix} = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \begin{pmatrix} |gg\rangle \\ |s\bar{s}\rangle \\ |n\bar{n}\rangle \end{pmatrix}, \tag{4.13}$$

The mixing amplitudes, a_i, b_i, and c_i can be written in terms of the physical masses (M_{f_j}), the bare masses (M_{gg}, $M_{s\bar{s}}$, and $M_{n\bar{n}}$), and the glueball–bare quarkonia mixing strengths (f and r). For the present analysis we will adopt the values obtained by Lee and Weingarten, and by Close and Kirk.

Beginning with the lattice QCD–motivated assumption that the bare scalar $s\bar{s}$ is lighter than the scalar glueball (*i.e.* $M_{n\bar{n}} < M_{s\bar{s}} < M_{gg}$), Lee and Weingarten obtained the following mixing amplitudes [42]:

$$\begin{pmatrix} 0.86 \pm 0.05 & 0.30 \pm 0.05 & 0.41 \pm 0.09 \\ -0.13 \pm 0.05 & 0.91 \pm 0.04 & -0.40 \pm 0.11 \\ -0.50 \pm 0.12 & 0.29 \pm 0.09 & 0.82 \pm 0.09 \end{pmatrix}. \tag{4.14}$$

Using a different approach, Close and Kirk [43] examined the constraints placed on the flavor content of $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ by decay branching ratios to pairs of pseudoscalar mesons. From a χ^2 analysis of the available branching ratio data, they obtained various solutions depending on which parameters were left free at the outset. One solution was consistent with a glueball which lies just above the bare $n\bar{n}$ (*i.e.* $M_{n\bar{n}} \lesssim M_{gg} < M_{s\bar{s}}$). It's associated mixing amplitudes are

$$\begin{pmatrix} 0.39 \pm 0.03 & 0.91 \pm 0.02 & 0.15 \pm 0.02 \\ -0.65 \pm 0.04 & 0.33 \pm 0.04 & -0.70 \pm 0.07 \\ -0.69 \pm 0.07 & 0.15 \pm 0.01 & 0.70 \pm 0.07 \end{pmatrix}. \tag{4.15}$$

Another of their solutions was consistent with an even lighter glueball which lies below the mass of the $n\bar{n}$ (*i.e.* $M_{gg} < M_{n\bar{n}} < M_{s\bar{s}}$). The mixing amplitudes for this solution are

$$\begin{pmatrix} 0.25 & 0.96 & 0.10 \\ -0.37 & 0.13 & -0.92 \\ -0.89 & 0.14 & 0.44 \end{pmatrix}. \tag{4.16}$$

In the next section, we shall obtain expressions for the transition form factors for various radiative decays. Then, we will use the mixing amplitudes given in Eqs. (4.14)–(4.16) above to numerically evaluate the form factors and corresponding decay widths for the heavy, medium, and light weight glueball cases respectively.

4.4 Form Factors for Radiative Decays

4.4.1 The Process $V(S) \to S(V) + \gamma$

The coupling constant $g_{AX\gamma}$ for the radiative $P_A(q_1\bar{q}) \to P_X(q_2\bar{q})\gamma$ decays between vector (V) and scalar (S) mesons, *i.e.* $(A, X) = ({}^3S_1, {}^3P_0)$ or $({}^3P_0, {}^3S_1)$, is obtained by the matrix element of the electromagnetic current J^μ which is defined by

$$\begin{aligned} \mathcal{M}_1 &= \langle X(P_2)|\epsilon_\gamma \cdot J|A(P_1)\rangle \\ &= eg_{AX\gamma}[(\epsilon_\gamma \cdot \epsilon_V)(P_1 \cdot q) - (\epsilon_\gamma \cdot P_1)(\epsilon_V \cdot q)], \end{aligned} \tag{4.17}$$

where ϵ_γ and ϵ_V are the polarization vectors of the photon and the vector meson, respectively. Since the $J_z = 0$ state of the vector meson cannot convert into a real photon, the ϵ_V should be transversely polarized($J_z = \pm 1$) to extract the coupling constant $g_{AX\gamma}$. In other words, the possible helicity combinations in the transition $V(S) \to S(V)\gamma$ are either from $(J_z^\gamma = +1, J_z^V = -1)$ or from $(J_z^\gamma = -1, J_z^V = +1)$. The decay width for $A \to X + \gamma$ is given by [49]

$$\Gamma(A \to X + \gamma) = \alpha \frac{g_{AX\gamma}^2}{2J_A + 1}\left[\frac{M_A^2 - M_X^2}{2M_A}\right]^3, \tag{4.18}$$

where α is the fine structure constant and J_A is the total angular momentum of the initial particle.

In LFQM calculations, we shall analyze the virtual photon(γ^*) decay process so that we calculate the momentum dependent transition form factor, $F_{AX\gamma^*}(Q^2)$. The coupling constant, $g_{AX\gamma}$, can then be determined in the limit as $Q^2 \to 0$ (*i.e.*, $g_{AX\gamma} = F_{AX\gamma^*}(Q^2 = 0)$). Figure 4.1 shows the primary Feynman diagram for this process. The

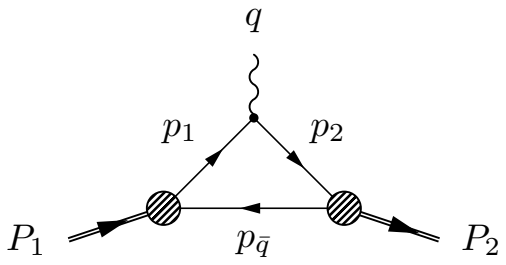

Figure 4.1: Primary diagram for $A(P_1) \to X(P_2) + \gamma(q)$. There is an additional diagram in which the virtual photon interacts with the antiquark.

amplitude is

$$\begin{aligned} \mathcal{M}_1^\mu &= \langle X(P_2)|J^\mu|A(P_1)\rangle \\ &= eF_{AX\gamma}(q^2)[\epsilon_V^\mu(P_1\cdot q) - P_1^\mu(\epsilon_V\cdot q)]. \end{aligned} \tag{4.19}$$

Our analysis is based on the standard light-front frame($q^+ = 0$)[30].

$$\begin{aligned} P_1 &= (P_1^+, P_1^-, \mathbf{P}_{1\perp}) = (P_1^+, \frac{M_A^2}{P_1^+}, \mathbf{0}_\perp), \\ q &= (0, \frac{M_A^2 - M_X^2 - Q^2}{P^+}, \mathbf{q}_\perp), \\ P_2 &= P_1 - q = (P_1^+, \frac{M_X^2 + Q^2}{P_1^+}, -\mathbf{q}_\perp), \end{aligned} \tag{4.20}$$

where $\mathbf{q}_\perp^2 = Q^2 = -q^2$ is the space-like photon momentum transfer.

The quark momentum variables in the $q^+ = 0$ frame are given by

$$\begin{aligned} p_1^+ &= (1-x)P_1^+,\ p_{\bar{q}}^+ = xP_1^+, \\ \mathbf{p}_{1\perp} &= (1-x)\mathbf{P}_{1\perp} + \mathbf{k}_\perp,\ \mathbf{p}_{\bar{q}\perp} = x\mathbf{P}_{1\perp} - \mathbf{k}_\perp, \\ p_2^+ &= (1-x)P_2^+,\ p_{\bar{q}}^{\prime +} = xP_2^+, \\ \mathbf{p}_{2\perp} &= (1-x)\mathbf{P}_{2\perp} + \mathbf{k'}_\perp,\ \mathbf{p'}_{\bar{q}\perp} = x\mathbf{P}_{2\perp} - \mathbf{k'}_\perp, \end{aligned} \tag{4.21}$$

which require that $p_{\bar{q}}^+ = p_{\bar{q}}^{\prime +}$ and $\mathbf{p}_{\bar{q}\perp} = \mathbf{p'}_{\bar{q}\perp}$.

The hadronic matrix elements in the radiative decay between two particles A and

X are of the form

$$\begin{aligned}
&\langle X(P_2)|J^\mu|A(P_1)\rangle \\
&= \sum_{j,\lambda' s} \mathcal{A}_j \int_0^1 dx \int \frac{d^2\mathbf{k}_\perp}{16\pi^3}\phi_2(x,\mathbf{k'}_\perp)\phi_1(x,\mathbf{k}_\perp) \\
&\times \mathcal{R}^{X\dagger}_{\lambda_2\bar{\lambda}} \frac{\bar{u}_{\lambda_2}(p_2)}{\sqrt{p_2^+}}\gamma^\mu \frac{u_{\lambda_1}(p_1)}{\sqrt{p_1^+}}\mathcal{R}^{A}_{\lambda_1\bar{\lambda}} \\
&= e\sum_j \mathcal{A}_j I_1(m_j,Q^2)[\epsilon_V^\mu (P_1\cdot q) - P_1^\mu(\epsilon_V\cdot q)],
\end{aligned} \tag{4.22}$$

where $\mathbf{k'}_\perp = \mathbf{k}_\perp - x\mathbf{q}_\perp$, $\mathcal{A}_j$ is the overlap of the j^{th} flavor portion of the flavor wave functions and contains all of the relevant charge factors and mixing amplitudes. Comparing the last line of Eq. (4.22) with Eq. (4.19) it can be seen that we have defined the form factor in terms of the charge and mixing amplitude independent quantity $I_1(m_j,Q^2)$.

The sum of the light-front spinors over the helicities in Eq. (4.22) is obtained as

$$\begin{aligned}
S^\mu &= \sum_{\lambda' s} \mathcal{R}^{X\dagger}_{\lambda_2\bar{\lambda}} \frac{\bar{u}_{\lambda_2}(p_2)}{\sqrt{p_2^+}}\gamma^\mu \frac{u_{\lambda_1}(p_1)}{\sqrt{p_1^+}}\mathcal{R}^{A}_{\lambda_1\bar{\lambda}} \\
&= \frac{\mathrm{Tr}[(\not{p}_{\bar{q}} - m_j)\Gamma_X(\not{p}_2 + m_j)\gamma^\mu(\not{p}_1 + m_j)\Gamma_A]}{\sqrt{p_1^+ p_2^+}}.
\end{aligned} \tag{4.23}$$

The explicit form of the trace is summarized in Appendix 1. For $V(P_1)\to S(P_2)\gamma^*(q)$ decay, the (transverse) polarization vector ϵ_V of the vector meson is given by $\epsilon_V(\pm) = [0,0,\epsilon_\perp(\pm)]$ and the trace term with the plus component of the current is given by

$$\begin{aligned}
S^+_{V\to S} = \frac{-1}{(1-x)M_0\sqrt{2}}\Big\{ & m_j[q^R - 4(1-x)k^R] \\
& - \frac{2k^R}{x(M_0+2m_j)}[(2x-1)^2 m_j^2 + \mathbf{k}_\perp^2 - x\mathbf{k}_\perp\cdot\mathbf{q}_\perp]\Big\},
\end{aligned} \tag{4.24}$$

where we use $\epsilon_V(+)$. Therefore, we obtain the one-loop integral, $I_1(m_j, Q^2)$, as follows

$$\begin{aligned} I_1(m_j,Q^2) &= \int_0^1 dx \int \frac{d^2\mathbf{k}_\perp}{16\pi^3}\phi_S(x,\mathbf{k'}_\perp)\phi_V(x,\mathbf{k}_\perp)\frac{1}{x(1-x)M_0} \\ &\times\Bigg\{xm_j\left[1-4(1-x)\frac{k^Rq^L}{\mathbf{q}_\perp^2}\right]-\frac{2(k^Rq^L)/\mathbf{q}_\perp^2}{(M_0+2m_j)} \\ &\times[(2x-1)^2m_j^2+\mathbf{k}_\perp^2-x\mathbf{k}_\perp\cdot\mathbf{q}_\perp] \\ &+(x\to 1-x,\mathbf{k}_\perp\to-\mathbf{k}_\perp)\Bigg\}, \end{aligned} \tag{4.25}$$

where $k^Rq^L = \mathbf{k}_\perp\cdot\mathbf{q}_\perp - i|\mathbf{k}_\perp\times\mathbf{q}_\perp|_z$ and even though the cross term does not contribute to the integral, the dot product term does contribute.

Then, the transition form factor $F_{VS\gamma^*}(Q^2)$ is given by

$$F_{VS\gamma^*}(Q^2) = \mathcal{A}_n I_1(m_n,Q^2) + \mathcal{A}_s I_1(m_s,Q^2), \tag{4.26}$$

where $\mathcal{A}_n$ and $\mathcal{A}_s$ are the overlaps of the up–down and strange portions of the flavor wave functions respectively. For example, in the case where $V=\phi$ and $S=(u\bar{u}-d\bar{d})/\sqrt{2}$, $\mathcal{A}_n = -(\sin\delta_{\omega-\phi})(e_u-e_d)/2$ and $\mathcal{A}_s = 0$. Also, for $V=\phi$ and $S=s\bar{s}$, $\mathcal{A}_n = 0$ and $\mathcal{A}_s = -(\cos\delta_{\omega-\phi})e_s$.

The transition form factor $F_{SV\gamma^*}(Q^2)$ for $S(P_1)\to V(P_2)\gamma^*(q)$ can be obtained from Eq. (4.25) by replacing $\mathbf{q}_\perp\to-\mathbf{q}_\perp$ and $\mathbf{k}_\perp\to\mathbf{k'}_\perp$ and the explicit form for the one-loop integral corresponding to Eq. (4.25) is given by

$$\begin{aligned} I_1'(m_j,Q^2) &= \int_0^1 dx \int \frac{d^2\mathbf{k}_\perp}{16\pi^3}\phi_V(x,\mathbf{k'}_\perp)\phi_S(x,\mathbf{k}_\perp)\frac{1}{x(1-x)M_0'} \\ &\times\Bigg\{xm_j\left[-(2x-1)^2-4(1-x)\frac{k^Rq^L}{\mathbf{q}_\perp^2}\right]-\frac{2(k^Rq^L-x\mathbf{q}_\perp^2)/\mathbf{q}_\perp^2}{(M_0'+2m_j)} \\ &\times[(2x-1)^2m_j^2+\mathbf{k}_\perp^2-x\mathbf{k}_\perp\cdot\mathbf{q}_\perp]+(x\to 1-x;\mathbf{k}\to-\mathbf{k}_\perp)\Bigg\}, \end{aligned} \tag{4.27}$$

where $M_0'^2 = (m_j^2+\mathbf{k'}_\perp^2)/x(1-x)$ and again the cross term in k^Rq^L does not contribute to the integral. As one may expect, however, we found that $I_1'(m_j,Q^2) = -I_1(m_j,Q^2)$ and thus $F_{SV\gamma^*}(Q^2) = -F_{VS\gamma^*}(Q^2)$.

4.4.2 The Process $S \to \gamma\gamma$

We now apply this model calculation to the two photon decays of scalar mesons. In this case, the coupling constant $g_{S\gamma\gamma}$ for $S \to \gamma\gamma$ is defined by

$$\begin{aligned} \mathcal{M}_2 &= \langle \gamma(P_2)|\epsilon_1 \cdot J|S(P_1)\rangle \\ &= e^2 g_{S\gamma\gamma}[(\epsilon_1 \cdot \epsilon_2)(P_1 \cdot q) - (\epsilon_1 \cdot P_1)(\epsilon_2 \cdot q)], \end{aligned} \tag{4.28}$$

where $\epsilon_1 = \epsilon_\gamma(q)$ and $\epsilon_2 = \epsilon_\gamma(P_2)$. In terms of $g_{S\gamma\gamma}$, the decay width for this process is given by

$$\Gamma = \frac{\pi}{4}\alpha^2 g_{S\gamma\gamma}^2 M_S^3 \,. \tag{4.29}$$

Here again, instead of calculating the two real photon decays, we first calculate the matrix element for $S \to \gamma\gamma^*$, which is given by

$$\begin{aligned} \mathcal{M}_2^\mu &= \langle \gamma(P_2)|J^\mu|S(P_1)\rangle, \\ &= e^2 F_{S\gamma\gamma^*}(Q^2)[\epsilon_2^\mu (P_1 \cdot q) - P_1^\mu (\epsilon_2 \cdot q)], \end{aligned} \tag{4.30}$$

and take the limit $Q^2 \to 0$ to compute the decay rate for the two real photon decays. Using the same quark momentum variables in $q^+ = 0$ frame as Eq. (4.21) with the plus component of the current, we then obtain

$$\begin{aligned} \mathcal{M}_2^+ &= \sqrt{n_c}\sum_j \mathcal{A}_j \sum_{\lambda_1,\lambda_2,\bar{\lambda}} \int_0^1 dx \int \frac{d^2\mathbf{k}_\perp}{16\pi^3}\phi_S(x_i,\mathbf{k}_\perp)\Bigg[\left(\frac{\bar{v}_{\bar{\lambda}}(p_{\bar{q}})}{\sqrt{p_{\bar{q}}^+}} \not{\epsilon}_2 \frac{u_{\lambda_2}(p_2)}{\sqrt{p_2^+}}\right) \\ &\quad \times\left(\frac{\bar{u}_{\lambda_2}(p_2)}{\sqrt{p_2^+}}\gamma^+\frac{u_{\lambda_1}(p_1)}{\sqrt{p_1^+}}\right)\left(\frac{1}{\mathbf{q}_\perp^2 - [\mathbf{p}_{2\perp}^2 + m_j^2]/p_2^+ - [\mathbf{p}_{\bar{q}\perp}^2 + m_j^2]/p_{\bar{q}}^+}\right) \\ &\quad +(x \to 1-x, \mathbf{k}_\perp \to -\mathbf{k}_\perp)\Bigg]\,\mathcal{R}^S_{\lambda_1\lambda_{\bar{q}}} \\ &= e^2\left(\sum_j \mathcal{A}_j I_2(m_j,Q^2)\right)[\epsilon_2^\mu(P_1\cdot q) - P_1^\mu(\epsilon_2\cdot q)], \end{aligned} \tag{4.31}$$

where $\mathcal{A}_j$ now contains factors of e_j^2 due to the presence of two electromagnetic vertices. As in the case of Eq. (4.23), the sum of the light-front spinors over the helicities in the first term of Eq. (4.31) is obtained as

$$\begin{aligned} T^\mu &= \sum_{\lambda_1,\lambda_2,\bar{\lambda}} \frac{\bar{v}_{\bar{\lambda}}(p_{\bar{q}})}{\sqrt{p_{\bar{q}}^+}} \not{\epsilon}_2 \frac{u_{\lambda_2}(p_2)}{\sqrt{p_2^+}} \frac{\bar{u}_{\lambda_2}(p_2)}{\sqrt{p_2^+}} \gamma^+ \frac{u_{\lambda_1}(p_1)}{\sqrt{p_1^+}} \mathcal{R}^S_{\lambda_1 \lambda_{\bar{q}}} \\ &= \frac{\mathrm{Tr}[(\not{p}_{\bar{q}} - m_j)\ \not{\epsilon}_2 (\not{p}_2 + m_j)\gamma^+ (\not{p}_1 + m_j)]}{2\sqrt{2p_{\bar{q}}^+ (p_2^+)^2 p_1^+}} \\ &= \frac{4m_j[p_1^+(p_{\bar{q}}\cdot\epsilon_2 - p_2\cdot\epsilon_2) + p_2^+(p_{\bar{q}}\cdot\epsilon_2 - p_1\cdot\epsilon_2) + p_{\bar{q}}^+(p_2\cdot\epsilon_2 - p_1\cdot\epsilon_2)]}{2\sqrt{2p_{\bar{q}}^+ (p_2^+)^2 p_1^+}}, \end{aligned} \tag{4.32}$$

where we have used the fact that $\epsilon_2^+(\pm) = 0$. Now, using $\epsilon_2(+) = (0, \sqrt{2}q^R/P_1^+, \epsilon_\perp)$, we finally obtain the one loop integral,

$$\begin{aligned} I_2(m_j, Q^2) &= -\sqrt{6}\int_0^1 dx \int \frac{d^2\mathbf{k}_\perp}{16\pi^3} \phi_S(x, \mathbf{k}_\perp) \Bigg\{ \frac{m_j[(2x-1)^2 + 4(1-x)(k^R q^L/\mathbf{q}_\perp^2)]}{(1-x)\sqrt{x(1-x)}} \\ &\quad \times \frac{x(1-x)}{m_j^2 + (\mathbf{k}_\perp - x\mathbf{q}_\perp)^2} + (x \to 1-x, \mathbf{k}_\perp \to -\mathbf{k}_\perp) \Bigg\}, \end{aligned} \tag{4.33}$$

and the transition form form factor is given by

$$F_{S\gamma\gamma^*}(Q^2) = \mathcal{A}_n I_2(m_n, Q^2) + \mathcal{A}_s I_2(m_s, Q^2). \tag{4.34}$$

As an example of the coefficients $\mathcal{A}_{n,s}$, consider the case where $S = f_0(1370)$. Here, $\mathcal{A}_n = c_3[(e_u^2 + e_d^2)/\sqrt{2}]$ and $\mathcal{A}_s = b_3 e_s^2$, where c_3 and b_3 are the glueball–quarkonia mixing amplitudes of Eq. (4.13).

4.5 Numerical Results

4.5.1 Decays Involving $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$

The expressions for the one-loop integrals, $I_1(m_j, Q^2)$ and $I_2(m_j, Q^2)$, are evaluated numerically and used in Eqs. (4.26) and (4.34) to compute the Q^2-dependent transition form factors for $\gamma\gamma^*$, $\phi\gamma^*$, and $\rho\gamma^*$ decays of the scalar mesons. As an example, we give the results for the case of the heavy glueball (*i.e.* $M_{n\bar{n}} < M_{s\bar{s}} < M_{gg}$). The transition form factors for the $\gamma\gamma^*$ process are shown in Fig. 4.2, and those for the $\phi\gamma^*$ and $\rho\gamma^*$ processes are

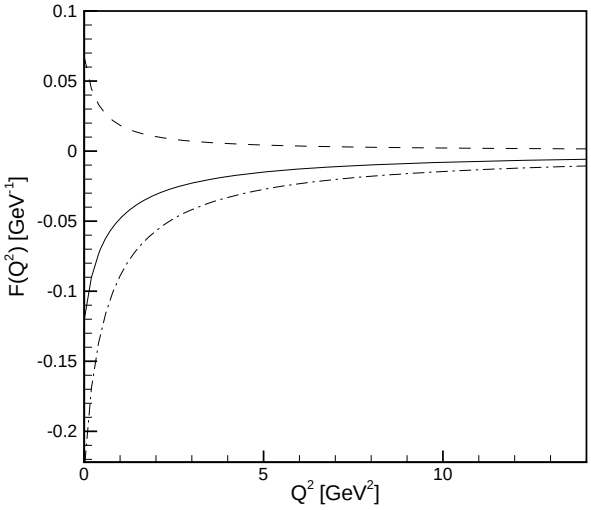

Figure 4.2: $f_0 \to \gamma\gamma^*$ transition form factors for $f_0(1370)$ [dash-dotted], $f_0(1500)$ [dashed], and $f_0(1710)$ [solid].

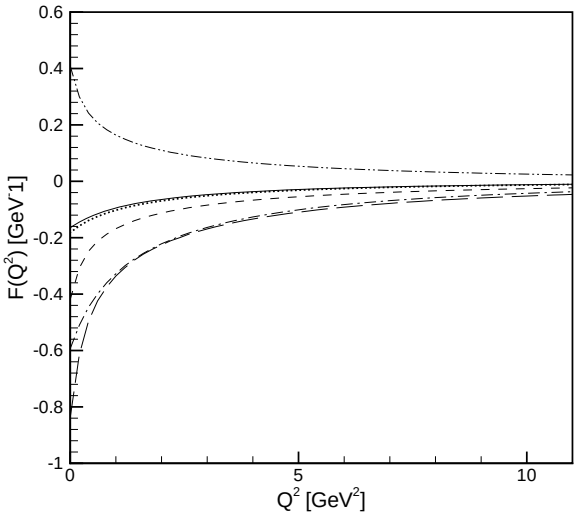

Figure 4.3: $f_0 \to \rho\gamma^*$ transition form factors for $f_0(1370)$ [long-dashed], $f_0(1500)$ [dash-dot-dotted], and $f_0(1710)$ [short-dashed]; $f_0 \to \phi\gamma^*$ transition form factors for $f_0(1370)$ [solid], $f_0(1500)$ [dash-dotted], and $f_0(1710)$ [dotted]. Here we have used $\delta_{\omega-\phi} = +7.8\,^\circ$.

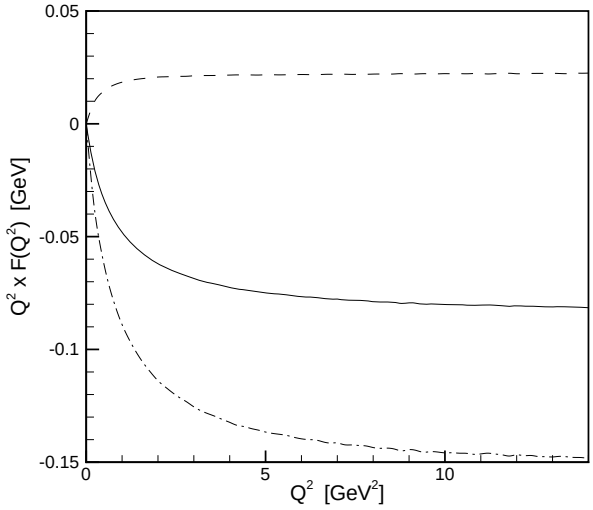

Figure 4.4: Q^2 times the $f_0 \to \gamma\gamma^*$ transition form factors (Fig. 4.2) for $f_0(1370)$ [dash-dotted], $f_0(1500)$ [dashed], and $f_0(1710)$ [solid].

collectively shown in Fig. 4.3. In the case of the two photon decay, the form factor should fall off like $1/Q^2$ due to an intermediate quark propagator which becomes highly off–shell at large Q^2. Figure 4.4 shows the behavior of $Q^2 \times F_{f_0\gamma\gamma^*}(Q^2)$ for each scalar state. Each of the curves clearly shows a tendency to flatten out, demonstrating $1/Q^2$ dependence in the form factors.

The decay constants for the real photon processes can be obtained from the form factors in the limit as $Q^2 \to 0$ (*i.e.* $g = F(Q^2 = 0)$). In this limit, the values of the one-loop integrals are

$$\begin{aligned} I_1(m_n, Q^2 = 0) &= 2.05\, GeV^{-1} \\ I_1(m_s, Q^2 = 0) &= 1.93\, GeV^{-1} \\ I_2(m_n, Q^2 = 0) &= -0.672\, GeV^{-1} \\ I_2(m_s, Q^2 = 0) &= -0.375\, GeV^{-1}. \end{aligned} \tag{4.35}$$

The decay constants are obtained by substituting these values into Eqs. (4.26) and (4.34). The decay widths are then calculated using Eqs. (4.18) and (4.29). The widths for the $\gamma\gamma$ decays are listed in Table 4.1. The widths for the $\phi\gamma$, and $\rho\gamma$ decays are listed under the

heading 'LFQM' in Tables 4.2 and 4.3 respectively. The subheadings Heavy, Medium and Light refer to the different glueball mass scenarios. The uncertainties result solely from the uncertainties in the mixing amplitudes in Eqs. (4.14) and (4.15). We have not accounted for the uncertainties in the meson masses. The uncertainties in the masses of $f_0(1500)$ and $f_0(1710)$ ($\sim$0.4%) are very small compared to the uncertainties in the mixing amplitudes (6%–40%), and can be neglected. However, the uncertainty in the mass of $f_0(1370)$ is about 10%, and would, therefore, contribute significantly to the uncertainties in the decay widths.

Experimental data for radiative decays of the isoscalars $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ are poor. As one example, in the recent past the PDG had reported partial widths of $3.8 \pm 1.5\, keV$ and $5.4 \pm 2.3\, keV$ for the process $f_0(1370) \to \gamma\gamma$ [50]. The PDG currently attributes these two values to $f_0(600)$, but at the same time they state in a footnote that this data could equally well be assigned to $f_0(1370)$ [2]. If these data which are on the order of a few keV do belong to $f_0(1370)$, this would be encouraging given that our results are consistent with this order of magnitude. However, the ambiguity noted above makes any such comparison irrelevant, and a great deal more experimental investigation is necessary before any definitive conclusions can be reached about the validity of any of the glueball mixing schemes.

In the absence of good experimental data with which to compare our results, we turn to other theoretical predictions concerning these decay processes. In Ref. [43], Close and Kirk give predictions for ratios of $f_0 \to \gamma\gamma$ widths which depend only on charge factors and mixing amplitudes, and ignore all mass-dependent effects. For the ratios $\Gamma(f_0(1710) \to$

Table 4.1: Decay widths for the process $f_0 \to \gamma\gamma$. The unit of the decay width is [keV]. The uncertainties result from the uncertainties in the mixing amplitudes in Eqs. (4.14) and (4.15).

	Light	Medium	Heavy
$f_0(1370)$	1.6	$3.9^{+0.8}_{-0.7}$	$5.6^{+1.4}_{-1.3}$
$f_0(1500)$	8.0	$4.1^{+1.0}_{-0.9}$	$0.65^{+0.72}_{-0.45}$
$f_0(1710)$	0.92	$1.3^{+0.2}_{-0.2}$	$3.0^{+1.4}_{-1.2}$

Table 4.2: Decay widths for the process $f_0 \to \phi\gamma$. The unit of the decay width is [keV]. Our results are under the heading 'LFQM', while CDK's results are under the heading 'NR Model'. The LFQM results are for $\delta_{\omega\text{-}\phi} = +7.8\,^\circ$ and CDK's results are for $\delta_{\omega\text{-}\phi} = 0^\circ$. The uncertainties on our LFQM results are due to the uncertainties in the mixing amplitudes in Eqs. (4.14) and (4.15).

	LFQM			**NR Model**		
	Light	Medium	Heavy	Light	Medium	Heavy
$f_0(1370)$	0.98	$0.83^{+0.27}_{-0.23}$	$4.5^{+4.5}_{-3.0}$	8	9	32
$f_0(1500)$	7.5	28^{+7}_{-6}	170^{+20}_{-20}	9	60	454
$f_0(1710)$	450	400^{+20}_{-20}	36^{+17}_{-14}	800	718	78

Table 4.3: Decay widths for the process $f_0 \to \rho\gamma$. The unit of the decay width is [keV]. Our results are under the heading 'LFQM', while CDK's results are under the heading 'NR Model'. The uncertainties in our LFQM results are due to the uncertainties in the mixing amplitudes in Eqs. (4.14) and (4.15).

	LFQM			**NR Model**		
	Light	Medium	Heavy	Light	Medium	Heavy
$f_0(1370)$	150	390^{+80}_{-70}	530^{+120}_{-110}	443	1121	1540
$f_0(1500)$	1100	630^{+130}_{-120}	210^{+130}_{-100}	2519	1458	476
$f_0(1710)$	24	55^{+16}_{-14}	410^{+200}_{-160}	42	94	705

Table 4.4: Decay widths in keV for the process $f_0 \to \phi\gamma$ for $\delta_{\omega\text{-}\phi} = +7.8^\circ, 0^\circ, and -7.8^\circ$. The results shown are for the medium-weight glueball scenario.

	$+7.8^\circ$	0°	-7.8°
$f_0(1370)$	0.83	1.9	3.4
$f_0(1500)$	28	21	15
$f_0(1710)$	400	410	420

$\gamma\gamma$):$\Gamma(f_0(1500) \to \gamma\gamma)$:$\Gamma(f_0(1370) \to \gamma\gamma)$ they obtain

$$\begin{aligned} \text{Light Glueball} &= 1 : 5.1 : 2.8 \\ \text{Medium Glueball} &= 1 : 2.4 : 3.6 \\ \text{Heavy Glueball} &= 1 : 0.1 : 3.7 \ . \end{aligned} \tag{4.36}$$

Our analysis which includes all of the relevant mass dependent effects yields

$$\begin{aligned} \text{Light Glueball} &= 1 : 8.7 : 1.7 \\ \text{Medium Glueball} &= 1 : 3.2 : 3.0 \\ \text{Heavy Glueball} &= 1 : 0.2 : 1.9 \ . \end{aligned} \tag{4.37}$$

Our results differ slightly from those of Close and Kirk, however the same overall qualitative pattern is preserved.

In Ref. [51], Close, Donnachie, and Kalashnikova (CDK) compute the ϕ and ρ radiative decay widths for $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ in the NR quark model. Assuming that $\delta_{\omega\text{-}\phi} = 0$, they obtain the values listed under the heading 'NR Model' in Tables 4.2 and 4.3. Comparing these values with our LFQM results, it is clear that the relativistic corrections introduced by our model reduce the overall magnitudes of the decay widths by about 50–70%. We note somewhat greater reduction for the process of $f_0(1370) \to \phi\gamma$ due to our non-zero $\delta_{\omega\text{-}\phi}$. A reduction in the widths would be expected given that the relativistic motion of the constituents tends to spread out the meson's momentum–space wave function, thereby decreasing its peak value. We have verified this qualitative behavior by examining the decay width's dependence on the model parameter β. As β is increased, the wave function, $\sim exp[-k^2/(2\beta^2)]$, is more spread out (more relativistic) and the decay width does in fact decrease. As β is decreased, the wave function becomes more sharply peaked and the decay width increases accordingly. Despite the differences in the overall magnitudes of the decay widths, however, the relative strengths between the different decay processes are fairly well preserved. Just as in CDK's analysis, we find that the largest branching ratio is likely to be that of $f_0(1500) \to \rho\gamma$. In our model, this branching ratio is about 1% for the light glueball case, 0.6% for the medium glueball case, and 0.2% for the heavy glueball case.

Additionally, we find that our predictions for the decay widths involving ϕ are not

highly sensitive to the value of the ω-ϕ mixing angle. As an example, in Table 4.4 we have listed the widths for the processes $f_0 \to \phi\gamma$ (medium glueball mass case) for various mixing angles. The decay involving $f_0(1370)$ is the most sensitive in terms of the percentage shift in the width. This is, of course, due to the fact that its wave function contains a relatively small amount of $s\bar{s}$ (see Eq. (4.15)). However, as we vary the mixing angle from $+7.8^\circ$ to -7.8°, we see that the change in our model prediction is not significant when compared to the difference between our model prediction and CDK's NR model prediction.

