Methods for integrable systems

Moorhouse, Thomas

Abstract

Methods for Integrable Systems

by

Thomas Moorhouse

This work concerns the study of certain methods for investigating integrable systems, and the application of these methods to specific problems and examples.

After introducing the notion of integrability in chapters 1 and 2, we go on, in chapter 3, to develop a novel type of discrete integrable equation by considering ways of enforcing Leibniz's rule for finite difference operators. We look at several approaches to the problem, derive some solutions and study several examples.

Chapter 4 describes a numerical implementation of a method for solving initial value problems for an integrable equation in 2+1 dimensions, exploiting the integrability of the equation. The introduction of twistors enables a powerful scheme to be developed.

In chapter 5 Darboux transformations derived from the factorisation of a scattering problem are examined, and a general operator form considered.

The topic of chapter 6 is the relationship between the Darboux transform for the sine-Gordon and related equations and certain ansatze established by twistor methods.

Finally in chapter 7 a geometric setting for partial differential equations is introduced and used to investigate the structure of Bäcklund transformations and generalised symmetries.

Preface

The material presented in this thesis is based on work carried out by the author at the University of Durham between October 1991 and September 1994. No part of this work has previously been submitted as part of the requirements of any degree, either to the University of Durham or to any other establishment.

With the exception of review material in chapters 1,2 and parts of chapters 7 and it is believed that the bulk of the work is original.

The material in chapter 4 is a development of a joint paper by myself and R.S. Ward which has been accepted for publication in the Journal of Mathematical Physics. Chapter 6 also contains material which has been submitted for publication.

I would like to thank Professor Richard Ward and Dr. David Fairlie for helpful comments and advice. My thanks also go to my wife Pauline for her patience and support, to Nick Myers for some interesting conversations, and to the Harold Hyam Wingate Foundation for assistance with the purchase of books.

The financial support of the EPSRC (formerly SERC) is gratefully acknowledged.

Methods for Integrable Systems

by
Thomas Moorhouse

A thesis submitted in part fulfillment of the
requirements for the degree of
Doctor of Philosophy

Department of Mathematical Sciences
University of Durham
South Road
Durham DH1 5LE

September 1994

Contents

List of Figures

Chapter 1

Introduction

Integrability is a property, or set of properties, possessed by a relatively small proportion of the equations of mathematical science. Its applications span much of modern theoretical physics, including field theory, lattice models and matrix models in string theory. It is also relevant to many physical phenomena, such as wave motion, plasma physics, optics and the dynamics of complex molecules.

Any attempt to define what is meant by an integrable system in full generality must address the range of characteristics occurring in such systems. As yet no comprehensive definition exists: perhaps that goal will never be reached. However, there are certain properties common to a large number of systems which may in some sense be considered the hallmarks of integrability. When these properties are present they often lead to solutions of the system in question.

In chapter 2 we present a short, and by no means comprehensive, review of some of these properties as applied to well known systems. This provides the context for the work to follow, introducing several of the ideas to be developed in later chapters.

Chapter 3 introduces the problem of obtaining integrable discrete versions of differential equations possessing some of the properties reviewed in chapter 2, and the various interesting solutions which arise. A central question addressed in this chapter is whether it is possible to apply Leibniz's rule in some form when the differential operators are replaced by their finite difference forms. We shall demonstrate that this can be answered affirmatively, and this allows us to develop the desired

integrable systems and to find solutions.

In the fourth chapter a numerical procedure for the solution of the initial value problem for an integrable equation in 2+1 dimensions is presented. The basis of the numerical method is the twistor description arising from the existence of a Lax pair, which, as will be related in chapter 4, is a characteristic property of integrable systems. We look at several initial configurations which may be treated in this way, and at some of the problems in the implementation, as well as indicating possible resolutions and developments.

We introduce Darboux transformations in chapter 5 by way of a factorisation method for the scattering problem associated to the Korteweg-de Vries (KdV) equation, and look at a general operator equation which contains the basic ingredients of the method. The algebra of the Darboux transformations is related to a Lie algebra, and we look at possible extensions.

Drawing together elements from earlier chapters we discuss a twistor inspired Darboux transformation for the sine-Gordon equation in chapter 6. A number of properties of integrable equations are linked, and a relationship between Darboux transformations and the construction of monopole ansatze examined.

In chapter 7 we look at a geometric setting for partial differential equations, from which viewpoint we examine the geometric aspects of generalised symmetries for an integrable equation, and of Bäcklund transformations. The basic ideas of jet theory are introduced and used to develop coordinate-free descriptions of the various structures. Some ideas from earlier chapters are reviewed in this context.

Chapter 2

A Brief Review of Integrability.

2.1 Characterisations of Integrability.

2.1.1 An Overview.

We should remark at the outset that there is no single adequate definition of integrability, despite much attention over the past twenty or so years. What is known is that the vast majority of equations uncovered in mathematical science do not enjoy this property, and that those which do have a special role to play in the study of all branches of science. It is perhaps misleading to place too much emphasis on integrable equations, but their role in modern science is nevertheless quite central.

In the next section we will examine some of the accepted definitions of integrability, with the understanding that each one has its limitations and exceptions, and each one its own areas of success. Whilst there appears not to be one definition covering every case of integrability, we will see that several of the ideas will be sufficient to explore many interesting examples and give us tools for exploiting integrability when it occurs in scientific problems.

Some examples of integrable systems will be presented in section 2.2. Whilst little will be said about physical applications of the equations it should be noted that many of them do arise in physical contexts, and provide a rich testing ground for physical theories. A wide range of applications to diverse fields exists, and is well described in the literature (see, for example, [23]).

2.1.2 Definitions of Integrable Systems.

In this section we will be necessarily vague in our 'definitions', simply indicating what features might underlie the idea of an integrable equation or system.

As a first definition, most of the equations accepted as integrable possess special kinds of solution which retain their profiles indefinitely and scatter (or pass through one another) without, asymptotically, changing shape: the only effect of the scattering is a phase shift. Such solutions are often referred to as solitons, and their stability is taken to be a defining property. Figure 1 shows the behaviour of a two-soliton solution to the KdV equation, illustrating the characteristic interaction. Here the large amplitude soliton is moving faster than the small one. During the interaction both solitons are phase shifted: the small one seems to pulled back, the large one is pushed forward, but when the interaction is over both solitons move at the same speeds they had before they met, and have the same shapes.

Secondly, such equations possess an infinite number (without making precise what we mean by 'infinite') of independent conserved quantities. In some cases this may be traced back to the existence of several compatible hamiltonian structures [38]. The conserved quantities are not all derivable from the Lie symmetries of the equation.

A third definition is that an integrable system is one which is solvable through an inverse scattering problem, in which the field variable becomes the potential in a scattering problem (Schrödinger equation). Related to this is a definition which will be important in later chapters, namely the existence of a Lax pair formulation, which will be described in the next section.

The existence of certain transformations, known as Bäcklund transformations, from the given equation to itself or a second, perhaps linear equation, is also used to define integrability, and this idea will be the subject of chapter 7. The importance of these transformations lies in their application to the generation of new solutions from old, and their relationship to the hierarchies of conserved currents mentioned above.

Bäcklund transformations and scattering formulations arise in the application of the method of Wahlquist and Estabrook [54], versions of which are given in

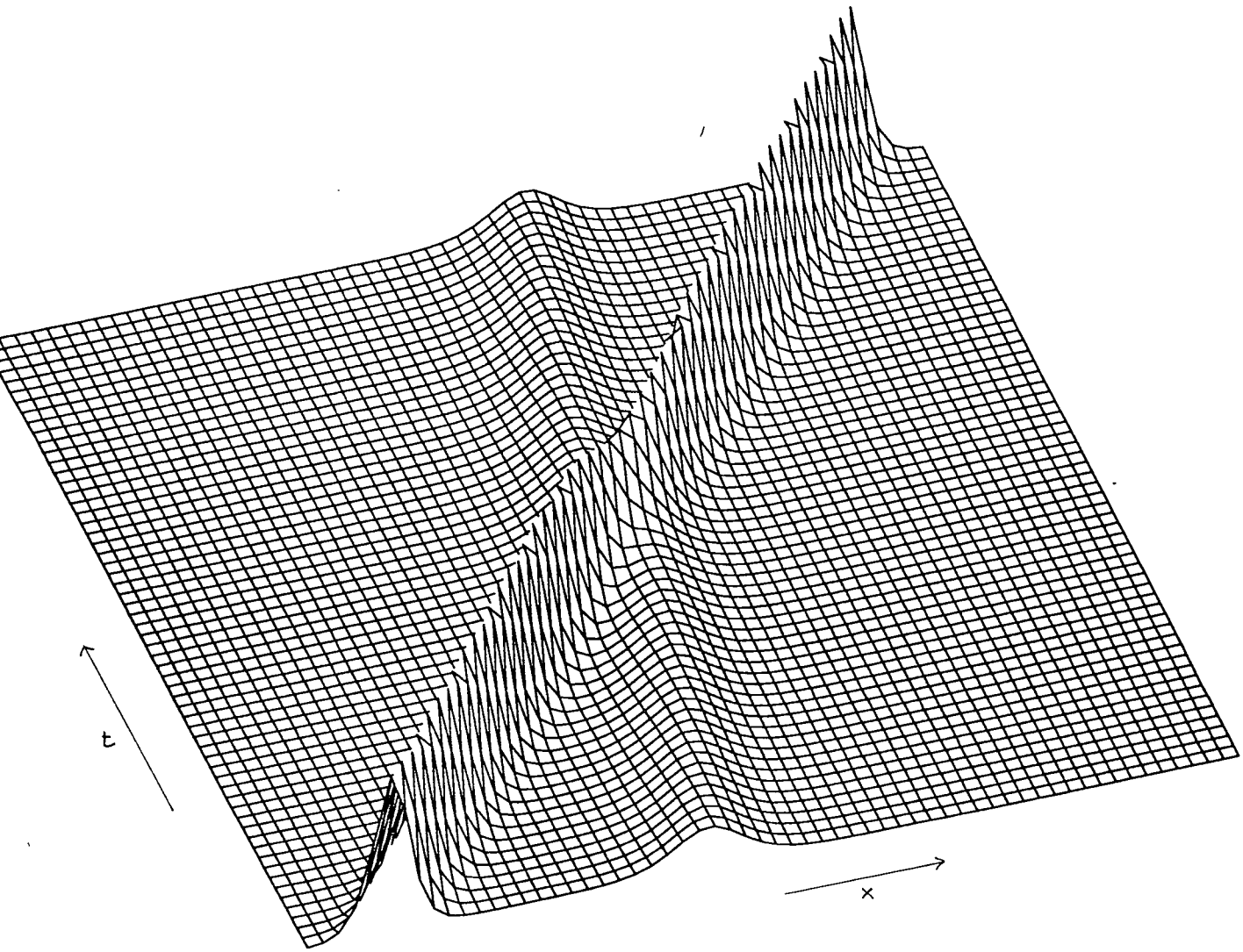

Figure 2.1: The interaction of two KdV solitons.

[16, 12]. The application of this 'prolongation' method leads, in the case of some integrable equations, to a scattering problem and an interesting algebraic structure, but collapses for most non integrable equations. In this sense it is a test for integrability, and yields useful information when successful. The method has been applied to higher dimensional and supersymmetric equations [34].

A further definition involves the so-called Painleve test: in its simplest form (which has been extended to more general situations) this requires that there exist similarity reductions of the equation which are special cases of the Painleve equations. This definition is based on the behaviour of singularities of the equation as initial data are varied, and this fact may be used in generalising the method [31, 61].

A definition of a slightly different character involves the existence of certain bilinear form of the equation due to R. Hirota (see, for example, [14]). The basis of Hirota's method was quite mysterious when first discovered, but is now understood in terms of so-called tau functions. The nature of these tau functions is a recurrent theme in modern studies of integrability.