4.5.2 Decays Involving $a_0(980)$ and $f_0(980)$

If we assume $a_0(980)$ to be a conventional $q\bar{q}$, then the flavor structure should be $(u\bar{u} - d\bar{d})/\sqrt{2}$. For the processes $a_0(980) \to \gamma\gamma$ and $\phi \to a_0(980)\gamma$, the decay constants and associated widths are calculated to be

$$\begin{aligned} g_{\phi a_0\gamma} &= -0.14\, GeV^{-1}, \quad & \Gamma_{\phi a_0\gamma} &= 2.8\, eV \\ g_{a_0\gamma\gamma} &= -0.16\, GeV^{-1}, \quad & \Gamma_{a_0\gamma\gamma} &= 990\, eV\ . \end{aligned} \tag{4.38}$$

None of these calculated values for the widths are consistent with experimental data. The ϕ radiative width of 2.8 eV gives a branching ratio of $BR(\phi \to a_0\gamma) = 6.7 \times 10^{-7}$ which is significantly smaller than the PDG average of $0.88^{+0.17}_{-0.17} \times 10^{-4}$; and, the two-photon width of 990 eV is roughly 3 times larger than the value reported by Amsler of $0.30 \pm 0.10\, keV$ [52]. The flavor content of the isoscalar $f_0(980)$ is less clear. If we consider the two possible extremes, $f_0(980) = n\bar{n}$ and $f_0(980) = s\bar{s}$, we obtain

$$\begin{aligned} n\bar{n} \quad &= \quad \begin{cases} g_{\phi f_0\gamma} = -0.05\, GeV^{-1}, & \Gamma_{\phi f_0\gamma} = 0.37\, eV \\ g_{f_0\gamma\gamma} = -0.26\, GeV^{-1}, & \Gamma_{f_0\gamma\gamma} = 2.7\, keV \end{cases} \\ s\bar{s} \quad &= \quad \begin{cases} g_{\phi f_0\gamma} = +0.64\, GeV^{-1}, & \Gamma_{\phi f_0\gamma} = 60\, eV \\ g_{f_0\gamma\gamma} = -0.04\, GeV^{-1}, & \Gamma_{f_0\gamma\gamma} = 63\, eV. \end{cases} \end{aligned} \tag{4.39}$$

The PDG average for the two photon width is $\Gamma_{f_0\gamma\gamma} = 0.39^{+0.10}_{-0.13}\, keV$. Since the $s\bar{s}$ result falls below this value and the $n\bar{n}$ result sits above it, it would be possible for some mixed $n\bar{n}$–$s\bar{s}$ state to reproduce the data. Working out the mixing required for this, we find that $f_0(980)$ would be about 6% $n\bar{n}$ and 94% $s\bar{s}$. This alone would allow for $f_0(980)$ to be

interpreted as a conventional $q\bar{q}$. However, the ϕ radiative widths we calculated lead to the branching ratios $BR(\phi \to f_0\gamma) = 8.7 \times 10^{-8}$ for the $n\bar{n}$ and $BR(\phi \to f_0\gamma) = 1.4 \times 10^{-5}$ for the $s\bar{s}$. Both of these values fall well below the PDG average of $3.3^{+0.8}_{-0.5} \times 10^{-4}$. Also, if we compute the ratio $BR(\phi \to f_0\gamma)/BR(\phi \to a_0\gamma)$ for our model predictions, we get 0.13 for $f_0 = n\bar{n}$ and 21 for $f_0 = s\bar{s}$, while the experimental ratio is around 4. This again hints at the possibility of a mixed $n\bar{n}$–$s\bar{s}$ being able to reproduce the data. However, the mixing needed to reproduce this ratio requires that $f_0(980)$ be 87% $n\bar{n}$ and 13% $s\bar{s}$. This is the exact opposite of the mixing needed to reproduce the two photon width above. Overall our results are not consistent with well established data on $a_0(980)$ and $f_0(980)$.

4.6 Conclusions

We have performed the first LFQM calculations involving scalar mesons. First, the $^3P_0$ light–front wave function was constructed. It was shown that, in general, the covariant operator used to obtain the spin–orbit wave function depends explicitly on the relative momentum between the meson's constituents, and is, therefore, more complicated than the naive form that is commonly used. This wave function was used to compute radiative decays involving $f_0(1370)$, $f_0(1500)$, $f_0(1710)$, $f_0(980)$, and $a_0(980)$.

In the case of the three heavy isoscalars, the effects of glueball–$q\bar{q}$ mixing were taken into account. Specifically, three different mixing schemes corresponding to a heavy, medium, and light glueball were used. The lack of good experimental data made it difficult to draw any conclusions about which of the three mixing scenarios, if any, could be the correct one. We note, however, that we have improved upon the earlier NR model predictions of Close *et al.* [51]. Relativistic corrections introduced by the LFQM resulted in decay widths that were about 50–70% smaller than those obtained in the NR calculations. Yet, very little change was observed in the pattern of relative strengths which is apparently quite robust. We also demonstrated that our model gives the correct qualitative behavior for the decay width as a function of the parameter β, in the sense that the decay width increases(decreases) as β is decreased(increased). In a future work, we intend to perform a more detailed comparison of CDK's NR model calculation and our LFQM calculation to isolate the relativistic correction terms that make these differences.

For the calculations involving $a_0(980)$ and $f_0(980)$, we assumed these states to be $q\bar{q}$. In contrast to the case of the heavy scalars, there does exist well–established data for

these light scalars. While one or two of the properties we calculated, when taken in isolation, could be considered consistent with the data, it is clear that our results as a whole do not match the data. How are we to interpret the inconsistencies between our results and the experimental data for $a_0(980)$ and $f_0(980)$? Are they due to some deficiency in our LFQM or are they due to the nature of $a_0(980)$ and $f_0(980)$ not being simple constituent $q\bar{q}$? We are leaning toward the latter conclusion for two reasons. First, as we discussed in the introduction, the LFQM has had many past successes in the phenomenology of pseudoscalar and vector meson systems. These past successes have led us to appreciate that the agreement with the data was not fortuitous, but natural in the sense that the advantage of LFD played its role. We can see no obvious reason why the model would work so well for the pseudoscalar and vector mesons but not for the low-lying scalar mesons. The second, and perhaps more persuasive reason is that models other than the LFQM predict very similar discrepancies between the $q\bar{q}$ picture and the available data. For example, in the previous section we obtained a value of 6.7×10^{-7} for the the branching ratio $BR(\phi \to a_0(980)\gamma)$. Gokalp and Yilmaz [53], using light–cone QCD sum rules, obtain a value of the coupling constant which yields $BR(\phi \to a_0(980)\gamma) = 4.1 \times 10^{-7}$, while Titov *et al.* [54], using simple meson–exchange model based on effective Lagrangians, obtain a value of the coupling constant which yields $BR(\phi \to a_0(980)\gamma) = 8.8 \times 10^{-7}$. All three of these values, based on the $q\bar{q}$ picture, are roughly two orders of magnitude smaller than the current PDG average of 0.88×10^{-4}. So, it seems that the discrepancies between the $q\bar{q}$ picture and experimental data do not depend too heavily on particular model details. In addition to comparing our results with those of other models, we have also tried varying our own model parameters (e.g. the $f_0(980)$ mixing angle between $n\bar{n}$ and $s\bar{s}$ as described in the previous section). However, when we have an agreement with one observable we do not get the same level of agreement with other observables using the same model parameters. Certainly, further investigation is necessary in determining the model parameters, but it seems unlikely that simply fine tuning them would alleviate the orders of magnitude problem. We are in the process of computing the scalar meson spectroscopy using the variational principle and this procedure would give a more definite conclusion regarding this issue. At the moment, because of the reasons listed above, we believe that the inconsistencies between our predictions and the available data for $a_0(980)$ and $f_0(980)$ are not due to any general deficiency in our LFQM, but rather to the fact that these mesons are not simple constituent $q\bar{q}$.

In addition to refining the model parameters, the spectrum analysis mentioned

above will also give the masses of the bare $n\bar{n}$ and $s\bar{s}$ P–wave quarkonia. With these masses, we will be able to perform our own glueball–$q\bar{q}$ mixing analysis involving the isoscalars $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$. These mixing amplitudes, obtained using the LFQM, could then be compared with those of Lee–Weingarten and Close–Kirk.

Chapter 5

Meson Spectroscopy and Decays

5.1 QCD-Inspired Model Hamiltonian

The model Hamiltonian consists of a confining potential, as well as appropriate spin-orbit (including Thomas precession) and spin-spin interactions. The specific form of the potential is as follows

$$\mathbb{H}^{NR} = \mathbb{H}_0^{NR} + \mathbb{H}_{conf}^{NR} + \mathbb{H}_{L\cdot S}^{NR} + \mathbb{H}_{S\cdot S}^{NR} \tag{5.1}$$

The kinetic energy term has the familiar form

$$\mathbb{H}_0^{NR} = \frac{\vec{p}^{\,2}}{2m_q} + \frac{\vec{p}^{\,2}}{2m_{\bar{q}}} \tag{5.2}$$

The QCD-inspired confining potential consists of a Coulomb term and a linear term

$$\mathbb{H}_{conf}^{NR} = \mathbb{G}(r) + \mathbb{S}(r) \tag{5.3}$$

where

$$\mathbb{G}(r) = -\frac{4\alpha_s}{3r} \qquad \text{and} \qquad \mathbb{S}(r) = c + br \tag{5.4}$$

The spin-orbit coupling

$$\mathbb{H}_{L\cdot S}^{NR} = \frac{1}{r}\frac{\partial \mathbb{G}(r)}{\partial r}\left(\frac{1}{m_q} + \frac{1}{m_{\bar{q}}}\right)\left(\frac{\vec{S}_q}{m_q} + \frac{\vec{S}_{\bar{q}}}{m_{\bar{q}}}\right)\cdot\vec{L} - \frac{1}{2r}\frac{\partial \mathbb{H}_{conf}}{\partial r}\left(\frac{\vec{S}_q}{m_q^2} + \frac{\vec{S}_{\bar{q}}}{m_{\bar{q}}^2}\right)\cdot\vec{L} \tag{5.5}$$

The hyperfine interaction

$$\mathbb{H}^{NR}_{S\cdot S} = \frac{2(\vec{S}_q \cdot \vec{S}_{\bar{q}})}{3m_q m_{\bar{q}}}\vec{\nabla}^2\mathbb{G}(r) - \left[\frac{(\vec{S}_q \cdot \hat{r})(\vec{S}_{\overline{q}} \cdot \hat{r}) - \frac{1}{3}\vec{S}_q \cdot \vec{S}_{\overline{q}}}{m_q m_{\bar{q}}}\right]\left(\frac{\partial^2}{\partial r^2} - \frac{1}{r}\frac{\partial}{\partial r}\right)\mathbb{G}(r) \tag{5.6}$$

Since the mesons under investigation consist of light quarks (u, d, and s), it would be desirable to have a relativized form of the Hamiltonian. We use the relativized Hamiltonian of Godfrey and Isgur [28]. Godfrey and Isgur have shown that the relativistic potentials differ from their non-relativistic counterparts in two ways. Firstly, the coordinate, $\vec{r}$, becomes smeared out over distance of the order of the inverse quark masses. Secondly, the coefficients of the various potentials become dependent on the momentum of the interacting quarks. To accomplish the first of these modifications, Godfrey and Isgur smeared a potential, $\mathbb{V}(r)$, in the following way

$$\tilde{\mathbb{V}}_{q\bar{q}}(r) = \int d^3\vec{r}\,'\rho_{q\bar{q}}(\vec{r} - \vec{r}\,')\mathbb{V}(r) \tag{5.7}$$

The smearing function, ρ, is given by

$$\rho_{q\bar{q}}(\vec{r} - \vec{r}\,') = \frac{\sigma^3_{q\bar{q}}}{\pi^{3/2}}e^{-\sigma^2_{q\bar{q}}(\vec{r}-\vec{r}\,')^2} \tag{5.8}$$

where

$$\sigma^2_{q\bar{q}} = \sigma_0^2\left(\frac{1}{2} + \frac{1}{2}\left(\frac{4m_q m_{\bar{q}}}{(m_q + m_{\bar{q}})^2}\right)^4\right) + s^2\left(\frac{2m_q m_{\bar{q}}}{m_q + m_{\bar{q}}}\right)^2 \tag{5.9}$$

and the parameters σ_0 and s are given in Ref. [28].

As far as the momentum dependences are concerned, the Coulomb potential is modified as follows

$$\mathbb{G}(r) \to \left(1 + \frac{p^2}{E_q E_{\bar{q}}}\right)^{1/2}\mathbb{G}(r)\left(1 + \frac{p^2}{E_q E_{\bar{q}}}\right)^{1/2} \tag{5.10}$$

where $E_i = \sqrt{\vec{p}^{\,2} + m_i^2}$. On the basis of solutions to the similar example of QED in one dimension, which confines with a linear potential, Godfrey and Isgur concluded that the QCD linear confining term would not be modified by momentum dependent terms. The

spin-dependent potentials are modified as follows

$$\frac{\mathbb{V}(r)}{m_q m_{\bar{q}}} \to \left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{1/2+\epsilon} \frac{\mathbb{V}(r)}{m_q m_{\bar{q}}} \left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{1/2+\epsilon} \tag{5.11}$$

The parameter ϵ can be different for each type of spin-dependent potential: contact spin-spin(ϵ_c), tensor spin-spin (ϵ_t), vector spin-orbit (ϵ_v), and scalar spin-orbit (ϵ_s). If $\epsilon = 0$, then these modifications have the effect of replacing the quark masses, m, with the quark energies, E. These parameters do turn out to be very small, and their values are given in Ref. [28]. Note that in the non-relativistic limit, all of these momentum dependent terms reduce to unity, giving back the non-relativistic potentials.

In addition to these modifications,wealso adopt Godfrey and Isgur's parameterization of the running coupling constant α_s.

$$\alpha_s(Q^2) = \sum_{k=1}^{3} \alpha_k e^{-Q^2/(4\gamma_k^2)}. \tag{5.12}$$

The parameters are given by $\alpha_k = \{0.25, 0.15, 0.20\}$ and $\gamma_k^2 = \{0.25, 2.5, 250\}$. In the limit as $Q^2 \to 0$, the coupling constant saturates at a value of $\alpha_s^{critical} = 0.6$. This can also be transformed into a function of r,

$$\alpha_s(r) = \sum_{k=1}^{3} \alpha_k \,\mathrm{erf}(\gamma_k r) \tag{5.13}$$

where the error function, $\mathrm{erf}(t) = (2/\sqrt{\pi}) \int_0^t e^{-x^2} dx$.

Applying these modifications to the non-relativistic Hamiltonian, gives the full relativistic Hamiltonian [28]

$$\mathbb{H}^R = \mathbb{H}_0^R + \mathbb{H}_{conf}^R + \mathbb{H}_{L\cdot S}^R + \mathbb{H}_{S\cdot S}^R \tag{5.14}$$

$$\mathbb{H}_0^R = \sqrt{\vec{p}^{\,2} + m_q^2} + \sqrt{\vec{p}^{\,2} + m_{\bar{q}}^2} \tag{5.15}$$

$$\mathbb{H}_{conf}^R = \left(1 + \frac{p^2}{E_q E_{\bar{q}}}\right)^{1/2} \tilde{\mathbb{G}}(r) \left(1 + \frac{p^2}{E_q E_{\bar{q}}}\right)^{1/2} + \tilde{\mathbb{S}}(r) \tag{5.16}$$

$$\tilde{\mathbb{G}}(r) = -\sum_{k=1}^{3} \frac{4\alpha_k}{3r} \,\mathrm{erf}(\tau_{kq\bar{q}}\, r) \tag{5.17}$$

$$\tau_{kq\bar{q}} = \frac{\gamma_k \sigma_{q\bar{q}}}{\sqrt{\gamma_k^2 + \sigma_{q\bar{q}}^2}} \tag{5.18}$$

$$\tilde{\mathbb{S}}(r) = br\left[\frac{e^{-\sigma_{q\bar{q}}^2 r^2}}{\sqrt{\pi}\sigma_{q\bar{q}} r} + \left(1 + \frac{1}{2\sigma_{q\bar{q}}^2 r^2}\right)\mathrm{erf}(\sigma_{q\bar{q}} r)\right] + c \tag{5.19}$$

$$\begin{aligned}
\mathbb{H}^R_{L\cdot S} = &\left(\frac{m_q^2}{E_q^2}\right)^{\frac{1}{2}+\epsilon_v}\left(\frac{\vec{S}_q\cdot\vec{L}}{2m_q^2}\right)\left(\frac{1}{r}\frac{\partial\tilde{\mathbb{G}}(r)}{\partial r}\right)\left(\frac{m_q^2}{E_q^2}\right)^{\frac{1}{2}+\epsilon_v} \\
&+\left(\frac{m_{\bar{q}}^2}{E_{\bar{q}}^2}\right)^{\frac{1}{2}+\epsilon_v}\left(\frac{\vec{S}_{\bar{q}}\cdot\vec{L}}{2m_{\bar{q}}^2}\right)\left(\frac{1}{r}\frac{\partial\tilde{\mathbb{G}}(r)}{\partial r}\right)\left(\frac{m_{\bar{q}}^2}{E_{\bar{q}}^2}\right)^{\frac{1}{2}+\epsilon_v} \\
&+\left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{\frac{1}{2}+\epsilon_v}\left(\frac{(\vec{S}_q+\vec{S}_{\bar{q}})\cdot\vec{L}}{m_q m_{\bar{q}}}\right)\left(\frac{1}{r}\frac{\partial\tilde{\mathbb{G}}(r)}{\partial r}\right)\left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{\frac{1}{2}+\epsilon_v} \\
&-\left(\frac{m_q^2}{E_q^2}\right)^{\frac{1}{2}+\epsilon_s}\left(\frac{\vec{S}_q\cdot\vec{L}}{2m_q^2}\right)\left(\frac{1}{r}\frac{\partial\tilde{\mathbb{S}}(r)}{\partial r}\right)\left(\frac{m_q^2}{E_q^2}\right)^{\frac{1}{2}+\epsilon_s} \\
&-\left(\frac{m_{\bar{q}}^2}{E_{\bar{q}}^2}\right)^{\frac{1}{2}+\epsilon_s}\left(\frac{\vec{S}_{\bar{q}}\cdot\vec{L}}{2m_{\bar{q}}^2}\right)\left(\frac{1}{r}\frac{\partial\tilde{\mathbb{S}}(r)}{\partial r}\right)\left(\frac{m_{\bar{q}}^2}{E_{\bar{q}}^2}\right)^{\frac{1}{2}+\epsilon_s}
\end{aligned} \tag{5.20}$$

$$\begin{aligned}
\mathbb{H}^R_{S\cdot S} = &\left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{\frac{1}{2}+\epsilon_c}\left(\frac{2(\vec{S}_q\cdot\vec{S}_{\bar{q}})}{3m_q m_{\bar{q}}}\right)\left(\vec{\nabla}^2\tilde{\mathbb{G}}(r)\right)\left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{\frac{1}{2}+\epsilon_c} \\
&-\left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{\frac{1}{2}+\epsilon_t}\left(\frac{(\vec{S}_q\cdot\hat{r})(\vec{S}_{\bar{q}}\cdot\hat{r}) - \frac{1}{3}\vec{S}_q\cdot\vec{S}_{\bar{q}}}{m_q m_{\bar{q}}}\right)\left(\frac{\partial^2}{\partial r^2} - \frac{1}{r}\frac{\partial}{\partial r}\right)\tilde{\mathbb{G}}(r)\left(\frac{m_q m_{\bar{q}}}{E_q E_{\bar{q}}}\right)^{\frac{1}{2}+\epsilon_t}
\end{aligned} \tag{5.21}$$

5.2 Spectrum Calculation

The spectrum calculation is done explicitly in the ordinary equal-time formulation, and the resulting wave functions are then transformed into light-front wave functions. This works because the spectrum is, of course, calculated in the rest frame of the meson. The Hamiltonian is the total energy of the meson, which in the equal-time formulation is simply P^0_{meson} and in the light-front formalism is $P^-_{meson} = P^0_{meson} - P^3_{meson}$. Since $P^3_{meson} = 0$ in the meson's rest frame, the equal-time and light-front Hamiltonians are equivalent here.

To compute the meson spectrum, the Hamiltonian matrix is written in the simple-harmonic oscillator(SHO) basis and then diagonalized. In Godfrey and Isgur's original analysis they used as many as 15 SHO basis states. In this work,we have employed 25 SHO

basis states with very similar results. In position–space, the SHO wave functions are

$$\Psi_{n\ell m}(r,\theta_{\hat{r}},\phi_{\hat{r}}) = \mathcal{R}_{n,\ell}(r)Y_\ell^m(\theta_{\hat{r}},\phi_{\hat{r}}) \tag{5.22}$$

$$\mathcal{R}_{n,\ell}(r) = N_r^{(n,\ell)} r^\ell e^{-\frac{\beta^2 r^2}{2}} L_n^{\ell+\frac{1}{2}}(\beta^2 r^2) \tag{5.23}$$

where

$$N_r^{(n,\ell)} = \sqrt{\frac{2\beta^{2\ell+3}}{\Gamma(\ell+\frac{3}{2})\frac{(n+\ell+\frac{1}{2})!}{n!(\ell+\frac{1}{2})!}}} \tag{5.24}$$

is a normalization constant, $L_n^{\ell+\frac{1}{2}}(\beta^2 r^2)$ is the generalized Laguerre polynomial, and the function $Y_\ell^m(\theta_{\hat{r}},\phi_{\hat{r}})$ is the standard spherical harmonic. Here, $n \in \{0,1,2,\cdots\}$, $\ell \in \{0,1,2,\cdots\}$, and $m \in \{\cdots,-2,-1,0,1,2,\cdots\}$. Note that the radial and angular parts are normalized separately such that $\int_0^\infty r^2|\mathcal{R}_{n,\ell}(r)|^2 dr = 1$ and $\int |Y_\ell^m(\theta_{\hat{r}},\phi_{\hat{r}})|^2 d\Omega = 1$. This can be Fourier transformed into momentum space which yields

$$\Phi_{n\ell m}(p,\theta_{\hat{p}},\phi_{\hat{p}}) = \mathcal{P}_{n,\ell}(p)Y_\ell^m(\theta_{\hat{p}},\phi_{\hat{p}}) \tag{5.25}$$

$$\mathcal{P}_{n,\ell}(p) = N_p^{(n,\ell)}(-1)^n(ip)^\ell e^{-\frac{p^2}{2\beta^2}} L_n^{\ell+\frac{1}{2}}\left(\frac{p^2}{\beta^2}\right) \tag{5.26}$$

where the normalization constant $N_p^{(n,\ell)} = [(2\pi)^{3/2}/\beta^{2\ell+3}]N_r^{(n,\ell)}$, and the normalizations $\int_0^\infty p^2|\mathcal{P}_{n,\ell}(p)|^2 dp = 1$ and $\int |Y_\ell^m(\theta_{\hat{p}},\phi_{\hat{p}})|^2 d\Omega = 1$ apply.

Since the mesons are assumed to be superpositions of 25 SHO basis states, the position-space radial wave function for a meson with orbital angular momentum, L, total spin, S, and total angular momentum, J, is

$$\phi_{L,S}^J(r) = \sum_{n=0}^{24} (\mathcal{C}_{L,S}^J)_n \mathcal{R}_{n,L} \tag{5.27}$$

where the coefficients $(\mathcal{C}_{L,S}^J)_n$ are determined by diagonalizing the Hamiltonian. The spin-orbit wave functions for each of the sectors are given below and are written using spectroscopic notation in the form $|^{(2S+1)}L_J; M\rangle$ where M is the z-component of the total angular momentum, J. In this work,wefocus on the $L = 0$ and $L = 1$ mesons, so we have only

presented the spin-orbit wave functions for these sectors. For pseudoscalar mesons,

$$\underline{{}^1S_0(J^{PC}=0^{-+})}$$
$$|{}^1S_0;\quad 0\rangle = Y_0^0(\theta,\phi)|0,0\rangle. \tag{5.28}$$

For the vector mesons,

$$\underline{{}^3S_1(J^{PC}=1^{--})}$$
$$|{}^3S_1;+1\rangle = Y_0^0(\theta,\phi)|1,1\rangle$$
$$|{}^3S_1;\quad 0\rangle = Y_0^0(\theta,\phi)|1,0\rangle$$
$$|{}^3S_1;-1\rangle = Y_0^0(\theta,\phi)|1,-1\rangle. \tag{5.29}$$

For the scalar mesons,

$$\underline{{}^3P_0(J^{PC}=0^{++})}$$
$$|{}^3P_0;\quad 0\rangle = \frac{1}{\sqrt{3}}\left[Y_1^{-1}(\theta,\phi)|1,1\rangle - Y_1^0(\theta,\phi)|1,0\rangle + Y_1^1(\theta,\phi)|1,-1\rangle\right]. \tag{5.30}$$

For the axial vector mesons with odd charge conjugation,

$$\underline{{}^1P_1(J^{PC}=1^{+-})}$$
$$|{}^1P_1;+1\rangle = Y_1^1(\theta,\phi)|0,0\rangle$$
$$|{}^1P_1;\quad 0\rangle = Y_1^0(\theta,\phi)|0,0\rangle$$
$$|{}^1P_1;-1\rangle = Y_1^{-1}(\theta,\phi)|0,0\rangle. \tag{5.31}$$

For the axial-vector mesons with even charge conjugation,

$$\underline{{}^3P_1(J^{PC}=1^{++})}$$
$$|{}^3P_1;+1\rangle = \frac{1}{\sqrt{2}}\left[Y_1^0(\theta,\phi)|1,1\rangle - Y_1^1(\theta,\phi)|1,0\rangle\right]$$
$$|{}^3P_1;\quad 0\rangle = \frac{1}{\sqrt{2}}\left[Y_1^{-1}(\theta,\phi)|1,1\rangle - Y_1^1(\theta,\phi)|1,-1\rangle\right]$$
$$|{}^3P_1;-1\rangle = \frac{1}{\sqrt{2}}\left[Y_1^{-1}(\theta,\phi)|1,0\rangle - Y_1^0(\theta,\phi)|1,-1\rangle\right]. \tag{5.32}$$

And, for the tensor mesons,

$$\begin{aligned}
&\underline{{}^3P_2(J^{PC}=2^{++})}\\
|{}^3P_2;+2\rangle &= Y_1^1(\theta,\phi)|1,1\rangle\\
|{}^3P_2;+1\rangle &= \frac{1}{\sqrt{2}}\left[Y_1^0(\theta,\phi)|1,1\rangle + Y_1^1(\theta,\phi)|1,0\rangle\right]\\
|{}^3P_2;\quad 0\rangle &= \frac{1}{\sqrt{6}}\left[Y_1^{-1}(\theta,\phi)|1,1\rangle + 2Y_1^0(\theta,\phi)|1,0\rangle + Y_1^1(\theta,\phi)|1,-1\rangle\right]\\
|{}^3P_2;-1\rangle &= \frac{1}{\sqrt{2}}\left[Y_1^{-1}(\theta,\phi)|1,0\rangle + Y_1^0(\theta,\phi)|1,-1\rangle\right]\\
|{}^3P_2;-2\rangle &= Y_1^{-1}(\theta,\phi)|1,-1\rangle.
\end{aligned} \tag{5.33}$$

The spin states in the above spin-orbit wave functions are the standard ones,

$$\begin{aligned}
|1,1\rangle &= \left|\frac{1}{2},\frac{1}{2}\right\rangle_q \left|\frac{1}{2},\frac{1}{2}\right\rangle_{\bar{q}}\\
|1,0\rangle &= \frac{1}{\sqrt{2}}\left[\left|\frac{1}{2},\frac{1}{2}\right\rangle_q \left|\frac{1}{2},-\frac{1}{2}\right\rangle_{\bar{q}} + \left|\frac{1}{2},-\frac{1}{2}\right\rangle_q \left|\frac{1}{2},\frac{1}{2}\right\rangle_{\bar{q}}\right]\\
|1,-1\rangle &= \left|\frac{1}{2},-\frac{1}{2}\right\rangle_q \left|\frac{1}{2},-\frac{1}{2}\right\rangle_{\bar{q}}\\
|0,0\rangle &= \frac{1}{\sqrt{2}}\left[\left|\frac{1}{2},\frac{1}{2}\right\rangle_q \left|\frac{1}{2},-\frac{1}{2}\right\rangle_{\bar{q}} - \left|\frac{1}{2},-\frac{1}{2}\right\rangle_q \left|\frac{1}{2},\frac{1}{2}\right\rangle_{\bar{q}}\right]
\end{aligned} \tag{5.34}$$

Using the meson wave functions, the matrix elements of the Hamiltonian

$$\langle {}^{(2S+1)}L_J;M|\langle n,L|\mathbb{H}^R|n',L\rangle|{}^{(2S+1)}L_J;M\rangle \tag{5.35}$$

are computed in each sector separately. Here, $|n,L\rangle$ denotes the SHO radial wave functions. In other words, $\langle r|n,L\rangle = \mathcal{R}_{n,L}$ or $\langle p|n,L\rangle = \mathcal{P}_{n,L}$. The expectation values of the spin-dependent operators can be found analytically, and the results of these calculations are summarized in Table 5.1.

After the expectation values of the spin-dependent parts of the potentials have been evaluated, it remains to evaluate the expectation values of the position-dependent and momentum-dependent parts. What complicates this slightly is that many of the potentials contain products of position and momentum operators. The form of these potentials is $\mathbb{V}(r,p) = f(p)g(r)f(p)$, where the function $f(p)$ contains only momentum operators and

Table 5.1: Expectation values of spin-dependent terms in the Hamiltonian.

	$^1S_0(0^{-+})$	$^3S_1(1^{--})$	$^1P_1(1^{+-})$	$^3P_0(0^{++})$	$^3P_1(1^{++})$	$^3P_2(2^{++})$
$\langle \vec{S}_q \cdot \vec{L} \rangle$	0	0	0	-1	$-\frac{1}{2}$	$+\frac{1}{2}$
$\langle \vec{S}_{\bar{q}} \cdot \vec{L} \rangle$	0	0	0	-1	$-\frac{1}{2}$	$+\frac{1}{2}$
$\langle (\vec{S}_q + \vec{S}_{\bar{q}}) \cdot \vec{L} \rangle$	0	0	0	-2	-1	$+1$
$\langle \vec{S}_q \cdot \vec{S}_{\bar{q}} \rangle$	$-\frac{3}{4}$	$+\frac{1}{4}$	$-\frac{3}{4}$	$+\frac{1}{4}$	$+\frac{1}{4}$	$\frac{1}{4}$
$\langle (\vec{S}_q \cdot \hat{r})(\vec{S}_{\bar{q}} \cdot \hat{r}) - \frac{1}{3}\vec{S}_q \cdot \vec{S}_{\bar{q}} \rangle$	0	0	0	$-\frac{1}{3}$	$+\frac{1}{6}$	$-\frac{1}{30}$

the function $g(r)$ contains only position operators. To evaluate the expectation values of such potentials, they are first rewritten by inserting two complete sets of basis states as follows [28],

$$\langle n, L|f(p)g(r)f(p)|n', L\rangle = \sum_{n''}\sum_{n'''}\langle n, L|f(p)|n'', L\rangle\langle n'', L|g(r)|n''', L\rangle\langle n''', L|f(p)|n', L\rangle. \tag{5.36}$$

Written in this way, the expectation value of the momentum-dependent parts of the potential can be evaluated separately from the position-dependent parts of the potential. While the sums over n'' and n''' should go from $0 \rightarrow \infty$, this is not practically possible. Therefore, the sum runs as high as is needed for the full expectation value of the potential to converge sufficiently. In this work where n and n' run from $0 \rightarrow 24$, the maximum values of n'' or n''' were typically around 30–35. These expectation values, and hence the matrix elements of the full Hamiltonian, were evaluated numerically using a set of FORTRAN codes. Copies of the codes can be found in Appendices 2-4.

The first of these codes, "vf.f", computes the expectation values for the momenutm-dependent parts of all the terms in the Hamiltonian. For each term, it outputs a matrix whose elements are $\langle n, L|f(p)|n'', L\rangle$, which are, of course, equivalent to the matrix elements $\langle n''', L|f(p)|n', L\rangle$. The code "vg.f" computes the expectation values for the position-dependent parts, and for each potential outputs a matrix whose elements are $\langle n'', L|g(r)|n''', L\rangle$. The third code, "diag.f", reads in the matrices output by "vf.f" and

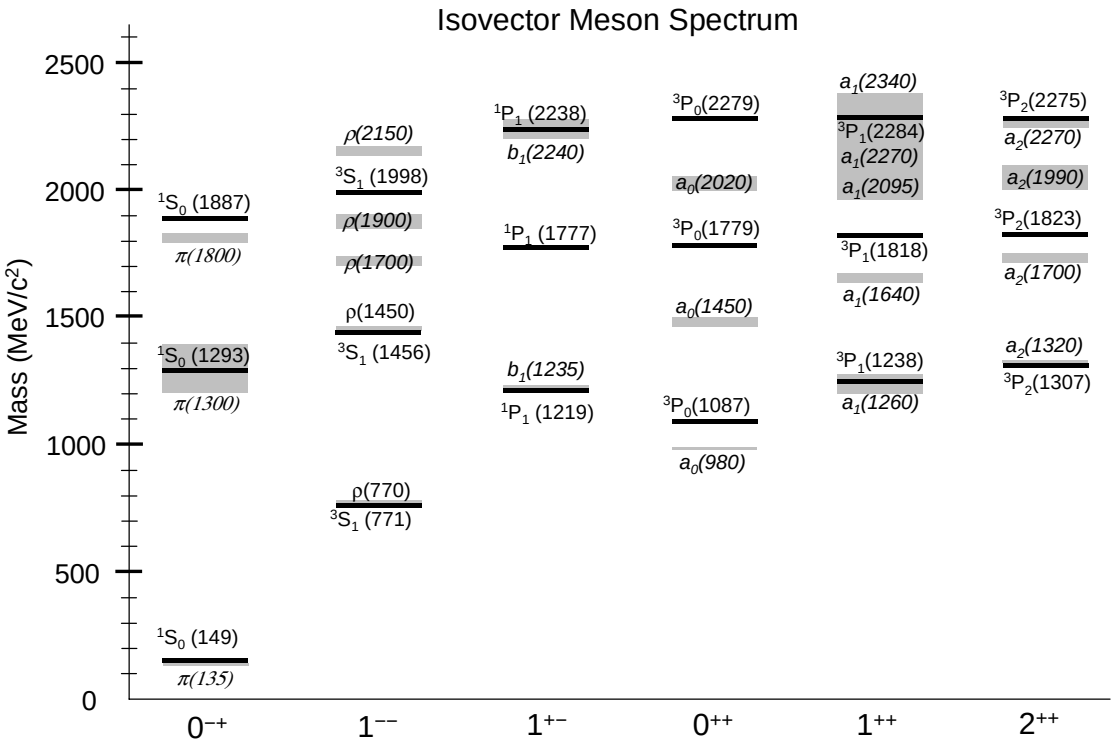

Figure 5.1: The spectrum of isovector mesons compared with experimental data. The black bars represent quark model predictions, while the grey shaded areas show the experimental data including uncertainties.

"vg.f", and multiplies them together so as to produce the full matrix elements for each term, $\langle n, L|f(p)g(r)f(p)|n', L\rangle$. Next, the code multiplies the matrix for each spin-dependent potential by the appropriate spin-dependent term from Table 5.1, and then adds all of these matrices to the matrices for the kinetic energy to produce the full Hamiltonian matrix. The code then diagonalizes the Hamiltonian matrix and outputs the meson masses (eigenvalues) and wave functions (eigenvectors).

The results of the spectrum calculation are presented in Figures 5.1– 5.3. Figure 5.1 gives the spectrum of isovector mesons ($u\bar{d}, d\bar{u}, \frac{1}{\sqrt{2}}(u\bar{u} - d\bar{d})$). With the exception of the scalar mesons (0^{++}), the ground states are in very good agreement with the data. The results for the radial excitations are somewhat mixed. For the 0^{-+} and 1^{--} mesons, the first excited states are in fairly good agreement while the second excited states are about 80–90 MeV higher than the experimentally observed states. Note that the calculated state $^3S_1(1998)$ should be identified with the $\rho(1900)$. The $\rho(1700)$ can be identified with the

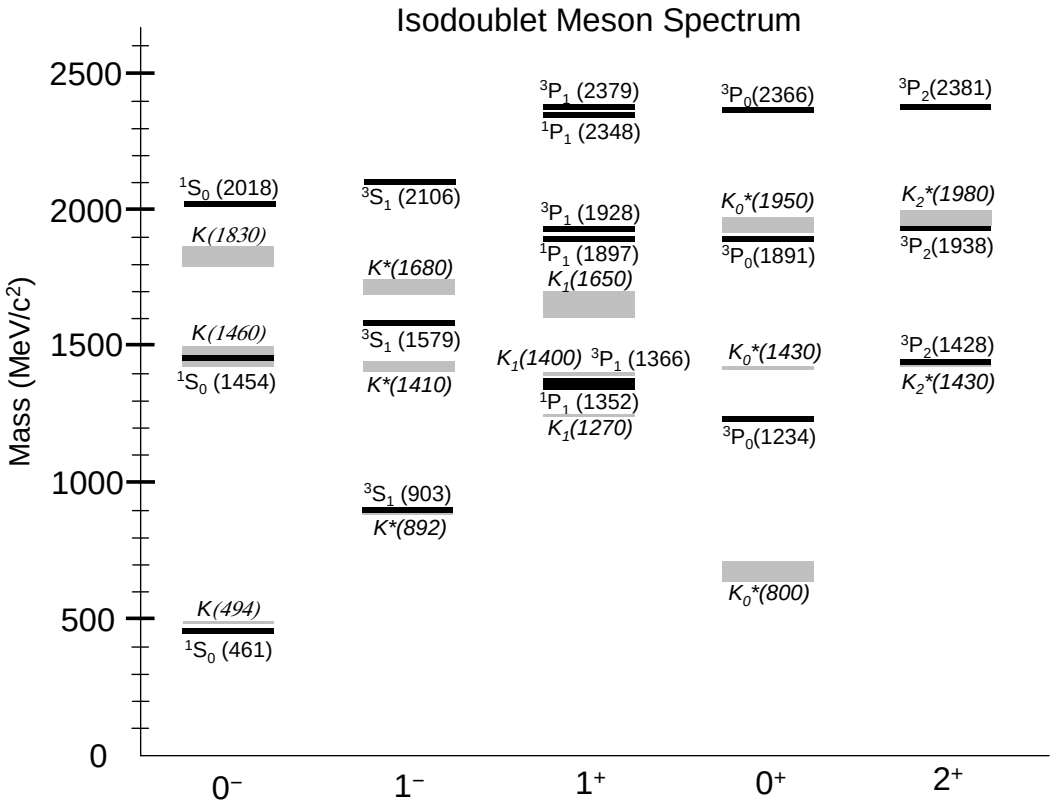

Figure 5.2: The spectrum of isodoublet mesons compared with experimental data. The black bars represent quark model predictions, while the grey shaded areas show the experimental data including uncertainties.

ground state D-wave 1^{--} meson which has not been calculated in this work. Godfrey and Isgur obtained a mass of 1660 MeV for this state in their work [28]. For the 1^{++} and 2^{++} mesons the first excited states are anywhere from 120–170 MeV higher than the observed states, while the second excited states are in fairly good agreement with the data. Here, the $^3P_2(1823)$ is identified with the $a_2(1700)$. The $a_2(1990)$ can be identified with the ground state F-wave 2^{++} meson which has not been calculated here, but for which Godfrey and Isgur obtained a mass of 2050 MeV [28]. All of the 0^{++} states are in poor agreement with the data which lends further support to the idea that they must be described by some structure other than simply $q\bar{q}$.

The strange mesons ($u\bar{s}$,$d\bar{s}$,$s\bar{u}$,$s\bar{d}$) are shown in Figure 5.2. With the exception of the 1^+ $^1P_1$ and 0^+ mesons, the ground states are all in excellent agreement with the data. The $^1P_1$ ground state lies about 80 MeV above the experimental ground state, which is

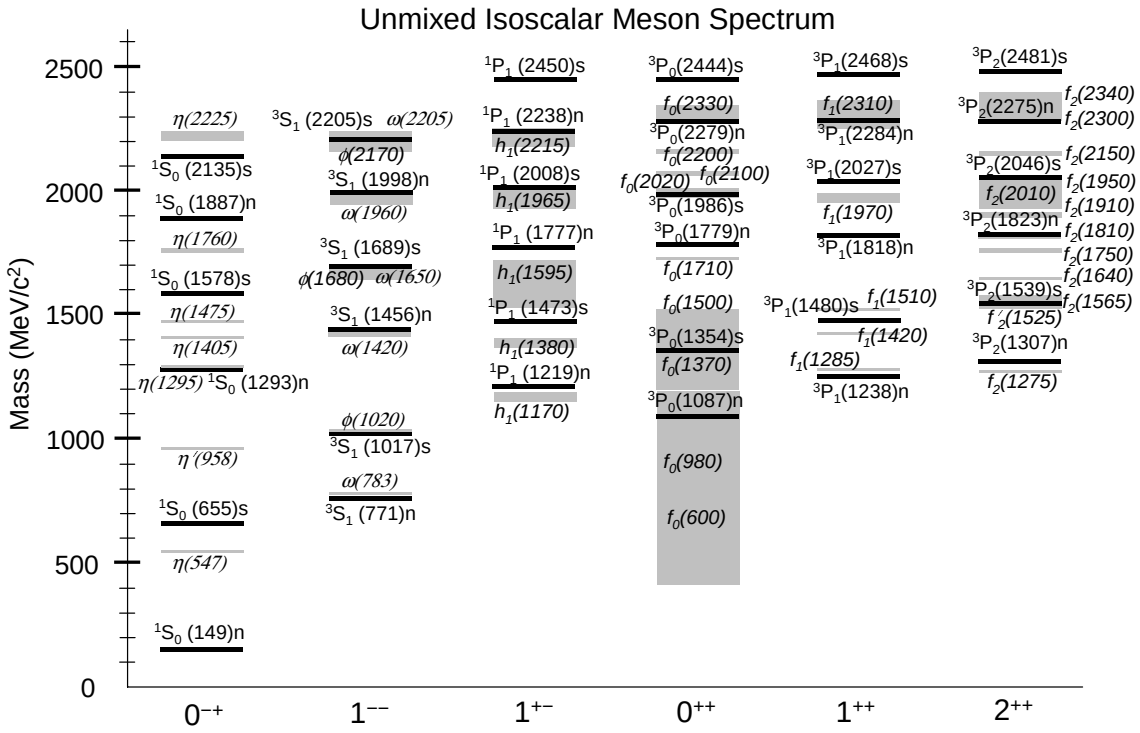

Figure 5.3: The spectrum of isoscalar mesons compared with experimental data. The black bars represent quark model predictions, while the grey shaded areas show the experimental data including uncertainties.

relatively fair agreement when compared with the 560 MeV difference between the $K_0^*(800)$ and the calculated 0^+ ground state at 1234 MeV. There is mixed success for the radially excited states. While the first excited states of the 0^- and 2^+ mesons are in excellent agreement, the calculated $1-$ state lies about 170 MeV above the $K^*(1410)$. Note that the $K^*(1680)$ can be identified with the D-wave ground state meson which has not been found here, but for which Godfrey and Isgur obtained a mass of about 1780 MeV [28]. In the 1^+ sector, both the $^1P_1(1897)$ and $^3P_1(1928)$ states lie over 200 MeV above the $K_1(1650)$. In the scalar sector, it is conceivable that one could identify the $^3P_0(1891)$ with the $K_0^*(1950)$. However this would require some other non-$q\bar{q}$ structure to mix with the $^3P_0(1234)$ to produce the $K_0^*(800)$ and the $K_0^*(1430)$.

The unmixed isoscalar mesons ($n\bar{n} = \frac{1}{\sqrt{2}}(u\bar{u} + d\bar{d})$ and $s\bar{s}$) are shown in Figure 5.3. It is clear that the vector mesons are in excellent agreement with the data. As such, there is very little, if any, mixing among these isoscalar states. In contrast, the pseudoscalar

mesons require a significant amount of mixing to reproduce the $\eta(547)$ and $\eta'(958)$ as well as the radially excited states. The 1^{+-}, 1^{++}, and 2^{++} mesons are in fair agreement, but will clearly require a small degree of mixing to match the data. It is difficult to compare the scalar mesons given that there are such large uncertainties in the data.

5.3 Mixing for Pseudoscalar Mesons

Since we will be calculating the decays of scalar mesons to pairs of pseudoscalar mesons, we first examine the simpler case of pseudoscalar mixing. To investigate the mixing of the ground state isoscalar $u\bar{u}$, $d\bar{d}$, and $s\bar{s}$ pseudoscalar mesons we follow the method of Scadron [48, 55]. Scadron parameterizes the annihilation graphs which contribute to the mixing with two parameters, β and X. The parameter β describes the strength of the u/d annihilation diagrams. Therefore, diagrams in which a $u\bar{u}$ pair annihilates and then produces another $u\bar{u}$ pair are collectively taken to have the value β. Likewise for $d\bar{d} \to d\bar{d}$. The parameter X is a multiplicative SU(3) symmetry breaking factor which is motivated by the factorized structure of the annihilation diagrams. In diagrams where a $u\bar{u}$ pair annihilates and produces an $s\bar{s}$, the strength is assigned a value of $X\beta$. Annihilation diagrams for $s\bar{s} \to s\bar{s}$ would be assigned a value of $X^2\beta$. When these contributions are taken to be small perturbations to the mass-squared matrix, they can be written in the following way:

$$\begin{aligned} \hat{\mathfrak{M}} &= \begin{pmatrix} m_{u\bar{u}}^2 & 0 & 0 \\ 0 & m_{d\bar{d}}^2 & 0 \\ 0 & 0 & m_{s\bar{s}}^2 \end{pmatrix} + \begin{pmatrix} \beta & \beta & X\beta \\ \beta & \beta & X\beta \\ X\beta & X\beta & X^2\beta \end{pmatrix} \\ &= \begin{pmatrix} m_{u\bar{u}}^2 + \beta & \beta & X\beta \\ \beta & m_{d\bar{d}}^2 + \beta & X\beta \\ X\beta & X\beta & m_{s\bar{s}}^2 + X^2\beta \end{pmatrix}. \end{aligned} \tag{5.37}$$

To separate out the isoscalar component from the isovector component one must transform the $u\bar{u}$ and $d\bar{d}$ components into $\frac{1}{\sqrt{2}}(u\bar{u}+d\bar{d})$ and $\frac{1}{\sqrt{2}}(u\bar{u}-d\bar{d})$. Doing this yields

$$\hat{\mathfrak{M}} = \begin{pmatrix} \frac{1}{2}(m_{u\bar{u}}^2 + m_{d\bar{d}}^2) & 0 & 0 \\ 0 & \frac{1}{2}(m_{u\bar{u}}^2 + m_{d\bar{d}}^2) + 2\beta & \sqrt{2}X\beta \\ 0 & \sqrt{2}X\beta & m_{s\bar{s}}^2 + X^2\beta \end{pmatrix} \tag{5.38}$$

Here, the isovector (I=1) has been isolated, and the 2×2 matrix remaining in the lower right corner of the matrix above describes the isoscalars (I=0). Since $m_{u\bar{u}} = m_{d\bar{d}}$,we will define this as $m_{n\bar{n}}$. So, the 2×2 matrix that describes the mixing of the isoscalar mesons in a given sector is

$$\hat{\mathfrak{M}}_{(I=0)} = \begin{pmatrix} m_{n\bar{n}}^2 + 2\beta & \sqrt{2}X\beta \\ \sqrt{2}X\beta & m_{s\bar{s}}^2 + X^2\beta \end{pmatrix}. \tag{5.39}$$

When this mass-squared matrix is diagonalized, the resulting eigenvalues are the physical meson masses. Upon diagonalization, both the trace and the determinant of the matrix are unchanged. From these two conditions, one can solve for the parameters X and β in terms of the unmixed masses and the physical meson masses (M_1 and M_2). This gives [55]

$$\beta = \frac{(M_2^2 - m_{n\bar{n}}^2)(M_1^2 - m_{n\bar{n}}^2)}{2(m_{s\bar{s}}^2 - m_{n\bar{n}}^2)} \tag{5.40}$$

$$X = \left(\frac{2(m_{s\bar{s}}^2 - M_1^2)(M_2^2 - m_{s\bar{s}}^2)}{(M_1^2 - m_{n\bar{n}}^2)(M_2^2 - m_{n\bar{n}}^2)} \right)^{\frac{1}{2}} \tag{5.41}$$

Using the isoscalar $n\bar{n}$ and $s\bar{s}$ masses from the spectrum calculation of the previous section to solve for the mixing parameters in the case of the pseudoscalar mesons, η and η', gives

$$m_{n\bar{n}} = 0.149\ GeV \quad m_{s\bar{s}} = 0.655\ GeV \quad M_1 = 0.548\ GeV \quad M_2 = 0.958\ GeV$$

$$\beta_{0^{-+}} = 0.306\ GeV^2$$

$$X_{0^{-+}} = 0.711$$

Substituting these parameters back into Eq 5.39 and diagonalizing yields the following eigenvectors

$$|\eta\rangle = 0.736|n\bar{n}\rangle - 0.677|s\bar{s}\rangle$$
$$|\eta'\rangle = 0.677|n\bar{n}\rangle + 0.736|s\bar{s}\rangle \ . \tag{5.42}$$

Writing the basis transformation in terms of a mixing angle, ϕ,

$$|\eta\rangle = \cos(\phi)|n\bar{n}\rangle - \sin(\phi)|s\bar{s}\rangle$$
$$|\eta'\rangle = \sin(\phi)|n\bar{n}\rangle + \cos(\phi)|s\bar{s}\rangle \ . \tag{5.43}$$

it can be seen that $\phi = 42.6°$. The physical states can also be written in the octet–singlet basis as

$$|\eta\rangle = \cos(\theta)|8\rangle - \sin(\theta)|1\rangle$$
$$|\eta'\rangle = \sin(\theta)|8\rangle + \cos(\theta)|1\rangle \ , \tag{5.44}$$

where $|8\rangle = \frac{|u\bar{u}\rangle + |d\bar{d}\rangle - 2|s\bar{s}\rangle}{\sqrt{6}}$ and $|1\rangle = \frac{|u\bar{u}\rangle + |d\bar{d}\rangle + |s\bar{s}\rangle}{\sqrt{3}}$. The octet–singlet mixing angle, θ, is related to ϕ by

$$\theta = \phi - \tan^{-1}(\sqrt{2}) \equiv \phi - 55° \ .$$

Therefore, $\theta = -12.4°$. These are in excellent agreement with Scadron's values of $\phi_{Scadron} = 42°$ and $\theta_{Scadron} = -13°$ [28]. In Scadron's analysis he had replaced $m^2_{n\bar{n}}$ with m^2_π, and $m^2_{s\bar{s}}$ with $(2m^2_K - m^2_\pi)$. The similarity in the mixing angles is due to the fact that the quark model value of $m_{s\bar{s}} = 655$ MeV is so close to the value of $(2m^2_K - m^2_\pi) = 684$ MeV.

5.4 Scalar Meson and Glueball Mixing

5.4.1 Assumptions

Before diving into the details of the mixing and decay calculations, we would first like to lay out the basic assumptions that underlie our analysis.

1. We assume that the unmixed $n\bar{n}$ and $s\bar{s}$ masses are given by the quark model spectrum. This gives the values of $m_{n\bar{n}} = 1090\ MeV$ and $m_{s\bar{s}} = 1354\ MeV$. The glueball is

taken to have a mass of $m_{gg} = 1611\ MeV$, consistent with Lattice QCD predictions. Many other scalar meson analyses [42, 56, 57, 58, 59, 60] have unmixed $n\bar{n}$ and $s\bar{s}$ masses that are anywhere from 200–400 MeV higher than the quark model values.

2. Previous analyses have considered the mixing between the $n\bar{n}$ and $s\bar{s}$ quarkonia to be negligible compared to glueball–quarkonia mixing (*i.e.* glueball dominance). It is certainly because of this assumption that the unmixed $n\bar{n}$ and $s\bar{s}$ masses are significantly larger that the quark model values we use. We do not assume glueball dominance, but leave the $q\bar{q}$-$q\bar{q}$ mixing in to make the scheme as general as possible.

3. We assume that hadronic decay processes will be dominated by the quarkonia, so that glueball contributions can be neglected. Later, in Section 5.4.6, we relax this assumption and estimate the effect of including the glueball contributions. This turns out to be necessary in order to satisfactorily explain the hadronic decay data.

With these points understood, we proceed to describe the specific mixing scheme used.

5.4.2 The Mixing Scheme

The focus of this analysis is to test whether or not the isoscalar 0^{++} mesons $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ are consistent with being mixtures of $n\bar{n}$, $s\bar{s}$, and the scalar glueball, gg. This can be written as

$$\begin{pmatrix} |f_0(1370)\rangle \\ |f_0(1500)\rangle \\ |f_0(1710)\rangle \end{pmatrix} = U \begin{pmatrix} |n\bar{n}\rangle \\ |s\bar{s}\rangle \\ |gg\rangle \end{pmatrix} \equiv \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \begin{pmatrix} |n\bar{n}\rangle \\ |s\bar{s}\rangle \\ |gg\rangle \end{pmatrix} \tag{5.45}$$

where the basis transformation has been denoted by U since it is in fact unitary.

To begin, the mass-squared matrix for these states must be written. The anihilation diagrams which contribute to the mixing will be parameterized in a similar way as for the pseudoscalar mesons. The parameter β will be used to represent $n\bar{n} \to n\bar{n}$ mixing, where $n = u$ or d. Also, X will represent the multiplicative SU(3) symmetry breaking factor, so that the contribution of $n\bar{n} \to s\bar{s}$ mixing will be given by $X\beta$ and the contribution of $s\bar{s} \to s\bar{s}$ mixing will be given as $X^2\beta$. For the $gg \to n\bar{n}$ mixing,weintroduce the parameter γ, and so $gg \to s\bar{s}$ mixing will be given by $X\gamma$. Finally, there will be annihilation diagrams

for $gg \to gg$ which we will call δ. So, the mass-squared matrix becomes

$$\hat{\mathfrak{M}} = \begin{pmatrix} m_{u\bar{u}}^2 & 0 & 0 & 0 \\ 0 & m_{d\bar{d}}^2 & 0 & 0 \\ 0 & 0 & m_{s\bar{s}}^2 & 0 \\ 0 & 0 & 0 & m_{gg}^2 \end{pmatrix} + \begin{pmatrix} \beta & \beta & X\beta & \gamma \\ \beta & \beta & X\beta & \gamma \\ X\beta & X\beta & X^2\beta & X\gamma \\ \gamma & \gamma & X\gamma & \delta \end{pmatrix}$$

$$= \begin{pmatrix} m_{u\bar{u}}^2 + \beta & \beta & X\beta & \gamma \\ \beta & m_{d\bar{d}}^2 + \beta & X\beta & \gamma \\ X\beta & X\beta & m_{s\bar{s}}^2 + X^2\beta & X\gamma \\ \gamma & \gamma & X\gamma & m_{gg}^2 + \delta \end{pmatrix}. \tag{5.46}$$

Rewriting this in the $\frac{1}{\sqrt{2}}(u\bar{u} + d\bar{d})$ and $\frac{1}{\sqrt{2}}(u\bar{u} - d\bar{d})$ basis to separate out the isoscalar component from the isovector component gives

$$\hat{\mathfrak{M}}_{(I=0)} = \begin{pmatrix} m_{n\bar{n}}^2 + 2\beta & \sqrt{2}X\beta & \sqrt{2}\gamma \\ \sqrt{2}X\beta & m_{s\bar{s}}^2 + X^2\beta & X\gamma \\ \sqrt{2}\gamma & X\gamma & m_{gg}^2 + \delta \end{pmatrix}. \tag{5.47}$$

Upon diagonalization, this should give the physical isoscalar meson masses.

$$U^\dagger \hat{\mathfrak{M}} U = \mathfrak{M} = \begin{pmatrix} M_{f_0(1370)}^2 & 0 & 0 \\ 0 & M_{f_0(1500)}^2 & 0 \\ 0 & 0 & M_{f_0(1710)}^2 \end{pmatrix} \tag{5.48}$$

This mass-squared matrix consists of four parameters which describe the strength of the mixing. There are the following relationships between the undiagonalized matrix, $\hat{\mathfrak{M}}$ and the diagonal matrix $\mathfrak{M}$:

$$\mathrm{Tr}(\hat{\mathfrak{M}}) = \mathrm{Tr}(\mathfrak{M}) \ , \tag{5.49}$$

$$\det(\hat{\mathfrak{M}}) = \det(\mathfrak{M}) \ , \tag{5.50}$$

$$\det(\hat{\mathfrak{M}} - IM_i^2) = 0. \tag{5.51}$$

Equation 5.51 constitues three equations, one for each eigenvalue($i = 1, 2, 3$). Unfortunately, only three of the five equations above are independent. To uniquely determine the mixing

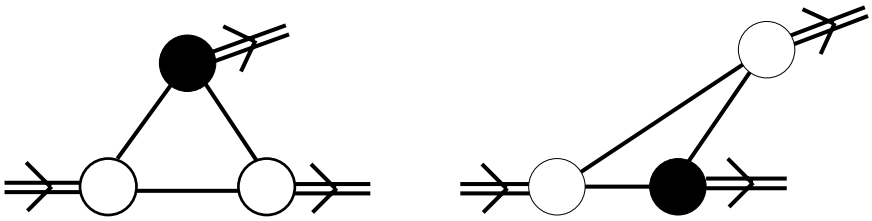

Figure 5.4: Light–Front time ordered diagrams for the decay of a scalar meson to two pseudoscalar mesons. The white circles represent light–front wave function vertices, and the black circles represent light–front non–wave function vertices.

parameters, a fouth constraint is needed. This final constraint will be taken to be the $\pi\pi$ decay width of the $f_0(1500)$. To see how this final constraint will be taken into account,we now turn to a detailed discussion of how the hadronic decay widths are computed in the light–front quark model.

5.4.3 Hadronic Decays

Calculating hadronic decays in the light–front quark model presents a challenge because all of the relevant diagrams contain a non–wave function vertex which cannot simply be assumed to be given by the light–front wave function. Figure 5.4 shows the two light–front time ordered diagrams involved in the computation of the decay of a scalar meson to a pair of pseudoscalar mesons. The white circle represents the ordinary light–front wave function, while the black circle represents the non–wave function vertex. In Ref. [61], an effective treatment of this non-wave function vertex is developed and applied to $K_{\ell 3}$ and $D^0 \to K^-\ell^+\nu_\ell$. The authors follow a Schwinger-Dyson type of approach to relate the non-wave function vertex to the ordinary light–front wave function. The essential idea is illustrated in Fig. 5.5. Here, the meson non-wave function vertex, denoted by G_{meson}, can be written as

$$G_{meson}(x,\vec{k}_\perp) = \int dy \int d^2\vec{\ell}_\perp \mathcal{K}(x,\vec{k}_\perp;y,\vec{\ell}_\perp)\Psi(y,\vec{\ell}_\perp). \tag{5.52}$$

While the kernel, $\mathcal{K}$, is in general dependent on all the internal momenta (x,$\vec{k}_\perp$,y, and $\vec{\ell}_\perp$), an average over y and $\vec{\ell}_\perp$ would depend only on x and $\vec{k}_\perp$. If the value of this average does not change significantly over the range of x and $\vec{k}_\perp$, then G_{meson} can be approximated simply as a constant. We will illustrate how this can be applied to the simple case of the elastic pseudoscalar meson form factor, and then discuss how this can be extended to

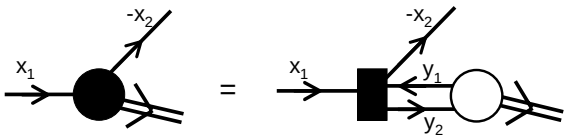

Figure 5.5: Non-wave function vertex (black circle) linked to an ordinary light–front wave function.

hadronic decays. The current for the elastic scattering of a pseudoscalar meson is

$$\langle (P-q)|J^{\mu}|P\rangle = iN_c \int \frac{d^4k}{(2\pi)^4} \frac{Tr[\gamma^5(\not{k}+m_1)\gamma^{\mu}(\not{k}-\not{q}+m_1)\gamma^5(\not{k}-\not{P}+m_2)]H_iH_f}{[k^2-m_1^2+i\epsilon][(k-q)^2-m_1^2+i\epsilon][(k-P)^2-m_1^2+i\epsilon]} \tag{5.53}$$

where H_i and H_f are the initial and final state meson–quark vertex functions. The form factor is related to the amplitude by $\langle J^{\mu}\rangle = (2P-q)^{\mu}F(Q^2)$. Writing this in light–front coordinates (k^+, k^-, and $\vec{k}_{\perp}$), choosing the $\mu = +$ component of the current, and integrating over the light–front energy, k^-, yields the following form factor,

$$\begin{aligned} F(Q^2) = \frac{N_c}{(2-\alpha)P^+}\Bigg\{ e_q\Bigg[\int_{\alpha}^{1} dx \int \frac{d^2\vec{k}_{\perp}}{16\pi^3} \frac{S_V^+}{x(1-x)x'} \Phi(x,\vec{k}_{\perp})\Phi\,'(x',\vec{k}'_{\perp}) \\ + \int_0^{\alpha} dx \int \frac{d^2\vec{k}_{\perp}}{16\pi^3} \left(\frac{S_{NV}^+}{x(1-x)x'}\right)\left(\frac{1}{D_{\gamma}}\right)\Phi(x,\vec{k}_{\perp})\left(\frac{4\pi^{3/2}G_{PS}}{\sqrt{2N_c}}\right)\Bigg] \\ + e_{\bar{q}}\Bigg[x \leftrightarrow (1-x), \vec{k}_{\perp} \leftrightarrow -\vec{k}_{\perp}, m_1 \leftrightarrow m_2\Bigg]\Bigg\}, \end{aligned} \tag{5.54}$$

where $x \equiv \frac{k^+}{P^+}$, $\alpha \equiv \frac{q^+}{P^+}$, $x' \equiv \frac{x-\alpha}{1-\alpha}$, and where

$$S_V^+ = \frac{4P^+}{1-x'}\Big[\vec{k}_{\perp}\cdot\vec{k}'_{\perp} + (xm_2+(1-x)m_1)(x'm_2+(1-x')m_1)\Big]$$

$$S_{NV}^+ = S_V^+ + \left(\frac{4P^+}{1-x'}\right)xx'(1-x)(M_{PS}^2 - M_0^2)$$

$$D_{\gamma} = \frac{1-\alpha}{\alpha}(q^2 - M_0''^2)$$

$$\Phi(x,\vec{k}_\perp) = \frac{H_i}{M_{PS}^2 - M_0^2} = \frac{1}{\sqrt{2N_c}} \frac{\sqrt{x(1-x)}}{[M_0^2 - (m_1-m_2)^2]^{\frac{1}{2}}} \left(\frac{\partial k_z}{\partial x}\right)^{\frac{1}{2}} \phi(|\vec{k}|,\beta)$$

$$\Phi'(x',\vec{k}'_\perp) = \frac{H_f}{M_{PS}^2 - M_0'^2} = \frac{1}{\sqrt{2N_c}} \frac{\sqrt{x'(1-x')}}{[M_0'^2 - (m_1-m_2)^2]^{\frac{1}{2}}} \left(\frac{\partial k'_z}{\partial x'}\right)^{\frac{1}{2}} \phi(|\vec{k}'|,\beta)$$

$$\left(\frac{\partial k_z}{\partial x}\right)^{\frac{1}{2}} = \sqrt{\frac{M_0}{4x(1-x)}} \left[1 - \frac{(m_2^2-m_1^2)^2}{M_0^4}\right]^{\frac{1}{2}}$$

$$\left(\frac{\partial k'_z}{\partial x'}\right)^{\frac{1}{2}} = \sqrt{\frac{M'_0}{4x'(1-x')}} \left[1 - \frac{(m_2^2-m_1^2)^2}{M_0'^4}\right]^{\frac{1}{2}}$$

$$|\vec{k}| = \frac{1}{2}\left[M_0^2 - 2(m_1^2+m_2^2) + \frac{(m_2^2-m_1^2)^2}{M_0^2}\right]^{\frac{1}{2}}$$

$$|\vec{k}'| = \frac{1}{2}\left[M_0'^2 - 2(m_1^2+m_2^2) + \frac{(m_2^2-m_1^2)^2}{M_0'^2}\right]^{\frac{1}{2}}$$

$$M_0^2 = \frac{\vec{k}_\perp^2 + m_1^2}{x} + \frac{\vec{k}_\perp^2 + m_2^2}{1-x}$$

$$M_0'^2 = \frac{\vec{k}_\perp'^2 + m_1^2}{x'} + \frac{\vec{k}_\perp'^2 + m_2^2}{1-x'}$$

$$M_0''^2 = \frac{\vec{k}_\perp''^2 + m_1^2}{x''} + \frac{\vec{k}_\perp''^2 + m_1^2}{1-x''}$$

$$\vec{k}'_\perp = \vec{k}_\perp - (1-x')\vec{q}_\perp$$

$$\vec{k}''_\perp = \vec{k}_\perp - x''\vec{q}_\perp$$

$$q^2 = -Q^2 = -\frac{\vec{q}_\perp^2 + \alpha^2 M_{PS}^2}{1-\alpha}\ ,$$

and where $\phi(|\vec{k}|,\beta)$ is the momentum space representation of the meson's radial wave function. In the form factor, the first term in square brackets multiplied by e_q corresponds to the contribution where the photon interacts with the quark. The second term in square brackets multiplied by $e_{\bar{q}}$ corresponds to the contribution where the photon interacts with the antiquark. This has the same form as the first term in square brackets, but with the quark and antiquark masses and momenta switched. The first term in square brackets contains two separate integrals. The first integral corresponds to the light–front time–ordered diagram in which the final–state meson is produced at a wave function vertex. The second integral corresponds to the diagram in which the final–state meson is produced at a non–wave function vertex, and therefore contains the constant G_{PS} which is the pseudoscalar (PS) non-wave function vertex.

To determine the value of G_{PS} for a particular meson,weemploy the following method. First, note that in the frame where the longitudinal momentum of the photon (q^+) vanishes, $\alpha = 0$ and the non-wave function integral vanishes. In this case, the wave function integral completely describes the form factor. In a different frame where q^+, and hence α, are non–zero the form factor receives contributions from both the wave function and non–wave function integrals. However, the form factor should not depend on our choice of frame. Therefore, the sum of the contributions of the wave function and non–wave function integrals in a frame where $q^+ \neq 0$ will be equal to the form factor calculated in the $q^+ = 0$ frame. For a number of frames in the range $0 < \alpha < 1$, the value of G_{PS} is chosen such that the average root mean square error for the form factor in all of the frames is a minimum.

For the pion, this yields a value of $G_\pi = 0.31$. For the kaon, we treat the case where the quark coming into the non-wave function vertex is non-strange (n), separately form the case where the incoming quark is strange(s). This gives $G_K^n = 0.26$ and $G_K^s = 0.29$. The resulting pion and kaon form factors are plotted for a number of different frames and are shown in Figures 5.6 and 5.7. Note that taking the non–wave function vertex to be a constant has very little effect. Since the η meson is neutral, it would not have an elastic form factor. In this case, the radiative decay process $\phi \to \eta\gamma$ is used to compute the non–wave function vertex, G_η. When the incoming quark is non–strange, $G_\eta^n = 0.29$ and when the incoming quark is strange, $G_\eta^s = 0.39$.

Once the values of the non–wave function vertices have been determined,we take them to be universal and apply them to the hadronic decay processes involving the heavy isoscalar mesons. The decay width for a scalar meson decaying to a pair of pseudoscalar mesons is given by

$$\Gamma(S \to PS + PS) = \frac{\mathcal{S}}{16\pi} \frac{\sqrt{M_S^2 - 4M_{PS}^2}}{M_S^2} |\mathcal{M}|^2 \ , \tag{5.55}$$

where M_S is the mass of the scalar meson, M_{PS} is the mass of the pseudoscalar meson, $\mathcal{M}$ is the Feynman amplitude for the decay process, and $\mathcal{S}$ is a statistical factor which is equal to 1 for distinguishable pseudoscalar mesons (*i.e.* $\pi^+\pi^-$) and 1/2 for indistinguishable pseudoscalar mesons (*i.e.* $\pi^0\pi^0$). Since the scalar mesons are mixtures of $n\bar{n}$, $s\bar{s}$, and gg as shown in Eq. 5.45, the Feynman amplitude for the decay of a scalar meson (S) to a pair of

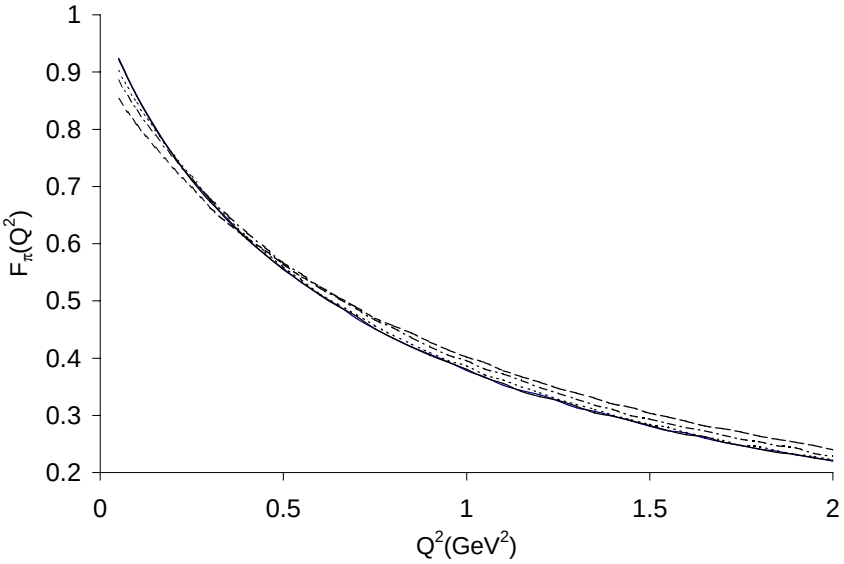

Figure 5.6: Pion form factor computed in the following frames: $\alpha = 0$ (solid line), $\alpha = 0.2$ (short dashed line), $\alpha = 0.4$ (dash–dot line), $\alpha = 0.6$ (long dashed line).