There are many relationships between the definitions, and it is clear that their must be some underlying structure. Much progress has been made in the search for such structure, but as we noted above, no comprehensive universal connection has been uncovered.

In a different context the idea of integrability may be applied to quantum systems [7], some of which are based on classical integrable equations. One prominent example is the sine-Gordon model, which has been quantised in various ways. The integrability implies a special factorised form for the S matrix of such models, which in some cases may be solved exactly. Although we will have little to say about these ideas they constitute a very active field of research.

2.1.3 Examples.

In the following we use the notation $u_x \equiv \partial u/\partial x$. Some well known integrable systems are evolution equations, such as

the Korteweg-de Vries (KdV) equation,

$$u_t = u_{xxx} + 6uu_x,$$

the Burgers equation,

$$u_t = u_{xx} + u^2,$$

and the nonlinear Schrödinger equation (NLS) for the complex field u,

$$iu_t + u_{xx} + |u|^2 u = 0.$$

The first of these has become something of a testing ground for characterisations of integrability, and will be considered further in a later chapter. Much work has been done on the properties of the NLS and its close link with the Toda lattice equations [4, 28].

Non-evolution type equations often considered in this context include the sine-Gordon equation,

$$u_{xt} = \sin u,$$

the Liouville equation

$$u_{xt} = \mathrm{e}^u,$$

the Boussinesq equation

$$u_{tt} - u_{xx} + (u^2)_{xx} + \frac{1}{3}u_{xxxx} = 0,$$

and the Toda lattice equations, of which the sinh-Gordon equation (that is, the sine-Gordon equation with the sine term replaced by sinh) is a special case.

Higher dimensional integrable equations include the (Anti-) Self-Dual Yang-Mills equations ((A)SDYM), the Kadomtsev-Petviashvili (KP) equation, and a modified chiral equation which we will consider in a later section. The SDYM and its relations are of particular interest because many of the known integrable equations are special cases of these systems [30, 59, 1], and it has been conjectured that all the integrable equations arise in this way. Some of the equations above are the first (or second) members of a hierarchy, or even of several hierarchies, of related equations: examples of this are the KdV and KP equations, which occur

in such hierarchies. The Self-dual Yang-Mills equations may also be placed in a collection of hierarchies [47].

Of course, this list is far from being a catalogue of all the known integrable nonlinear equations, but it introduces those systems which will concern us in what follows. Many more examples may be found in the extensive literature, for example [16, 2].

2.2 Some Methods for Integrable Equations.

2.2.1 The Inverse Scattering Transform.

One of the first successful methods exploited in the study of nonlinear evolution equations was the Inverse Scattering Transform (IST) [18]. This method makes use of the fact that for many such equations a Schrödinger equation with time-independent spectral parameter and the variable u playing the role of the scattering potential forms one half of a pair of equations the consistency condition for which is the nonlinear equation for u. This Schrödinger equation, together with conditions on the asymptotic behaviour of u, leads to a set of scattering data which may be simply evolved and inverted to obtain information about u. The method, as applied to quantum mechanical problems is well established, and its application to many potentials, via variants of the Marchenko equations, is discussed in [65].

2.2.2 The Darboux Transform.

The occurrence of a Schrödinger equation allows us to use another piece of machinery, namely the Darboux transform [32], to obtain new potentials, and hence new solutions to the original equation. The other half of the linear problem, the time part, must be such that the compatibility condition is just the original equation, as mentioned above. A discussion of one approach to the Darboux transformation will appear in chapter 6.

2.2.3 Bäcklund Transformations.

It is a characteristic property of integrable equations that they possess a special type of transformation, a Bäcklund transformation, which links solutions of the equation to solutions of another equation. In some cases the solutions all belong to the same equation, in which case the transformation is called an auto-Bäcklund transformation.

The linking equations are usually differential equations of lower order than the original equations, and the consistency conditions on these low order equations give rise to the desired integrable equations. A geometric interpretation is given in chapter 7, where the machinery of jet bundles is used to describe differential equations, Bäcklund transformations and generalised symmetries.

Once a Bäcklund transformation is known it may be used to construct new solutions to the equation in question from old, possibly trivial solutions. It may also be used to find conserved currents, and both these possibilities will be discussed in chapter 7.

2.2.4 Twistor Methods For Integrable Systems.

Solutions to the self-dual Yang-Mills equations in four dimensions may be expressed in terms of complex bundles over a complex twistor space. The correspondence may be reduced to lower dimensions, and since many of the well-known integrable equations arise as such reductions it is useful to explore this idea. In chapter 4 the theory of twistors in 2+1 dimensions is briefly examined and used to develop a numerical scheme for solving the initial-value problem for a specific integrable equation in 2+1 dimensions.

2.2.5 Other Methods

The evolution type equations may sometimes be derived from a Hamiltonian system. That is, there is a skew-adjoint operator $\mathcal{D}$ and a functional h of u such that

$$u_t = \mathcal{D}\delta_u h,$$

where δ_u is the variational derivative operator. This is an extension of the usual Hamiltonian method in classical mechanics. Of particular interest are systems having more than one hamiltonian formulation, as these systems may then be shown to possess an infinite tower of conserved currents: as we stated above, this can be a hallmark of an integrable equation.

Many other powerful methods for tackling nonlinear integrable equations exist, but we will satisfy ourselves with this brief review, as it introduces the major topics of later chapters.

Chapter 3

A Construction for Discrete Integrable Systems.

3.1 Introduction.

This chapter concerns novel types of discrete integrable system in two dimensions. The systems will be differential- difference equations related to classical integrable partial differential equations, but involving a discretised 'space' variable. The type of system considered here is to be contrasted with those, such as the Toda lattice systems, which are integrable 'as they stand' [3, 4].

We examine a situation where the notions of derivations and multiplication of functions must be re-examined. The motivation here is not simply to facilitate numerical computation, although the ideas may be of interest in this area, but to investigate the the idea of integrability for the type of discretised systems under consideration [62].

Drell *et al* proved [15] that no lattice gradient operator can satisfy Leibniz's rule when the product of two functions is pointwise (or 'local'). To get round the conclusions of this theorem Bouguenaya and Fairlie [8] generalised the notion of the product to encompass nonlocal effects, and noted a kind of duality between the degree of localisation of the gradient operator and that of the associated product.

In the next section we review the background to this problem, and establish the reasons for wishing a version of Leibniz's rule to hold when constructing a discrete integrable system.

The third section establishes notation, and we then go on to set up conditions on pairs $(\nabla, *)$ of derivations and products such that the Leibniz rule holds. Some solutions are obtained before a second method is introduced, and a relation between the two formulations is presented explicitly for a particular case.

In the final section some general results about systems constructed in this way are presented, and the consequences examined in a few simple cases.

Some of the calculations exploit the properties of the multinomial coefficients, and the relevant identities are briefly derived in Appendix A. A second appendix lists some results and computer code.

3.2 Discrete Derivations and Leibniz's rule.

The study of discrete versions of differential equations by finite difference methods has an extensive history, and recently the possibility of probing consequences of strong interactions using the methods of lattice gauge theory has been intensively investigated. It was in this context, in particular for the problem of putting matter fields on a lattice, that Drell *et al* [15] showed that any linear discrete version of the derivative would fail to satisfy Leibniz's rule when the product was taken to be pointwise multiplication.

The continuous version of Leibniz's rule is

$$\nabla_i(fg) = (\nabla_i f)g + f(\nabla_i g), \tag{3.1}$$

where f and g are functions of the independent variables $\{x^i\}$ and $\nabla_i = \partial/\partial x^i$. This rule is central to the Lax pair formulation of integrability, so if we wish to extend this definition to discrete systems we should see whether we can escape the conclusions of Drell *et al.* Let us first see how the rule is involved in the Lax pair formulation, and how its loss affects the discrete analogue.

In the Lax pair formulation a pair of linear equations, usually involving a free parameter, give rise to the integrable equation of interest through a compatibility condition. Examples are given in chapters 4 and 5. A simple example of a Lax pair is

$$\partial_x \Psi = A\Psi,$$

$$\partial_t \Psi = B\Psi,$$

where A and B may depend upon some parameter. Cross differentiating we arrive at the 'zero curvature' condition

$$A_t - B_x + AB - BA = 0.$$

In some cases powers of the parameter occurring in the Lax pair are equated separately to zero to obtain a set of equations, but we will use this simplified system to illustrate our point. It is clear that, in deriving this equation, we have used Leibniz's rule to expand the derivatives $\partial_t(A\Psi)$ and $\partial_x(B\Psi)$. For a finite difference version of the x derivative we can no longer do this.

Given a one dimensional lattice of points x_n, $n = 1, \ldots N$, the pointwise product of f and g is

$$(fg)_n = f_n g_n,$$

where $f_n = f(x_n)$. Suppose we take as our discrete version of ∂_x the forward difference operator ∇_f, where $(\nabla_\mathrm{f} f)_n = f_{n+1} - f_n$. Then

$$(\nabla_\mathrm{f}(A\Psi))_n = (\nabla_\mathrm{f} A)_n \Psi_{n+1} + A_n (\nabla_\mathrm{f} \Psi)_n.$$

The first term on the right may be expanded using the explicit expression for $\nabla_\mathrm{f}\Psi$, and we find that in addition to the analogue of the usual integrable equation, we have an extra constraint: that the 'gauge' fields are constant in space. To avoid the occurrence of an overdetermined system we look to restore a version of (3.1).

The stipulation that the product should be local, that is, defined by pointwise multiplication at each lattice site, seems very natural, but if relaxed can lead to restoration of Leibniz's rule, as pointed out by Bouguenaya and Fairlie [8]. In their work a family of compatible products and derivations was found, starting from a fairly simple ansatz.

The formulae used in [8] to define certain desirable properties of a derivation and compatible product will be used in the next section, cast in a slightly modified notation. The question of boundary conditions will be considered later. We note that not all of the conditions imposed by Bouguenaya and Fairlie are linearly

independent, and the objects examined in later sections of their paper could be degenerate in some cases. Subsequent development of the method will follow [62], addressing some of the questions raised there.

3.3 Notation and Preliminaries

3.3.1 Notation.

To set up our discrete systems the space variable x will be taken as a set of ordered equally-spaced points on the real line, indexed by the integers. Throughout this chapter we will take the grid spacing to be unity and omit it in our formulae, with the understanding that it may be reinstated if desired. The time variable will remain continuous, although we could choose to discretise all the variables occurring in the problem. A function of x assigns to each point x_n the number (real or complex) $f_n = f(x_n)$. The evaluation map

$$E_n : f \mapsto f_n \tag{3.2}$$

will be used extensively below. Initially we will make no explicit mention of boundary conditions, but we should be aware that their inclusion will have significant consequences.

If f is a function of x we require that ∇f, the derivative of f, also be a function, and define the 'components' of ∇, d_{nm} by

$$E_n(\nabla f) = \sum_m d_{nm} E_m(f). \tag{3.3}$$

For example, the forward difference operator has $d_{nm} = \delta_{n+1,m} - \delta_{n,m}$, where $\delta_{n,m}$ is the Kronecker delta.

The product $f * g$ of two functions will be defined by

$$E_n(f * g) = \sum_{i,j} M_{nij} E_i(f) E_j(g). \tag{3.4}$$

The Leibniz rule then becomes a condition on (d_{nm}, M_{nij}), and solutions to this and other conditions to be introduced below will lead to the desired systems.

3.3.2 Conditions on ∇ and $*$.

The three main conditions on our operations correspond to properties which are desirable for differentiation and multiplication in any context (see [8]).

Firstly, the derivative of a constant must vanish. It is clear from (3.3) that this leads to the condition

$$\sum_m d_{nm} = 0. \tag{3.5}$$

This tells is that, if we consider the matrix with components $\{d_{ij}\}$, the rows of this matrix are linearly dependent.

Secondly, the product of a constant and a function should be just the pointwise product:

$$E_n(k * g) = k \sum_{ij} M_{nij} E_j(g) = k E_n(g), \tag{3.6}$$

so

$$\sum_i M_{nij} = \delta_{n,j}. \tag{3.7}$$

We will require pointwise commutativity, which is just the condition $M_{nij} = M_{nji}$, and associativity, which produces

$$\sum_j M_{nij} M_{jkl} = \sum_j M_{nlj} M_{jki}. \tag{3.8}$$

This will clearly be a difficult condition to check in general.

The central requirement will be Leibniz's rule

$$E_n(\nabla(f * g)) = E_n(f * \nabla g + (\nabla f) * g). \tag{3.9}$$

On the left of (3.9) we have

$$\sum_m d_{nm} E_m(f * g) = \sum_{mij} d_{nm} M_{mij} E_i(f) E_j(g),$$

and on the right

$$\sum_{im} E_i(f) M_{nim} E_m(\nabla g) + \sum_{mj} E_m(\nabla f) M_{nmj} E_j(g)$$

which is

$$\sum_{mij}(d_{mj}M_{nim} + d_{mi}M_{nmj})E_i(f)E_j(g).$$

Since f and g are arbitrary we must have

$$\sum_m d_{nm}M_{mij} = \sum_m (d_{mj}M_{nim} + d_{mi}M_{nmj}). \tag{3.10}$$

This relation will be important in what follows. Our approach will be to satisfy (3.10) first, then verify that our product is consistent with (3.7) and (3.8).

3.4 Some Examples.

3.4.1 The Forward Difference Scheme.

To illustrate the strategy for finding the M_{nij} we will look at the simplest case. This example will be used again in later sections. Here we take $d_{mn} = \delta_{m+1n} - \delta_{mn}$, so that Leibniz's rule (3.10) becomes

$$M_{n+1ij} = M_{nij-1} - M_{nij} + M_{ni-1j}. \tag{3.11}$$

This three-index recurrence relation can be solved using a generating function.

Let

$$u(x,y,z) = M_{000} + M_{001}z + M_{010}y + M_{100}x + ... + M_{ijk}x^i y^j z^k + ...$$

Take the product to be local at $x_0 = 0$, so $M_{0ij} = \delta_{i,0}\delta_{j,0}$. Then, from (3.11) we have

$$u = \frac{1}{1 - x(y + z - 1)},$$

$$u = \sum_{pij} \binom{p}{i\ j\ p-i-j} (-1)^{p-i-j} x^p y^i z^j.$$

This gives

$$M_{nij} = (-1)^{n-i-j} \frac{n!}{i!j!(n-i-j)!}. \tag{3.12}$$

With M_{nij} so defined, the Leibniz rule is respected, but we should check the other conditions.

Clearly $\sum_m d_{nm} = 0$, so we need to look at (3.7):

$$\sum_i M_{nij} = \sum_i (-1)^{n-i-j} \binom{n}{i\ j\ n-i-j}$$

which is

$$\frac{n!}{j!(n-j)!} \left(\sum_i (-1)^i \binom{n-j}{i} \right) = \delta_{n,j}.$$

Condition (3.8) is difficult to check directly, and a demonstration that this definition of the product satisfies this condition will await the introduction of a second method.

3.4.2 A Second-order Scheme.

In order to illustrate some of the drawbacks of this direct approach we will consider a less straightforward case. Take $d_{mn} = \delta_{m+1,n} - 2\delta_{m,n} + \delta_{m-1,n}$, for which

$$M_{n+1ij} = M_{nij+1} + M_{nij-1} + M_{ni+1} + M_{ni-1j} - M_{n-1ij} - 2M_{nij}.$$

It is clear that more boundary conditions are needed here: these will be accommodated by letting

$$u(x,y,z) = M_{000} + \text{low terms} + \sum_{nij} M_{n+1ij} x^{n+1} y^i z^j.$$

We find

$$u = \frac{\Lambda}{1 - x\{(y + y^{-1}) + (z + z^{-1}) - (x+2)\}},$$

where Λ depends upon the choice of boundary conditions.

From the form of u we see that y and y^{-1} are on exactly the same footing in the denominator, and that in M_{kij} no terms with $j > k$ or $l > k$ will occur.

The infinite formal sum for u, with $\Lambda = 1$ may be decomposed as follows:

$$u = \sum_{k=0}^{\infty} x^k ((y + y^{-1}) + (z + z^{-1}) - (x+2))^k$$

$$= \sum_{k=0}^{\infty} \sum_{m=0}^{k} x^k \binom{k}{m} P^m x^{k-m} (-1)^{k-m},$$

where $P =\equiv (y + y^{-1}) + (z + z^{-1}) - 2$.

Then

$$u = \sum_{k=0}^{\infty} \sum_{m=0}^{k} \binom{k}{m} P^m x^{2k-m} (-1)^{k-m}.$$

To pick out the powers of x in u, say the nth power, set $n = 2k - m$: the sum then runs over k and m such that $2k - m = n$. Terms in x^n do not occur beyond the $k = n$ term in u, nor below the $k = [\frac{(n+1)}{2}]$ term (where $[a]$ denotes the integer part of a,) so the coefficient of x^n is

$$\sum_{k=[\frac{(n+1)}{2}]}^{n} \binom{k}{2k-n} P^{2k-n} (-1)^{n-k}.$$

To expand P we use multinomial coefficients:

$$P^m = \sum_{\substack{i=0 \\ j=0}} (-2)^{m-i-j} \binom{m}{i\ j\ m-i-j} (y + y^{-1})^i (z + z^{-1})^j.$$

Now

$$(y + y^{-1})^i = \sum_{s=0}^{i} \binom{i}{s} y^{i-2s},$$

so we must pick out powers of y and z by summing subseries of u. The result is

$$M_{npq} = \sum_{k=[\frac{(n+1)}{2}]}^{n} \sum_{ij} (-2)^{2k-n-i-j} (-1)^{n-k} \times$$

$$\binom{2k-n}{i\ j\ 2k-n-i-j} \binom{i}{\frac{(i-p)}{2}} \binom{j}{\frac{(j-q)}{2}}. \tag{3.13}$$

It is understood here that the multinomial coefficients vanish when any of the components is half an odd integer or less than zero. This makes calculation of low terms simpler since only terms in which $i-p$ and $j-q$ are even can contribute, and $i + j < 2k - n + 1$. Some of these terms are shown in Appendix A. A FORTRAN code was used to generate these terms and, in general, such a program could be used to obtain results where explicit formulae cumbersome to work with.

It is evident that checking any of the conditions of section 2 for this M_{nij} is a laborious task, as indeed was the calculation leading up to (3.13). For this reason, and others which will become apparent, the second method alluded to above will be introduced.

3.5 Powers of x.

3.5.1 The Basic Ideas.

The difficulties encountered in using the M_{nij} described above can be alleviated by looking not directly at the general multiplication scheme but at the behaviour of powers of x. Thus we would consider $x^{[2]} = x * x$ and so on, using these powers to define functions as 'power series' in x. Of course the notation $[n]$ should not be confused with the 'integer part' symbol.

The essential content of Leibniz's rule becomes

$$E_i(\nabla x^{[n]}) = nE_i(x^{[n-1]}), \tag{3.14}$$

which is a simpler expression to work with than (3.10). The advantage here is that associativity is guaranteed for the product so defined, and laborious checking can be avoided. It is also a more direct way of obtaining function definitions, as we will see. We will here use the notation

$$a_{nj} = E_j(x^{[n]}). \tag{3.15}$$

3.5.2 Examples.

The first example will again be the forward difference operator, for which (3.14) becomes

$$a_{nj+1} - a_{nj} = na_{n-1j}. \tag{3.16}$$

In principle the generating function

$$u(x, y) = a_{00} + a_{01}y + a_{10}x + ... + a_{ij}x^i y^j + ...$$

could be used to solve the recurrence (3.16), but leads to a differential equation for u the solution of which involves the exponential integral. A cleaner approach is a direct method.

We will require that $x^{[1]} = x$ in this example, although this condition will not necessarily hold in general, so that $a_{1j} = j$. Then

$$a_{2j} = a_{2j-1} + 2(j - 1)$$

$$= a_{20} + 2((j-1) + (j-2) + ... + 1) = a_{20} + j(j-1).$$

For a_{3j} we find

$$a_{3j} = a_{3j-1} + 3(a_{20} + (j-1)(j-2))$$

$$= a_{30} + 3ja_{20} + j(j-1)(j-2).$$

In general, by a process of induction, we have

$$a_{Nj} = \sum_{r=0}^{N} r! \binom{N}{r} \binom{j}{r} a_{N-r0}, \tag{3.17}$$

where $a_{00} = 1, a_{10} = 0$ and the a_{m0} are constants. These constants may be chosen arbitrarily, and this point will be examined further in later sections.

If we choose $a_{m0} = 0$ for $m > 0$ (which corresponds intuitively, when the product is local at 0, to $0^{[m]} = \delta_{m0}$, from (3.15)), we find

$$a_{Nj} = N! \binom{j}{N} = \frac{j!}{(j-N)!}.$$

It is a straightforward calculation to verify that the a_{Nj} satisfy (3.16).

Now we may examine the link between these a_{Nj} and the M_{nij}. We will show that, in the case of the forward difference operator the expressions obtained above for the a_{Nj} and M_{nij} are compatible, and therefore the M_{nij} satisfy (3.8).

We start from the expression

$$a_{p+qn} = \sum_{ij} M_{nij} a_{pi} a_{qj}, \tag{3.18}$$

which is just a statement of the identity $x^{[p]} * x^{[q]} = x^{[p+q]}$. The boundary conditions on M and a are equivalent, as may be seen by evaluating (3.18) for $n = 0$. Now

$$M_{nij} = (-1)^{n-i-j} \binom{n}{i \; j \; n-i-j},$$

$$a_{nj} = n! \binom{j}{n},$$

so that the r.h.s. of (3.18) is

$$\sum_{ij} (-1)^{n-i-j} \frac{n!}{(n-i-j)!(i-p)!(j-q)!}.$$

Only $i > p$ and $j > q$ contribute to the sum, so setting $k = i - p$, $l = j - p$ we have

$$\sum_{k+l=0}^{n-p-q} (-1)^{n-k-l-p-q} \frac{(n-p-q)!p!q!}{k!l!(n-p-q-k-l)!} \binom{n}{p\ q\ n-p-q}.$$

Now, from Appendix A,

$$1 = (1+1-1)^{n-p-q} = \sum_{k+l=0}^{n-p-q} (-1)^{n-p-q-k-l} \frac{(n-p-q)!}{k!l!(n-p-q-k-l)!},$$

so the r.h.s is

$$p!q!\binom{n}{p\ q\ n-p-q} = \frac{n!}{(n-p-q)!} = a_{p+qn},$$

as required.

This demonstrates that the M_{nij} do indeed satisfy the conditions of section 2, and demonstrates that the a_{Nj} give a neater formulation of the problem.

3.5.3 A Simple Method for Obtaining a_{Nj}.

As the difference operator becomes more complex, so too does the recurrence relation for a_{Nj}, and as this is a two-index relation the calculations become less tractable. Even the case of

$$d_{nm} = \frac{\delta_{n+1,m} - \delta_{n-1,m}}{2}$$

presents considerable difficulties.