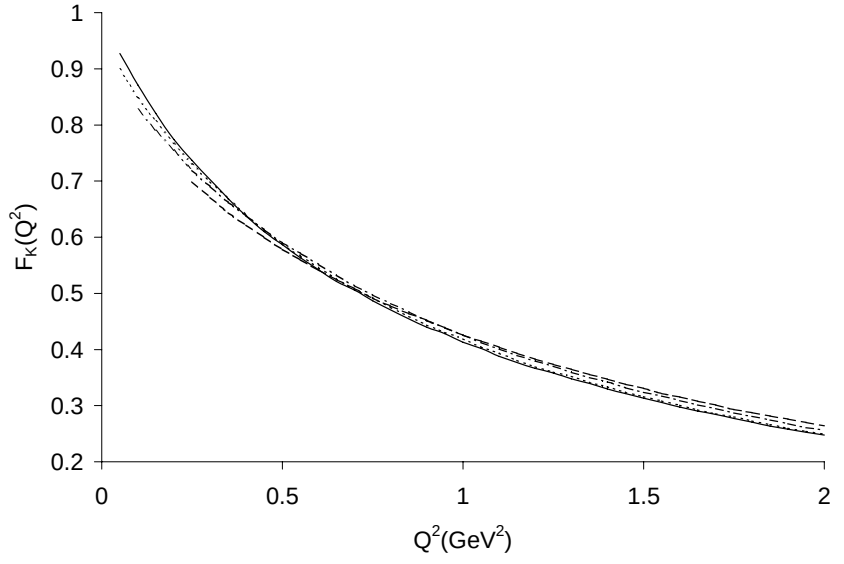

Figure 5.7: Kaon form factor computed in the following frames: $\alpha = 0$ (solid line), $\alpha = 0.2$ (short dashed line), $\alpha = 0.4$ (dash–dot line), $\alpha = 0.6$ (long dashed line).

pseudoscalar mesons (PS)is

$$\mathcal{M}(S_i \to PS + PS) = a_i \mathcal{C}_{n\bar{n}} I^{n\bar{n}}_{i\to PS} + b_i \mathcal{C}_{s\bar{s}} I^{s\bar{s}}_{i\to PS} \tag{5.56}$$

where the $\mathcal{C}^s$ are flavor factors, the index $i = 1, 2, 3$ denotes each of the three scalar mesons considered, $I^{q\bar{q}}_{i\to PS} = I^{q\bar{q}\,\prime}_{i\to PS} + I^{q\bar{q}\,\prime\prime}_{i\to PS}$,

$$I^{q\bar{q}\,\prime}_{i\to PS} = iN_c \int \frac{d^4k}{(2\pi)^4} \frac{Tr\left[\gamma^5(\not{k}+m_1)(\not{k}-\not{P}+m_1)\gamma^5(\not{k}-\not{q}+m_2)\right] H_S H'_{PS} H''_{PS}}{\left[k^2-m_1^2+i\epsilon\right]\left[(k-P)^2-m_1^2+i\epsilon\right]\left[(k-q)^2-m_2^2+i\epsilon\right]} \ , \tag{5.57}$$

and where $I^{q\bar{q}\,\prime\prime}_{i\to PS} = I^{q\bar{q}\,\prime}_{i\to PS}(q \leftrightarrow \bar{q})$. The H^s above denote the meson–quark vertex functions. Writing $I^{q\bar{q}\,\prime}_{i\to PS}$ in light–front coordinates (k^+, k^-, and $\vec{k}_\perp$) and integrating over the light–front energy, k^-, yields

$$\begin{aligned} I^{q\bar{q}\,\prime}_{i\to PS} &= 2\pi^{3/2}\sqrt{2N_c}\int \frac{d^2k_\perp}{16\pi^3}\Bigg[\int_0^\alpha dx \frac{S_1}{x(1-x)(1-x'')} G'_{PS}\Phi_S\Phi''_{PS} \\ &\qquad + \int_\alpha^1 dx \frac{S_2}{x(1-x)x'} G''_{PS}\Phi_S\Phi'_{PS}\Bigg] \end{aligned} \tag{5.58}$$

where

$$\begin{aligned} S_1 &= 4\Bigg\{m_1(1-x'')M_0''^2 - m_1^3 + m_1x''M_{PS}^2 - \frac{1}{2}m_2M_{S_i}^2 \\ &\qquad + (m_1-m_2)\left[\frac{1}{2}(1-x)M_0^2 + m_1m_2\right]\Bigg\} \\ S_2 &= 4\Bigg\{m_1x'M_0'^2 - m_1m_2^2 + m_1x''M_{PS}^2 - \left[(1-\alpha)x'm_1 + \frac{1}{2}m_2\right]M_{S_i}^2 \\ &\qquad + \frac{x'm_1\vec{q}_\perp^{\,2}}{\alpha} + (m_1-m_2)\left[\frac{1}{2}x(M_0^2-M_{S_i}^2) - m_1^2\right]\Bigg\} \\ \Phi_S &= \frac{H_S}{M_{S_i}^2 - M_0^2} = \frac{\sqrt{M_0}}{4\sqrt{2N_c}}\phi_S\left(|\vec{k}|,\beta_S\right) \\ \Phi'_{PS} &= \frac{H'_{PS}}{M_{PS}^2 - M_0'^2} = \frac{1}{\sqrt{2N_c}}\frac{\sqrt{x'(1-x')}}{[M_0'^2-(m_1-m_2)^2]^{\frac{1}{2}}}\left(\frac{\partial k'_z}{\partial x'}\right)^{\frac{1}{2}}\phi_{PS}\left(|\vec{k}'|,\beta_{PS}\right) \\ \Phi''_{PS} &= \frac{H''_{PS}}{M_{PS}^2 - M_0''^2} = \frac{1}{\sqrt{2N_c}}\frac{\sqrt{x''(1-x'')}}{[M_0''^2-(m_1-m_2)^2]^{\frac{1}{2}}}\left(\frac{\partial k''_z}{\partial x''}\right)^{\frac{1}{2}}\phi_{PS}\left(|\vec{k}''|,\beta_{PS}\right) \end{aligned} \tag{5.59}$$

and where the G^s are the non–wave function vertices determined above. Therefore, $I^{q\bar{q}\,\prime\prime}_{i\to PS} = I^{q\bar{q}\,\prime}_{i\to PS}\left(x \leftrightarrow (1-x), \vec{k}_\perp \leftrightarrow -\vec{k}_\perp\right)$.

Using Eqs. 5.55–5.59, the hadronic decay widths for the scalar mesons to pairs of pseudoscalar mesons can be evaluated once the mixing amplitudes, a_i and b_i, are determined. Recall that the final constraint which will give the solution for the mixing angles is the decay width $\Gamma(f_0(1500) \to \pi\pi) = 38$ MeV. Only the $n\bar{n}$ component of the mesons contributes to the $\pi\pi$ decay. Evaluating $I^{n\bar{n}}_{f_0(1500)\to\pi}$ and writing the decay width in terms of the mixing amplitude, a_2, yields

$$\begin{aligned}\Gamma(f_0(1500) \to \pi\pi) &= \Gamma(f_0(1500) \to \pi^+\pi^-) + \Gamma(f_0(1500) \to \pi^0\pi^0) \\ &= \left(55a_2^2 + 27a_2^2\right)\ \text{MeV} \\ &= 82a_2^2\ \text{MeV}. \end{aligned} \tag{5.60}$$

Given the experimental value of 38 MeV, solving for the mixing amplitude gives $|a_2| = 0.68$.

5.4.4 Mixing Amplitude Solutions

Having the magnitude of a_2 fixed at 0.68 provides the final constraint needed to solve for the mixing parameters,X,β,γ,δ, and therefore the other mixing amplitudes. It turns out that the solution is highly sensitive to the values of the meson masses. For the analysis, the masses of the $f_0(1500)$ and $f_0(1710)$ are fixed at $M_2 = 1507$ MeV and $M_3 = 1724$ MeV respectively. Because of it's large uncertainty (1200–1500 MeV), the mass of the $f_0(1370)$ is varied and separate solutions are found for each value of the mass, M_1. Real solutions for the mixing amplitudes are found only when the mass of the $f_0(1370)$ is in the range from about 1275 MeV to about 1350 MeV. Some sample solutions are shown below and their predictions for the decay widths are shown in Table 5.2. For each of the solutions, $M_2 = 1.507\ GeV$, $M_3 = 1.724\ GeV$, $m_{n\bar{n}} = 1.09\ GeV$, $m_{s\bar{s}} = 1.354\ GeV$, and $m_{gg} = 1.611\ GeV$.

SOLUTION #1

$$M_1 = 1.275\ GeV \quad X = 0.620 \quad \beta = 0.393\ GeV^2 \quad \gamma = 0.125\ GeV^2 \quad \delta = 0.324\ GeV^2$$

$$U = \begin{pmatrix} 0.715 & -0.697 & -0.056 \\ 0.662 & 0.701 & -0.264 \\ 0.223 & 0.152 & 0.963 \end{pmatrix}$$

SOLUTION #2

$$M_1 = 1.295\ GeV \quad X = 0.515 \quad \beta = 0.471\ GeV^2 \quad \gamma = 0.186\ GeV^2 \quad \delta = 0.248\ GeV^2$$

$$U = \begin{pmatrix} 0.630 & -0.772 & -0.079 \\ 0.683 & 0.601 & -0.415 \\ 0.368 & 0.208 & 0.906 \end{pmatrix}$$

SOLUTION #3

$$M_1 = 1.315\ GeV \quad X = 0.403 \quad \beta = 0.563\ GeV^2 \quad \gamma = 0.228\ GeV^2 \quad \delta = 0.149\ GeV^2$$

$$U = \begin{pmatrix} 0.515 & -0.852 & -0.087 \\ 0.683 & 0.470 & -0.558 \\ 0.516 & 0.228 & 0.825 \end{pmatrix}$$

SOLUTION #4

$$M_1 = 1.335\ GeV \quad X = 0.259 \quad \beta = 0.657\ GeV^2 \quad \gamma = 0.245\ GeV^2 \quad \delta = 0.062\ GeV^2$$

$$U = \begin{pmatrix} 0.346 & -0.935 & -0.069 \\ 0.683 & 0.302 & -0.664 \\ 0.642 & 0.182 & 0.744 \end{pmatrix}$$

In previous analyses [42, 56, 57, 58, 59, 60] the mixing amplitudes (given by U) displayed a fairly consistent phase structure. The $n\bar{n}$ and $s\bar{s}$ components of the $f_0(1370)$ were in phase,

Table 5.2: Calculated decay rates in MeV compared with experimental values.

Process	**Soln.#1**	**Soln#2**	**Soln#3**	**Soln#4**	**Expt.**
$\Gamma(f_0(1500) \to \pi\pi)$	36.0	38.3	38.3	38.3	38.0 ± 3.5
$\Gamma(f_0(1500) \to K\bar{K})$	16.7	20.5	23.7	28.1	9.4 ± 1.2
$\Gamma(f_0(1500) \to \eta\eta)$	18.2	15.1	11.2	7.0	5.6 ± 1.0
$\Gamma(f_0(1710) \to \pi\pi)$	6.7	18.3	36.0	55.7	21^{+10}_{-13}
$\Gamma(f_0(1710) \to K\bar{K})$	7.3	19.4	37.5	56.6	52^{+12}_{-26}
$\Gamma(f_0(1710) \to \eta\eta)$	2.7	5.9	8.9	9.2	25^{+10}_{-15}
$\Gamma(f_0(1370) \to \pi\pi)$	15.2	11.8	7.9	3.6	< 175
$\Gamma(f_0(1370) \to K\bar{K})$	75.8	74.6	70.7	61.9	$44 \to 240$
$\Gamma(f_0(1370) \to \eta\eta)$	0.02	0.08	0.2	0.4	> 0.54

while those of the $f_0(1500)$ were out of phase. The solutions here show the exact opposite behavior. This is due to the fact that the unmixed masses given by the quark model are significantly lower than those used in these previous analyses. In many of these previous analyses, the $n\bar{n}$, $s\bar{s}$, and gg masses were left as parameters that were fit to decay data. The fits returned masses that were a few of hundred MeV higher than the quark model values. This was mainly due to the assumption of glueball dominance which motivated the authors to set the $q\bar{q}$-$q\bar{q}$ mixing to zero. In our analysis, the smaller values of the input masses result in a different phase structure for the mixing amplitudes. Additionally, a consistent feature of our solutions is that the $q\bar{q}$-$q\bar{q}$ mixing is stronger than the $q\bar{q}$-gg($i.e. \beta > \gamma$). Clearly, the reason for this is that, again, the $n\bar{n}$ and $s\bar{s}$ masses are so low compared to the masses of the $f_0(1370)$ and $f_0(1500)$. Even with glueball mixing present, strong quarkonia mixing is needed to raise these values to the physical meson masses.

Looking at the predictions of the solutions in Table 5.2, it is obvious that none of the solutions matches the data very well. Overall, Solution #3 probably comes the closest, but it is still not satisfactory. It is somewhat reassuring that the calculations are of the correct order of magnitude. But, it would be nice to see closer agreement. There could be a couple of reasons for the discrepancies. First, it could be that the parameterization of the mixing in Eq. 5.47 is incorrect. While a similar analysis was successful in the case of the pseudoscalar mesons, one could argue that the scheme is too restrictive in that it makes too many assumptions about the relationships between the various annihilation diagrams. Second, it is possible that the assumption that the glueball does not contribute significantly to the hadronic decays is bad. We explore these possibilities in the next two sub–sections.

5.4.5 Alternative Mixing Amplitude Solutions

It is possible to relax the assumptions that go into the mixing scheme of Eq. 5.47 and calculate the mixing amplitudes directly in terms of mixing angles. The mixing angles can be obtained from the SU(2) group transformation $U(\theta, \phi, \psi)$ as follows

$$\vec{\sigma} \cdot \vec{R}' = U^{\dagger} \left(\vec{\sigma} \cdot \vec{R} \right) U \tag{5.61}$$

where $\vec{R}' = (|f_0(1370)\rangle, |f_0(1500)\rangle, |f_0(1710)\rangle)$ and $\vec{R} = (|n\bar{n}\rangle, |s\bar{s}\rangle, |gg\rangle)$. The SU(2) transformation U is defined by

$$\begin{aligned} U(\theta,\phi,\psi) &= \exp\left(-\frac{i\vec{\sigma}\cdot\hat{n}\theta}{2}\right) \\ &= \mathbf{1}\cos\left(\frac{\theta}{2}\right) - i\vec{\sigma}\cdot\hat{n}\sin\left(\frac{\theta}{2}\right) \end{aligned} \tag{5.62}$$

where the unit vector $\hat{n} = (\sin\psi\cos\phi, \sin(\psi)\sin(\phi), \cos(\psi))$. Using this notation, the mixing amplitudes from Eq. 5.45 are given by

$$\begin{aligned} a_1 &= \cos^2(\theta/2) - \cos^2(\psi)\sin^2(\theta/2) + \sin^2(\psi)\sin^2(\theta/2)\cos(2\phi) \\ b_1 &= \sin^2(\psi)\sin^2(\theta/2)\sin(2\phi) + \cos(\psi)\sin(\theta) \\ c_1 &= \cos(\phi)\sin(2\psi)\sin^2(\theta/2) - \sin(\theta)\sin(\phi)\sin(\psi) \end{aligned}$$

$$\begin{aligned} a_2 &= \sin^2(\psi)\sin^2(\theta/2)\sin(2\phi) - \sin(\theta)\cos(\psi) \\ b_2 &= \cos^2(\theta/2) - \cos^2(\psi)\sin^2(\theta/2) - \sin^2(\psi)\sin^2(\theta/2)\cos(2\phi) \\ c_2 &= \sin(\theta)\cos(\phi)\sin(\psi) + \sin^2(\theta/2)\sin(\phi)\sin(2\psi) \\ a_3 &= \sin^2(\theta/2)\cos(\phi)\sin(2\psi) + \sin(\theta)\sin(\phi)\sin(\psi) \\ b_3 &= \sin^2(\theta/2)\sin(\phi)\sin(2\psi) - \sin(\theta)\cos(\phi)\sin(\psi) \\ c_3 &= \cos^2(\theta/2) + \sin^2(\theta/2)\cos(2\psi)\ . \end{aligned} \tag{5.63}$$

Using the above expresions, the decay widths for the $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ to $\pi\pi$, $K\bar{K}$, and $\eta\eta$ can be written in terms of the mixing amplitudes, a_i and b_i. The expressions in Eq. 5.63 can then be substituted, so that the decay widths are given in terms of the three mixing angles (θ, ϕ, ψ). Therefore, any three of these expressions for the decay widths can be set equal to the experimental values, giving three independent equations which can be solved for the three mixing angles. In this way, the mixing amplitudes can be determined without reference to, and therefore without any additional assumptions about, the form of the mass–squared matrix. In this case, the only thing that constrains the form of the mass–squared matrix is the fact that the basis transformation between the unmixed and mixed states is unitary. This does force certain relationships among the off–diagonal elements of the mass–squared matrix, but it is not as restrictive as the assumptions that go

into Eq. 5.47. The data for the decays of the $f_0(1500)$ and $f_0(1710)$ were used to find the mixing angle solutions. For these mesons, there is a total of six decay widths. All possible groups of three were chosen from these to fit the three mixing angles. Four of the solutions are given below. The predictions of the decay rates are given in Table 5.3.

The results shown below are representative of all the solutions. As can be seen, the decay width predictions do not match the data very well. Even though the mixing angles in this case are fit to more data points than were the mixing parameters (X,β,γ,δ) in the first method, the results are no better. Clearly this points to the fact that not all of the necessary components have been included. In the next sub–section, we do a simple preliminary analysis to determine the effect of including the glueball contributions in the decay calculations.

SOLUTION #1

$$\theta = -85^\circ \quad \phi = 165^\circ \quad \psi = -39^\circ$$

$$U = \begin{pmatrix} 0.433 & -0.860 & 0.267 \\ 0.681 & 0.118 & -0.723 \\ 0.590 & 0.495 & 0.637 \end{pmatrix}$$

SOLUTION #2

$$\theta = -70^\circ \quad \phi = 172^\circ \quad \psi = -57^\circ$$

$$U = \begin{pmatrix} 0.800 & -0.569 & 0.189 \\ 0.444 & 0.349 & -0.825 \\ 0.403 & 0.744 & 0.532 \end{pmatrix}$$

SOLUTION #3

$$\theta = -68^\circ \quad \phi = 154^\circ \quad \psi = -49^\circ$$

$$U = \begin{pmatrix} 0.657 & -0.753 & -0.023 \\ 0.468 & 0.432 & -0.770 \\ 0.590 & 0.495 & 0.637 \end{pmatrix}$$

SOLUTION #4

$$\theta = -76^\circ \quad \phi = 178^\circ \quad \psi = -45^\circ$$

$$U = \begin{pmatrix} 0.611 & -0.705 & 0.359 \\ 0.680 & 0.236 & -0.693 \\ 0.403 & 0.668 & 0.624 \end{pmatrix}$$

Table 5.3: Decay rates in MeV calculated using the mixing angle method. For each solution, the underlined values are the ones used to fit the mixing angles. The other decay widths are predictions using the mixing angles obtained from the fit.

Process	**Soln.#1**	**Soln#2**	**Soln#3**	**Soln#4**	**Expt.**
$\Gamma(f_0(1500) \to \pi\pi)$	$\underline{38.0}$	16.2	18.0	$\underline{38.0}$	38.0±3.5
$\Gamma(f_0(1500) \to K\bar{K})$	33.4	$\underline{9.4}$	$\underline{9.4}$	30.0	9.4±1.2
$\Gamma(f_0(1500) \to \eta\eta)$	3.4	$\underline{5.5}$	7.5	$\underline{5.5}$	5.6±1.0
$\Gamma(f_0(1710) \to \pi\pi)$	47.0	$\underline{21.0}$	47.0	$\underline{21.0}$	21^{+10}_{-13}
$\Gamma(f_0(1710) \to K\bar{K})$	$\underline{52.0}$	28.2	$\underline{52.0}$	27.4	52^{+12}_{-26}
$\Gamma(f_0(1710) \to \eta\eta)$	$\underline{25.0}$	37.0	$\underline{25.0}$	31.2	25^{+10}_{-15}
$\Gamma(f_0(1370) \to \pi\pi)$	5.6	19.0	12.8	11.1	< 175
$\Gamma(f_0(1370) \to K\bar{K})$	63.1	71.4	75.4	65.6	$44 \to 240$
$\Gamma(f_0(1370) \to \eta\eta)$	0.3	0.002	0.06	0.05	> 0.54

5.4.6 Mixing with Glueball Contributions

It is conceivable that the failure of the above methods in predicting the hadronic decay widths of the scalar mesons, is due to the assumption that the glueball contribution is negligible in these processes. This assumption was motivated by large N_c arguments, but it is possible that it may be incorrect. Even though the glueball contribution to these decays has not been directly calculated, it is possible to determine whether the inclusion of these contributions would improve or worsen the predictions of the decay rates. To accomplish this,wesimply parameterize the glueball contribution in the following way. Including the glueball component, a given decay amplitude will have the form

$$\begin{aligned} \mathcal{M} &= a_i \mathcal{C}_{n\bar{n}} I^{n\bar{n}}_{i\to PS} + b_i \mathcal{C}_{s\bar{s}} I^{s\bar{s}}_{i\to PS} + c_i \mathcal{C}_{gg} I^{gg}_{i\to PS} \\ &= \left(a_i \mathcal{C}_{n\bar{n}} I^{n\bar{n}}_{i\to PS} + b_i \mathcal{C}_{s\bar{s}} I^{s\bar{s}}_{i\to PS}\right) \left[1 + c_i \frac{\mathcal{C}_{gg} I^{gg}_{i\to PS}}{a_i \mathcal{C}_{n\bar{n}} I^{n\bar{n}}_{i\to PS} + b_i \mathcal{C}_{s\bar{s}} I^{s\bar{s}}_{i\to PS}}\right] \\ &= \mathcal{M}_{q\bar{q}} \left[1 + c_i \kappa_{PS}\right] \ , \end{aligned} \tag{5.64}$$

Table 5.4: Hadronic decay widths including the glueball contributions. The $|c_i\kappa|$ are the glueball contributions to the amplitude as a fraction of the quarkonia contributions. The final column gives the the glueball contribution as a percentage of the total amplitude.

Decay Process	Calculated Width (MeV)	Expt. (MeV)	$\lvert c_i\kappa\rvert$	Percentage of Total Amplitude
$\Gamma(f_0(1500) \to \pi\pi)$	37.5	38.0 ± 3.5	0	0%
$\Gamma(f_0(1500) \to K\bar{K})$	10.1	9.4 ± 1.2	0.29	23%
$\Gamma(f_0(1500) \to \eta\eta)$	5.5	5.6 ± 1.0	0.38	28%
$\Gamma(f_0(1710) \to \pi\pi)$	22.0	21^{+10}_{-13}	0	0%
$\Gamma(f_0(1710) \to K\bar{K})$	57.8	52^{12}_{26}	0.58	37%
$\Gamma(f_0(1710) \to \eta\eta)$	21.2	25^{+10}_{-15}	0.75	43%
$\Gamma(f_0(1370) \to \pi\pi)$	11.3	< 175	0	0%
$\Gamma(f_0(1370) \to K\bar{K})$	83.0	$44 \to 240$	0.057	5%
$\Gamma(f_0(1370) \to \eta\eta)$	0.08	> 0.54	0.074	7%

where we have defined $\mathcal{M}_{q\bar{q}} \equiv (a_i\mathcal{C}_{n\bar{n}}I^{n\bar{n}}_{i\to PS} + b_i\mathcal{C}_{s\bar{s}}I^{s\bar{s}}_{i\to PS})$ as well as the parameter $\kappa_{PS} \equiv (\mathcal{C}_{gg}I^{gg}_{i\to PS})/(a_i\mathcal{C}_{n\bar{n}}I^{n\bar{n}}_{i\to PS} + b_i\mathcal{C}_{s\bar{s}}I^{s\bar{s}}_{i\to PS})$. The quantity $\mathcal{M}_{q\bar{q}}$ is the quark contribution to the decays which was found earlier. The product $c_i\kappa_{PS}$ is the ratio of the glueball contribution to the quark contribution. The parameter κ_{PS} will be different for each of the final states $\pi\pi(\kappa_\pi)$, $K\bar{K}(\kappa_K)$, and $\eta\eta(\kappa_\eta)$. However, these values are assumed to be the same for each of the initial scalar mesons. The mixing amplitudes, c_i, will scale the value of κ_{PS} so that the scalar mesons with a larger glueball content will receive larger corrections. The best fit to the data is,

BEST SOLUTION

$$M_1 = 1.297\ GeV \quad X = 0.50 \quad \beta = 0.489\ GeV^2 \quad \gamma = 0.193\ GeV^2$$

$$\delta = 0.208\ GeV^2 \qquad \kappa_\pi = 0 \qquad \kappa_K = 0.65 \qquad \kappa_\eta = 0.84$$

$$U = \begin{pmatrix} 0.617 & -0.782 & -0.088 \\ 0.676 & 0.581 & -0.453 \\ 0.403 & 0.224 & 0.887 \end{pmatrix}.$$

The resulting decay widths are shown in Table 5.4. The inclusion of the glueball contributions has improved the situation considerably. The contributions to the $K\bar{K}$ and $\eta\eta$ decays of the $f_0(1500)$ and $f_0(1710)$ are significant. Recall that $|c_i\kappa|$ is the glueball contribution as a fraction of the quarkonia contribution, so that the percentage of the total amplitude will

be given by $\frac{|c_i\kappa|}{1+|c_i\kappa|}$. These values are listed in Table 5.4 as well. The largest contribution is in the process $f_0(1710) \rightarrow \eta\eta$, where the glueball contribution is 43% of the total amplitude. One interesting feature of this solution is the order of the glueball contributions, $\kappa_\pi < \kappa_K < \kappa_\eta$. We point out that this finding is in agreement with the consequences of chiral suppression found by Chanowitz [1]. Recall that Chanowitz determined that the amplitude for the scalar glueball to decay to pairs of mesons is proportional to the quark mass. So, decays to $K\bar{K}$ or $\eta\eta$ would be enhanced over decays to $\pi\pi$. Chanowitz also pointed out that this suppression allows for a reinterpretation of the structure of the $f_0(1710)$. It has been generally accepted that the $f_0(1710)$ is a dominantly $s\bar{s}$ state because it strongly decays to $K\bar{K}$ but not to $\pi\pi$. With chiral suppression, one could just as easily explain this behavior by taking the $f_0(1710)$ to be the scalar glueball. This is consistent with our final mixing solution which assigns nearly 80% of the structure of the $f_0(1710)$ to the scalar glueball.

If this prediction is not borne out in the direct glueball decay calculation, it would mean that the $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ cannot be described as a mixtures of $n\bar{n}$, $s\bar{s}$, and gg. To fully describe these mesons, one would most likely need to include the four–quark states in the mixing. In the Conclusion that follows,we briefly outline how this could be accomplished in a future work if it in fact becomes necessary.

Chapter 6

Summary and Conclusion

In this work,we have attempted to explain the scalar meson decay data by assuming that the heavy isoscalar mesons $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$ are mixtures of $n\bar{n}$, $s\bar{s}$, and gg. While our previous work focused on radiative decays of these mesons for which there is no good data, this work focuses on hadronic decays. In particular,we focus on the decays $f_0 \to \pi\pi$, $f_0 \to K\bar{K}$, and $f_0 \to \eta\eta$. Here, the experimental situation is much cleaner, at least in the cases of the $f_0(1500)$ and $f_0(1710)$. The data for the $f_0(1370)$ is still somewhat ambiguous.

Computing the hadronic decay amplitudes presented a challenge because of the presence of non–wave function vertices in the diagrams. In Ref [61], it was shown that while the non-wave function vertex is in general a function of all the internal momenta, it could successfully be approximated as a constant. Using a similar idea,we have shown that when computing pion and kaon elastic form factors the assumption that this non-wave function vertex is a constant results in negligible errors. A similar result follows in the calculation of $\phi \to \eta\gamma$. Therefore, the constant non-wave function vertices determined from these calculations were used in the computation of hadronic decays of scalar mesons.

Once a method was established for computing the hadronic decay amplitudes, the mixing was analyzed. The mixing was handled using two different methods. In the first, a mass-squared matrix was written which involved parameterizing the contributions of the various annihilation diagrams that connect $q\bar{q}$ to $q'\bar{q}'$ or gg to $q\bar{q}$. This resulted in four mixing parameters. The basis transformation that connects the unmixed states with the physical states diagonalizes this mass-squared matrix and thus provides three constraints on the mixing parameters. Using the decay width $\Gamma(f_0(1500) \to \pi\pi)$ as a fourth constraint,

the four mixing parameters were calculated. These parameters determined the $n\bar{n}$, $s\bar{s}$, and gg content for the $f_0(1370)$, $f_0(1500)$, and $f_0(1710)$. This information was then used to predict the remaining hadronic decay widths.

The second method of computing the mixing was more general and did not assume any particular form for the mass-squared matrix. Instead, it involved writing the basis transformation between the unmixed and mixed states in terms of mixing angles. This gives the $n\bar{n}$, $s\bar{s}$, and gg content of the mesons in terms of three mixing angles. Using three decay widths, the three mixing angles were fitted and then used to predict the remaining decay widths. While this method was more general and involved fewer assumptions, it required three data points to fix the mixing parameters whereas the first method only required one.

Unfortunately, neither method was successful in predicting the remaining decay widths satisfactorily once the mixing parameters/angles had been fixed. This is likely due to the fact that the glueball contributions to the hadronic decay widths were neglected. To check whether the inclusion of the glueball would improve the predictions, we made a simple parameterization of these contributions. The best results seem to occur when the $gg \to \pi\pi$ contribution is essentially zero, while the $gg \to K\bar{K}$ is fairly large and the $gg \to \eta\eta$ contribution is even larger. The decay data is very well described when these glueball contributions are added. One might expect that the relative sizes of the contributions would be in the reverse order where the pion contribution is largest and the eta meson contribution is smallest due to phase space differences. However, it may be too difficult to see how the reverse might be true, especially in light of Chanowitz's chiral suppression [1].

To lowest order, the glueball decay to two pseudoscalar mesons can proceed through two basic diagrams. In the first diagram, the glueball produces a single quark-antiquark pair, $q_1\bar{q}_1$. Then, either the quark or antiquark can emit a gluon which subsequently produces another quark-antiquark pair, $q_2\bar{q}_2$. These then combine as $q_1\bar{q}_2$ and $q_2\bar{q}_1$ to form the final state mesons. In the second diagram, each gluon directly produces a quark-antiquark pair, $q_1\bar{q}_1$ and $q_2\bar{q}_2$. The quark from the first gluon would then combine with the antiquark from the second ($q_1\bar{q}_2$) to produce one of the final state mesons, while the quark from the second gluon and the antiquark from the first gluon $q_2\bar{q}_1$ would form the second final state meson. In either diagram, the quark and antiquark forming a final state meson would need to be collinear. This would be much less probable for the second diagram than for the first. We would therefore expect the first diagram to be the dominant one. However, according to

Chanowitz, it is exactly this diagram that would be chirally suppressed. By this reasoning, we would expect to see scalar glueball decays to $\pi\pi$ suppressed far greater than decays to $K\bar{K}$ or $\eta\eta$. It is also worth remembering that this would be in line with the lattice results of Sexton, Vaccarino, and Weingarten [19] who demonstrate that the coupling of a scalar glueball to two pseudoscalar mesons increases as the pseudoscalar meson mass increases.

Finally, we note that the mixing amplitudes of our final solution give that the $f_0(1710)$ is mostly glueball (nearly 80%) while the $f_0(1500)$ and $f_0(1370)$ are dominantly mixtures of $n\bar{n}$ and $s\bar{s}$. This is contrary to the popular opinion that the $f_0(1500)$ is the best scalar glueball candidate. However, the interpretation given by our final solution is contingent upon the results of the scalar glueball decay calculations. These must be performed in order to verify that decay to $\pi\pi$ is suppressed to a greater extent than are decays to $K\bar{K}$ or $\eta\eta$. If this is not the case, then our mixing solution will not match the data, and the possibility exists that one must include four–quark states in the mixing to arrive at a successful interpretation.

Future work will include directly computing the decay of the scalar glueball to pairs of pseudoscalar mesons. To this end,we have done a preliminary calculation of the $L = 0$ sector of a two constituent gluon glueball spectrum. The Hamiltonian is a simple non-relativistic one. It has the same form as the Hamiltonian for constituent quarks, except the $q\bar{q}$ color factor $f_C = \frac{4}{3}$ is replaced with the gg color factor $f_C = 3$. A single ground–state simple harmonic oscillator wave function was used as a trial wave function for the variational principle. The ground state glueball (0^{++}) was adjusted to be 1611 MeV, which was the lattice value used in this thesis. With this, the spin–spin splitting was adjusted to give the mass of the 2^{++} glueball as 2287 MeV. The wave function obtained from this analysis can be used to obtain a preliminary estimate of the glueball contribution to the hadronic decays. To improve upon this estimate a more complete glueball spectrum can be computed with the full relativized model Hamiltonian.

If this does not reconcile the discrepancy between the hadronic decay predictions and the data, then the final possibility would be to include four–quark states in the mixing. This could be accomplished by treating the four–quark states as bound states of diquarks. As a first step, the diquark spectrum would need to be analyzed. For this, the color factor in the Hamiltonian would simply have to be modified. The color factor for a qq or $\bar{q}\bar{q}$ is $\frac{2}{3}$. After computing the diquark spectrum, the diquarks themselves would then be combined to form $(qq)(\bar{q}\bar{q})$ bound states. To bind the color anti–triplet (qq) and color triplet $(\bar{q}\bar{q})$ into

a color singlet, the same Hamiltonian can be used, but with the color factor set back to $\frac{4}{3}$. The masses can then be used to extend the mixing calculation. It is thought that the scalar states below 1 GeV, the $f_0(600)$ and the $f_0(980)$, are part of a four-quark nonet. The mixing would be between the $f_0(600)$,$f_0(980)$,$f_0(1370)$,$f_0(1500)$,and $f_0(1710)$. The four–quark wave functions can then be used to compute the contributions to hadronic decays.