An alternative approach is as follows: we solve the problem

$$E_n(\nabla f) = \alpha E_n(f), \tag{3.19}$$

with suitable boundary conditions, to obtain the equivalent of $\exp(\alpha x)$, which we denote $\exp *(\alpha x)$. This solution to (3.19) may be used to find $x^{[n]}$, although we will find that for some purposes this is not essential. The expression

$$\frac{\partial^r}{\partial \alpha^r} \exp(\alpha x)\Big|_{\alpha=0} = x^r,$$

which is readily checked, leads us to define

$$\lim_{\alpha \to 0} [\frac{\partial^n}{\partial \alpha^n} \exp * (\alpha x)] = x^{[n]}. \tag{3.20}$$

This definition reflects the desired properties of the object $x^{[n]}$ and more importantly is, in principle, tractable for most reasonable difference operators.

As a first example take

$$E_n(\nabla f) = f_{n+1} - f_n.$$

The 'eigenfunction' problem becomes

$$f_{n+1} = (1+\alpha)f_n = ... = (1+\alpha)^{n+1} f_0. \tag{3.21}$$

With the boundary condition $f_0 = 1$ we have

$$j^{[n]} = \lim_{\alpha \to 0} \frac{\partial^n}{\partial \alpha^n} (1+\alpha)^j,$$

which is

$$j(j-1)...(j-n+1) = \frac{j!}{(j-n)!},$$

in agreement with the expression derived in section 3.4. The arbitrary constants in the a_{Nj} of section 3.4 occur here in the boundary condition for f_0, which is actually an arbitrary function of α. The Taylor coefficients of $f_0(\alpha)$ correspond to the a_{m0}. To see this in the case of the forward difference scheme, we will examine (3.20) with

$$\exp *(\alpha x) = \sum_{r=0}^{j} \binom{j}{r} \alpha^r f_0(\alpha).$$

Then from (3.20) we have

$$E_j(x^{[n]}) = \sum_{p=0}^{n} \sum_{r=0}^{j} \binom{j}{r} \binom{n}{p} \frac{\partial^p}{\partial \alpha^p} (\alpha^r) \frac{\partial^{n-p} f_0}{\partial \alpha^{n-p}} \Big|_{\alpha=0},$$

which is

$$\sum_{r=0}^{n} \binom{j}{r} \binom{n}{r} r! \frac{\partial^{n-r} f_0}{\partial \alpha^{n-r}} \Big|_{\alpha=0}.$$

We recognise the expression of (3.17) with $a_{N-r}0$ given by terms in the Taylor expansion of f_0 about 0.

The second example follows the pattern of the first for the difference operator $E_n(\nabla f) = (f_{n+1} - f_{n-1})/2$.

The eigenfunction problem is

$$f_{n+1} = 2\alpha f_n + f_{n-1}, \tag{3.22}$$

which is related to the problem of finding Fibonacci's numbers.

Let

$$u(x) = f_0 + f_1 x + ... + f_{n+1}x^{n+1} + ...$$

be a generating function for the problem. We see that, taking as boundary conditions $f_0 = 1, f_1 = 2\alpha$,

$$u(x) = \frac{1 + 2\alpha x}{1 - 2\alpha x - x^2}.$$

One can use partial fractions to find $\{f_n\}$ in terms of α, but it is simpler to observe that u is closely related to a generating function for the Chebychev polynomials, and the f_n's may be obtained from these.

The first few f_n's are:

$$f_0 = 1;$$

$$f_1 = 2\alpha;$$

$$f_2 = 4\alpha^2 + 1;$$

$$f_3 = 8\alpha^3 + 4\alpha.$$

Then we find directly that

$$j^{[n]} = 2^n n! \binom{\frac{j+n}{2}}{\frac{j-n}{2}}, \tag{3.23}$$

which is nonzero only for $j \pm n$ even, according to the conventions listed in Appendix A.

An example which involves the Chebychev polynomials directly is furnished by the operator

$$E_n(\nabla f) = \frac{f_{n+1} - 2f_n + f_{n-1}}{2}.$$

The eigenfunctions are those polynomials in $\beta = 1 + \alpha$ satisfying

$$P_{n+1} + P_{n-1} = 2\beta P_n,$$

and these include the type I Chebychev polynomials. In this case

$$j^{[n]} = \lim_{\beta \to 1} (\frac{\partial}{\partial \beta})^n P_j(\beta).$$

The difference operator here is the analogue of the double derivative in the continuous case, but we can treat it as a formal discrete derivation and demand that it be compatible with a Leibniz rule in order to obtain an integrable system. Of course its continuum analogue, ∂_x^2, does not satisfy Leibniz's rule, so the discrete version leads to a novel type of system.

3.6 Power Series and Applications.

3.6.1 Introduction.

The most important use for the a_{Nj} is in the definition of functions of x. An obvious step is to develop the analogy between discrete and continuous systems and, having obtained the a_{Nj}, define functions formally in terms of their 'Taylor series'. Thus, we will have

$$E_i(f) = \sum_m \frac{Q_m}{m!} a_{mi}, \tag{3.24}$$

where the Q_m completely determine f. For example, if $f = \beta \exp *(\alpha x)$ we have $Q_m = \beta \alpha^m$.

With functions so defined we can use the methods developed above to expand powers of a given function f. This allows us to examine functions of a function, which will be important in applications. Consider

$$E_i(f * f) = \sum_{jk} M_{ijk} E_j(f) E_k(f).$$

We can expand $E_r(f)$ using (3.24), to find

$$E_i(f^{[2]}) = \sum_{jk} M_{ijk} \sum_r \frac{Q_r}{r!} a_{rj} \sum_s \frac{Q_s}{s!} a_{sk}.$$

From (3.18) we see that this expression is just

$$\sum_{rs} \frac{Q_r Q_s}{r! s!} a_{s+r\ i}.$$

We want to write this in the form of (3.24), for which purpose we define

$$Q_m^{(2)} = \sum_{k+l=m} Q_k Q_l \binom{m}{k},$$

and, similarly,

$$Q_m^{(s)} = \sum_{k_1+k_2+\ldots+k_s=m} \frac{m! Q_{k_1} Q_{k_2} \ldots Q_{k_s}}{k_1! k_2! \ldots k_s!}. \tag{3.25}$$

In this notation

$$E_i(f^{[s]}) = \sum_m \frac{Q_m^{(s)}}{m!} a_{mi}. \tag{3.26}$$

In the next subsection we calculate $Q_m^{(s)}$ explicitly for certain functions occurring in an interesting example.

3.6.2 The Discrete sine-Gordon Equation.

We have been leading up to the construction of integrable discrete versions of standard integrable equations. The overall strategy in realising this goal is to replace the continuum derivative by a discrete version, and any products by the discrete product compatible with the derivative. This compatibility ensures that constructions for the continuum case carry over to the discretised version, and that, in particular, solutions of the standard equation may in principle be translated into solutions of the corresponding discretised equation. Explicit examples will be given below.

The first discrete system to which we turn our attention will be the analogue of the sine-Gordon equation written in laboratory coordinates:

$$\partial_x^2 \phi - \partial_t^2 \phi = \sin \phi. \tag{3.27}$$

The discrete version is obtained by writing ∂_x as ∇ and sin as sin$*$. The discrete 'one-soliton' solution is, in analogy with the continuum case,

$$\phi(x,t) = 4\arctan * (\beta \exp * \alpha x), \tag{3.28}$$

where

$$\beta = \mathrm{e}^{-\alpha v t}, \text{ and } \alpha = \frac{1}{\sqrt{(1-v^2)}}.$$

Here v is to be interpreted as a speed, and α as the usual relativistic 'γ' factor. Now define

$$E_i(\exp * f) = \sum_s \frac{f^{[s]}}{s!},$$

$$= \sum_s \sum_m \frac{Q_m^{(s)}}{m!\ s!} a_{mi}.$$

Analogous expressions for other functions are obtained by substituting the relevant Taylor coefficients, derived from the continuum versions of the functions, into the expansion. We find

$$E_i(\arctan * f) = \sum_s \sum_m \frac{(-1)^s}{(2s+1)} \frac{Q_m^{(2s+1)}}{m!} a_{mi}. \tag{3.29}$$

Taking $f = \beta \exp *(\alpha x)$ we have

$$Q_m^{(2)} = \sum_k \beta^2 \alpha^m \binom{m}{k} = \beta^2 (2\alpha)^m,$$

and, using multinomial identities from Appendix A,

$$Q_m^{(s)} = \beta^s (s\alpha)^m.$$

From (3.29) we find

$$E_i(\arctan * f) = \sum_s \sum_m \frac{(-1)^s ((2s+1)\alpha)^m \beta^{2s+1}}{(2s+1)m!} a_{mi}.$$

Provided that $\beta < 1$ we can sum this series for arbitrary a_{mi} by noting that

$$\sum_r \frac{(-1)^r \beta^{2r+1} ((2r+1)\alpha)^m}{(2r+1)} = (\alpha\beta)^m \frac{\partial^m}{\partial \beta^m} \arctan(\ \beta)$$

$$= \alpha^m \frac{\partial^m}{\partial \gamma^m} \arctan\ e^{\ \gamma},$$

where $\gamma = \log\ \beta$. This tells us immediately that

$$E_i(\arctan * (\beta \exp *(\alpha x))) = E_i(\exp * (\alpha x \frac{\partial}{\partial \gamma})) \arctan\ \mathrm{e}^{\gamma}. \tag{3.30}$$

We may verify that this holds in the continuum limit, when $\exp(\alpha x \frac{\partial}{\partial \gamma})$ is just the translation operator, sending $f(\gamma)$ to $f(\gamma + \alpha x)$. Since $\gamma = -\alpha v t$ we find

$$\phi(x,t) = 4 \arctan \exp(\frac{x - vt}{\sqrt{1 - v^2}}),$$

which is one form of the one-soliton solution. This observation will be taken up in the next subsection.

As a discrete example, take ∇ to be the forward-difference operator, with $E_i(\exp * \alpha x)$ given by (3.21) with $f_0 = 1$. Then (3.30)is

$$(1 + \alpha\frac{\partial}{\partial\gamma})^i \arctan\ e^{\ \gamma},$$

which is essentially a result quoted by Ward [62]. When $v = 0$, $\alpha = 1$, and if $\gamma = -\frac{N}{2}$ with N large this becomes (since e^γ is small)

$$2^i e^{-\frac{N}{2}} = \exp(-\frac{N}{2} + i\log 2), \tag{3.31}$$

which grows exponentially for large i.

We may now consider the effect of including the arbitrary constants in exp$*$. In the limiting case above the effect will be to multiply (3.31) by $f_0(1)$, which clearly does not improve the behaviour for large x for nontrivial f_0.

The expression (3.30) is quite general, and in this example avoids the need to compute $x^{[n]}$ explicitly. Other results for exp$*$ may be substituted into (3.30) and the result examined for its properties. In the case of the central difference scheme, the result is, for the limiting case,

$$\phi(x,t) \approx 4P_i(1)e^{-\frac{N}{2}},$$

where $P_i(1)$ is the polynomial obtained in section 4 evaluated at $\beta = 1$. For large i it grows faster than 2^i, and so the large-x behaviour is still exponential growth.

3.6.3 Further Applications.

The observation that the translation operator $t \mapsto t + a$ is realised by $\exp(a\partial_t)$ leads to the interesting result that travelling solutions to many equations may be constructed for the discrete case by substituting exp$*$ for exp, as in the following examples. Here we restrict our attention to the forward difference operator $E_n(\nabla f) = E_{n+1}(f) - E_n(f)$.

First, we will look at some linear examples.