Bibliography

[1] M. Chanowitz. *Physics Review Letters*, 95:172001, 2005.

[2] K. Hagiwara *et al. Physical Review*, D66:010001, 2002.

[3] J. Weinstein and N. Isgur. *Physical Review Letters*, 48:659, 1982.

[4] J. Weinstein and N. Isgur. *Physical Review*, D41:2236, 1990.

[5] S. Godfrey and N. Isgur. *Physical Review*, D32:189, 1985.

[6] R. Kokoski and N. Isgur. *Physical Review*, D35:907, 1987.

[7] F. E. Barnes. *Physics Letters*, B165:434, 1985.

[8] C. Amsler and F. E. Close. *Physical Review*, D53:295, 1996.

[9] V. V. Anisovich and A. V. Sarantsev. *Physics Letters*, B382:429, 1996.

[10] R. L. Jaffe and K. Johnson. *Physics Letters*, B60:201, 1976.

[11] R. L. Jaffe. *Physical Review*, D15:267, 1977.

[12] R. L. Jaffe. *Physical Review*, D15:281, 1977.

[13] R. L. Jaffe. *Physical Review*, D17:1444, 1978.

[14] F.E. Close, N. Isgur, and S. Kumano. *Nuclear Physics*, B389:513, 1993.

[15] N. A. Törnqvist. *Z. Phys.*, C68:647, 1995.

[16] N. A. Törnqvist and Matts Roos. *Physics Review Letters*, 76:1575, 1996.

[17] G.Bali *et al. Physics Letters*, B309:378, 1993.

[18] D. Weingarten. *Nuclear Physics B (Proceedings Supplment)*, B34:29, 1994.

[19] J. Sexton, A. Vaccarino, and D. Weingarten. *Physics Review Letters*, 75:4563, 1995.

[20] F. E. Close and N. A. Törnqvist. *J. Phys.*, G28:R249–R267, 2002.

[21] C. Carlson and C.-R. Ji. *Physical Review*, D67:116002, 2003.

[22] H.-M. Choi and C.-R. Ji. *Physical Review*, D59:074015, 1999.

[23] C.-R. Ji and H.-M. Choi. *Heavy Ion Physics*, 4:369, 1996.

[24] C.-R. Ji and H.-M. Choi. *Few Body Systems Supplement*, 10:131, 1999.

[25] H.-M. Choi and C.-R. Ji. *Physical Review*, D59:034001, 1999.

[26] H.-M. Choi and C.-R. Ji. *Physics Letters*, B460:461, 1999.

[27] W. Jaus. *Physical Review*, D53:1349, 1996.

[28] Stephen Godfrey and Nathan Isgur. *Physical Review*, D32:189, 1985.

[29] P. A. M. Dirac. *Reviews of Modern Physics*, 21:392, 1949.

[30] S. J. Brodsky, H.-C. Pauli, and S. S. Pinsky. *Physics Report*, 301:299–486, 1998.

[31] D. Arndt. Light-cone quark model analysis of pseudoscalar and vector mesons for radially excited states. Master's thesis, North Carolina State University, 1999.

[32] G.P. Lepage and S.J. Brodsky. *Physical Review*, D22:2157, 1980.

[33] H. J. Melosh. *Physics Review*, D9:1095, 1974.

[34] P. L. Chung, F. Coester, and W. N. Polyzou. *Physics Letters*, B205:545, 1988.

[35] W. Jaus. *Physical Review*, D44:2851, 1991.

[36] W. Jaus. *Physical Review*, D41:3394, 1990.

[37] B. L. G. Bakker and C.-R. Ji. *Physical Review*, D62:074014, 2000.

[38] B. L. G. Bakker, H.-M. Choi, and C.-R. Ji. *Physical Review*, D63:074014, 2001.

[39] A. Misra and S. Warawdekar. *Physical Review*, D71:125011, 2005.

[40] H.-C. Pauli and S. J. Brodsky. *Physical Review*, D32:1993, 1985.

[41] M. Burkardt. *Physical Review*, D57:1136, 1998.

[42] W. Lee and D. Weingarten. *Physical Review*, D61:014015, 1999.

[43] F. E. Close and A. Kirk. *Eur. Phys. J.*, C21:531, 2001.

[44] C.-R. Ji, L. Chung P, and S. R. Cotanch. *Physical Review*, D45:4214, 1992.

[45] T. Das, V. S. Mathur, and S. Okubo. *Physics Review Letters*, 19:470, 1967.

[46] J. J. Sakura. *Physics Review Letters*, 19:803, 1967.

[47] R. J. Oakes and J. J. Sakura. *Physics Review Letters*, 19:1266, 1967.

[48] M. D. Scadron. *Physical Review*, D29:2076, 1984.

[49] H-.M. Choi and C.-R. Ji. *Nuclear Physics*, A618:291, 1997.

[50] D. E. Groom *et al. European Journal of Physics*, C15:1, 2000.

[51] F.E. Close, A. Donnachie, and Y. S. Kalashnikova. *Physical Review*, D67:074031, 2003.

[52] C. Amsler. *Reviews of Modern Physics*, 70:1293, 1998.

[53] A. Gokalp and O. Yilmaz. *Physics Letters*, B525:273, 2002.

[54] A. I. Titov, T.-S. H. Lee, H. Toki, and O. Streltsova. *Physical Review*, C60:035205, 1999.

[55] H.F. Jones and M. D. Scadron. *Nuclear Physics*, B155:409, 1979.

[56] C. Amsler and F. E. Close. *Physics Letters*, B353:385, 1995.

[57] F. E. Close, G. R. Farrar, and Z.-P. Li. *Physical Review*, D55:5749, 1997.

[58] D.-M. Li, H. Yu, and Q.-X. Shen. *Communications in Theoretical Physics*, 34:507, 2000.

[59] M. Genovese. *Physical Review*, D46:5204, 1992.

[60] F. E. Close and A. Kirk. *Physics Letters*, B483:345, 2000.

[61] C.-R. Ji and H.-M. Choi. *Physics Letters*, B513:330, 2001.

Appendices

Appendix 1

Spinor Structure for $V(S) \to S(V)\gamma^*$

In this appendix, we show the explict form of the trace given by Eq. (4.23). For the $V \to S\gamma^*$ transition, the following two trace calculations are necessary

$$\begin{aligned}
S^+_{VS1} &= \mathrm{Tr}[(\not{p}_{\bar{q}} - m)(\not{p}_2 + m)\gamma^+(\not{p}_1 + m)\ \not{\epsilon}] \\
&= 4m[p_1^+(\epsilon \cdot p_{\bar{q}} - \epsilon \cdot p_2) + p_2^+(\epsilon \cdot p_{\bar{q}} - \epsilon \cdot p_1) \\
&\quad + p_{\bar{q}}^+(\epsilon \cdot p_1 - \epsilon \cdot p_2) \\
&\quad + \epsilon^+(p_1 \cdot p_2 + p_2 \cdot p_{\bar{q}} - p_1 \cdot p_{\bar{q}} - m^2)], \\
S^+_{SV2} &= \mathrm{Tr}[(\not{p}_{\bar{q}} - m)(\not{p}_2 + m)\gamma^+(\not{p}_1 + m)] \\
&= 4[p_1^+(p_2 \cdot p_{\bar{q}} - m^2) + p_2^+(p_1 \cdot p_{\bar{q}} - m^2) \\
&\quad + p_{\bar{q}}^+(m^2 - p_1 \cdot p_2)],
\end{aligned} \tag{6.1}$$

to get

$$S^+_{V\to S} = \frac{-1}{4(1-x)M_0}\left[S^+_{VS1} - \frac{\epsilon \cdot (p_1 - p_{\bar{q}})}{M_0 + 2m} S^+_{VS2}\right], \tag{6.2}$$

where $\epsilon = \epsilon(P_1)$ and we used the transverse polarizations in the calculation of the form factor and decay width.

On the other hand, for the $S \to V\gamma^*$ transitions, we have

$$\begin{aligned}
S^+_{SV1} &= \mathrm{Tr}[(\not{p}_{\bar{q}} - m)\ \not{\epsilon}'(\not{p}_2 + m)\gamma^+(\not{p}_1 + m)] \\
&= 4m[p_1^+(\epsilon' \cdot p_{\bar{q}} - \epsilon' \cdot p_2) + p_2^+(\epsilon' \cdot p_{\bar{q}} - \epsilon' \cdot p_1) \\
&\quad -p_{\bar{q}}^+(\epsilon' \cdot p_1 - \epsilon' \cdot p_2) \\
&\quad +\epsilon'^+(p_1 \cdot p_2 + p_1 \cdot p_{\bar{q}} - p_2 \cdot p_{\bar{q}} - m^2)], \\
S^+_{SV2} &= \mathrm{Tr}[(\not{p}_{\bar{q}} - m)(\not{p}_2 + m)\gamma^+(\not{p}_1 + m)] \\
&= 4[p_1^+(p_2 \cdot p_{\bar{q}} - m^2) + p_2^+(p_1 \cdot p_{\bar{q}} - m^2) \\
&\quad +p_{\bar{q}}^+(m^2 - p_1 \cdot p_2)],
\end{aligned} \tag{6.3}$$

to get

$$S^+_{S\to V} = \frac{-1}{4(1-x)M_0'}\left[S^+_{SV1} - \frac{\epsilon' \cdot (p_2 - p_{\bar{q}})}{M_0' + 2m} S^+_{SV2}\right], \tag{6.4}$$

where $\epsilon' = \epsilon'(P_2)$ and again we used the transverse polarizations in the calculation of the form factor and decay width.

Appendix 2

Code for Momentum-Dependent Matrix Elements

```
      program vf
      include 'mpif.h'
      double precision dummy
      integer l1,l2,j1,j2
      real*16 laguerre,PI,beta,m1,m2,m1sq,m2sq,esov,esos,et,ec,
     &alpha(3),tau(3),sigma0,sigma12,s,ss,bb,cc
      integer MAX_ROWS, MAX_COLS, rows, cols, max_tag
      parameter (MAX_ROWS=100,MAX_COLS=100,rows=25,cols=40)
      double precision ff1(MAX_ROWS,MAX_COLS),ff2(MAX_ROWS,MAX_COLS),
     &ff3(MAX_ROWS,MAX_COLS),ff4(MAX_ROWS,MAX_COLS),
     &ff5(MAX_ROWS,MAX_COLS),ff7(MAX_ROWS,MAX_COLS),
     &ff8(MAX_ROWS,MAX_COLS),ff9(MAX_ROWS,MAX_COLS),
     &ff10(MAX_ROWS,MAX_COLS)
      DOUBLE PRECISION c(cols)
      integer myid, master, numprocs, ierr, status(MPI_STATUS_SIZE)
      integer i, j, numsent, sender
      integer anstype, tagnum, frmt
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /fnums/esov,et,esos,ec
c*************************************************************
c Parameters for the integration routine
      external F1,F2,F3,F4,F5,F7,F8,F9,F10
      real*16 a,b
      a = 0.0q0
```

```
      b = 30.0q0
c*****************************************************************
c Additional stuff
      PI=3.14159265358979323846264338328q0
      beta=0.75q0
      l1=0
      l2=0
      m1=0.22q0
      m2=0.22q0
      m1sq=m1**2.q0
      m2sq=m2**2.q0
      esov=-0.035q0
      esos=0.055q0
      et=0.025q0
      ec=-0.168q0
c*****************************************************************
      call MPI_INIT( ierr )
      call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
      call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
      master = 0
      max_tag = 9*rows
c*****************************************************************
      if ( myid .eq. master ) then
      print*,'Start!'
c    master initializes and then dispatches
c    initialize aa and bb  (arbitrary)
      do 30 i = 1,rows
         do 31 j = 1,cols
            ff1(i,j) = 1.d0
            ff2(i,j) = 1.d0
    ff3(i,j) = 1.d0
    ff4(i,j) = 1.d0
    ff5(i,j) = 1.d0
            ff7(i,j) = 1.d0
    ff8(i,j) = 1.d0
    ff9(i,j) = 1.d0
    ff10(i,j)= 1.d0
31       continue
30    continue
      numsent = 0
c     send b to each slave process
c     send a row to each slave process; tag with tagnum number
      do 40 i = 1,min(numprocs-1,max_tag)
        call MPI_SEND(dummy,1, MPI_DOUBLE_PRECISION, i,
```

```
    &                 i, MPI_COMM_WORLD, ierr)
       numsent = numsent+1
40   continue
     do 70 i = 1,max_tag
       call MPI_RECV(c, cols , MPI_DOUBLE_PRECISION,
    &                 MPI_ANY_SOURCE, MPI_ANY_TAG,
    &                 MPI_COMM_WORLD, status, ierr)
       sender     = status(MPI_SOURCE)
       anstype    = status(MPI_TAG)       ! anstype is tag value
   if (anstype.le.rows) then
          do 55 k=1,cols
     ff1(anstype,k) = c(k)
55        continue
       elseif (anstype.le.rows+rows) then
          do 56 k=1,cols
             ff2(anstype-rows,k) = c(k)
56        continue
       elseif (anstype.le.rows+2*rows) then
   do 57 k=1,cols
             ff3(anstype-2*rows,k) = c(k)
57        continue
       elseif (anstype.le.rows+3*rows) then
   do 58 k=1,cols
             ff4(anstype-3*rows,k) = c(k)
58        continue
       elseif (anstype.le.rows+4*rows) then
   do 59 k=1,cols
             ff5(anstype-4*rows,k) = c(k)
59        continue
       elseif (anstype.le.rows+5*rows) then
   do 60 k=1,cols
             ff7(anstype-5*rows,k) = c(k)
60        continue
       elseif (anstype.le.rows+6*rows) then
   do 61 k=1,cols
             ff8(anstype-6*rows,k) = c(k)
61        continue
       elseif (anstype.le.rows+7*rows) then
   do 62 k=1,cols
             ff9(anstype-7*rows,k) = c(k)
62        continue
       elseif (anstype.le.rows+8*rows) then
   do 63 k=1,cols
     ff10(anstype-8*rows,k) = c(k)
```

```
63     continue
       endif
       if (numsent .lt. max_tag) then        ! send another row
          call MPI_SEND(dummy,1, MPI_DOUBLE_PRECISION,
    &                   sender, numsent+1, MPI_COMM_WORLD, ierr)
          numsent = numsent+1
       else      ! Tell sender that there is no more work
          call MPI_SEND(MPI_BOTTOM, 0, MPI_DOUBLE_PRECISION,
    &                    sender, 0, MPI_COMM_WORLD, ierr)
       endif
70    continue
      open(201,file='ff1.matrix',status='unknown')
      open(202,file='ff2.matrix',status='unknown')
      open(203,file='ff3.matrix',status='unknown')
      open(204,file='ff4.matrix',status='unknown')
      open(205,file='ff5.matrix',status='unknown')
      open(207,file='ff7.matrix',status='unknown')
      open(208,file='ff8.matrix',status='unknown')
      open(209,file='ff9.matrix',status='unknown')
      open(210,file='ff10.matrix',status='unknown')
      open(221,file='ff1.col',status='unknown')
      open(222,file='ff2.col',status='unknown')
      open(223,file='ff3.col',status='unknown')
      open(224,file='ff4.col',status='unknown')
      open(225,file='ff5.col',status='unknown')
      open(227,file='ff7.col',status='unknown')
      open(228,file='ff8.col',status='unknown')
      open(229,file='ff9.col',status='unknown')
      open(230,file='ff10.col',status='unknown')
      do 71 i=1,rows
         do 72 k=1,cols
   if (k.eq.cols) then
      ASSIGN 255 TO frmt
   else
      ASSIGN 250 to frmt
   endif
           WRITE(201,frmt)ff1(i,k)
           WRITE(202,frmt)ff2(i,k)
           WRITE(203,frmt)ff3(i,k)
           WRITE(204,frmt)ff4(i,k)
           WRITE(205,frmt)ff5(i,k)
           WRITE(207,frmt)ff7(i,k)
           WRITE(208,frmt)ff8(i,k)
           WRITE(209,frmt)ff9(i,k)
```

```
   WRITE(210,frmt)ff10(i,k)
   WRITE(221,255)ff1(i,k)
           WRITE(222,255)ff2(i,k)
           WRITE(223,255)ff3(i,k)
           WRITE(224,255)ff4(i,k)
           WRITE(225,255)ff5(i,k)
           WRITE(227,255)ff7(i,k)
           WRITE(228,255)ff8(i,k)
           WRITE(229,255)ff9(i,k)
   WRITE(230,255)ff10(i,k)
 72      continue
 71    continue
 250   FORMAT(G27.18E2,$)
 255   FORMAT(G27.18E2)
       close(201)
       close(202)
       close(203)
       close(204)
       close(205)
       close(207)
       close(208)
       close(209)
       close(210)
       close(221)
       close(222)
       close(223)
       close(224)
       close(225)
       close(227)
       close(228)
       close(229)
       close(230)
       print*,'End!'
c**********************************************************************
      else
c        slaves receive b, then compute dot products until
c        done message received
c        skip if more processes than work
         if (rank .gt. max_tag)
     &       goto 200
 90      call MPI_RECV(dummy, 1, MPI_DOUBLE_PRECISION, master,
     &                 MPI_ANY_TAG, MPI_COMM_WORLD, status, ierr)
         if (status(MPI_TAG) .eq. 0) then
            go to 200
```

```
        else
           tagnum=status(MPI_TAG)
           if (tagnum.le.rows) then
              do 100 i = 1,cols
 j1=tagnum
 j2=i
                 call qsimp(F1,a,b,s)
 c(i) = s
100           continue
           elseif (tagnum.le.rows+rows) then
              do 101 i = 1,cols
                 j1=tagnum-rows
 j2=i
                 call qsimp(F2,a,b,s)
 c(i) = s
101           continue
           elseif (tagnum.le.rows+2*rows) then
              do 102 i = 1,cols
                 j1=tagnum-2*rows
 j2=i
                 call qsimp(F3,a,b,s)
 c(i) = s
102           continue
           elseif (tagnum.le.rows+3*rows) then
              do 103 i = 1,cols
                 j1=tagnum-3*rows
 j2=i
                 call qsimp(F4,a,b,s)
 c(i) = s
103           continue
           elseif (tagnum.le.rows+4*rows) then
              do 104 i = 1,cols
                 j1=tagnum-4*rows
 j2=i
                 call qsimp(F5,a,b,s)
 c(i) = s
104           continue
           elseif (tagnum.le.rows+5*rows) then
              do 105 i = 1,cols
                 j1=tagnum-5*rows
 j2=i
                 call qsimp(F7,a,b,s)
 c(i) = s
105           continue
```

```
           elseif (tagnum.le.rows+6*rows) then
              do 106 i = 1,cols
                 j1=tagnum-6*rows
 j2=i
                 call qsimp(F8,a,b,s)
 c(i) = s
106           continue
           elseif (tagnum.le.rows+7*rows) then
              do 107 i = 1,cols
                 j1=tagnum-7*rows
 j2=i
                 call qsimp(F9,a,b,s)
 c(i) = s
107           continue
           elseif (tagnum.le.rows+8*rows) then
              do 108 i = 1,cols
                 j1=tagnum-8*rows
 j2=i
                 call qsimp(F10,a,b,s)
 c(i) = s
108           continue
           endif
           call MPI_SEND(c, cols, MPI_DOUBLE_PRECISION, master,
    &                    tagnum, MPI_COMM_WORLD, ierr)
           go to 90
         endif
200      continue
       endif
       call MPI_FINALIZE(ierr)
       stop
       end

      SUBROUTINE qsimp(func,a,b,s)
      INTEGER JMAX
      REAL*16 a,b,s,EPS,func
      PARAMETER (EPS=1.q-9, JMAX=15)
      INTEGER j
      REAL*16 os,ost,st
      ost=-1.q30
      os=-1.q30
      do 16 j=1,JMAX
         call trapzd(func,a,b,st,j)
 s=(4.q0*st-ost)/3.q0
 if (abs(s-os).lt.EPS*abs(os)) return
```

```
  os=s
  ost=st
16    continue
      return
      END

      SUBROUTINE trapzd(func,a,b,s,n)
      INTEGER n
      REAL*16 a,b,s,func
      INTEGER it,j
      REAL*16 del,sum,tnm,xx
      if (n.eq.1) then
         s=0.5q0*(b-a)*(func(a)+func(b))
      else
         it=2**(n-2)
  tnm=it
  del=(b-a)/tnm
  xx=a+0.5q0*del
  sum=0.q0
  do 11 j=1,it
    sum=sum+func(xx)
    xx=xx+del
11       continue
         s=0.5q0*(s+(b-a)*sum/tnm)
      endif
      RETURN
      END

       REAL*16 FUNCTION F1(x)
       REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
     & VF1,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
     & esos,GAM0,GAM1,GAM2,GAM3
       EXTERNAL laguerre
       COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
       COMMON /fnums/esov,et,esos,ec
       COMMON /coeff0/binom0(100)
       COMMON /coeff1/binom1(100)
       COMMON /coeff2/binom2(100)
       COMMON /coeff3/binom3(100)
       GAM0=0.88622692545275801364908374167 1q0
       GAM1=1.32934038817913702047362561251q0
       GAM2=3.32335097044784255118406403126q0
       GAM3=11.63172839656744892914422410 94q0
       NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
```

```
  if (l1.eq.0) then
     NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.1) then
     NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.2) then
     NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.3) then
     NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
  endif
  if (l2.eq.0) then
     NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.1) then
     NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.2) then
     NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.3) then
     NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
  endif
  NK1=NKN/NK1D
  NK2=NKN/NK2D
  x1=(x/beta)**2.q0
  E1=sqrt(x**2.q0+m1sq)
  E2=sqrt(x**2.q0+m2sq)
  VF1=sqrt(1.q0+(x**2.q0)/(E1*E2))
  R1=NK1*NK2/(2.q0*PI)**3.q0
  R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
& laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
  F1=R1*R2*VF1
  RETURN
  END

  REAL*16 FUNCTION F2(x)
  REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
& VF2,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
& esos,GAM0,GAM1,GAM2,GAM3
  EXTERNAL laguerre
  COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
  COMMON /fnums/esov,et,esos,ec
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.886226925452758013649083741671q0
  GAM1=1.329340388179137020473625612510q0
```

```
      GAM2=3.32335097044784255118406403126q0
      GAM3=11.631728396567448929l442241094q0
      NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
      if (l1.eq.0) then
         NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.1) then
         NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.2) then
         NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.3) then
         NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
      endif
      if (l2.eq.0) then
         NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.1) then
         NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.2) then
         NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.3) then
         NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
      endif
      NK1=NKN/NK1D
      NK2=NKN/NK2D
      x1=(x/beta)**2.q0
      E1=sqrt(x**2.q0+m1sq)
      E2=sqrt(x**2.q0+m2sq)
      VF2=(m1sq/E1**2.q0)**(0.5q0+esov)
      R1=NK1*NK2/(2.q0*PI)**3.q0
      R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
     & laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
      F2=R1*R2*VF2
      RETURN
      END

      REAL*16 FUNCTION F3(x)
      REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
     & VF3,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
     & esos,GAM0,GAM1,GAM2,GAM3
      EXTERNAL laguerre
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /fnums/esov,et,esos,ec
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
```

```
      COMMON /coeff3/binom3(100)
      GAM0=0.886226925452758013649083741671q0
      GAM1=1.329340388179137020473625612510q0
      GAM2=3.323350970447842551184064031260q0
      GAM3=11.631728396567448929144224109400q0
      NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
      if (l1.eq.0) then
         NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.1) then
         NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.2) then
         NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.3) then
         NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
      endif
      if (l2.eq.0) then
         NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.1) then
         NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.2) then
         NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.3) then
         NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
      endif
      NK1=NKN/NK1D
      NK2=NKN/NK2D
      x1=(x/beta)**2.q0
      E1=sqrt(x**2.q0+m1sq)
      E2=sqrt(x**2.q0+m2sq)
      VF3=(m2sq/E2**2.q0)**(0.5q0+esov)
      R1=NK1*NK2/(2.q0*PI)**3.q0
      R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
     & laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
      F3=R1*R2*VF3
      RETURN
      END

      REAL*16 FUNCTION F4(x)
      REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
     & VF4,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
     & esos,GAM0,GAM1,GAM2,GAM3
      EXTERNAL laguerre
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /fnums/esov,et,esos,ec
```

```
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.88622692545275801364908374167 1q0
  GAM1=1.32934038817913702047362561251q0
  GAM2=3.32335097044784255118406403126q0
  GAM3=11.63172839656744892914422410 94q0
  NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
  if (l1.eq.0) then
     NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.1) then
     NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.2) then
     NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.3) then
     NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
  endif
  if (l2.eq.0) then
     NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.1) then
     NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.2) then
     NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.3) then
     NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
  endif
  NK1=NKN/NK1D
  NK2=NKN/NK2D
  x1=(x/beta)**2.q0
  E1=sqrt(x**2.q0+m1sq)
  E2=sqrt(x**2.q0+m2sq)
  VF4=(m1*m2/(E1*E2))**(0.5q0+esov)
  R1=NK1*NK2/(2.q0*PI)**3.q0
  R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
& laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
  F4=R1*R2*VF4
  RETURN
  END

  REAL*16 FUNCTION F5(x)
  REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
& VF5,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
& esos,GAM0,GAM1,GAM2,GAM3
```

```
      EXTERNAL laguerre
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /fnums/esov,et,esos,ec
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      GAM0=0.88622692545275801364908374167 1q0
      GAM1=1.32934038817913702047362561251q0
      GAM2=3.32335097044784255118406403126q0
      GAM3=11.63172839656744892914422410 94q0
      NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
      if (l1.eq.0) then
         NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.1) then
         NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.2) then
         NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.3) then
         NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
      endif
      if (l2.eq.0) then
         NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.1) then
         NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.2) then
         NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.3) then
         NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
      endif
      NK1=NKN/NK1D
      NK2=NKN/NK2D
      x1=(x/beta)**2.q0
      E1=sqrt(x**2.q0+m1sq)
      E2=sqrt(x**2.q0+m2sq)
      VF5=(m1*m2/(E1*E2))**(0.5q0+et)
      R1=NK1*NK2/(2.q0*PI)**3.q0
      R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
     & laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
      F5=R1*R2*VF5
      RETURN
      END

      REAL*16 FUNCTION F7(x)
```

```
      REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
     & VF7,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
     & esos,GAM0,GAM1,GAM2,GAM3
      EXTERNAL laguerre
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /fnums/esov,et,esos,ec
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      GAM0=0.88622692545275801364908374167 1q0
      GAM1=1.32934038817913702047362561251q0
      GAM2=3.32335097044784255118406403126q0
      GAM3=11.63172839656744892914422410 94q0
      NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
      if (l1.eq.0) then
         NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.1) then
         NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.2) then
         NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.3) then
         NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
      endif
      if (l2.eq.0) then
         NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.1) then
         NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.2) then
         NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.3) then
         NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
      endif
      NK1=NKN/NK1D
      NK2=NKN/NK2D
      x1=(x/beta)**2.q0
      E1=sqrt(x**2.q0+m1sq)
      E2=sqrt(x**2.q0+m2sq)
      VF7=(m1sq/E1**2.q0)**(0.5q0+esos)
      R1=NK1*NK2/(2.q0*PI)**3.q0
      R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*laguerre(l1,j1,x1)*
     &     laguerre(l2,j2,x1)
      F7=R1*R2*VF7
      RETURN
```

```
      END

      REAL*16 FUNCTION F8(x)
      REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
     & VF8,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
     & esos,GAM0,GAM1,GAM2,GAM3
      EXTERNAL laguerre
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /fnums/esov,et,esos,ec
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      GAM0=0.88622692545275801364908374167l1q0
      GAM1=1.32934038817913702047362561251q0
      GAM2=3.32335097044784255118406403126q0
      GAM3=11.63172839656744892914422410941q0
      NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
      if (l1.eq.0) then
         NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.1) then
         NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.2) then
         NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.3) then
         NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
      endif
      if (l2.eq.0) then
         NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.1) then
         NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.2) then
         NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.3) then
         NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
      endif
      NK1=NKN/NK1D
      NK2=NKN/NK2D
      x1=(x/beta)**2.q0
      E1=sqrt(x**2.q0+m1sq)
      E2=sqrt(x**2.q0+m2sq)
      VF8=(m2sq/E2**2.q0)**(0.5q0+esos)
      R1=NK1*NK2/(2.q0*PI)**3.q0
      R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
```

```
& laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
  F8=R1*R2*VF8
  RETURN
  END

  REAL*16 FUNCTION F9(x)
  REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
& VF9,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
& esos,GAM0,GAM1,GAM2,GAM3
  EXTERNAL laguerre
  COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
  COMMON /fnums/esov,et,esos,ec
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.88622692545275801364908374167 1q0
  GAM1=1.32934038817913702047362561251q0
  GAM2=3.32335097044784255118406403126q0
  GAM3=11.63172839656744892914422410 94q0
  NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
  if (l1.eq.0) then
     NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.1) then
     NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.2) then
     NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
  elseif (l1.eq.3) then
     NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
  endif
  if (l2.eq.0) then
     NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.1) then
     NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.2) then
     NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
  elseif (l2.eq.3) then
     NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
  endif
  NK1=NKN/NK1D
  NK2=NKN/NK2D
  x1=(x/beta)**2.q0
  E1=sqrt(x**2.q0+m1sq)
  E2=sqrt(x**2.q0+m2sq)
```

```
      VF9=E1+E2
      R1=NK1*NK2/(2.q0*PI)**3.q0
      R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
     & laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
      F9=R1*R2*VF9
      RETURN
      END

      REAL*16 FUNCTION F10(x)
      REAL*16 PI,NKN,NK1D,NK2D,NK1,NK2,E1,E2,x,x1,ec,
     & VF10,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,esov,et,
     & esos,GAM0,GAM1,GAM2,GAM3
      EXTERNAL laguerre
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /fnums/esov,et,esos,ec
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      GAM0=0.88622692545275801364908374167 1q0
      GAM1=1.32934038817913702047362561251q0
      GAM2=3.32335097044784255118406403126q0
      GAM3=11.63172839656744892914422410 94q0
      NKN=sqrt(2.q0)*(2.q0*PI)**1.5q0
      if (l1.eq.0) then
         NK1D=sqrt(GAM0*binom0(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.1) then
         NK1D=sqrt(GAM1*binom1(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.2) then
         NK1D=sqrt(GAM2*binom2(j1)*(beta**(2.q0*l1+3.q0)))
      elseif (l1.eq.3) then
         NK1D=sqrt(GAM3*binom3(j1)*(beta**(2.q0*l1+3.q0)))
      endif
      if (l2.eq.0) then
         NK2D=sqrt(GAM0*binom0(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.1) then
         NK2D=sqrt(GAM1*binom1(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.2) then
         NK2D=sqrt(GAM2*binom2(j2)*beta**(2.q0*l2+3.q0))
      elseif (l2.eq.3) then
         NK2D=sqrt(GAM3*binom3(j2)*beta**(2.q0*l2+3.q0))
      endif
      NK1=NKN/NK1D
      NK2=NKN/NK2D
```

```
      x1=(x/beta)**2.q0
      E1=sqrt(x**2.q0+m1sq)
      E2=sqrt(x**2.q0+m2sq)
      VF10=(m1*m2/(E1*E2))**(0.5q0+ec)
      R1=NK1*NK2/(2.q0*PI)**3.q0
      R2=((-1)**(j1+j2-2))*(x**(l1+l2+2))*exp(-x1)*
     & laguerre(l1,j1,x1)*laguerre(l2,j2,x1)
      F10=R1*R2*VF10
      RETURN
      END

      REAL*16 FUNCTION laguerre(ll,jj,x)
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      REAL*16 x,am,bm,cm
      integer jj,ll,M
      am = 1.q0
      do 16 M=(jj-1),1,-1
         bm=1.q0*(jj-M)
     cm=1.q0*M*(M+ll+0.5q0)
         am=1.q0-(bm/cm)*(x*am)
16    continue
      if (ll.eq.0) then
      laguerre=am*binom0(jj)
      elseif (ll.eq.1) then
      laguerre=am*binom1(jj)
      elseif (ll.eq.2) then
      laguerre=am*binom2(jj)
      elseif (ll.eq.3) then
      laguerre=am*binom3(jj)
      endif
      RETURN
      END

      BLOCK DATA binomial0
      COMMON /coeff0/binom0(100)
      DATA binom0/1.q0,1.5q0,1.875q0,2.1875q0,2.4609375q0,
     & 2.70703125q0,2.9326171875q0,3.14208984375q0,
     & 3.338470458984375q0,3.5239410400390625q0,
     & 3.700138092041015625q0,3.8683261871337890625q0,
     & 4.02950644493103027343750q0,4.184487462043762207031250q0,
     & 4.33393344283103942871093750q0,4.4783978909254074096679687500q0,
```

```
& 4.6183478250168263912200927734375q0,
& 4.7541815845761448144949127197265625q0,
& 4.8862421841477043926715850830078125q0,
& 5.0148275047831702977418899536132812q0,
& 5.1401981924027495551854372024536132q0,
& 5.2625838636504340684041380882263183q0,
& 5.3821880423697621154133230447769165q0,
& 5.4991921302473656396614387631416320q0,
& 5.6137586329608524238210520707070827q0,
& 5.7260338056200694722974731121212244q0,
& 5.8361498403435323467647322104312479q0,
& 5.9442266892387829457788939180318266q0,
& 6.0503735944037612126678027379966806q0,
& 6.1546903805141708887482820955483475q0,
& 6.2572685535227404035607534638074866q0,
& 6.3581922398698813778117333583850268q0,
& 6.4575389936178482743400416921097928q0,
& 6.5553804935211490057694362632023654q0,
& 6.6517831478376364911483985611906355q0,
& 6.7468086213781741553076613977790732q0,
& 6.8405142966750932407980455838593381q0,
& 6.9329536790625945008088299836412210q0,
& 7.0241767537871023231878935360575529q0,
& 7.1142303019125779939979947352377779q0,
& 7.2031581806864852189229696694282501q0,
& 7.2910015731338813801293473483237166q0,
& 7.3777992109092847298927919596132847q0,
& 7.4635875738268345523334058196087880,
& 7.5484010689839576722462854311952515q0,
& 7.6322721919726683130490219359863098q0,
& 7.7152316723201973164517286961600740q0,
& 7.7973086050044547347118534695234791q0,
& 7.8785305696399178048651019431643487q0,
& 7.9589237387178761498127050242170461q0,
& 8.0385129761050549113108320744592166q0,
& 8.1173219268511829006374088595029344q0,
& 8.1953730992247519669896916369981549q0,
& 8.2726879397834760421499717467811564q0,
& 8.3492869021888785980958048185106115q0,
& 8.4251895103905956762603121350425262q0,
& 8.5004144167333688519412077791054059q0,
& 8.5749794554766440173091131105010674q0,
& 8.6489016921617875002169502924881455q0,
& 8.7221974692140060383543820746278756q0,
```

```
& 8.7948824481241227553406685919164412q0,
& 8.8669716485185827779254281705387071q0,
& 8.9384794843937326390377300106236967q0,
& 9.0094197977619368663316802488032498q0,
& 9.0798058899319519980998965007470252q0,
& 9.1496505506237362442391264738296946q0,
& 9.2189660850981584885136653107526469q0,
& 9.2877643394645626563383941563552786q0,
& 9.3560567243135667935173529369167144q0,
& 9.4238542368085926398471888277639370q0,
& 9.4911674813572254444175258908193937q0,
& 9.5580066889724171728993394534307979q0,
& 9.6243817354236145143778070885240673q0,
& 9.6903021582689817370790249452947801q0,
& 9.7557771728518802623295588976278529q0,
& 9.8208156873375594640784226236120386q0,
& 9.8854263168595170921315701408726441q0,
& 9.9496173968391243459765803365925964q0,
& 10.013396995536811040502071236186139q0,
& 10.076772925888309844555881813630355q0,
& 10.139752756675111781084356074965544q0,
& 10.202343823074340866152778026045579q0,
& 10.264553236629672212897612038399515q0,
& 10.326387894681658190083742231401922q0,
& 10.387854489292858536453288316112648q0,
& 10.448959515700463586667719423854487q0,
& 10.509709280326629072636717792597827q0,
& 10.570109908374483262709342607497814q0,
& 10.630167351035701917611100235949506q0,
& 10.689887392333655299170713158623380q0,
& 10.749275655624397828610550453949065q0,
& 10.808337609776180234262256775124610q0,
& 10.867078575046702952926725561945939q0,
& 10.925503728675986302136009032709089q0,
& 10.983618110211496867572902697776691q0,
& 11.041426626581031061612760080396568q0,
& 11.098934056927807265058659872481967q0,
& 11.156145057221249570548652964608163q0,
& 11.213064164656051864275942010345959q0,
& 11.269695801851284449449052828580030q0/
  END

  BLOCK DATA binomial1
  COMMON /coeff1/binom1(100)
```

```
      DATA binom1/1.q0,2.5q0,4.375q0,6.5625q0,9.0234375q0,
     & 11.73046875q0,
     & 14.6630859375q0,17.80517578125q0,21.143646240234375q0,
     & 24.6675872802734375q0,28.367725372314453125q0,
     & 32.2360515594482421875q0,36.2655580043792724609375q0,
     & 40.450045664230346679687 5q0,44.7839789092540740966796875q0,
     & 49.26237680017948150634765625q0,
     & 53.880724625196307897567749023 4375q0,
     & 58.6349062097724527120590209960 9375q0,
     & 63.52114839392015710473060607910 1562q0,
     & 68.53597589870332740247249603271 4843q0,
     & 73.67617409110607695765793323516 8457q0,
     & 78.93875795475651102606207132339 4775q0,
     & 84.32094599712627314147539436817 1691q0,
     & 89.82013812737363878113683313131 3324q0,
     & 95.43389676033449120495788520202 0406q0,
     & 101.15993056595456067725535831414163q0,
     & 106.99608040629809302402009052457288q0,
     & 112.94030709553687596979898444260470q0,
     & 118.99068068994063718246678718060138q0,
     & 125.14537107045480807121506927614973q0,
     & 131.40263962397754847477582273995722q0,
     & 137.76083186384742985258755609834224q0,
     & 144.21837085746527812692759779045204q0,
     & 150.77375135098642713269703405365440q0,
     & 157.42553449882406362384543261484504q0,
     & 164.17234312020223777915309401262411q0,
     & 171.01285741687733101995113959648345q0,
     & 177.94581109593992552075996958012467q0,
     & 184.96998784972702784394786311618222q0,
     & 192.08421815163960583794585785142000q0,
     & 199.28737633232609105686882752084825q0,
     & 206.57837905459972436998174869171 97q0,
     & 213.95617711636925716689096682878525q0,
     & 221.41976469019609171922437264839404q0,
     & 228.96816575918004939147065807958929q0,
     & 236.60043795115271770451968001557560q0,
     & 244.31566962347291502097140871173568q0,
     & 252.11297822847736975568326218125915q0,
     & 259.99150879811728756054836412442350q0,
     & 267.95043253683516371036106914864055q0,
     & 275.98894551294021862167190122309977q0,
     & 284.10626743979140152230931008260270q0,
     & 292.30164053901615348929900171960086q0,
```

```
& 300.57432847879962953144897346638202q0,
& 308.92361538098850812954477828489263q0,
& 317.34880489137910380580509041993515q0,
& 325.84921930811247265774629819904056q0,
& 334.42419876358911667505541130954163q0,
& 343.07310045575090417527236160202977q0,
& 351.79529792496491021362674367665765q0,
& 360.59018037308903296896741226857409q0,
& 369.45715202160761574689284043911280q0,
& 378.39563150600134838593057044973649q0,
& 387.40505130376328525226225069853974q0,
& 396.48485719369523725036214719928677q0,
& 405.63450774431897349460127367311646q0,
& 414.85347382941713198311493898386911q0,
& 424.14123816888169463945333314022439q0,
& 433.49729489319526143297068607714110q0,
& 442.92114913000385407281787490490504q0,
& 452.41231661136107951723540079572443q0,
& 461.97032330033349669013474024915523q0,
& 471.59470503575711120451254733767930q0,
& 481.28500719402609294159157228297408q0,
& 491.04078436687797320392113118060193q0,
& 500.86160005421553266799955380421397q0,
& 510.74702637107504976013112394508661q0,
& 520.69664376791417410610770428167921q0,
& 530.71004076345098514660977551786535q0,
& 540.78681368933929499116565733149570q0,
& 550.92656644601440677225001340646125q0,
& 561.12891026908874763840279143250683q0,
& 571.39346350571841985130040347090634q0,
& 581.71985140040007804138414570230826q0,
& 592.10770588969293657783743401842091q0,
& 602.55666540539340016450515344227540q0,
& 613.06637468572002923714187123487322q0,
& 623.63648459409451249985121384237104q0,
& 634.26665194513021441746231407832055q0,
& 644.95653933746386971663302723694393q0,
& 655.70581499308826754524357769089299q0,
& 666.51415260286444777950583446601760q0,
& 677.38123117791115073243256002796354q0,
& 688.30673490658713703456856906067263q0,
& 699.29035301679863390214147175844932q0,
& 710.33177964337966496375423183884589q0,
& 721.43071370030747222881289171132786q0,
```

```
& 732.5868587575287217993615446759360 2q0,
& 743.7999229221847736636374866862819 8q0,
& 755.0696187240360581130865395148620 1q0/
  END

  BLOCK DATA binomial2
  COMMON /coeff2/binom2(100)
  DATA binom2/1.q0,3.5q0,7.875q0,14.4375q0,
& 23.4609375q0,35.19140625q0,
& 49.8544921875q0,67.65966796875q0,88.803314208984375q0,
& 113.4709014892578125q0,141.838626861572265625q0,
& 174.0746784210205078125q0,210.3402364253997802734375q0,
& 250.7902818918228149414062 5q0,295.5742608010768890380859375q0,
& 344.8366376012563705444335 9375q0,
& 398.7173622264526784420013427734375q0,
& 457.3522684362251311540603637695312 5q0,
& 520.8734168301452882587909698486328 1q0,
& 589.4093927288486156612634658813476 5q0,
& 663.0855668199546926189213991165161 1q0,
& 742.0243247747112036449834704399108 8q0,
& 826.3452707718374767864588648080825 8q0,
& 916.1654088992111155675956979393959 0q0,
& 1011.599305659545606772553583141416 3q0,
& 1112.759236225500167449808941455557 9q0,
& 1219.755316631798260473829031980130 8q0,
& 1332.695623727335136443628016422735 5q0,
& 1451.686304417275773626094803603336 9q0,
& 1576.831675487730581697309872879486 6q0,
& 1708.234315111708130172085695619443 8q0,
& 1845.995146975555560024673251717786 1q0,
& 1990.213517833020838151600849508238 1q0,
& 2140.987269184007265284297883561892 5q0,
& 2298.412803682831328908143316176737 6q0,
& 2462.585146803033566687296410189361 7q0,
& 2633.598004219910897707247549785845 1q0,
& 2811.543815315850823228007519365969 8q0,
& 2996.513803165577851071955382482152 0q0,
& 3188.598021317217456909901240333572 0q0,
& 3387.885397649543547966770067854420 3q0,
& 3594.463775555003520403768242723592 3q0,
& 3808.419952671372777570659209552377 5q0,
& 4029.839717361568869289883582200771 6q0,
& 4258.807883120748918681354240280360 8q0,
& 4495.408321071901636385873920295936 5q0,
```

```
& 4739.72399069537455140684532900767 21q0,
& 4991.83696892385192116252859118893 13q0,
& 5251.82847772196920872307695531335 48q0,
& 5519.77891025880437243343802446199 53q0,
& 5795.76785577174459105510992568509 52q0,
& 6079.87412321153599257741923576769 79q0,
& 6372.17576375055214606671823748729 87q0,
& 6672.75009222935177559816721095368 07q0,
& 6981.67370761034028372771198923857 34q0,
& 7299.02251250171938753351707965850 85q0,
& 7624.87173180983186019126337785754 91q0,
& 7959.29593057342097686631878916709 07q0,
& 8302.36903102917188104159115076912 05q0,
& 8654.16432895413679125521789444577 81q0,
& 9014.75450932722582422418530671435 22q0,
& 9384.21166134883343997107814715346 50q0,
& 9762.60729285483478835700871760320 15q0,
& 10150.01234415859807360927096830174 1q0,
& 10546.49720135229331085963311550102 8q0,
& 10952.13170909661228435423438917414 4q0,
& 11366.98518292602941633734932815801 4q0,
& 11791.12642109491111097680266129823 8q0,
& 12224.62371598810637240977334737537 9q0,
& 12667.54486511811022648259122228028 4q0,
& 13119.95718172947130599982662307600 9q0,
& 13581.92750502980480268996136332516 4q0,
& 14053.52221006556191389447391066284 3q0,
& 14534.80721725958800683606548294581 7q0,
& 15025.84800162646598003998661412641 9q0,
& 15526.70960168068151270798616793063 3q0,
& 16037.45662805175656246811729187572 0q0,
& 16558.15327181967073657422499615739 9q0,
& 17088.86331258312172172083477167526 4q0,
& 17629.65012627246101671200042900676 0q0,
& 18180.57669271847542348425044241322 1q0,
& 18741.70560298756417112265323384572 8q0,
& 19313.09906649328259097395363731663 4q0,
& 19894.81891789368266901533778301894 3q0,
& 20486.92662378337560559317521703736 4q0,
& 21089.48328918876900575768037047963 9q0,
& 21702.54966387448903499482224171451 2q0,
& 22326.18614846858354749467345555688 3q0,
& 22960.45280041371376191213576963520 4q0,
& 23605.40933975117763162876879687214 8q0,
```

```
& 24261.11515474426589917401237456304 1q0,
& 24927.62930734713034695351820902905 8q0,
& 25605.01053852504149768595076905702 2q0,
& 26293.31727343162863472051933811769 4q0,
& 26992.60762644842726862266080987614 4q0,
& 27702.93940609180693358641504171499 0q0,
& 28424.37011979211440581522793342631 7q0,
& 29156.95697854964312761458947810225 4q0,
& 29900.75690147182790127822696478853 5q0,
& 30655.82652019586395939131350430339 7q0/
  END

  BLOCK DATA binomial3
  COMMON /coeff3/binom3(100)
  DATA binom3/1.q0,4.5q0,12.3750q0,26.8125q0,
& 50.2734375q0,85.46484375q0,
& 135.3193359375q0,202.97900390625q0, 291.782318115234375q0,
& 405.2532196044921875q0,547.091846466064453125q0,
& 721.1665248870849609375q0, 931.5067613124847412109375q0,
& 1182.297043204307556152343 75q0,
& 1477.871304005384445190429 6875q0,
& 1822.707941606640815734863 28125q0,
& 2221.425303833093494176864 6240234375q0,
& 2678.777572269318625330924 9877929688q0,
& 3199.650989099463913589715 9576416016q0,
& 3789.060381828312529250979 4235229492q0,
& 4452.145948648267221869900 8226394653q0,
& 5194.170273422978425514884 2930793762q0,
& 6020.515544194815902301343 1578874588q0,
& 6936.680953094027017868938 8558268547q0,
& 7948.280258753572624641492 4389682710q0,
& 9061.039494979072792091301 3804238290q0,
& 10280.794811610871052565130 412403960q0,
& 11613.490435338206189008758 428826695q0,
& 13065.176739755481962634853 232430032q0,
& 14642.008415243212544332163 105309519q0,
& 16350.242730354920674504248 800928963q0,
& 18196.237877330476234528922 052646749q0,
& 20186.451395163497072680522 902154987q0,
& 22327.438664347504337964820 785716880q0,
& 24625.851468030335666872964 101893617q0,
& 27088.436614833369233560260 512082979q0,
& 29722.034619053280131267508 061868824q0,
& 32533.578434369130954495515 581234794q0,
```

```
& 35530.092237534708805567470963716946q0,
& 38718.690258851926262477372204050518q0,
& 42106.575656501469810444142271904938q0,
& 45701.039432056473330847910514628531q0,
& 49509.459384727846108418569724180908q0,
& 53539.299102089414977708453306381680q0,
& 57798.106985210163896389807546662041q0,
& 62293.515306282065532775681466957977q0,
& 67033.239296977440084182526795965649q0,
& 72025.076265901292005345055387154581q0,
& 77276.904743623261214068132342467936q0,
& 82796.683653882065586501570366929931q0,
& 88592.451509653810177556680292615026q0,
& 94672.325632865346170134099528382724q0,
& 101044.501396615898316200817765870002q0,
& 107717.251488845250091798984976823700q0,
& 114698.925196455590375526696966062280q0,
& 121997.947708957309763060214045720790q0,
& 129622.819440767141623251477423578330q0,
& 137582.115371340562600117796212745430q0,
& 145884.484402369734481159387363514550q0,
& 154538.648731323871272414605257960320q0,
& 163553.403240651097096638790564674680q0,
& 172937.614901999930536609868711828140q0,
& 182700.222194854765324966877429431340q0,
& 192850.234539013363398576148397733080q0,
& 203396.731740365656709435781513234110q0,
& 214348.863449462268993790015902408260q0,
& 225715.848632388298410127365230566270q0,
& 237506.975053483209521104167891864510q0,
& 249731.598769471315893513941239239890q0,
& 262399.143634589426119996532461520170q0,
& 275519.100816318897425996359084596180q0,
& 289101.028321348702228686320447921340q0,
& 303154.550531414264142580794358584190q0,
& 317689.357748673852149416859841530000q0,
& 332715.205750300318129456846455656420q0,
& 348241.915351980999642164832623587060q0,
& 364279.371980032756204632949915462780q0,
& 380837.525251852426941207174911620170q0,
& 397926.388564435548662928009683295440q0,
& 415556.038690708009679640010112302200q0,
& 433736.615383426485103124260554715420q0,
& 452478.320986414049274246913788561150q0,
```

```
& 471791.42005290733186522086742587778q0,
& 491686.23897080101453423620520889672q0,
& 512173.16559458439013982938042593409q0,
& 533262.64888377315914558706079641373q0,
& 554965.19854764764818058188303812824q0,
& 577291.38469611623172807655649368512q0,
& 600251.83749652994548998869226332033q0,
& 623857.24683628112312161746106019247q0,
& 648118.36199102538902079147343475551q0,
& 673045.99129837251936774499164378457q0,
& 698651.00183689756086543094241284159q0,
& 724944.31911032918950015146175095929q0,
& 751936.92673677761676877412256083543q0,
& 779639.86614286942370236053760255042q0,
& 808064.23626266153810817576553597674q0,
& 837221.19324121118123579035501407899q0,
& 867121.95014268300913706858197886753q0,
& 897777.77666287887309645989548317093/
  END
```