The general solution to the 1+1-dimensional wave equation

$$\partial_x^2 f - \partial_t^2 f = 0$$

is

$$f(x,t) = u_+(t+x) + u_-(t-x) = e^{x\partial_t}u_+(t) + e^{-x\partial_t}u_-(t),$$

where $u_\pm$ are arbitrary functions of their argument. This can be seen in a formal way by treating ∂_t as a constant in the x-differentiation. The spatially discrete analogue of this general solution is

$$E_i(f) = E_i(\exp * x\partial_t)u_+(t) + E_i(\exp * x(-\partial_t))u_-(t),$$

so in the discrete case the wave-equation

$$f_{n+2} - 2f_{n+1} + f_n - \partial_t^2 f = 0$$

has solutions

$$E_n(f) = (1+\partial_t)^n u_+(t) + (1-\partial_t)^n u_-(t),$$

where $u_\pm$ are arbitrary smooth functions of t.

Similarly, the function g such that

$$E_n(g) = (1 + \frac{1}{c}\partial_t)^n e^{c^2 t} = (1+c)^n e^{c^2 t}$$

with arbitrary c solves the spatially discrete heat equation

$$\nabla^2 g - \partial_t g = 0.$$

This can be quickly demonstrated by direct calculation, but the analogy technique is neater.

A 1-soliton solution to the Korteweg-deVries equation is

$$u(x,t) = -2\frac{k^2}{\cosh^2(kx - 4k^3 t)},$$

which leads to the discrete KdV solution

$$u = -2k^2 \exp * (\frac{-x}{4k^2}\partial_t)\mathrm{sech}^2(-4k^3 t),$$

where the discrete KdV equation is written

$$\partial_t u - \nabla^3 u + 6u * \nabla u = 0.$$

Since the function exp$*$ is relatively easy to obtain from the derivative ∇, the study of this type of solution is fairly straightforward. It should be emphasised that, given any solution to the standard equation for which we know a power series expansion, we can write down the corresponding solution to the discrete equation by replacing x^n with $x^{[n]}$. The difficulty is in obtaining and manipulating the $x^{[n]}$, and the special feature of the travelling waves is that we do not need to use the explicit expression. It may be that in more general cases we will be able exploit similar techniqes, but the solutions need not be wave-like, and we may have to go back to the basic formulae. Despite this, travelling wave-like solutions are often of special interest in applications and there is some justification for giving them special consideration.

3.7 Conclusions.

The methods discussed in this chapter show that, at least formally, interesting discrete analogues of nonlinear integrable systems can be written down, these discrete systems retaining integrability. Some problems with the asymptotic behaviour of the solutions to the resulting equations may place restrictions on the usefulness of the systems, but further investigation building on the methods presented may point to a resolution of these difficulties.

The overall strategy of enforcing Leibniz's rule and 'translating' continuum solutions into the corresponding discrete solutions allows a very wide range of equations to be treated.

Chapter 4

Numerical Implementation of the twistor method.

4.1 Introduction

The subject of this chapter is the numerical implementation of the twistor method in the solution of the initial-value problem for a modified chiral model in 2+1 dimensions. The method itself is derived from Ward's construction of anti self-dual Yang- Mills fields using twistor methods [55], and the numerical development follows the theory very closely. The motivation for this line of development stems from work done using standard numerical methods on a related system possessing $SO(2)$ spatial symmetry, where perturbations of a static lump were examined [49], and on this system itself where interesting and sometimes nontrivial scattering behaviour of various configurations was observed [50, 60].

Whilst the simulations indicated interesting behaviour, no use was made of the fact that the modified chiral equation may be obtained as the compatibility condition for a Lax pair, and hence is integrable. The construction presented in this chapter explicitly incorporates this property.

Models of this kind in 2+1 dimensions have a number of roles. Firstly, they act as a testing ground for models in higher dimensions: an example of such a model is the Skyrme model for baryons, which is described in the context of the Standard Model in [13]. The interactions of similar structures in 2+1 dimensions is investigated in [51]. Whilst there are similarities between the Skyrme structures

and those occurring in this integrable model, it is important to understand that the reasons for stability in each case are quite different. Whereas the modified chiral equation considered below is integrable, and possesses soliton solutions similar to those in other integrable systems such as the sine-Gordon and KdV equations, the Skyrme-type models are not, and stable lumps arise due to the inclusion of additional terms [49].

Secondly, as higher dimensional versions of integrable systems their study can extend understanding of the nature of such systems and methods which may be applied to their study.

The next section introduces some background to the use of twistors in the solution of equations in 2+1 dimensions. The chiral equation of interest is then presented, and the existence of a Lax pair is used to develop the twistor description. The structure of the numerical implementation is examined, and some examples, techniques and strategies considered. A description of the computer code concludes the chapter, and the code is laid out in the appendices.

4.2 A Brief Introduction to Twistors in 2+1 Dimensions.

Twistors were introduced into physics by Penrose [39], and have found a remarkable number of applications both in physics and in mathematics (see for example [6]). The original applications were to problems in 3+1 dimensional spacetimes, but the ideas extend to many other situations [35].

One such extension is to 'minitwistor' space, which is based upon a three complex dimensional spacetime. The metric of signature (+ + +) is appropriate for the study of monopoles [21], whilst that of signature (+ - -) is relevant to the modified chiral model to be examined here. Minitwistor space may be constructed from full twistor space by a symmetry reduction, but we can also present a more direct correspondence with 2+1 dimensional complexified spacetime.

What follows is a summary of some of the ideas needed for this chapter and the next. More details may be found in [59, 22, 40].

4.2.1 The Basic Ideas.

We will be concerned with a three dimensional spacetime with metric

$$\mathrm{d}s^2 = \mathrm{d}t^2 - \mathrm{d}x^2 - \mathrm{d}y^2. \tag{4.1}$$

In order to introduce the ideas of twistor theory which we will use we allow t, x and y to take values in $\mathbf{C}$. The above metric will then be non-Hermitian.

The null planes in this spacetime are conveniently parametrised by points of the holomorphic tangent bundle to the extended complex plane. To see how this comes about let us for the moment examine the case of t, x and y all real. Then, referring to Fig 4.1, we see that a null plane may be identified from its t-intercept and an angle ϵ, both referred to a fixed coordinate system. This allows us to identify the plane using a pair (h, ϵ) in $\mathbf{R} \times S^1$, the tangent bundle to the circle, since given (h, ϵ) we have the equation of the plane, referred to the fixed origin,

$$t + x \sin\epsilon + y \cos\epsilon = h.$$

The complex extension of this leads to the identification of the space of null planes in $\mathbf{C}^{2+1}$ with points of $T\mathbf{P}^1$. A point in $\mathbf{C}^{2+1}$ lies on a $\mathbf{P}^1$ of null planes, and so corresponds to a section of the vector bundle $T\mathbf{P}^1$. In fact, the space of null planes should strictly be described as an affine bundle, since the choice of an arbitrary origin determines a zero section of $\mathbf{T} = T\mathbf{P}^1$.

The equation for the t-intercept in terms of t, z and $\tilde{z}$, where $z = x + iy$ and $\tilde{z} = x - iy$ (note that these are not complex conjugates in general, since x and y are complex), is

$$\gamma = 2t - i\lambda^{-1} z + i\lambda\tilde{z}. \tag{4.2}$$

Here we take γ and λ to be coordinates on $\mathbf{T}$. This equation will be used later in this chapter. Of course, it does not make sense as it stands for $\lambda = 0$ or $\lambda = \infty$ (both of which are points on the extended complex plane): we should use two sets of coordinates patched over the equator, and use $\lambda\gamma$, $\lambda^{-1}\gamma$ as fibre coordinates. The patching function is then λ^2, reflecting the fact that $\mathbf{T}$ is the hyperplane

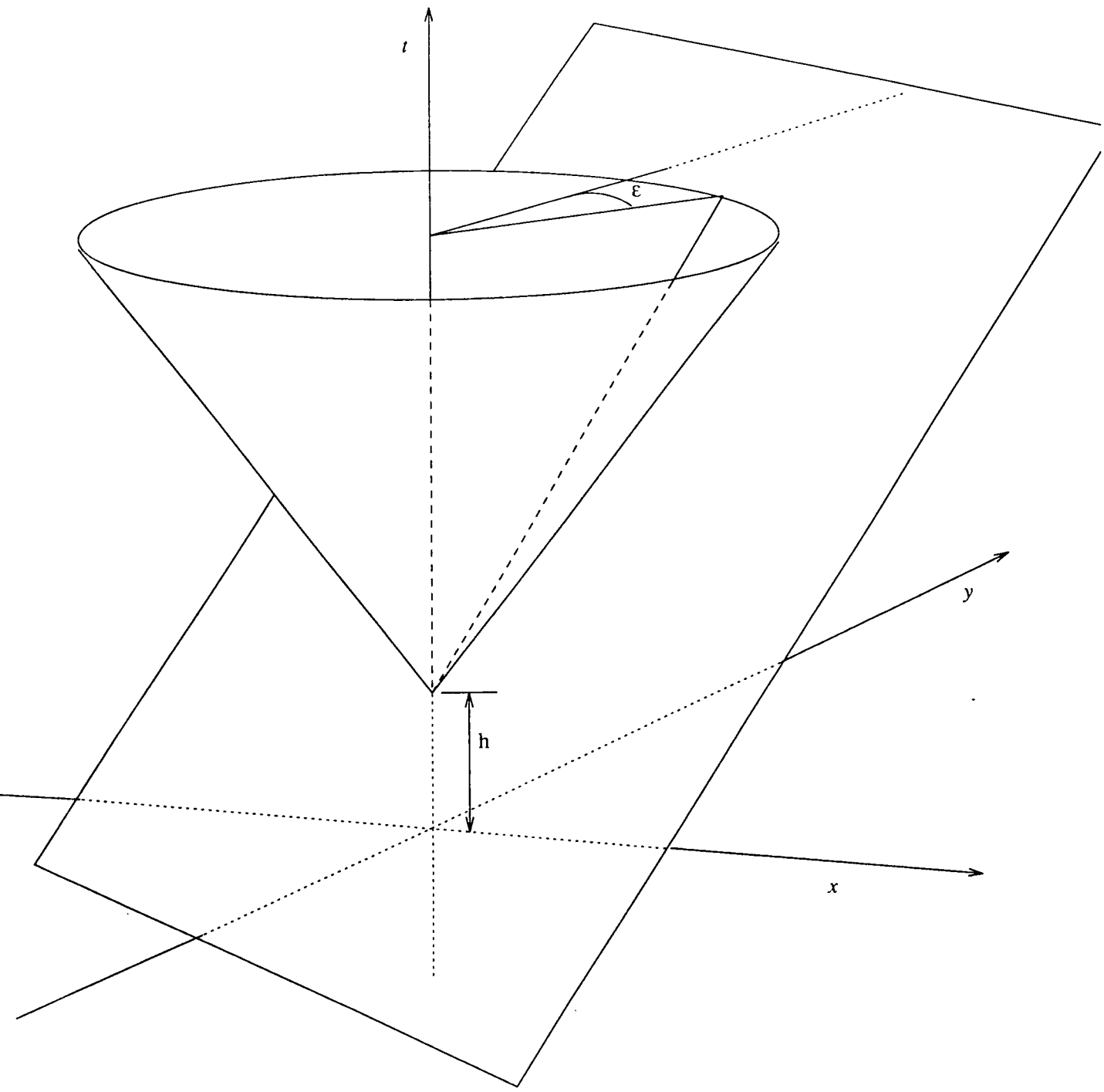

Figure 4.1: A null cone and tangent plane

section bundle of chern number 2 on $\mathbf{P}^1$. Thus, for instance, when we refer to the properties of objects restricted to special sections, it is to the sections of this bundle, $\mathbf{T}$, that we refer.

4.2.2 The 2+1 Dimensional Wave Equation.

In the chapter 6 we will be interested in the twistor solution of the equation

$$(\partial_t^2 - \partial_x^2 - \partial_y^2)u = 0. \tag{4.3}$$

This is an equation for the free spin-0 field u, and has spin-$n/2$ counterparts, which we will describe below.