Appendix 3

Code for Position-Dependent Matrix Elements

```
      program vg
      include 'mpif.h'
      double precision dummy
      integer l1,l2,j1,j2
      real*16 laguerre,PI,beta,m1,m2,m1sq,m2sq,
     &alpha(3),tau(3),sigma0,sigma12,s,ss,bb,cc
      integer MAX_ROWS, MAX_COLS, rows, cols, max_tag
      parameter (MAX_ROWS=100,MAX_COLS=100,rows=40,cols=40)
      double precision gg1(MAX_ROWS,MAX_COLS),gg2(MAX_ROWS,MAX_COLS),
     &gg5(MAX_ROWS,MAX_COLS),gg6(MAX_ROWS,MAX_COLS),
     &gg7(MAX_ROWS,MAX_COLS),gg10(MAX_ROWS,MAX_COLS)
      double precision c(cols)
      integer myid, master, numprocs, ierr, status(MPI_STATUS_SIZE)
      integer i, j, numsent, sender
      integer anstype, tagnum, frmt
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
      COMMON /gnums/alpha,tau,bb,cc,sigma12
c*************************************************************
c Parameters for the integration routine
      external G1,G2,G5,G6,G7,G10
      real*16 a,b
      a = 0.0001q0
      b = 50.0q0
c*************************************************************
```

```
c Additional stuff
      PI=3.14159265358979323846264338328q0
      beta=0.75q0
      l1=0
      l2=0
      m1=0.22q0
      m2=0.22q0
      m1sq=m1**2.q0
      m2sq=m2**2.q0
      bb=0.18q0
      cc=-0.253q0
      ss=1.55q0
      sigma0=1.8q0
      sigma12=sqrt(sigma0**2.q0*(0.5q0+
     &0.5q0*(4.q0*m1*m2/(m1+m2)**2.q0)**4.q0)+ss**2.q0*
     &(2.q0*m1*m2/(m1+m2))**2.q0)
      alpha(1)=0.25q0
      alpha(2)=0.15q0
      alpha(3)=0.2q0
      tau(1)=1.q0/(sqrt(1.q0/0.25q0+1/sigma12**2.q0))
      tau(2)=1.q0/(sqrt(1.q0/2.5q0+1/sigma12**2.q0))
      tau(3)=1.q0/(sqrt(1.q0/250.q0+1/sigma12**2.q0))
c*************************************************************
      call MPI_INIT( ierr )
      call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
      call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
      master = 0
      max_tag = 6*rows
c*****************************************************************
      if ( myid .eq. master ) then
      print*,'Start!'
c    master initializes and then dispatches
c    initialize aa and bb  (arbitrary)
      do 30 i = 1,rows
         do 31 j = 1,cols
    gg1(i,j) = 1.d0
            gg2(i,j) = 1.d0
    gg5(i,j) = 1.d0
    gg6(i,j) = 1.d0
    gg7(i,j) = 1.d0
    gg10(i,j)= 1.d0
31       continue
30    continue
      numsent = 0
```

```
c     send b to each slave process
c     send a row to each slave process; tag with tagnum number
      do 40 i = 1,min(numprocs-1,max_tag)
        call MPI_SEND(dummy,1, MPI_DOUBLE_PRECISION, i,
     &                i, MPI_COMM_WORLD, ierr)
        numsent = numsent+1
 40   continue
      do 70 i = 1,max_tag
        call MPI_RECV(c, cols , MPI_DOUBLE_PRECISION,
     &                MPI_ANY_SOURCE, MPI_ANY_TAG,
     &                MPI_COMM_WORLD, status, ierr)
        sender     = status(MPI_SOURCE)
        anstype    = status(MPI_TAG)       ! anstype is tag value
    if (anstype.le.rows) then
           do 55 k=1,cols
      gg1(anstype,k) = c(k)
 55        continue
        elseif (anstype.le.rows+rows) then
           do 56 k=1,cols
              gg2(anstype-rows,k) = c(k)
 56        continue
        elseif (anstype.le.rows+2*rows) then
    do 57 k=1,cols
              gg5(anstype-2*rows,k) = c(k)
 57        continue
        elseif (anstype.le.rows+3*rows) then
    do 58 k=1,cols
              gg6(anstype-3*rows,k) = c(k)
 58        continue
        elseif (anstype.le.rows+4*rows) then
    do 59 k=1,cols
              gg7(anstype-4*rows,k) = c(k)
 59        continue
        elseif (anstype.le.rows+5*rows) then
    do 60 k=1,cols
      gg10(anstype-5*rows,k) = c(k)
 60    continue
        endif
        if (numsent .lt. max_tag) then        ! send another row
           call MPI_SEND(dummy,1, MPI_DOUBLE_PRECISION,
     &                   sender, numsent+1, MPI_COMM_WORLD, ierr)
           numsent = numsent+1
        else      ! Tell sender that there is no more work
           call MPI_SEND(MPI_BOTTOM, 0, MPI_DOUBLE_PRECISION,
```

```
     &                     sender, 0, MPI_COMM_WORLD, ierr)
        endif
70    continue
      open(211,file='gg1.matrix',status='unknown')
      open(212,file='gg2.matrix',status='unknown')
      open(215,file='gg5.matrix',status='unknown')
      open(216,file='gg6.matrix',status='unknown')
      open(217,file='gg7.matrix',status='unknown')
      open(218,file='gg10.matrix',status='unknown')
      open(231,file='gg1.col',status='unknown')
      open(232,file='gg2.col',status='unknown')
      open(235,file='gg5.col',status='unknown')
      open(236,file='gg6.col',status='unknown')
      open(237,file='gg7.col',status='unknown')
      open(238,file='gg10.col',status='unknown')
      do 71 i=1,rows
         do 72 k=1,cols
   if (k.eq.cols) then
      ASSIGN 255 TO frmt
   else
      ASSIGN 250 to frmt
   endif
   WRITE(211,frmt)gg1(i,k)
           WRITE(212,frmt)gg2(i,k)
           WRITE(215,frmt)gg5(i,k)
           WRITE(216,frmt)gg6(i,k)
           WRITE(217,frmt)gg7(i,k)
   WRITE(218,frmt)gg10(i,k)
   WRITE(231,255)gg1(i,k)
           WRITE(232,255)gg2(i,k)
           WRITE(235,255)gg5(i,k)
           WRITE(236,255)gg6(i,k)
           WRITE(237,255)gg7(i,k)
   WRITE(238,255)gg10(i,k)
72       continue
71    continue
250   FORMAT(G27.18E2,$)
255   FORMAT(G27.18E2)
      close(211)
      close(212)
      close(215)
      close(216)
      close(217)
      close(218)
```

```
      close(231)
      close(232)
      close(235)
      close(236)
      close(237)
      close(238)
      print*,'End!'
c*********************************************************************
      else
c        slaves receive b, then compute dot products until
c        done message received
c        skip if more processes than work
         if (rank .gt. max_tag)
     &       goto 200
 90      call MPI_RECV(dummy, 1, MPI_DOUBLE_PRECISION, master,
     &                 MPI_ANY_TAG, MPI_COMM_WORLD, status, ierr)
         if (status(MPI_TAG) .eq. 0) then
            go to 200
         else
            tagnum=status(MPI_TAG)
            if (tagnum.le.rows) then
               do 100 i = 1,cols
                  j1=tagnum
  j2=i
                  call qsimp(G1,a,b,s)
  c(i) = s
 100           continue
            elseif (tagnum.le.rows+rows) then
               do 101 i = 1,cols
                  j1=tagnum-rows
  j2=i
                  call qsimp(G2,a,b,s)
  c(i) = s
 101           continue
            elseif (tagnum.le.rows+2*rows) then
               do 102 i = 1,cols
                  j1=tagnum-2*rows
  j2=i
                  call qsimp(G5,a,b,s)
  c(i) = s
 102           continue
            elseif (tagnum.le.rows+3*rows) then
               do 103 i = 1,cols
                  j1=tagnum-3*rows
```

```
 j2=i
                  call qsimp(G6,a,b,s)
 c(i) = s
103           continue
           elseif (tagnum.le.rows+4*rows) then
              do 104 i = 1,cols
                 j1=tagnum-4*rows
 j2=i
                 call qsimp(G7,a,b,s)
 c(i) = s
104           continue
           elseif (tagnum.le.rows+5*rows) then
              do 105 i = 1,cols
                 j1=tagnum-5*rows
 j2=i
                 call qsimp(G10,a,b,s)
 c(i) = s
105           continue
           endif
           call MPI_SEND(c, cols, MPI_DOUBLE_PRECISION, master,
    &                    tagnum, MPI_COMM_WORLD, ierr)
           go to 90
        endif
200     continue
      endif
      call MPI_FINALIZE(ierr)
      stop
      end

     SUBROUTINE qsimp(func,a,b,s)
     INTEGER JMAX
     REAL*16 a,b,s,EPS,func
     PARAMETER (EPS=1.q-9, JMAX=17)
     INTEGER j
     REAL*16 os,ost,st
     ost=-1.q30
     os=-1.q30
     do 16 j=1,JMAX
        call trapzd(func,a,b,st,j)
 s=(4.q0*st-ost)/3.q0
 if (abs(s-os).lt.EPS*abs(os)) return
 os=s
 ost=st
16    continue
```

```
      return
      END

      SUBROUTINE trapzd(func,a,b,s,n)
      INTEGER n
      REAL*16 a,b,s,func
      INTEGER it,j
      REAL*16 del,sum,tnm,xx
      if (n.eq.1) then
         s=0.5q0*(b-a)*(func(a)+func(b))
      else
         it=2**(n-2)
  tnm=it
  del=(b-a)/tnm
  xx=a+0.5q0*del
  sum=0.q0
  do 11 j=1,it
    sum=sum+func(xx)
    xx=xx+del
11       continue
         s=0.5q0*(s+(b-a)*sum/tnm)
      endif
      RETURN
      END

       REAL*16 FUNCTION G1(x)
       REAL*16 PI,NR1N,NR2N,NR1D,NR2D,NR1,NR2,E1,E2,GX1,GX2,GX3,
     & VG1,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,alpha(3),tau(3),
     & bb,cc,sigma12,x,x1,GAM0,GAM1,GAM2,GAM3
       EXTERNAL laguerre
       COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
       COMMON /gnums/alpha,tau,bb,cc,sigma12
       COMMON /coeff0/binom0(100)
       COMMON /coeff1/binom1(100)
       COMMON /coeff2/binom2(100)
       COMMON /coeff3/binom3(100)
       GAM0=0.88622692545275801364908374167 1q0
       GAM1=1.32934038817913702047362561251q0
       GAM2=3.32335097044784255118406403126q0
       GAM3=11.63172839656744892914422410 94q0
       NR1N=sqrt(2.q0*(beta**(2.q0*l1+3.q0)))
       NR2N=sqrt(2.q0*(beta**(2.q0*l2+3.q0)))
       if (l1.eq.0) then
          NR1D=sqrt(GAM0*binom0(j1))
```

```
  elseif (l1.eq.1) then
     NR1D=sqrt(GAM1*binom1(j1))
  elseif (l1.eq.2) then
     NR1D=sqrt(GAM2*binom2(j1))
  elseif (l1.eq.3) then
     NR1D=sqrt(GAM3*binom3(j1))
  endif
  if (l2.eq.0) then
     NR2D=sqrt(GAM0*binom0(j2))
  elseif (l2.eq.1) then
     NR2D=sqrt(GAM1*binom1(j2))
  elseif (l2.eq.2) then
     NR2D=sqrt(GAM2*binom2(j2))
  elseif (l2.eq.3) then
     NR2D=sqrt(GAM3*binom3(j2))
  endif
  NR1=NR1N/NR1D
  NR2=NR2N/NR2D
  GX1=-(4.q0*alpha(1)/(3.q0*x))*ERF(tau(1)*x)
  GX2=-(4.q0*alpha(2)/(3.q0*x))*ERF(tau(2)*x)
  GX3=-(4.q0*alpha(3)/(3.q0*x))*ERF(tau(3)*x)
  VG1=GX1+GX2+GX3
  x1=(x*beta)**2.q0
  R1=NR1*NR2
  R2=(x**(l1+l2+2))*exp(-x1)*laguerre(l1,j1,x1)*
&      laguerre(l2,j2,x1)
  G1=R1*R2*VG1
  RETURN
  END

  REAL*16 FUNCTION G2(x)
  REAL*16 PI,NR1N,NR2N,NR1D,NR2D,NR1,NR2,E1,E2,GX1,GX2,GX3,
& VG2,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,alpha(3),tau(3),
& bb,cc,sigma12,x,x1,GAM0,GAM1,GAM2,GAM3
  EXTERNAL laguerre
  COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
  COMMON /gnums/alpha,tau,bb,cc,sigma12
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.88622692545275801364908374167 1q0
  GAM1=1.32934038817913702047362561251q0
  GAM2=3.32335097044784255118406403126q0
```

```
      GAM3=11.631728396567448929144224109 4q0
      NR1N=sqrt(2.q0*(beta**(2.q0*l1+3.q0)))
      NR2N=sqrt(2.q0*(beta**(2.q0*l2+3.q0)))
      if (l1.eq.0) then
         NR1D=sqrt(GAM0*binom0(j1))
      elseif (l1.eq.1) then
         NR1D=sqrt(GAM1*binom1(j1))
      elseif (l1.eq.2) then
         NR1D=sqrt(GAM2*binom2(j1))
      elseif (l1.eq.3) then
         NR1D=sqrt(GAM3*binom3(j1))
      endif
      if (l2.eq.0) then
         NR2D=sqrt(GAM0*binom0(j2))
      elseif (l2.eq.1) then
         NR2D=sqrt(GAM1*binom1(j2))
      elseif (l2.eq.2) then
         NR2D=sqrt(GAM2*binom2(j2))
      elseif (l2.eq.3) then
         NR2D=sqrt(GAM3*binom3(j2))
      endif
      NR1=NR1N/NR1D
      NR2=NR2N/NR2D
      GX1=(4.q0*alpha(1)/(3.q0*x**3.q0))*ERF(tau(1)*x)-
     &      (8.q0*tau(1)*alpha(1)/(3.q0*sqrt(PI)*x**2.q0))*
     &      exp(-(tau(1)*x)**2.q0)
      GX2=(4.q0*alpha(2)/(3.q0*x**3.q0))*ERF(tau(2)*x)-
     &      (8.q0*tau(2)*alpha(2)/(3.q0*sqrt(PI)*x**2.q0))*
     &      exp(-(tau(2)*x)**2.q0)
      GX3=(4.q0*alpha(3)/(3.q0*x**3.q0))*ERF(tau(3)*x)-
     &      (8.q0*tau(3)*alpha(3)/(3.q0*sqrt(PI)*x**2.q0))*
     &      exp(-(tau(3)*x)**2.q0)
      VG2=(1/m1sq)*(GX1+GX2+GX3)
      x1=(x*beta)**2.q0
      R1=NR1*NR2
      R2=(x**(l1+l2+2))*exp(-x1)*laguerre(l1,j1,x1)*
     &      laguerre(l2,j2,x1)
       G2=R1*R2*VG2
      RETURN
      END

      REAL*16 FUNCTION G5(x)
      REAL*16 PI,NR1N,NR2N,NR1D,NR2D,NR1,NR2,E1,E2,GX1,GX2,GX3,
     & VG5,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,alpha(3),tau(3),
```

```
& bb,cc,sigma12,x,x1,GAM0,GAM1,GAM2,GAM3
  EXTERNAL laguerre
  COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
  COMMON /gnums/alpha,tau,bb,cc,sigma12
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.88622692545275801364908374167 1q0
  GAM1=1.32934038817913702047362561251q0
  GAM2=3.32335097044784255118406403126q0
  GAM3=11.63172839656744892914422410 94q0
  NR1N=sqrt(2.q0*(beta**(2.q0*l1+3.q0)))
  NR2N=sqrt(2.q0*(beta**(2.q0*l2+3.q0)))
  if (l1.eq.0) then
     NR1D=sqrt(GAM0*binom0(j1))
  elseif (l1.eq.1) then
     NR1D=sqrt(GAM1*binom1(j1))
  elseif (l1.eq.2) then
     NR1D=sqrt(GAM2*binom2(j1))
  elseif (l1.eq.3) then
     NR1D=sqrt(GAM3*binom3(j1))
  endif
  if (l2.eq.0) then
     NR2D=sqrt(GAM0*binom0(j2))
  elseif (l2.eq.1) then
     NR2D=sqrt(GAM1*binom1(j2))
  elseif (l2.eq.2) then
     NR2D=sqrt(GAM2*binom2(j2))
  elseif (l2.eq.3) then
     NR2D=sqrt(GAM3*binom3(j2))
  endif
  NR1=NR1N/NR1D
  NR2=NR2N/NR2D
  GX1=-(4.q0*alpha(1)/(x**3.q0))*ERF(tau(1)*x)+
&      (16.q0*alpha(1)/(3.q0*sqrt(PI)))*(tau(1)**3.q0+
&      (3.q0*tau(1))/(2.q0*x**2.q0))*exp(-(tau(1)*x)**2.q0)
  GX2=-(4.q0*alpha(2)/(x**3.q0))*ERF(tau(2)*x)+
&      (16.q0*alpha(2)/(3.q0*sqrt(PI)))*(tau(2)**3.q0+
&      (3.q0*tau(2))/(2.q0*x**2.q0))*exp(-(tau(2)*x)**2.q0)
  GX3=-(4.q0*alpha(3)/(x**3.q0))*ERF(tau(3)*x)+
&      (16.q0*alpha(3)/(3.q0*sqrt(PI)))*(tau(3)**3.q0+
&      (3.q0*tau(3))/(2.q0*x**2.q0))*exp(-(tau(3)*x)**2.q0)
  VG5=(1.q0/(m1*m2))*(GX1+GX2+GX3)
```

```
  x1=(x*beta)**2.q0
  R1=NR1*NR2
  R2=(x**(l1+l2+2))*exp(-x1)*laguerre(l1,j1,x1)*
&      laguerre(l2,j2,x1)
  G5=R1*R2*VG5
  RETURN
  END

  REAL*16 FUNCTION G6(x)
  REAL*16 PI,NR1N,NR2N,NR1D,NR2D,NR1,NR2,E1,E2,GX1,GX2,GX3,
& VG6,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,alpha(3),tau(3),
& bb,cc,sigma12,x,x1,GAM0,GAM1,GAM2,GAM3
  EXTERNAL laguerre
  COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
  COMMON /gnums/alpha,tau,bb,cc,sigma12
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.88622692545275801364908374167 1q0
  GAM1=1.32934038817913702047362561251q0
  GAM2=3.32335097044784255118406403126q0
  GAM3=11.63172839656744892914422410 94q0
  NR1N=sqrt(2.q0*(beta**(2.q0*l1+3.q0)))
  NR2N=sqrt(2.q0*(beta**(2.q0*l2+3.q0)))
  if (l1.eq.0) then
     NR1D=sqrt(GAM0*binom0(j1))
  elseif (l1.eq.1) then
     NR1D=sqrt(GAM1*binom1(j1))
  elseif (l1.eq.2) then
     NR1D=sqrt(GAM2*binom2(j1))
  elseif (l1.eq.3) then
     NR1D=sqrt(GAM3*binom3(j1))
  endif
  if (l2.eq.0) then
     NR2D=sqrt(GAM0*binom0(j2))
  elseif (l2.eq.1) then
     NR2D=sqrt(GAM1*binom1(j2))
  elseif (l2.eq.2) then
     NR2D=sqrt(GAM2*binom2(j2))
  elseif (l2.eq.3) then
     NR2D=sqrt(GAM3*binom3(j2))
  endif
  NR1=NR1N/NR1D
```

```
  NR2=NR2N/NR2D
  VG6=bb*x*(exp(-(sigma12*x)**2.q0)/(sqrt(PI)*sigma12*x)+
&     (1.q0+1.q0/(2.q0*(sigma12*x)**2.q0))*erf(sigma12*x))+cc
  x1=(x*beta)**2.q0
  R1=NR1*NR2
  R2=(x**(l1+l2+2))*exp(-x1)*laguerre(l1,j1,x1)*
&    laguerre(l2,j2,x1)
  G6=R1*R2*VG6
  RETURN
  END

  REAL*16 FUNCTION G7(x)
  REAL*16 PI,NR1N,NR2N,NR1D,NR2D,NR1,NR2,E1,E2,GX1,GX2,GX3,
& VG7,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,alpha(3),tau(3),
& bb,cc,sigma12,x,x1,GAM0,GAM1,GAM2,GAM3
  EXTERNAL laguerre
  COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
  COMMON /gnums/alpha,tau,bb,cc,sigma12
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.88622692545275801364908374167 1q0
  GAM1=1.32934038817913702047362561251q0
  GAM2=3.32335097044784255118406403126q0
  GAM3=11.63172839656744892914422410 94q0
  NR1N=sqrt(2.q0*(beta**(2.q0*l1+3.q0)))
  NR2N=sqrt(2.q0*(beta**(2.q0*l2+3.q0)))
  if (l1.eq.0) then
     NR1D=sqrt(GAM0*binom0(j1))
  elseif (l1.eq.1) then
     NR1D=sqrt(GAM1*binom1(j1))
  elseif (l1.eq.2) then
     NR1D=sqrt(GAM2*binom2(j1))
  elseif (l1.eq.3) then
     NR1D=sqrt(GAM3*binom3(j1))
  endif
  if (l2.eq.0) then
     NR2D=sqrt(GAM0*binom0(j2))
  elseif (l2.eq.1) then
     NR2D=sqrt(GAM1*binom1(j2))
  elseif (l2.eq.2) then
     NR2D=sqrt(GAM2*binom2(j2))
  elseif (l2.eq.3) then
```

```
     NR2D=sqrt(GAM3*binom3(j2))
  endif
  NR1=NR1N/NR1D
  NR2=NR2N/NR2D
  GX1=(1.q0/x-1.q0/(2.q0*(sigma12**2.q0)*x**3.q0))*
&       erf(sigma12*x)
  GX2=(exp(-(sigma12*x)**2.q0)/sqrt(PI))*(1.q0/(sigma12*x))
  VG7=(1/m1sq)*bb*(GX1+GX2)
  x1=(x*beta)**2.q0
  R1=NR1*NR2
  R2=(x**(l1+l2+2))*exp(-x1)*laguerre(l1,j1,x1)*
&      laguerre(l2,j2,x1)
  G7=R1*R2*VG7
  RETURN
  END

  REAL*16 FUNCTION G10(x)
  REAL*16 PI,NR1N,NR2N,NR1D,NR2D,NR1,NR2,E1,E2,GX1,GX2,GX3,
& VG10,R1,R2,laguerre,beta,m1sq,m2sq,m1,m2,alpha(3),tau(3),
& bb,cc,sigma12,x,x1,GAM0,GAM1,GAM2,GAM3
  EXTERNAL laguerre
  COMMON /nums/l1,l2,j1,j2,PI,beta,m1sq,m2sq,m1,m2
  COMMON /gnums/alpha,tau,bb,cc,sigma12
  COMMON /coeff0/binom0(100)
  COMMON /coeff1/binom1(100)
  COMMON /coeff2/binom2(100)
  COMMON /coeff3/binom3(100)
  GAM0=0.88622692545275801364908374167 1q0
  GAM1=1.32934038817913702047362561251q0
  GAM2=3.32335097044784255118406403126q0
  GAM3=11.63172839656744892914422410 94q0
  NR1N=sqrt(2.q0*(beta**(2.q0*l1+3.q0)))
  NR2N=sqrt(2.q0*(beta**(2.q0*l2+3.q0)))
  if (l1.eq.0) then
     NR1D=sqrt(GAM0*binom0(j1))
  elseif (l1.eq.1) then
     NR1D=sqrt(GAM1*binom1(j1))
  elseif (l1.eq.2) then
     NR1D=sqrt(GAM2*binom2(j1))
  elseif (l1.eq.3) then
     NR1D=sqrt(GAM3*binom3(j1))
  endif
  if (l2.eq.0) then
     NR2D=sqrt(GAM0*binom0(j2))
```

```
      elseif (l2.eq.1) then
         NR2D=sqrt(GAM1*binom1(j2))
      elseif (l2.eq.2) then
         NR2D=sqrt(GAM2*binom2(j2))
      elseif (l2.eq.3) then
         NR2D=sqrt(GAM3*binom3(j2))
      endif
      NR1=NR1N/NR1D
      NR2=NR2N/NR2D
      GX1=alpha(1)*(tau(1)**3.q0)*EXP(-(tau(1)*x)**2.q0)
      GX2=alpha(2)*(tau(2)**3.q0)*EXP(-(tau(2)*x)**2.q0)
      GX3=alpha(3)*(tau(3)**3.q0)*EXP(-(tau(3)*x)**2.q0)
      VG10=32.q0*(GX1+GX2+GX3)/(9.q0*sqrt(PI)*m1*m2)
      x1=(x*beta)**2.q0
      R1=NR1*NR2
      R2=(x**(l1+l2+2))*exp(-x1)*laguerre(l1,j1,x1)*
     &      laguerre(l2,j2,x1)
      G10=R1*R2*VG10
      RETURN
      END

      REAL*16 FUNCTION laguerre(ll,jj,x)
      COMMON /coeff0/binom0(100)
      COMMON /coeff1/binom1(100)
      COMMON /coeff2/binom2(100)
      COMMON /coeff3/binom3(100)
      REAL*16 x,am,bm,cm
      integer jj,ll,M
      am = 1.q0
      do 16 M=(jj-1),1,-1
         bm=1.q0*(jj-M)
     cm=1.q0*M*(M+ll+0.5q0)
         am=1.q0-(bm/cm)*(x*am)
16    continue
      if (ll.eq.0) then
      laguerre=am*binom0(jj)
      elseif (ll.eq.1) then
      laguerre=am*binom1(jj)
      elseif (ll.eq.2) then
      laguerre=am*binom2(jj)
      elseif (ll.eq.3) then
      laguerre=am*binom3(jj)
      endif
      RETURN
```

```
 END

  BLOCK DATA binomial0
  COMMON /coeff0/binom0(100)
  DATA binom0/1.q0,1.5q0,1.875q0,2.1875q0,2.4609375q0,
& 2.70703125q0,2.9326171875q0,3.14208984375q0,
& 3.338470458984375q0,3.5239410400390625q0,
& 3.7001380920410156 25q0,3.8683261871337890625q0,
& 4.0295064449310302734375q0,4.1844874620437622070312 5q0,
& 4.3339334428310394287109375q0,4.4783978909254074096679 6875q0,
& 4.6183478250168263912200927734375q0,
& 4.7541815845761448144912719726 5625q0,
& 4.8862421841477043926715850830078125q0,
& 5.0148275047831702977418899536132812q0,
& 5.1401981924027495551854372024536132q0,
& 5.2625838636504340684041380882263183q0,
& 5.3821880423697621154133230447769165q0,
& 5.4991921302473656396614387631416320q0,
& 5.6137586329608524238210520707070827q0,
& 5.7260338056200694722974731121212244q0,
& 5.8361498403435323467647322104312479q0,
& 5.9442266892387829457788939180318266q0,
& 6.0503735944037612126678027379966806q0,
& 6.1546903805141708887482820955483475q0,
& 6.2572685535227404035607534638074866q0,
& 6.3581922398698813778117333583850268q0,
& 6.4575389936178482743400416921097928q0,
& 6.5553804935211490057694362632023654q0,
& 6.6517831478376364911483985611906355q0,
& 6.7468086213781741553076613977790732q0,
& 6.8405142966750932407980455838593381q0,
& 6.9329536790625945008088299836412210q0,
& 7.0241767537871023231878935360575529q0,
& 7.1142303019125779939979947352377779q0,
& 7.2031581806864852189229696694282501q0,
& 7.2910015731338813801293473483237166q0,
& 7.3777992109092847298927919596132847q0,
& 7.463587573826834552333405819608788q0,
& 7.5484010689839576722462854311952515q0,
& 7.6322721919726683130490219359863098q0,
& 7.7152316723201973164517286961600740q0,
& 7.7973086050044547347118534695234791q0,
& 7.8785305696399178048651019431643487q0,
& 7.9589237387178761498127050242170461q0,
```

```
& 8.038512976105054911310832074459216 6q0,
& 8.117321926851182900637408859502934 4q0,
& 8.195373099224751966989691636998154 9q0,
& 8.272687939783476042149971746781156 4q0,
& 8.349286902188878598095804818510611 5q0,
& 8.425189510390595676260312135042526 2q0,
& 8.500414416733368851941207779105405 9q0,
& 8.574979455476644017309113110501067 4q0,
& 8.648901692161787500216950292488145 5q0,
& 8.722197469214006038354382074627875 6q0,
& 8.794882448124122755340668591916441 2q0,
& 8.866971648518582777925428170538707 1q0,
& 8.938479484393732639037730010623696 7q0,
& 9.009419797761936866331680248803249 8q0,
& 9.079805889931951998099896500747025 2q0,
& 9.149650550623736244239126473829694 6q0,
& 9.218966085098158488513665310752646 9q0,
& 9.287764339464562656338394156355278 6q0,
& 9.356056724313566793517352936916714 4q0,
& 9.423854236808592639847188827763937 0q0,
& 9.491167481357225444417525890819393 7q0,
& 9.558006688972417172899339453430797 9q0,
& 9.624381735423614514377807088524067 3q0,
& 9.690302158268981737079024945294780 1q0,
& 9.755777172851880262329558897627852 9q0,
& 9.820815687337559464078422623612038 6q0,
& 9.885426316859517092131570140872644 1q0,
& 9.949617396839124345976580336592596 4q0,
& 10.013396995536811040502071236186139q0,
& 10.076772925888309844555881813630355q0,
& 10.139752756675111781084356074965544q0,
& 10.202343823074340866152778026045579q0,
& 10.264553236629672212897612038399515q0,
& 10.326387894681658190083742231401922q0,
& 10.387854489292858536453288316112648q0,
& 10.448959515700463586667719423854487q0,
& 10.509709280326629072636717792597827q0,
& 10.570109908374483262709342607497814q0,
& 10.630167351035701917611100235949506q0,
& 10.689887392333655299170713158623380q0,
& 10.749275655624397828610550453949065q0,
& 10.808337609776180234262256775124610q0,
& 10.867078575046702952926725561945939q0,
& 10.925503728675986302136009032709089q0,
```

```
& 10.983618110211496867572902697776691q0,
& 11.041426626581031061612760080396568q0,
& 11.098934056927807265058659872481967q0,
& 11.156145057221249570548652964608163q0,
& 11.213064164656051864275942010345959q0,
& 11.269695801851284449449052828580030q0/
  END

  BLOCK DATA binomial1
  COMMON /coeff1/binom1(100)
  DATA binom1/1.q0,2.5q0,4.375q0,6.5625q0,9.0234375q0,
& 11.73046875q0,
& 14.6630859375q0,17.80517578125q0,21.143646240234375q0,
& 24.6675872802734375q0,28.367725372314453125q0,
& 32.236051559448242187 5q0,36.2655580043792724609375q0,
& 40.450045466423034667968 75q0,44.7839789092540740966796875q0,
& 49.26237680017948150634765625q0,
& 53.880724625196307897567749023 4375q0,
& 58.634906209772452712059020996 09375q0,
& 63.521148393920157104730606079101562q0,
& 68.535975898703327402472496032714843q0,
& 73.676174091106076957657933235168457q0,
& 78.938757954756511026062071323394775q0,
& 84.320945997126273141475394368171691q0,
& 89.820138127373638781136833131313324q0,
& 95.433896760334491204957885202020406q0,
& 101.15993056595456067725535831414163q0,
& 106.99608040629809302402009052457288q0,
& 112.94030709553687596979898444260470q0,
& 118.99068068994063718246678718060138q0,
& 125.14537107045480807121506927614973q0,
& 131.40263962397754847477582273995722q0,
& 137.76083186384742985258755609834224q0,
& 144.21837085746527812692759779045204q0,
& 150.77375135098642713269703405365440q0,
& 157.42553449882406362384543261484504q0,
& 164.17234312020223777915309401262411q0,
& 171.01285741687733101995113959648345q0,
& 177.94581109593992552075996958012467q0,
& 184.96998784972702784394786311618222q0,
& 192.08421815163960583794585785142000q0,
& 199.28737633232609105686882752084825q0,
& 206.57837790545997243699817486917197q0,
& 213.95617711636925716689096682878525q0,
```

```
& 221.41976469019609171922437264839404q0,
& 228.96816575918004939147065807958929q0,
& 236.60043795115271770451968001557560q0,
& 244.31566962347291502097140871173568q0,
& 252.11297822847736975568326218125915q0,
& 259.99150879811728756054836412442350q0,
& 267.95043253683516371036106914864055q0,
& 275.98894551294021862167190122309977q0,
& 284.10626743979140152230931008260270q0,
& 292.30164053901615348929900171960086q0,
& 300.57432847879962953144897346638202q0,
& 308.92361538098850812954477828489263q0,
& 317.34880489137910380580509041993515q0,
& 325.84921930811247265774629819904056q0,
& 334.42419876358911667505541130954163q0,
& 343.07310045575090417527236160202977q0,
& 351.79529792496491021362674367665765q0,
& 360.59018037308903296896741226857409q0,
& 369.45715202160761574689284043911280q0,
& 378.39563150600134838593057044973649q0,
& 387.40505130376328525226225069853974q0,
& 396.48485719369523725036214719928677q0,
& 405.63450774431897349460127367311646q0,
& 414.85347382941713198311493898386911q0,
& 424.14123816888169463945333314022439q0,
& 433.49729489319526143297068607714110q0,
& 442.92114913000385407281787490490504q0,
& 452.41231661136107951723540079572443q0,
& 461.97032330033349669013474024915523q0,
& 471.59470503575711120451254733767930q0,
& 481.28500719402609294159157228297408q0,
& 491.04078436687797320392113118060193q0,
& 500.86160005421553266799955380421397q0,
& 510.74702637107504976013112394508661q0,
& 520.69664376791417410610770428167921q0,
& 530.71004076345098514660977551786535q0,
& 540.78681368933929499116565733149570q0,
& 550.92656644601440677225001340646125q0,
& 561.12891026908874763840279143250683q0,
& 571.39346350571841985130040347090634q0,
& 581.71985140040007804138414570230826q0,
& 592.10770588969293657783743401842091q0,
& 602.55666540539340016450515344227540q0,
& 613.06637468572002923714187123487322q0,
```

```
& 623.6364845940945124998512138423710 4q0,
& 634.2666519451302144174623140783205 5q0,
& 644.9565393374638697166330272369439 3q0,
& 655.7058149930882675452435776908929 9q0,
& 666.5141526028644477795058344660176 0q0,
& 677.3812311779111507324325600279635 4q0,
& 688.3067349065871370345685690606726 3q0,
& 699.2903530167986339021414717584493 2q0,
& 710.3317796433796649637542318388458 9q0,
& 721.4307137003074722288128917113278 6q0,
& 732.5868587575287217993615446759360 2q0,
& 743.7999229221847736636374866862819 8q0,
& 755.0696187240360581130865395148620 1q0/
  END

  BLOCK DATA binomial2
  COMMON /coeff2/binom2(100)
  DATA binom2/1.q0,3.5q0,7.875q0,14.4375q0,
& 23.4609375q0,35.19140625q0,
& 49.8544921875q0,67.65966796875q0,88.803314208984375q0,
& 113.4709014892578125q0,141.838626861572265625q0,
& 174.0746784210205078125q0,210.34023642539978027343750q0,
& 250.7902818918228149414062500q0,295.5742608010768890380859375q0,
& 344.8366376012563705444335937500q0,
& 398.7173622264526784420013427734375q0,
& 457.3522684362251311540603637695312500q0,
& 520.8734168301452882587909698486328100q0,
& 589.4093927288486156612634658813476500q0,
& 663.0855668199546926189213991165161100q0,
& 742.0243247747112036449834704399108800q0,
& 826.3452707718374767864588648080825800q0,
& 916.1654088992111155675956979393959000q0,
& 1011.599305659545606772553583141416300q0,
& 1112.759236225500167449808941455557900q0,
& 1219.755316631798260473829031980130800q0,
& 1332.695623727335136443628016422735500q0,
& 1451.686304417275773626094803603336900q0,
& 1576.831675487730581697309872879486600q0,
& 1708.234315111708130172085695619443800q0,
& 1845.995146975555560024673251717786100q0,
& 1990.213517833020838151600849508238100q0,
& 2140.987269184007265284297883561892500q0,
& 2298.412803682831328908143316176737600q0,
& 2462.585146803033566687296410189361700q0,
```

```
& 2633.59800421991089770724754978584 51q0,
& 2811.54381531585082322800751936596 98q0,
& 2996.51380316557785107195538248215 20q0,
& 3188.59802131721745690990124033357 20q0,
& 3387.88539764954354796677006785442 03q0,
& 3594.46377555500352040376824272359 23q0,
& 3808.41995267137277757065920955237 75q0,
& 4029.83971736156886928988358220077 16q0,
& 4258.80788312074891868135424028036 08q0,
& 4495.40832107190163638587392029593 65q0,
& 4739.72399069537455140684532900767 21q0,
& 4991.83696892385192116252859118893 13q0,
& 5251.82847772196920872307695531335 48q0,
& 5519.77891025880437243343802446199 53q0,
& 5795.76785577174459105510992568509 52q0,
& 6079.87412321153599257741923576769 79q0,
& 6372.17576375055214606671823748729 87q0,
& 6672.75009222935177559816721095368 07q0,
& 6981.67370761034028372771198923857 34q0,
& 7299.02251250171938753351707965850 85q0,
& 7624.87173180983186019126337785754 91q0,
& 7959.29593057342097686631878916709 07q0,
& 8302.36903102917188104159115076912 05q0,
& 8654.16432895413679125521789444577 81q0,
& 9014.75450932722582422418530671435 22q0,
& 9384.21166134883343997107814715346 50q0,
& 9762.60729285483478835700871760320 15q0,
& 10150.01234415859807360927096830174 1q0,
& 10546.49720135229331085963311550102 8q0,
& 10952.13170909661228435423438917414 4q0,
& 11366.98518292602941633734932815801 4q0,
& 11791.12642109491111097680266129823 8q0,
& 12224.62371598810637240977334737537 9q0,
& 12667.54486511811022648259122228028 4q0,
& 13119.95718172947130599982662307600 9q0,
& 13581.92750502980480268999613633251 64q0,
& 14053.52221006556191389447391066284 3q0,
& 14534.80721725958800683606548294581 7q0,
& 15025.84800162646598003998661412641 9q0,
& 15526.70960168068151270798616793063 3q0,
& 16037.45662805175656246811729187572 0q0,
& 16558.15327181967073657422499615739 9q0,
& 17088.86331258312172172083477167526 4q0,
& 17629.65012627246101671200042900676 0q0,
```

```
& 18180.576692718475423484250442413221q0,
& 18741.705602987564171122653233845728q0,
& 19313.099066493282590973953637316634q0,
& 19894.818917893682669015337783018943q0,
& 20486.926623783375605593175217037364q0,
& 21089.483289188769005757680370479639q0,
& 21702.549663874489034994822241714512q0,
& 22326.186148468583547494673455556883q0,
& 22960.452800413713761912135769635204q0,
& 23605.409339751177631628768796872148q0,
& 24261.115154744265899174012374563041q0,
& 24927.629307347130346953518209029058q0,
& 25605.010538525041497685950769057022q0,
& 26293.317273431628634720519338117694q0,
& 26992.607626448427268622660809876144q0,
& 27702.939406091806933586415041714990q0,
& 28424.370119792114405815227933426317q0,
& 29156.956978549643127614589478102254q0,
& 29900.756901471827901278226964788535q0,
& 30655.826520195863959391313504303397q0/
  END

  BLOCK DATA binomial3
  COMMON /coeff3/binom3(100)
  DATA binom3/1.q0,4.5q0,12.3750q0,26.8125q0,
& 50.2734375q0,85.46484375q0,
& 135.3193359375q0,202.97900390625q0, 291.782318115234375q0,
& 405.2532196044921875q0,547.091846466064453125q0,
& 721.16652488708496093750q0, 931.5067613124847412109375q0,
& 1182.29704320430755615234375q0,
& 1477.87130400538444519042968750q0,
& 1822.70794160664081573486328125q0,
& 2221.42530383309349417686462402343750q0,
& 2678.77757226931862533092498779296875q0,
& 3199.65098909946391358971595764160156q0,
& 3789.06038182831252925097942352294921q0,
& 4452.14594864826722186990082263946533q0,
& 5194.17027342297842551488429307937622q0,
& 6020.51554419481590230134315788745880q0,
& 6936.68095309402701786893885582685470q0,
& 7948.28025875357262464149243896827101q0,
& 9061.03949497907279209130138042382907q0,
& 10280.79481161087105256513041240396023q0,
& 11613.49043533820618900875842882669544q0,
```

```
& 13065.176739755481962634853232430032q0,
& 14642.008415243212544332163105309519q0,
& 16350.242730354920674504248800928963q0,
& 18196.237877330476234528922052646749q0,
& 20186.451395163497072680522902154987q0,
& 22327.438664347504337964820785716880q0,
& 24625.851468030335666872964101893617q0,
& 27088.436614833369233560260512082979q0,
& 29722.034619053280131267508061868824q0,
& 32533.578434369130954495515581234794q0,
& 35530.092237534708805567470963716946q0,
& 38718.690258851926262477372204050518q0,
& 42106.575656501469810444142271904938q0,
& 45701.039432056473330847910514628531q0,
& 49509.459384727846108418569724180908q0,
& 53539.299102089414977708453306381680q0,
& 57798.106985210163896389807546662041q0,
& 62293.515306282065532775681466957977q0,
& 67033.239296977440084182526795965649q0,
& 72025.076265901292005345055387154581q0,
& 77276.904743623261214068132342467936q0,
& 82796.683653882065586501570366929931q0,
& 88592.451509653810177556680292615026q0,
& 94672.325632865346170134099528382724q0,
& 101044.501396615898316200817765870002q0,
& 107717.251488845250091798984976823 70q0,
& 114698.925196455590375526696966062 28q0,
& 121997.947708957309763060214045720 79q0,
& 129622.819440767141623251477423578 33q0,
& 137582.115371340562600117796212745 43q0,
& 145884.484402369734481159387363514 55q0,
& 154538.648731323871272414605257960 32q0,
& 163553.403240651097096638790564674 68q0,
& 172937.614901999930536609868711828 14q0,
& 182700.222194854765324966877429431 34q0,
& 192850.234539013363398576148397733 08q0,
& 203396.731740365656709435781513234 11q0,
& 214348.863449462268993790015902408 26q0,
& 225715.848632388298410127365230566 27q0,
& 237506.975053483209521104167891864 51q0,
& 249731.598769471315893513941239239 89q0,
& 262399.143634589426119996532461520 17q0,
& 275519.100816318897425996359084596 18q0,
& 289101.028321348702228686320447921 34q0,
```

```
& 303154.55053141426414258079435858419q0,
& 317689.35774867385214941685984153000q0,
& 332715.20575030031812945684645565642q0,
& 348241.91535198099964216483262358706q0,
& 364279.37198003275620463294991546278q0,
& 380837.52525185242694120717491162017q0,
& 397926.38856443554866292800968329544q0,
& 415556.03869070800967964001011230220q0,
& 433736.61538342648510312426055471542q0,
& 452478.32098641404927424691378856115q0,
& 471791.42005290733186522086742587778q0,
& 491686.23897080101453423620520889672q0,
& 512173.16559458439013982938042593409q0,
& 533262.64888377315914558706079641373q0,
& 554965.19854764764818058188303812824q0,
& 577291.38469611623172807655649368512q0,
& 600251.83749652994548998869226332033q0,
& 623857.24683628112312161746106019247q0,
& 648118.36199102538902079147343475551q0,
& 673045.99129837251936774499164378457q0,
& 698651.00183689756086543094241284159q0,
& 724944.31911032918950015146175095929q0,
& 751936.92673677761676877412256083543q0,
& 779639.86614286942370236053760255042q0,
& 808064.23626266153810817576553597674q0,
& 837221.19324121118123579035501407899q0,
& 867121.95014268300913706858197886753q0,
& 897777.77666287887309645989548317093/
  END
```

Appendix 4

Code to Diagonalize Hamiltonian

```
      PROGRAM diag
      INTEGER i,j,rows,cols,grows,gcols,frmt,LWORK,IFAIL,smax
      PARAMETER (rows=25,cols=40,grows=40,gcols=40,smax=5)
      PARAMETER (LWORK=64*rows,IFAIL=0)
      DOUBLE PRECISION WORK(LWORK)
      DOUBLE PRECISION ff1(rows,cols),ff2(rows,cols),ff3(rows,cols),
     &ff4(rows,cols),ff5(rows,cols),ff7(rows,cols),ff8(rows,cols),
     &ff9(rows,cols),ff10(rows,cols),gg1(grows,gcols),gg2(grows,gcols)
     &,gg3(grows,gcols),gg4(grows,gcols),gg5(grows,gcols),
     &gg6(grows,gcols),gg7(grows,gcols),gg8(grows,gcols),
     &gg10(grows,gcols),
     &fg1(rows,gcols),fg2(rows,gcols),fg3(rows,gcols),
     &fg4(rows,gcols),fg5(rows,gcols),fg7(rows,gcols),fg8(rows,gcols),
     &fg10(rows,gcols),fgf1(rows,rows),fgf2(rows,rows),
     &fgf3(rows,rows),fgf4(rows,rows),fgf5(rows,rows),
     &fgf7(rows,rows),fgf8(rows,rows),fgf10(rows,rows)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc                   L=0 Variable Declarations                      cc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION oneS0(rows,rows),threeS1(rows,rows)
      DOUBLE PRECISION oneS0evalsR(rows),threeS1evalsR(rows),
     &oneS0evectsR(rows,rows),threeS1evectsR(rows,rows)
      DOUBLE PRECISION oneS0evalsI(rows),threeS1evalsI(rows),
     &oneS0evectsI(rows,rows),threeS1evectsI(rows,rows)
      INTEGER IRANK1S0(rows),IRANK3S1(rows)
      DOUBLE PRECISION DUMMY1S0(rows),DUMMY3S1(rows)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
cc                    L=1 Variable Declarations                     cc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c      DOUBLE PRECISION oneP1(rows,rows),threeP1(rows,rows),
c     &threeP0(rows,rows),threeP2(rows,rows)
c      DOUBLE PRECISION oneP1evalsR(rows),threeP0evalsR(rows),
c     &threeP1evalsR(rows),threeP2evalsR(rows),
c     &oneP1evectsR(rows,rows),threeP0evectsR(rows,rows),
c     &threeP1evectsR(rows,rows),threeP2evectsR(rows,rows)
c      DOUBLE PRECISION oneP1evalsI(rows),threeP0evalsI(rows),
c     &threeP1evalsI(rows),threeP2evalsI(rows),
c     &oneP1evectsI(rows,rows),threeP0evectsI(rows,rows),
c     &threeP1evectsI(rows,rows),threeP2evectsI(rows,rows)
c      INTEGER IRANK1P1(rows),IRANK3P0(rows),IRANK3P1(rows),
c     &IRANK3P2(rows),a,b
c      DOUBLE PRECISION DUMMY1P1(rows),
c     &DUMMY3P0(rows),DUMMY3P1(rows),DUMMY3P2(rows)
c      DOUBLE PRECISION AA,MIXevalsR(2*smax),MIXevalsI(2*smax),
c     &MIXevectsR(2*smax,2*smax),MIXevectsI(2*smax,2*smax),
c     &MIX(2*smax,2*smax)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc                     L=2 Variable Declarations                    cc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c      DOUBLE PRECISION oneD2(rows,rows),threeD1(rows,rows),
c     &threeD2(rows,rows),threeD3(rows,rows)
c      DOUBLE PRECISION oneD2evalsR(rows),threeD1evalsR(rows),
c     &threeD2evalsR(rows),threeD3evalsR(rows),
c     &oneD2evectsR(rows,rows),threeD1evectsR(rows,rows),
c     &threeD2evectsR(rows,rows),threeD3evectsR(rows,rows)
c      DOUBLE PRECISION oneD2evalsI(rows),threeD1evalsI(rows),
c     &threeD2evalsI(rows),threeD3evalsI(rows),
c     &oneD2evectsI(rows,rows),threeD1evectsI(rows,rows),
c     &threeD2evectsI(rows,rows),threeD3evectsI(rows,rows)
c      INTEGER IRANK1D2(rows),IRANK3D1(rows),IRANK3D2(rows),
c     &IRANK3D3(rows),a,b
c      DOUBLE PRECISION DUMMY1D2(rows),DUMMY3D1(rows),
c     &DUMMY3D2(rows),DUMMY3D3(rows)
c      DOUBLE PRECISION AA,MIXevalsR(2*smax),MIXevalsI(2*smax),
c     &MIXevectsR(2*smax,2*smax),MIXevectsI(2*smax,2*smax),
c     &MIX(2*smax,2*smax)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
cc                      L=3 Variable Declarations                      cc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c      DOUBLE PRECISION oneF3(rows,rows),threeF2(rows,rows),
c     &threeF3(rows,rows),threeF4(rows,rows)
c      DOUBLE PRECISION oneF3evalsR(rows),threeF2evalsR(rows),
c     &threeF3evalsR(rows),threeF4evalsR(rows),
c     &oneF3evectsR(rows,rows),threeF2evectsR(rows,rows),
c     &threeF3evectsR(rows,rows),threeF4evectsR(rows,rows)
c      DOUBLE PRECISION oneF3evalsI(rows),threeF2evalsI(rows),
c     &threeF3evalsI(rows),threeF4evalsI(rows),
c     &oneF3evectsI(rows,rows),threeF2evectsI(rows,rows),
c     &threeF3evectsI(rows,rows),threeF4evectsI(rows,rows)
c      INTEGER IRANK1F3(rows),IRANK3F2(rows),IRANK3F3(rows),
c     &IRANK3F4(rows),a,b
c      DOUBLE PRECISION DUMMY1F3(rows),DUMMY3F2(rows),
c     &DUMMY3F3(rows),DUMMY3F4(rows)
c      DOUBLE PRECISION AA,MIXevalsR(2*smax),MIXevalsI(2*smax),
c     &MIXevectsR(2*smax,2*smax),MIXevectsI(2*smax,2*smax),
c     &MIX(2*smax,2*smax)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION m1,m2,m1sq,m2sq
      m1=0.22d0
      m2=0.22d0
      m1sq=m1**2.d0
      m2sq=m2**2.d0
      OPEN(201, file='ff1.col', STATUS='old')
      OPEN(202, file='ff2.col', STATUS='old')
      OPEN(203, file='ff3.col', STATUS='old')
      OPEN(204, file='ff4.col', STATUS='old')
      OPEN(205, file='ff5.col', STATUS='old')
      OPEN(206, file='ff7.col', STATUS='old')
      OPEN(207, file='ff8.col', STATUS='old')
      OPEN(208, file='ff9.col', STATUS='old')
      OPEN(214, file='ff10.col',STATUS='old')
      OPEN(209, file='gg1.col', STATUS='old')
      OPEN(210, file='gg2.col', STATUS='old')
      OPEN(211, file='gg5.col', STATUS='old')
      OPEN(212, file='gg6.col', STATUS='old')
      OPEN(213, file='gg7.col', STATUS='old')
      OPEN(215, file='gg10.col',STATUS='old')
```

```
      do 10 i=1,rows
         do 11 j=1,cols
            READ(201,*)ff1(i,j)
    READ(202,*)ff2(i,j)
    READ(203,*)ff3(i,j)
    READ(204,*)ff4(i,j)
    READ(205,*)ff5(i,j)
    READ(206,*)ff7(i,j)
    READ(207,*)ff8(i,j)
    READ(208,*)ff9(i,j)
    READ(214,*)ff10(i,j)
11       continue
10    continue
      do 12 i=1,grows
         do 13 j=1,gcols
    READ(209,*)gg1(i,j)
    READ(210,*)gg2(i,j)
    gg3(i,j)=(m1sq/m2sq)*gg2(i,j)
    gg4(i,j)=(m1sq/(m1*m2))*gg2(i,j)
    READ(211,*)gg5(i,j)
    READ(212,*)gg6(i,j)
    READ(213,*)gg7(i,j)
    gg8(i,j)=(m1sq/m2sq)*gg7(i,j)
    READ(215,*)gg10(i,j)
13       continue
12    continue
      CLOSE(201)
      CLOSE(202)
      CLOSE(203)
      CLOSE(204)
      CLOSE(205)
      CLOSE(206)
      CLOSE(207)
      CLOSE(208)
      CLOSE(214)
      CLOSE(209)
      CLOSE(210)
      CLOSE(211)
      CLOSE(212)
      CLOSE(213)
      CLOSE(215)

      CALL DGEMUL(ff1,rows,'N',gg1,grows,'N',fg1,rows,rows,cols,gcols)
      CALL DGEMUL(ff2,rows,'N',gg2,grows,'N',fg2,rows,rows,cols,gcols)
```

```
      CALL DGEMUL(ff3,rows,'N',gg3,grows,'N',fg3,rows,rows,cols,gcols)
      CALL DGEMUL(ff4,rows,'N',gg4,grows,'N',fg4,rows,rows,cols,gcols)
      CALL DGEMUL(ff5,rows,'N',gg5,grows,'N',fg5,rows,rows,cols,gcols)
      CALL DGEMUL(ff7,rows,'N',gg7,grows,'N',fg7,rows,rows,cols,gcols)
      CALL DGEMUL(ff8,rows,'N',gg8,grows,'N',fg8,rows,rows,cols,gcols)
      CALL DGEMUL(ff10,rows,'N',gg10,grows,'N',fg10,rows,rows,cols,
     &gcols)
c
      CALL DGEMUL(fg1,rows,'N',ff1,rows,'T',fgf1,rows,rows,cols,rows)
      CALL DGEMUL(fg2,rows,'N',ff2,rows,'T',fgf2,rows,rows,cols,rows)
      CALL DGEMUL(fg3,rows,'N',ff3,rows,'T',fgf3,rows,rows,cols,rows)
      CALL DGEMUL(fg4,rows,'N',ff4,rows,'T',fgf4,rows,rows,cols,rows)
      CALL DGEMUL(fg5,rows,'N',ff5,rows,'T',fgf5,rows,rows,cols,rows)
      CALL DGEMUL(fg7,rows,'N',ff7,rows,'T',fgf7,rows,rows,cols,rows)
      CALL DGEMUL(fg8,rows,'N',ff8,rows,'T',fgf8,rows,rows,cols,rows)
      CALL DGEMUL(fg10,rows,'N',ff10,rows,'T',fgf10,rows,rows,cols,
     &rows)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccc                                                                 ccc
ccc                              L=0                                ccc
ccc                                                                 ccc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      do 15 i=1,rows
         do 16 j=1,rows
 oneS0(i,j)=ff9(i,j)+fgf1(i,j)+gg6(i,j)-
     &              (3.d0/4.d0)*fgf10(i,j)
         threeS1(i,j)=ff9(i,j)+fgf1(i,j)+gg6(i,j)+
     &                 (1.d0/4.d0)*fgf10(i,j)
16       continue
15    continue
      OPEN(301, file='1S0.matrix', STATUS='unknown')
      OPEN(302, file='3S1.matrix', STATUS='unknown')
      OPEN(305, file='fgf1.matrix', STATUS='unknown')
      OPEN(306, file='fgf2.matrix', STATUS='unknown')
      OPEN(307, file='fgf3.matrix', STATUS='unknown')
      OPEN(308, file='fgf4.matrix', STATUS='unknown')
      OPEN(309, file='fgf5.matrix', STATUS='unknown')
      OPEN(310, file='fgf7.matrix', STATUS='unknown')
      OPEN(311, file='fgf8.matrix', STATUS='unknown')
      OPEN(312, file='fgf10.matrix',STATUS='unknown')
      do 30 i=1,rows
         do 40 j=1,rows
```

```
    if (j.eq.rows) then
       ASSIGN 255 TO frmt
    else
       ASSIGN 250 TO frmt
    endif
    WRITE(301,frmt)oneS0(i,j)
    WRITE(302,frmt)threeS1(i,j)
    WRITE(305,frmt)fgf1(i,j)
    WRITE(306,frmt)fgf2(i,j)
    WRITE(307,frmt)fgf3(i,j)
    WRITE(308,frmt)fgf4(i,j)
    WRITE(309,frmt)fgf5(i,j)
    WRITE(310,frmt)fgf7(i,j)
    WRITE(311,frmt)fgf8(i,j)
    WRITE(312,frmt)fgf10(i,j)
40       continue
30    continue
250   FORMAT(G27.18E2,$)
255   FORMAT(G27.18E2)
      CLOSE(301)
      CLOSE(302)
      CLOSE(305)
      CLOSE(306)
      CLOSE(307)
      CLOSE(308)
      CLOSE(309)
      CLOSE(310)
      CLOSE(311)
      CLOSE(312)

      CALL F02EBF('V',rows,oneS0,rows,oneS0evalsR,oneS0evalsI,
     &oneS0evectsR,rows,oneS0evectsI,rows,WORK,LWORK,IFAIL)
      CALL F02EBF('V',rows,threeS1,rows,threeS1evalsR,
     &threeS1evalsI,threeS1evectsR,rows,threeS1evectsI,rows,
     &WORK,LWORK,IFAIL)

      CALL M01DAF(oneS0evalsR,1,rows,'A',IRANK1S0,IFAIL)
      CALL M01DAF(threeS1evalsR,1,rows,'A',IRANK3S1,IFAIL)

      CALL M01EAF(oneS0evalsR,1,rows,IRANK1S0,IFAIL)
      CALL M01EAF(threeS1evalsR,1,rows,IRANK3S1,IFAIL)

      do 35 i=1,rows
         do 36 j=1,rows
```

```
    DUMMY1S0(j)=oneS0evectsR(i,j)
    DUMMY3S1(j)=threeS1evectsR(i,j)
36       continue
         CALL M01EAF(DUMMY1S0,1,rows,IRANK1S0,IFAIL)
 CALL M01EAF(DUMMY3S1,1,rows,IRANK3S1,IFAIL)
 do 37 j=1,rows
 oneS0evectsR(i,j)=DUMMY1S0(j)
 threeS1evectsR(i,j)=DUMMY3S1(j)
37       continue
35    continue

      OPEN(401, file='1S0.evals', STATUS='unknown')
      OPEN(402, file='1S0.evects', STATUS='unknown')
      OPEN(403, file='3S1.evals', STATUS='unknown')
      OPEN(404, file='3S1.evects', STATUS='unknown')
      do 50 i=1,rows
         WRITE(401,*)oneS0evalsR(i)
 WRITE(403,*)threeS1evalsR(i)
       do 55 j=1,rows
    if (j.eq.rows) then
       ASSIGN 255 TO frmt
    else
       ASSIGN 250 TO frmt
    endif
       WRITE(402,frmt)oneS0evectsR(i,j)
       WRITE(404,frmt)threeS1evectsR(i,j)
55       continue
50    continue
      CLOSE(401)
      CLOSE(402)
      CLOSE(403)
      CLOSE(404)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccc                                                                   ccc
ccc                              L=1                                  ccc
ccc                                                                   ccc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c      do 15 i=1,rows
c         do 16 j=1,rows
c  oneP1(i,j)=ff9(i,j)+fgf1(i,j)+gg6(i,j)-
c     &              (3.q0/4.q0)*fgf10(i,j)
c         threeP0(i,j)=ff9(i,j)+fgf1(i,j)-(1.d0/2.d0)*fgf2(i,j)-
```

```
c     &                (1.d0/2.d0)*fgf3(i,j)-(2.d0)*fgf4(i,j)+
c     &                (1.d0/3.d0)*fgf5(i,j)+gg6(i,j)+
c     &                (1.d0/2.d0)*fgf7(i,j)+(1.d0/2.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c         threeP1(i,j)=ff9(i,j)+fgf1(i,j)-(1.d0/4.d0)*fgf2(i,j)-
c     &                (1.d0/4.d0)*fgf3(i,j)-fgf4(i,j)-
c     &                (1.d0/6.d0)*fgf5(i,j)+gg6(i,j)+
c     &                (1.d0/4.d0)*fgf7(i,j)+(1.d0/4.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c         threeP2(i,j)=ff9(i,j)+fgf1(i,j)+(1.d0/4.d0)*fgf2(i,j)+
c     &                (1.d0/4.d0)*fgf3(i,j)+fgf4(i,j)+
c     &                (1.d0/30.d0)*fgf5(i,j)+gg6(i,j)-
c     &                (1.d0/4.d0)*fgf7(i,j)-(1.d0/4.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c16       continue
c15    continue
c      OPEN(301, file='1P1.mat', STATUS='unknown')
c      OPEN(302, file='3P0.mat', STATUS='unknown')
c      OPEN(303, file='3P1.mat', STATUS='unknown')
c      OPEN(304, file='3P2.mat', STATUS='unknown')
c      OPEN(305, file='fgf1.mat', STATUS='unknown')
c      OPEN(306, file='fgf2.mat', STATUS='unknown')
c      OPEN(307, file='fgf3.mat', STATUS='unknown')
c      OPEN(308, file='fgf4.mat', STATUS='unknown')
c      OPEN(309, file='fgf5.mat', STATUS='unknown')
c      OPEN(310, file='fgf7.mat', STATUS='unknown')
c      OPEN(311, file='fgf8.mat', STATUS='unknown')
c      OPEN(312, file='fgf10.mat',STATUS='unknown')
c      do 30 i=1,rows
c         do 31 j=1,rows
c     if (j.eq.rows) then
c        ASSIGN 255 TO frmt
c     else
c        ASSIGN 250 TO frmt
c     endif
c     WRITE(301,frmt)oneP1(i,j)
c     WRITE(302,frmt)threeP0(i,j)
c     WRITE(303,frmt)threeP1(i,j)
c     WRITE(304,frmt)threeP2(i,j)
c     WRITE(305,frmt)fgf1(i,j)
c     WRITE(306,frmt)fgf2(i,j)
c     WRITE(307,frmt)fgf3(i,j)
c     WRITE(308,frmt)fgf4(i,j)
c     WRITE(309,frmt)fgf5(i,j)
```

```
c     WRITE(310,frmt)fgf7(i,j)
c     WRITE(311,frmt)fgf8(i,j)
c     WRITE(312,frmt)fgf10(i,j)
c31       continue
c30    continue
c250   FORMAT(G27.18E2,$)
c255   FORMAT(G27.18E2)
c      CLOSE(301)
c      CLOSE(302)
c      CLOSE(303)
c      CLOSE(304)
c      CLOSE(305)
c      CLOSE(306)
c      CLOSE(307)
c      CLOSE(308)
c      CLOSE(309)
c      CLOSE(310)
c      CLOSE(311)
c      CLOSE(312)
c      CALL F02EBF('V',rows,oneP1,rows,oneP1evalsR,oneP1evalsI,
c     &oneP1evectsR,rows,oneP1evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeP0,rows,threeP0evalsR,threeP0evalsI,
c     &threeP0evectsR,rows,threeP0evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeP1,rows,threeP1evalsR,threeP1evalsI,
c     &threeP1evectsR,rows,threeP1evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeP2,rows,threeP2evalsR,threeP2evalsI,
c     &threeP2evectsR,rows,threeP2evectsI,rows,WORK,LWORK,IFAIL)
c
c      CALL M01DAF(oneP1evalsR,1,rows,'A',IRANK1P1,IFAIL)
c      CALL M01DAF(threeP0evalsR,1,rows,'A',IRANK3P0,IFAIL)
c      CALL M01DAF(threeP1evalsR,1,rows,'A',IRANK3P1,IFAIL)
c      CALL M01DAF(threeP2evalsR,1,rows,'A',IRANK3P2,IFAIL)
c
c      CALL M01EAF(oneP1evalsR,1,rows,IRANK1P1,IFAIL)
c      CALL M01EAF(threeP0evalsR,1,rows,IRANK3P0,IFAIL)
c      CALL M01EAF(threeP1evalsR,1,rows,IRANK3P1,IFAIL)
c      CALL M01EAF(threeP2evalsR,1,rows,IRANK3P2,IFAIL)
c
c      do 35 i=1,rows
c         do 36 j=1,rows
c     DUMMY1P1(j)=oneP1evectsR(i,j)
c     DUMMY3P0(j)=threeP0evectsR(i,j)
c     DUMMY3P1(j)=threeP1evectsR(i,j)
c     DUMMY3P2(j)=threeP2evectsR(i,j)
```

```
c36       continue
c         CALL M01EAF(DUMMY1P1,1,rows,IRANK1P1,IFAIL)
c  CALL M01EAF(DUMMY3P0,1,rows,IRANK3P0,IFAIL)
c  CALL M01EAF(DUMMY3P1,1,rows,IRANK3P1,IFAIL)
c  CALL M01EAF(DUMMY3P2,1,rows,IRANK3P2,IFAIL)
c  do 37 j=1,rows
c  oneP1evectsR(i,j)=DUMMY1P1(j)
c  threeP0evectsR(i,j)=DUMMY3P0(j)
c  threeP1evectsR(i,j)=DUMMY3P1(j)
c  threeP2evectsR(i,j)=DUMMY3P2(j)
c37       continue
c35    continue
c
c
c      OPEN(401, file='1P1.evals', STATUS='unknown')
c      OPEN(402, file='1P1.evects', STATUS='unknown')
c      OPEN(403, file='3P0.evals', STATUS='unknown')
c      OPEN(404, file='3P0.evects', STATUS='unknown')
c      OPEN(405, file='3P1.evals', STATUS='unknown')
c      OPEN(406, file='3P1.evects', STATUS='unknown')
c      OPEN(407, file='3P2.evals', STATUS='unknown')
c      OPEN(408, file='3P2.evects', STATUS='unknown')
c      do 50 i=1,rows
c         WRITE(401,*)oneP1evalsR(i)
c  WRITE(403,*)threeP0evalsR(i)
c  WRITE(405,*)threeP1evalsR(i)
c  WRITE(407,*)threeP2evalsR(i)
c         do 55 j=1,rows
c     if (j.eq.rows) then
c        ASSIGN 255 TO frmt
c     else
c        ASSIGN 250 TO frmt
c     endif
c        WRITE(402,frmt)oneP1evectsR(i,j)
c        WRITE(404,frmt)threeP0evectsR(i,j)
c        WRITE(406,frmt)threeP1evectsR(i,j)
c        WRITE(408,frmt)threeP2evectsR(i,j)
c55       continue
c50    continue
c      CLOSE(401)
c      CLOSE(402)
c      CLOSE(403)
c      CLOSE(404)
c      CLOSE(405)
```

```
c      CLOSE(406)
c      CLOSE(407)
c      CLOSE(408)
c
c      do 60 a=1,smax
c         do 61 b=1,smax
c     if (a.eq.b) then
c     MIX(a,b)=oneP1evalsR(a)
c     MIX(a+smax,b+smax)=threeP1evalsR(a)
c     else
c     MIX(a,b)=0.d0
c     MIX(a+smax,b+smax)=0.d0
c     endif
c61       continue
c60    continue
c
c      do 63 a=1,smax
c      do 64 b=1,smax
c            AA=0.d0
c     do 65 i=1,rows
c     do 66 j=1,rows
c        AA=AA+oneP1evectsR(i,a)*threeP1evectsR(j,b)*
c     &             (fgf2(i,j)-fgf3(i,j)-fgf7(i,j)+fgf8(i,j))
c66          continue
c65          continue
c         MIX(a,b+smax)=AA/(2.d0*sqrt(2.d0))
c       MIX(a+smax,b)=AA/(2.d0*sqrt(2.d0))
c64    continue
c63    continue
c
c      OPEN(450, file='MIX.mat',status='unknown')
c      do 80 i=1,2*smax
c         do 81 j=1,2*smax
c     if (j.eq.2*smax) then
c        ASSIGN 255 to frmt
c     else
c        ASSIGN 250 to frmt
c     endif
c     WRITE(450,frmt)MIX(i,j)
c81       continue
c80    continue
c      CLOSE(450)
c
c      CALL F02EBF('V',2*smax,MIX,2*smax,MIXevalsR,MIXevalsI,
```

```
c     &MIXevectsR,2*smax,MIXevectsI,2*smax,WORK,LWORK,IFAIL)
c
c      OPEN(460,file='MIX.evals',status='unknown')
c      OPEN(470,file='MIX.evects',status='unknown')
c      do 90 i=1,2*smax
c         WRITE(460,*)MIXevalsR(i)
c         do 91 j=1,2*smax
c     if (j.eq.2*smax) then
c        ASSIGN 255 to frmt
c     else
c        ASSIGN 250 to frmt
c     endif
c     WRITE(470,frmt)MIXevectsR(i,j)
c91       continue
c90    continue
c      CLOSE(460)
c      CLOSE(470)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccc                                                                   ccc
ccc                                L=2                                ccc
ccc                                                                   ccc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c      do 15 i=1,rows
c         do 16 j=1,rows
c  oneD2(i,j)=ff9(i,j)+fgf1(i,j)+gg6(i,j)-(3.q0/4.q0)*fgf10(i,j)
c  threeD1(i,j)=ff9(i,j)+fgf1(i,j)-(3.d0/4.d0)*fgf2(i,j)-
c     &                (3.d0/4.d0)*fgf3(i,j)-(3.d0)*fgf4(i,j)+
c     &                (1.d0/6.d0)*fgf5(i,j)+gg6(i,j)+
c     &                (3.d0/4.d0)*fgf7(i,j)+(3.d0/4.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c         threeD2(i,j)=ff9(i,j)+fgf1(i,j)-(1.d0/4.d0)*fgf2(i,j)-
c     &                (1.d0/4.d0)*fgf3(i,j)-(1.d0)*fgf4(i,j)-
c     &                (1.d0/6.d0)*fgf5(i,j)+gg6(i,j)+
c     &                (1.d0/4.d0)*fgf7(i,j)+(1.d0/4.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c         threeD3(i,j)=ff9(i,j)+fgf1(i,j)+(1.d0/2.d0)*fgf2(i,j)+
c     &                (1.d0/2.d0)*fgf3(i,j)+(2.d0)*fgf4(i,j)+
c     &                (1.d0/21.d0)*fgf5(i,j)+gg6(i,j)-
c     &                (1.d0/2.d0)*fgf7(i,j)-(1.d0/2.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c16       continue
c15    continue
```

```
c      OPEN(301, file='1D2.mat', STATUS='unknown')
c      OPEN(302, file='3D1.mat', STATUS='unknown')
c      OPEN(303, file='3D2.mat', STATUS='unknown')
c      OPEN(304, file='3D3.mat', STATUS='unknown')
c      OPEN(305, file='fgf1.mat', STATUS='unknown')
c      OPEN(306, file='fgf2.mat', STATUS='unknown')
c      OPEN(307, file='fgf3.mat', STATUS='unknown')
c      OPEN(308, file='fgf4.mat', STATUS='unknown')
c      OPEN(309, file='fgf5.mat', STATUS='unknown')
c      OPEN(310, file='fgf7.mat', STATUS='unknown')
c      OPEN(311, file='fgf8.mat', STATUS='unknown')
c      OPEN(312, file='fgf10.mat',STATUS='unknown')
c      do 30 i=1,rows
c         do 40 j=1,rows
c     if (j.eq.rows) then
c        ASSIGN 255 TO frmt
c     else
c        ASSIGN 250 TO frmt
c     endif
c     WRITE(301,frmt)oneD2(i,j)
c     WRITE(302,frmt)threeD1(i,j)
c     WRITE(303,frmt)threeD2(i,j)
c     WRITE(304,frmt)threeD3(i,j)
c     WRITE(305,frmt)fgf1(i,j)
c     WRITE(306,frmt)fgf2(i,j)
c     WRITE(307,frmt)fgf3(i,j)
c     WRITE(308,frmt)fgf4(i,j)
c     WRITE(309,frmt)fgf5(i,j)
c     WRITE(310,frmt)fgf7(i,j)
c     WRITE(311,frmt)fgf8(i,j)
c     WRITE(312,frmt)fgf10(i,j)
c40       continue
c30    continue
c250   FORMAT(G27.18E2,$)
c255   FORMAT(G27.18E2)
c      CLOSE(301)
c      CLOSE(302)
c      CLOSE(303)
c      CLOSE(304)
c      CLOSE(305)
c      CLOSE(306)
c      CLOSE(307)
c      CLOSE(308)
c      CLOSE(309)
```

```
c      CLOSE(310)
c      CLOSE(311)
c      CLOSE(312)
c      CALL F02EBF('V',rows,oneD2,rows,oneD2evalsR,oneD2evalsI,
c     &oneD2evectsR,rows,oneD2evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeD1,rows,threeD1evalsR,threeD1evalsI,
c     &threeD1evectsR,rows,threeD1evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeD2,rows,threeD2evalsR,threeD2evalsI,
c     &threeD2evectsR,rows,threeD2evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeD3,rows,threeD3evalsR,threeD3evalsI,
c     &threeD3evectsR,rows,threeD3evectsI,rows,WORK,LWORK,IFAIL)
c
c      CALL M01DAF(oneD2evalsR,1,rows,'A',IRANK1D2,IFAIL)
c      CALL M01DAF(threeD1evalsR,1,rows,'A',IRANK3D1,IFAIL)
c      CALL M01DAF(threeD2evalsR,1,rows,'A',IRANK3D2,IFAIL)
c      CALL M01DAF(threeD3evalsR,1,rows,'A',IRANK3D3,IFAIL)
c
c      CALL M01EAF(oneD2evalsR,1,rows,IRANK1D2,IFAIL)
c      CALL M01EAF(threeD1evalsR,1,rows,IRANK3D1,IFAIL)
c      CALL M01EAF(threeD2evalsR,1,rows,IRANK3D2,IFAIL)
c      CALL M01EAF(threeD3evalsR,1,rows,IRANK3D3,IFAIL)
c
c      do 35 i=1,rows
c         do 36 j=1,rows
c     DUMMY1D2(j)=oneD2evectsR(i,j)
c     DUMMY3D1(j)=threeD1evectsR(i,j)
c     DUMMY3D2(j)=threeD2evectsR(i,j)
c     DUMMY3D3(j)=threeD3evectsR(i,j)
c36       continue
c         CALL M01EAF(DUMMY1D2,1,rows,IRANK1D2,IFAIL)
c  CALL M01EAF(DUMMY3D1,1,rows,IRANK3D1,IFAIL)
c  CALL M01EAF(DUMMY3D2,1,rows,IRANK3D2,IFAIL)
c  CALL M01EAF(DUMMY3D3,1,rows,IRANK3D3,IFAIL)
c  do 37 j=1,rows
c  oneD2evectsR(i,j)=DUMMY1D2(j)
c  threeD1evectsR(i,j)=DUMMY3D1(j)
c  threeD2evectsR(i,j)=DUMMY3D2(j)
c  threeD3evectsR(i,j)=DUMMY3D3(j)
c37       continue
c35    continue
c
c
c      OPEN(401, file='1D2.evals', STATUS='unknown')
c      OPEN(402, file='1D2.evects', STATUS='unknown')
```

```
c      OPEN(403, file='3D1.evals', STATUS='unknown')
c      OPEN(404, file='3D1.evects', STATUS='unknown')
c      OPEN(405, file='3D2.evals', STATUS='unknown')
c      OPEN(406, file='3D2.evects', STATUS='unknown')
c      OPEN(407, file='3D3.evals', STATUS='unknown')
c      OPEN(408, file='3D3.evects', STATUS='unknown')
c      do 50 i=1,rows
c         WRITE(401,*)oneD2evalsR(i)
c  WRITE(403,*)threeD1evalsR(i)
c  WRITE(405,*)threeD2evalsR(i)
c  WRITE(407,*)threeD3evalsR(i)
c         do 55 j=1,rows
c     if (j.eq.rows) then
c        ASSIGN 255 TO frmt
c     else
c        ASSIGN 250 TO frmt
c     endif
c        WRITE(402,frmt)oneD2evectsR(i,j)
c        WRITE(404,frmt)threeD1evectsR(i,j)
c        WRITE(406,frmt)threeD2evectsR(i,j)
c        WRITE(408,frmt)threeD3evectsR(i,j)
c55       continue
c50    continue
c      CLOSE(401)
c      CLOSE(402)
c      CLOSE(403)
c      CLOSE(404)
c      CLOSE(405)
c      CLOSE(406)
c      CLOSE(407)
c      CLOSE(408)
c
c      do 60 a=1,smax
c         do 61 b=1,smax
c     if (a.eq.b) then
c     MIX(a,b)=oneD2evalsR(a)
c     MIX(a+smax,b+smax)=threeD2evalsR(a)
c     else
c     MIX(a,b)=0.d0
c     MIX(a+smax,b+smax)=0.d0
c     endif
c61       continue
c60    continue
c
```

```
c      do 63 a=1,smax
c      do 64 b=1,smax
c            AA=0.d0
c     do 65 i=1,rows
c     do 66 j=1,rows
c        AA=AA+oneD2evectsR(i,a)*threeD2evectsR(j,b)*
c     &             (fgf2(i,j)-fgf3(i,j)-fgf7(i,j)+fgf8(i,j))
c66          continue
c65          continue
c         MIX(a,b+smax)=sqrt(3.d0/2.d0)*AA/2.d0
c  MIX(a+smax,b)=sqrt(3.d0/2.d0)*AA/2.d0
c64    continue
c63    continue
c
c      OPEN(450, file='MIX.mat',status='unknown')
c      do 80 i=1,2*smax
c         do 81 j=1,2*smax
c     if (j.eq.2*smax) then
c        ASSIGN 255 to frmt
c     else
c        ASSIGN 250 to frmt
c     endif
c     WRITE(450,frmt)MIX(i,j)
c81       continue
c80    continue
c      CLOSE(450)
c
c      CALL F02EBF('V',2*smax,MIX,2*smax,MIXevalsR,MIXevalsI,
c     &MIXevectsR,2*smax,MIXevectsI,2*smax,WORK,LWORK,IFAIL)
c
c      OPEN(460,file='MIX.evals',status='unknown')
c      OPEN(470,file='MIX.evects',status='unknown')
c      do 90 i=1,2*smax
c         WRITE(460,*)MIXevalsR(i)
c         do 91 j=1,2*smax
c     if (j.eq.2*smax) then
c        ASSIGN 255 to frmt
c     else
c        ASSIGN 250 to frmt
c     endif
c     WRITE(470,frmt)MIXevectsR(i,j)
c91       continue
c90    continue
c      CLOSE(460)
```

```
c      CLOSE(470)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccc                                                                   ccc
ccc                              L=3                                  ccc
ccc                                                                   ccc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c        do 15 i=1,rows
c         do 16 j=1,gcols
c  oneF3(i,j)=ff9(i,j)+fgf1(i,j)+gg6(i,j)-(3.q0/4.q0)*fgf10(i,j)
c  threeF2(i,j)=ff9(i,j)+fgf1(i,j)-fgf2(i,j)-
c     &                fgf3(i,j)-4.d0*fgf4(i,j)-
c     &                (2.d0/15.d0)*fgf5(i,j)+gg6(i,j)+
c     &                fgf7(i,j)+fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c         threeF3(i,j)=ff9(i,j)+fgf1(i,j)-(1.d0/4.d0)*fgf2(i,j)-
c     &                (1.d0/4.d0)*fgf3(i,j)-fgf4(i,j)-
c     &                (1.d0/6.d0)*fgf5(i,j)+gg6(i,j)+
c     &                (1.d0/4.d0)*fgf7(i,j)+(1.d0/4.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c         threeF4(i,j)=ff9(i,j)+fgf1(i,j)+(3.d0/4.d0)*fgf2(i,j)+
c     &                (3.d0/4.d0)*fgf3(i,j)+(3.0d0)*fgf4(i,j)+
c     &                (1.d0/18.d0)*fgf5(i,j)+gg6(i,j)-
c     &                (3.d0/4.d0)*fgf7(i,j)-(3.d0/4.d0)*fgf8(i,j)
c     &                +(1.d0/4.d0)*fgf10(i,j)
c16       continue
c15    continue
c      OPEN(301, file='1F3.mat', STATUS='unknown')
c      OPEN(302, file='3F2.mat', STATUS='unknown')
c      OPEN(303, file='3F3.mat', STATUS='unknown')
c      OPEN(304, file='3F4.mat', STATUS='unknown')
c      OPEN(305, file='fgf1.mat', STATUS='unknown')
c      OPEN(306, file='fgf2.mat', STATUS='unknown')
c      OPEN(307, file='fgf3.mat', STATUS='unknown')
c      OPEN(308, file='fgf4.mat', STATUS='unknown')
c      OPEN(309, file='fgf5.mat', STATUS='unknown')
c      OPEN(310, file='fgf7.mat', STATUS='unknown')
c      OPEN(311, file='fgf8.mat', STATUS='unknown')
c      OPEN(312, file='fgf10.mat',STATUS='unknown')
c      do 30 i=1,rows
c         do 31 j=1,rows
c     if (j.eq.rows) then
c        ASSIGN 255 TO frmt
```

```
c     else
c        ASSIGN 250 TO frmt
c     endif
c     WRITE(301,frmt)oneF3(i,j)
c     WRITE(302,frmt)threeF2(i,j)
c     WRITE(303,frmt)threeF3(i,j)
c     WRITE(304,frmt)threeF4(i,j)
c     WRITE(305,frmt)fgf1(i,j)
c     WRITE(306,frmt)fgf2(i,j)
c     WRITE(307,frmt)fgf3(i,j)
c     WRITE(308,frmt)fgf4(i,j)
c     WRITE(309,frmt)fgf5(i,j)
c     WRITE(310,frmt)fgf7(i,j)
c     WRITE(311,frmt)fgf8(i,j)
c     WRITE(312,frmt)fgf10(i,j)
c31       continue
c30    continue
c250   FORMAT(G27.18E2,$)
c255   FORMAT(G27.18E2)
c      CLOSE(301)
c      CLOSE(302)
c      CLOSE(303)
c      CLOSE(304)
c      CLOSE(305)
c      CLOSE(306)
c      CLOSE(307)
c      CLOSE(308)
c      CLOSE(309)
c      CLOSE(310)
c      CLOSE(311)
c      CLOSE(312)
c      CALL F02EBF('V',rows,oneF3,rows,oneF3evalsR,oneF3evalsI,
c     &oneF3evectsR,rows,oneF3evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeF2,rows,threeF2evalsR,threeF2evalsI,
c     &threeF2evectsR,rows,threeF2evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeF3,rows,threeF3evalsR,threeF3evalsI,
c     &threeF3evectsR,rows,threeF3evectsI,rows,WORK,LWORK,IFAIL)
c      CALL F02EBF('V',rows,threeF4,rows,threeF4evalsR,threeF4evalsI,
c     &threeF4evectsR,rows,threeF4evectsI,rows,WORK,LWORK,IFAIL)
c
c      CALL M01DAF(oneF3evalsR,1,rows,'A',IRANK1F3,IFAIL)
c      CALL M01DAF(threeF2evalsR,1,rows,'A',IRANK3F2,IFAIL)
c      CALL M01DAF(threeF3evalsR,1,rows,'A',IRANK3F3,IFAIL)
c      CALL M01DAF(threeF4evalsR,1,rows,'A',IRANK3F4,IFAIL)
```

```
c
c      CALL M01EAF(oneF3evalsR,1,rows,IRANK1F3,IFAIL)
c      CALL M01EAF(threeF2evalsR,1,rows,IRANK3F2,IFAIL)
c      CALL M01EAF(threeF3evalsR,1,rows,IRANK3F3,IFAIL)
c      CALL M01EAF(threeF4evalsR,1,rows,IRANK3F4,IFAIL)
c
c      do 35 i=1,rows
c         do 36 j=1,rows
c     DUMMY1F3(j)=oneF3evectsR(i,j)
c     DUMMY3F2(j)=threeF2evectsR(i,j)
c     DUMMY3F3(j)=threeF3evectsR(i,j)
c     DUMMY3F4(j)=threeF4evectsR(i,j)
c36       continue
c         CALL M01EAF(DUMMY1F3,1,rows,IRANK1F3,IFAIL)
c  CALL M01EAF(DUMMY3F2,1,rows,IRANK3F2,IFAIL)
c  CALL M01EAF(DUMMY3F3,1,rows,IRANK3F3,IFAIL)
c  CALL M01EAF(DUMMY3F4,1,rows,IRANK3F4,IFAIL)
c  do 37 j=1,rows
c  oneF3evectsR(i,j)=DUMMY1F3(j)
c  threeF2evectsR(i,j)=DUMMY3F2(j)
c  threeF3evectsR(i,j)=DUMMY3F3(j)
c  threeF4evectsR(i,j)=DUMMY3F4(j)
c37       continue
c35    continue
c
c
c      OPEN(401, file='1F3.evals', STATUS='unknown')
c      OPEN(402, file='1F3.evects', STATUS='unknown')
c      OPEN(403, file='3F2.evals', STATUS='unknown')
c      OPEN(404, file='3F2.evects', STATUS='unknown')
c      OPEN(405, file='3F3.evals', STATUS='unknown')
c      OPEN(406, file='3F3.evects', STATUS='unknown')
c      OPEN(407, file='3F4.evals', STATUS='unknown')
c      OPEN(408, file='3F4.evects', STATUS='unknown')
c      do 50 i=1,rows
c         WRITE(401,*)oneF3evalsR(i)
c  WRITE(403,*)threeF2evalsR(i)
c  WRITE(405,*)threeF3evalsR(i)
c  WRITE(407,*)threeF4evalsR(i)
c         do 55 j=1,rows
c     if (j.eq.rows) then
c        ASSIGN 255 TO frmt
c     else
c        ASSIGN 250 TO frmt
```

```
c     endif
c        WRITE(402,frmt)oneF3evectsR(i,j)
c        WRITE(404,frmt)threeF2evectsR(i,j)
c        WRITE(406,frmt)threeF3evectsR(i,j)
c        WRITE(408,frmt)threeF4evectsR(i,j)
c55       continue
c50    continue
c      CLOSE(401)
c      CLOSE(402)
c      CLOSE(403)
c      CLOSE(404)
c      CLOSE(405)
c      CLOSE(406)
c      CLOSE(407)
c      CLOSE(408)
c
c      do 60 a=1,smax
c         do 61 b=1,smax
c     if (a.eq.b) then
c     MIX(a,b)=oneF3evalsR(a)
c     MIX(a+smax,b+smax)=threeF3evalsR(a)
c     else
c     MIX(a,b)=0.d0
c     MIX(a+smax,b+smax)=0.d0
c     endif
c61       continue
c60    continue
c
c      do 63 a=1,smax
c      do 64 b=1,smax
c            AA=0.d0
c     do 65 i=1,rows
c     do 66 j=1,rows
c        AA=AA+oneF3evectsR(i,a)*threeF3evectsR(j,b)*
c     &             (fgf2(i,j)-fgf3(i,j)-fgf7(i,j)+fgf8(i,j))
c66          continue
c65          continue
c         MIX(a,b+smax)=sqrt(3.d0)*AA/2.d0
c  MIX(a+smax,b)=sqrt(3.d0)*AA/2.d0
c64    continue
c63    continue
c
c      OPEN(450, file='MIX.mat',status='unknown')
c      do 80 i=1,2*smax
```

```
c         do 81 j=1,2*smax
c     if (j.eq.2*smax) then
c        ASSIGN 255 to frmt
c     else
c        ASSIGN 250 to frmt
c     endif
c     WRITE(450,frmt)MIX(i,j)
c81       continue
c80    continue
c      CLOSE(450)
c
c      CALL F02EBF('V',2*smax,MIX,2*smax,MIXevalsR,MIXevalsI,
c     &MIXevectsR,2*smax,MIXevectsI,2*smax,WORK,LWORK,IFAIL)
c
c      OPEN(460,file='MIX.evals',status='unknown')
c      OPEN(470,file='MIX.evects',status='unknown')
c      do 90 i=1,2*smax
c         WRITE(460,*)MIXevalsR(i)
c         do 91 j=1,2*smax
c     if (j.eq.2*smax) then
c        ASSIGN 255 to frmt
c     else
c        ASSIGN 250 to frmt
c     endif
c     WRITE(470,frmt)MIXevectsR(i,j)
c91       continue
c90    continue
c      CLOSE(460)
c      CLOSE(470)

       END
```