The 2+1 dimensional coordinates may be combined into an object x^{AB} with components

$$\begin{pmatrix} t+y & x \\ x & t-y \end{pmatrix}. \tag{4.4}$$

This is analogous to the introduction of spinors in 3+1 dimensions using the Pauli matrices. The metric may be written

$$\mathrm{d}s^2 = -\frac{1}{2}\mathrm{d}x^{AB}\mathrm{d}x_{AB}, \tag{4.5}$$

where the spinor indices are raised and lowered with the skew tensors ϵ^{AB} and ϵ_{AB} satisfying $\epsilon^{AB}\epsilon_{AB} = -2$.

The wave operator is proportional to $\partial^{AB}\partial_{AB}$, and we may also consider the equations

$$\partial^{AB}\varphi_{BC\ldots D} = 0, \quad (n \text{ indices on } \varphi),$$

which are the analogues of helicity $n/2$ equations in 3+1 dimensions.

These equations may conveniently be solved using the twistor method. The 3+1 dimensional case is examined in [22], and the 2+1 dimensional scheme presented in [59]. Here we state the general form of solutions.

First, let $[\pi^0, \pi^1]$ be homogeneous coordinates on $\mathbf{P}^1$. The equation of a null plane in $\mathbf{C}^3$ is then $\pi^A\pi^B x_{AB} = \omega$.

Take f to be a homogeneous function of (ω, π) of degree $-n-2$, and consider the integral

$$I_{AB\ldots C} = \frac{1}{2\pi i}\int \pi_A \pi_B \ldots \pi_C f(\pi^D \pi^E x_{DE}, \pi)\pi_F \mathrm{d}\pi^F. \tag{4.6}$$

It is clear that, since $\pi_A \pi^A = 0$, we have

$$\partial^{AB} I_{B\ldots C} = 0,$$

and that many such solutions, depending on the properties of f are available.

Now we consider what the homogeneity of f signifies. In terms of (ω, π) we say that f is homogeneous of degree k if

$$f(a^2\omega, a\pi) = a^k f(\omega, \pi).$$

In order that the integral be well defined on $\mathbf{P}^1$, the integrand must be homogeneous of degree zero: this explains the form of (4.6): each factor of π^A introduces a power of the scaling factor a, and the integration measure $\pi_A \mathrm{d}\pi^A$ introduces a^2.

By Cauchy's theorem we can add to f any function h analytic in any one patch of (a small neighbourhood of a section of) $\mathbf{T}$ and still obtain the same solution to the wave equation: we can effectively shrink the contour of integration to a point without encountering poles of h. This may be interpreted in terms of sheaf cohomology, from which the notation appearing in chapter 6 is drawn. We say that our f is a representative of the cohomology group $H^1(\mathcal{M}, \mathcal{O}(-n-2))$ for a suitable $\mathcal{M}$. We will not discuss the ideas of cohomology theory in any detail, simply drawing attention to the fact that it is an extensively used tool in the literature.

4.3 The Twistor Approach to the Modified Chiral Equation

In this chapter the central object of study will be the equation

$$\partial_x(J^{-1}\partial_x J) + (\partial_y - \partial_t)(J^{-1}(\partial_y + \partial_t)J) = 0, \tag{4.7}$$

where J takes values in the Lie group $SU(2)$, though the ideas carry over to other Lie groups. An important aspect of the twistor approach is the exploitation of the integrability of the equation: a Lax pair may be written down for the problem, and this leads quite directly to the twistor formulation. The idea of the Lax pair formulation is that the consistency condition holds when J satisfies (4.7). The equation itself is a reduction of the ASDYM equations, and in a sense inherits its method of solution from that system.

As we saw above, the null planes in $\mathbf{C}^{2+1}$ (planes tangent to null cones) may be specified, once an origin is chosen, by giving a point on the line $x = y = 0$, say, (one complex number), and a point on the Riemann sphere. This latter point is a pair of complex numbers up to a scale factor. This information corresponds to a single point in $T\mathbf{P}^1$.

We have seen that the characterising equation of null planes through $(t, z, \tilde{z})$ is

$$\gamma = 2t + i\lambda\tilde{z} - i\lambda^{-1}z. \tag{4.8}$$

We note that if we consider $\tilde{z}$ and z to be independent complex coordinates the condition that $\tilde{z}$ is the complex conjugate of z selects out a real spacetime. There is a real structure on $\mathbf{T}$, allowing us to define real sections, which incorporates this fact: note that γ satisfies the condition

$$\gamma(t, \bar{z}, \bar{\tilde{z}}; \lambda) = \gamma(t, z, \tilde{z}; \bar{\lambda}^{-1}).$$

The function $f : (t, z, \tilde{z}; \lambda) \mapsto \gamma$ is annihilated by the vectors

$$\mathbf{L} = \partial_z + \frac{i}{2}\lambda^{-1}\partial_t,$$

$$\mathbf{M} = \partial_{\tilde{z}} - \frac{i}{2}\lambda\partial_t,$$

these vectors spanning the null plane defined by (γ, λ). We may now construct the Lax pair for the chiral equation

$$\mathbf{L}\Psi = \mathbf{A}_L\Psi, \quad \mathbf{M}\Psi = \mathbf{A}_M\Psi. \tag{4.9}$$

For the purposes of this section we define

$$\mathbf{A}_L = \frac{1}{4}(\lambda^{-1} - 1)(J^{-1}\partial_{\tilde{z}}J - iJ^{-1}\partial_t J),$$

$$\mathbf{A}_M = \frac{1}{4}(\lambda - 1)(J^{-1}\partial_z J + iJ^{-1}\partial_t J). \tag{4.10}$$

The consistency condition on (4.9) is the modified chiral equation.

Our interpretation of the consistency of (4.9) is that the curvature of the connection defined by $\mathbf{A}_L$ and $\mathbf{A}_M$ vanishes when restricted to the null plane spanned by $\mathbf{L}$ and $\mathbf{M}$, so the parallel transport of the objects Ψ is path- independent on these planes. This means that we can construct a vector bundle π over the space of null planes by specifying a fibre basis over one of its points p and noting that any vector referred to this basis at p may be consistently transported to any point in the plane by virtue of the above. This fibre becomes the fibre of a vector bundle π over $T\mathbf{P}^1$, and the patching matrix $\mathbf{F}$ is determined by the parallel transporter.

We will be interested in the initial- value problem for the modified chiral equation and so the initial data surface will be taken to be $t = 0$, and the path along which we will transport Ψ will be in the line formed by the intercept of $t = 0$ with a null plane. Such a line has tangent vector $\mathbf{V} = \lambda\mathbf{L} + \lambda^{-1}\mathbf{M}$. The equation

$$\mathbf{V}\Psi = (\lambda\mathbf{A}_L + \lambda^{-1}\mathbf{A}_M)\Psi \tag{4.11}$$

holds for this line, and may be used to find an expression for $\mathbf{F}$.

The bundle over $T\mathbf{P}^1$ will be patched over the equator $\lambda = e^{-i\theta}$ for simplicity. As we move Ψ from a point p to a point q we effectively change the coordinate basis in the fibre of E_Z, where Z is the point of twistor space corresponding to the null plane containing the line joining p and q. To propagate Ψ we path integrate the connection along the line l joining these points to find, formally,

$$\mathbf{F} = P\exp(\int_l \Xi).$$

We see by direct substitution that Ξ is determined by (4.10):

$$\Xi = \frac{1}{2}(1 - \cos\theta)J_0^{-1}J_{0x} + \frac{1}{2}\sin\theta J_0^{-1}(\dot{J}_0 + J_{0y}),$$

where $(J_0, \dot{J}_0)$ is the analytic initial data, on $t = 0$.

The choice of p and q can be important, and this will be taken up later. Here we take the point p to be given by $z_p = i\gamma\lambda, \tilde{z}_p = 0$, and q by $z_q = 0, \tilde{z}_q = -i\gamma\lambda^{-1}$,

and parametrise the path by

$$z(\sigma) = z_q\sigma + z_p(1-\sigma) = -i\gamma\lambda(\sigma - 1),$$

$$\tilde{z}(\sigma) = \tilde{z}_q\sigma + \tilde{z}_p(1-\sigma) = i\gamma\sigma\lambda^{-1}. \tag{4.12}$$

With this choice

$$\partial_z = i\gamma\lambda\frac{\mathrm{d}}{\mathrm{d}\sigma},$$

$$\partial_{\tilde{z}} = i\gamma\lambda^{-1}\frac{\mathrm{d}}{\mathrm{d}\sigma},$$

and

$$\frac{\mathrm{d}}{\mathrm{d}\sigma} = -i\gamma\mathbf{V}. \tag{4.13}$$

Then we can obtain $\mathbf{F}$ from the solution G to

$$\frac{\mathrm{d}G}{\mathrm{d}\sigma} = -i\frac{\gamma}{4}\{2(1-\cos\theta)J_0^{-1}J_{0x} + 2\sin\theta J_0^{-1}(\dot{J}_0 + J_{0y})\}G, \tag{4.14}$$

where the initial data J_0 and $\dot{J}_0$ are to be specified as functions of x and y, and the derivatives should be calculated formally treating these as real variables. The initial condition is $G(0) = \mathbf{1}$.

It will not be possible to solve the system (4.14) analytically in most cases, but in principle the solution $\mathbf{F} = G(1)$ is the required patching matrix, and this $\mathbf{F}$ contains the information needed to reconstruct the field J at all future times t. In practice, we take a finite range for γ, and this restricts the range of times we can use.

The method of obtaining J is based on the condition that the restriction of the bundle π to any real section of the bundle $\mathbf{T}$, (which, we recall, is $T\mathbf{P}^1$, the holomorphic tangent bundle of $\mathbf{P}^1$) is trivial. This means that the patching matrix factorises into $\hat{H}H^{-1}$ with $\hat{H}$ holomorphic in $\hat{U}$, and H in U, where

$$\hat{U} = \{\lambda : |\lambda| > 1 - \epsilon\},$$

$$U = \{\lambda : |\lambda| < 1 + \epsilon\}. \tag{4.15}$$

The small positive real number ϵ will not be crucial here, as we will be patching over the equator. The reconstruction of J from H and $\hat{H}$ involves consideration of extended solutions to the Lax pair, and will be presented in a later section.

4.4 The Numerical Procedure.

4.4.1 An Outline.

The above outline has been adapted to a numerical procedure for solving the initial-value problem. The overall method is the same: construct Ξ from J_0, $\dot{J}_0$, choose endpoints p and q, integrate numerically to obtain $\mathbf{F}$, and finally split $\mathbf{F}$ on real sections to obtain H and $\hat{H}$ and thus J. In fact there are a number of difficulties in this implementation: these will be discussed below as the details are presented.

4.4.2 From Initial Data to the Twistor Matrix.

Given the initial data as functions of x and y, the first step is to perform the necessary differentiations, treating the variables as real. Since the resulting matrices are to be manipulated to find Ξ, it is most appropriate to use a symbolic computing language to work through the calculations. After some experimentation, REDUCE [44] was chosen as the most suitable language for this work, and a program was developed.

A problem which may affect the next stage, the integration, is that Ξ can have singularities on the path between p and q, and the numerical integration will be unreliable whenever this occurs. Such cases have been investigated and the results have been unsatisfactory. If poles are present it is sometimes possible to pick p and q and/or parameters occurring in the initial data so that the path avoids these poles, and examples of these strategies will be given explicitly later.

Once a suitable path has been found, a FORTRAN program, based on the REDUCE output, is used to perform the integration. A NAG [36] routine forms the core of the program, the output of which is the matrix $\mathbf{F}$. Since $\mathbf{F}$ is a function of γ and θ (recall that we need only consider $\lambda = \exp(-i\theta)$) the integration is carried out at each point of a grid of (γ, θ) points. The x- and y- dependence of Ξ is accommodated by expressing (x, y) in terms of (γ, θ), using the expressions for the points p and q. This substitution is made at the REDUCE stage as it involves some error-prone algebra. The number of points in the grid may be quite small, and in the cases examined to date has not exceeded 23×23. For reasons to be

clarified later, the number of θ points is taken to be an odd number.

4.4.3 Splitting the Twistor Matrix.

The next stage in the procedure is to evaluate $\mathbf{F}$ on real sections. This is accomplished in practice by replacing γ by its expression in terms of x, y, t and θ;

$$\gamma = 2(t + x\sin\theta + y\cos\theta). \tag{4.16}$$

However, since we only know $\mathbf{F}$ at points on the (γ, θ) grid we have to ensure that our choice of (t, x, y) leads to a (γ, θ) for which we know $\mathbf{F}$. In practice this has meant interpolating $\mathbf{F}$ in γ, and substituting $(t, x, y; \theta)$ into (4.16). t is fixed for the entire splitting procedure, and the range of γ then dictates the ranges of x and y: this is built into the program on the assumption that a square grid of (x, y) points is required, although this is not an essential restriction.

The interpolation in γ may introduce errors into the procedure in several ways: the interpolating functions are taken to be polynomials, and this form will not always reflect the behaviour of the components of $\mathbf{F}$; and the choice of degree of these polynomials may introduce fluctuations in the values between data points. The degree of each element may be chosen by the user, and some experimentation may be needed to minimise the errors described above. In general the degrees should be kept fairly low where feasible.

We then have $\mathbf{F}(\lambda)$ at each (t, x, y) point, and at each such point we must perform the splitting described above. Once again we know $\mathbf{F}$ only on a discrete grid of θ points, so the analytic splitting methods cannot be directly applied.

At the early stages of development attempts were made to solve a matrix Fredholm integral equation to recover $H(\lambda)$ [2]. However, this involved interpolation in θ, across a singular point, and was found to be unreliable. The basis of this approach is the analytic Riemann problem with zeros, which leads to

$$\hat{H}(\lambda) - \int \frac{(F(\tau)^{-1}F(\lambda) - \mathbf{1})\hat{H}(\tau)}{2\pi i(\tau - \lambda)} \mathrm{d}\tau = 1. \tag{4.17}$$

The kernel of the discrete version is ill- defined for certain points, and although it is possible to try to interpolate the kernel through these points the results are disappointing.

A second procedure which meets with some success involves taking the nonlinear system with data **F** and solving it directly, using existing routines. The idea of expressing **F** in terms of a Fourier series is used to generate the nonlinear system. We take the number of θ points in our grid to be $2N+1$, and consider the set of functions $\{e_l\}$ of the discrete variable k, where k runs from 1 to $2N+1$, and

$$e_l(k) = \Omega^{lk}, \ \Omega = \exp(2\pi i/(2N+1)).$$

These functions form an orthonormal set with respect to the inner product

$$\langle f, g\rangle \equiv \frac{1}{N}\sum_k f(k)\bar{g(k)}, \tag{4.18}$$

by which we mean that

$$\langle e_a, e_b\rangle = \delta_{ab}.$$

Then we can analyse a given function f into its components in the e basis: for example

$$F(k) = \sum_{a=-N}^{N} f_a e_a(k),$$

where $f_a = \langle F, e_a\rangle$, and we use lower case letters as kernels of the expansion coefficients. The condition that H be holomorphic in U and $\hat{H}$ in $\hat{U}$ implies that we can make the expansions

$$\hat{H} = \sum_{a=0}^{N} \hat{h}_a e_{-a}, \ (N+1 \text{ unknowns}), \tag{4.19}$$

$$H^{-1} = \sum_{a=0}^{N} h_a e_a, \ (N+1 \text{ unknowns}). \tag{4.20}$$

Then, from the expansion of F and the explicit expansion of $\hat{H}H^{-1}$ we have a set of coupled quadratic matrix equations. To these we add the (arbitrary 'gauge') condition $\hat{h}_0 = 1$ to define our full nonlinear system

$$f_a = \sum_{b+c=a} \hat{h}_b h_c. \tag{4.21}$$

Standard numerical methods may be employed to solve this system for the remaining $2N + 1$ matrix unknowns.

Appendix H contains code for realising this approach. A serious drawback for more involved data is the long running time of the program. This means that repeat runs involving different parameters becomes time-consuming. The nonlinear nature of the splitting problem also occasionally causes the method to fail altogether, and it is to be expected that spurious solutions could be found in some cases.

The most successful method, and the one finally adopted, exploits the idea that the conditions on $\mathbf{F}$, H, and $\hat{H}$ may be expressed in terms of a system of linear equations at each $(x.y)$ point. The linearisation allows the use of fast and efficient methods in the splitting problem, and this in turn means that many cases can be tested in an acceptable time. Once again the idea of expressing the components of the splitting problem in terms of discrete Fourier series is central to the method.

Writing $H_k = H(\theta_k)$ and so on, we have

$$F_k = \hat{H}_k H_k^{-1}.$$

We simply rearrange to get

$$\hat{H}_k = F_k H_k. \tag{4.22}$$

Although this may not seem like a great simplification, it allows us to develop a linear system for the H_k and $\tilde{H}_k$. The other necessary information is the set of holomorphicity conditions: here we again assume that $F, H, \hat{H}$ may be expanded as Fourier series in θ, and that the number of θ points is $2N + 1$ for some N.

Recalling that

$$\Omega = \exp(2\pi i/(2N + 1)),$$

we demand that

$$\sum_{k=1}^{2N+1} \omega^{-nk} \hat{H}_k = 0, \ n = 1, ..., N,$$

$$\sum_{k=1}^{2N+1} \omega^{nk} H_k = 0, \ n = 1, ..., N. \tag{4.23}$$

These expressions are the discrete equivalent of the holomorphicity conditions on $\hat{H}$ and H, and are an alternative expression of the assumption that the expansions for $\hat{H}$ and H^{-1} are as in (4.19) and (4.20). The Fourier expansions are constrained by these $2N$ conditions. A further $2N+1$ conditions are provided by

$$\hat{H}_k = F_k H_k, \ k = 1, ..., 2N+1.$$

In order to avoid confusion is important to understand that these F_k are the values of F at θ_k and are not the components f_a. Explicitly,

$$f_a = \langle F, e_a \rangle = \frac{1}{N} \sum_k F_k \bar{e}_a(k).$$

Since there are precisely $4N+2$ components ($2N+1$ each of $\hat{H}$ and H) we must impose another arbitrary condition, or gauge choice, just as we did above. This is allowed because the twistor matrix determines only a gauge equivalence class of solutions, and the choice of $\hat{H}_0$ just fixes the particular gauge. We take the gauge condition simply to be $\hat{H}_0 = \mathbf{1}$. The resulting system of $4N+2$ complex 2×2 matrix equations may, of course, be expressed as a system of $4 \times (4N+2)$ complex linear equations. Efficient routines for the solution of this kind of system exist, and may be used to obtain $\hat{H}_k$ and H_k for each k. The particular routines used were LAPACK routines from the NAG library, and these are listed in the code in the appendices.

4.4.4 Reconstructing J.

To reconstruct J we need H at $\theta = \pi$, as we will see below, and since this was not one of our grid points we must use the Fourier analysis to obtain the value at this point.

Even now, however, the resulting J is not the required solution due to our choice of p and q. The problem is related to the fact that that $\mathbf{F}(-1)$ is not necessarily the unit matrix. The true solution may be obtained from J by premultiplying by a correcting factor obtained by the following analysis.

The linear problem for Ψ admits an extended solution χ depending on $(t, z, \tilde{z}; \lambda)$

and satisfying

$$\chi(1) = \mathbf{1}, \quad \chi(-1) = J, \tag{4.24}$$

$$\nabla\chi = -\chi A,$$

$$\tilde{\nabla}\chi = -\chi\tilde{A}.$$

We note that

$$\nabla(\chi\Psi) = 0, \text{ so } \Psi = \chi^{-1}\psi,$$

where ψ is constant. Now consider $\Psi = \chi^{-1}\psi$. We find that

$$\nabla\Psi = \nabla\chi^{-1}\psi = -\chi^{-1}(\nabla\chi)\chi^{-1}\psi = A\chi^{-1}\psi = A\Psi,$$

and similarly for $\tilde{\nabla}\Psi$. We have the following expressions:

$$\mathbf{F} = \chi_q^{-1}\chi_p,$$

$$H = \chi_p^{-1}\chi, \quad \hat{H} = \chi_q^{-1}\chi,$$

where χ_p indicates that the expressions for z_p ans $\tilde{z}_p$ are substituted into χ, and so on. The expressions for H and $\hat{H}$ are, of course, not unique, but the expression

$$H(-1)\hat{H}^{-1}(1) = \chi_p^{-1}(-1)\chi(-1)\chi(1)^{-1}\chi_q(1)$$

is gauge invariant. This expression is seen, using the properties (4.24) of χ, to be

$$H(-1)\hat{H}^{-1}(1) = J_p^{-1}J,$$

so that, using our gauge choice $\hat{H}(1) = \mathbf{1}$,

$$J = J_p H(-1).$$

To evaluate J_p, we substitute our expressions for z_p and $\tilde{z}_p$, together with $\lambda = -1$, into J_0. For the choice above, $z = i\gamma\lambda, \tilde{z} = 0$, we would substitute $z = 2i(y-t), \tilde{z} = 0$ into J_0.

This correction is effected by the program in Appendix F, where any consistent choice of p and q is automatically accommodated.

4.5 Examples and Observations.

4.5.1 Static Initial Data.

The above procedure has been tested on a number of examples: a static 1-lump solution at the origin; static charge 2 and charge 3 rings; moving 1-lump solutions; and perturbed 1-lump configurations. The static cases gave good results for small times, when the field was reconstructed from the twistor data. Figure 1a is a picture of the energy density for the static lump solution, and shows why the solution is so called.

The initial data for this problem is given by taking $f = z$ in

$$J[f] = \frac{i}{1+f\tilde{f}} \begin{pmatrix} 1-f\tilde{f} & 2f \\ 2\tilde{f} & f\tilde{f}-1 \end{pmatrix},$$

$$J_0 = J[f], \quad \dot{J}_0 = 0. \tag{4.25}$$

At small times the procedure gives good results, but as t is increased the final output deteriorates. It seems that the problem arises at the splitting stage, since checks on the numerical twistor matrix show it to be very close to the analytic version in this case.

4.5.2 Nonstatic Initial Data: a Constraint.

The restriction to static solutions is not essential, and we may choose moving initial configurations by letting $\dot{J}_0$ be a nonvanishing function of x and y. In allowing this, however, we need to take account of the behaviour of the resulting integrand in (4.14). For most static configurations the choice (4.12) is adequate, being well-behaved for $\sigma \in [0,1]$, but for a moving 1-lump the following analysis shows that we can expect singularities along the path of integration. It also shows that in general the choice of $(\tilde{z}, z)$ will not be obvious or easily derived from the initial data.

The initial data for the moving 1-lump is given by setting

$$f = x + i\beta(y + vt)$$

in $J[f]$, putting $t = 0$ for J_0 and

$$\dot{J}_0 = \partial_t J[f]\Big|_{t=0}.$$

We see that for a $(\tilde{z}, z)$ of the form

$$z = i\gamma\lambda(1-\sigma),\ \tilde{z} = -i\gamma\lambda^{-1}\sigma,$$

there will be a factor

$$\begin{aligned}\zeta &= 1 + x^2 + \beta^2 y^2 \\ &= 1 + (x^2 + y^2) + (\beta^2 - 1)y^2 \\ &= 1 + z\tilde{z} + (\beta^2 - 1)(\frac{1}{2i}(z - \tilde{z}))^2 \\ &= 1 + \frac{1}{2}(1 + \beta^2)z\tilde{z} - \frac{1}{4}(\beta^2 - 1)(z^2 + \tilde{z}^2)\end{aligned}$$

in the denominator of Ξ, where $\beta = (1 - v^2)^{-1/2}$, and v is identified with the speed of the lump, which is moving along the $y-$axis. In terms of σ this is

$$1 + \frac{1}{2}(\beta^2 + 1)\gamma^2\sigma(1-\sigma) + \frac{\gamma^2}{4}(\beta^2 - 1)((1 - 2\sigma + 2\sigma^2)\cos 2\theta - (1 - 2\sigma)i\sin 2\theta).$$

We must try to locate zeros of this polynomial lying on the integration path, that is, values of σ in $[0, 1]$ at which the expression vanishes, and find values of the parameters which avoid these zeros. The imaginary part of ζ is proportional to

$$(1 - 2\sigma)\sin 2\theta,$$

which vanishes when $\sigma = 1/2$ or $\sin 2\theta = 0$. If $\sigma = 1/2$ the real part of ζ is positive on the path of integration, so we consider the case $\theta = 0, \text{or } 3\pi/2$.

In this case we see that the speed parameter v is constrained. This follows from an examination of the real part of ζ:

$$\begin{aligned}1 + \frac{1}{2}(\beta^2 - 1)\gamma^2\sigma(1-\sigma) - \frac{\gamma^2}{4}(\beta^2 - 1)(1 - 2\sigma + 2\sigma^2) \\ = 1 + \frac{\gamma^2}{4} - \frac{\beta^2\gamma^2}{4}(1 - 2\sigma)^2.\end{aligned}$$

This vanishes when

$$\beta^2 = \frac{(4 + \gamma^2)}{\gamma^2(1 - 2\sigma)^2},$$

but since $(1-2\sigma)^2$ is greater than zero (recalling that $\sigma \neq 1/2$) and always less than or equal to 1, the real part of ζ can only vanish for

$$\beta^2 > \frac{4+\gamma^2}{\gamma^2}.$$

In terms of v this means that in order to ensure that ζ is never zero we must have

$$v < \sqrt{\frac{4}{4+\gamma^2}}. \tag{4.26}$$

For example, if the maximum value of γ is 16, then v must be less than about 0.12. The numerical procedure does indeed work for initial data conforming to this constraint, and for larger values of v the presence of the poles has been observed to lead to failure.

4.5.3 Avoiding the Constraint.

An alternative strategy is to decide upon the parameters in the initial conditions and try to find endpoints which do not lead to poles on the path of integration. We will again look at the case of a single moving lump with initial speed v along the y-axis. A plausible strategy is to choose the endpoints so that $\zeta = 1$ at both. Then we just need to check that $\zeta > 0$ along the whole path.

Using

$$\zeta = 1 + x^2 + \beta^2 y^2 = 1 + w\tilde{w},$$

where we have introduced $w = x + i\beta y$ and $\tilde{w} = x - i\beta y$, we find

$$x = \frac{1}{2}(w + \tilde{w}) = \frac{1}{2}(z + \tilde{z}),$$

$$y = \frac{1}{2i\beta}(w - \tilde{w}) = \frac{1}{2i}(z - \tilde{z}),$$

from which we deduce the linear change of variables

$$\begin{pmatrix} w \\ \tilde{w} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} (1+\beta) & (1-\beta) \\ -(1-\beta) & (1+\beta) \end{pmatrix}\begin{pmatrix} z \\ \tilde{z} \end{pmatrix}. \tag{4.27}$$

The inverse of the matrix on the right is

$$\frac{1}{2\beta}\begin{pmatrix} (1+\beta) & -(1-\beta) \\ -(1-\beta) & (1+\beta) \end{pmatrix}$$

so that, on the plane t=0

$$\gamma = i\lambda\tilde{z} - i\lambda^{-1}z = \frac{i}{2\beta}\left([\lambda(\beta-1) - \lambda^{-1}(\beta+1)]w + [\lambda(\beta+1) - \lambda^{-1}(\beta-1)]\tilde{w}\right).$$

If we choose at p, say $\tilde{w} = 0$ so that $\zeta = 1$ we then have

$$w_p = \frac{2i\beta\gamma\lambda}{(\beta+1) - \lambda^2(\beta-1)},$$

and, taking $w_q = 0$ we find

$$\tilde{w}_q = \frac{-2i\beta\gamma\lambda^{-1}}{(\beta+1) - \lambda^{-2}(\beta-1)}.$$

These choices guarantee that $\zeta = 1$ at p and q and so we have some hope that ζ will be nonzero all along the line. However, this needs to be checked.

Let us examine the behaviour of $w\tilde{w}$ along the path of integration. We have

$$w(\sigma) = (1-\sigma)\frac{2i\beta\gamma\lambda}{(\beta+1) - \lambda^2(\beta-1)},$$

$$\tilde{w}(\sigma) = \sigma\frac{-2i\beta\gamma\lambda^{-1}}{(\beta+1) - \lambda^{-2}(\beta-1)},$$

and

$$w\tilde{w} = \sigma(1-\sigma)\frac{4\beta^2\gamma^2}{2(\beta^2+1) - (\lambda^2+\lambda^{-2})(\beta^2-1)},$$

which becomes, using $\lambda = \exp(-i\theta)$,

$$w\tilde{w} = \sigma(1-\sigma)\frac{4\beta^2\gamma^2}{2[\beta^2(1-\cos 2\theta) + (1+\cos 2\theta)]}.$$

The expression multiplying $\sigma(1-\sigma)$ is manifestly non-negative, and so ζ never vanishes for σ on the interval $(0,1)$, independently of β, (which is always less than 1.) Thus restrictions such as (4.26) on the initial data are not needed. Of course, the expression (4.13) will be modified by the change of endpoints, but this can be written into the codes.

Whilst examination of the denominator of Ξ may lead to 'good' choices for $(\tilde{z}, z)_{p,q}$, the analysis is not straightforward, and not guaranteed to lead to a useful answer. It may be that a different approach is needed at this stage if the method is to be applied more widely.

4.5.4 Perturbation of the Static Lump.

In addition to the moving lump and various static configurations, including the static charge 2 ring of Fig 4.2, one can also consider initial data representing perturbations of static configurations which do not correspond to solutions known in analytic form. One such set of data is a perturbation of the width of a 1-lump configuration, for which we take

$$\dot{J}_0 = \frac{\epsilon}{1+r^2}(zJ_{0,z} + \tilde{z}J_{0,\tilde{z}}),$$

and J_0 to be the static 1-lump solution. The numerical evolution of this data indicates that the lump vibrates, radiating energy, but remains lump-like and stays in the same place. This type of behaviour is encouraging evidence of the stability of the lump. The evolution of the initial configuration is illustrated in Fig 4.3, where the difference between the potential energies of the unperturbed and perturbed lumps is plotted for various times, in arbitrary units.

4.6 Description of the Code.

The appendices contain the code used in the numerical experiments, annotated for ease of reference. However, it will be useful to present an overview of its construction in order to point out some of the less obvious features.

4.6.1 The REDUCE Code.

Firstly, the code for generating the both the twistor matrix $\mathbf{F}$ and the correcting factor is all contained in the REDUCE program in Appendix C. All the FORTRAN code for the construction of the twistor matrix is written into this program, and most of the rest of the REDUCE part is quite straightforward algebraic manipulation to obtain the constituent parts of Ξ. One aspect which may at first appear confusing is the use of \$ and ; as terminators for each line. The former indicates that the code in the line will be evaluated but not written out, whilst the latter symbol causes the evaluated code to be written out. For example, the line

$$\mathsf{x := xp * (1 - sig) + xph * sig;}$$

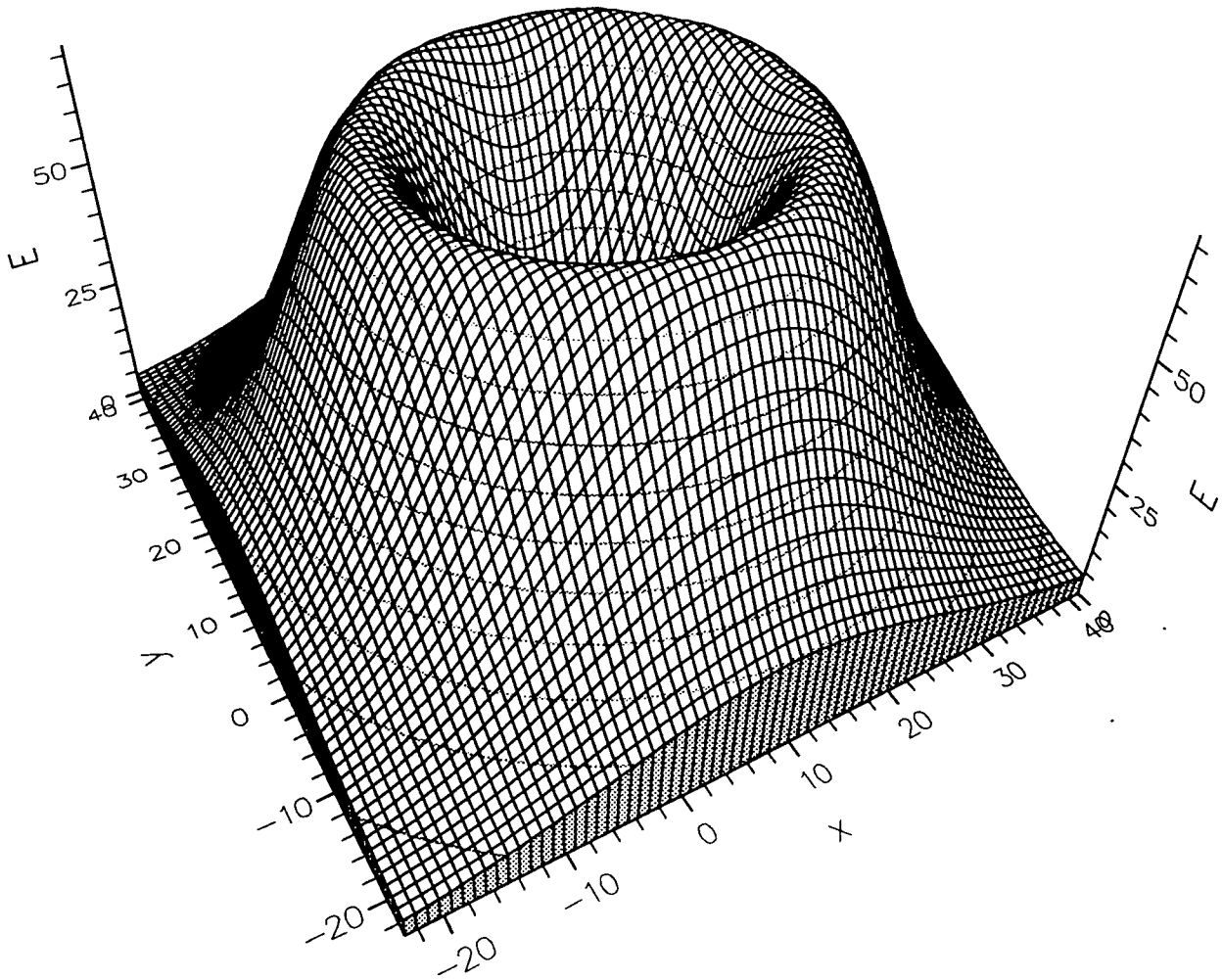

Figure 4.2: Static charge 2 ring.

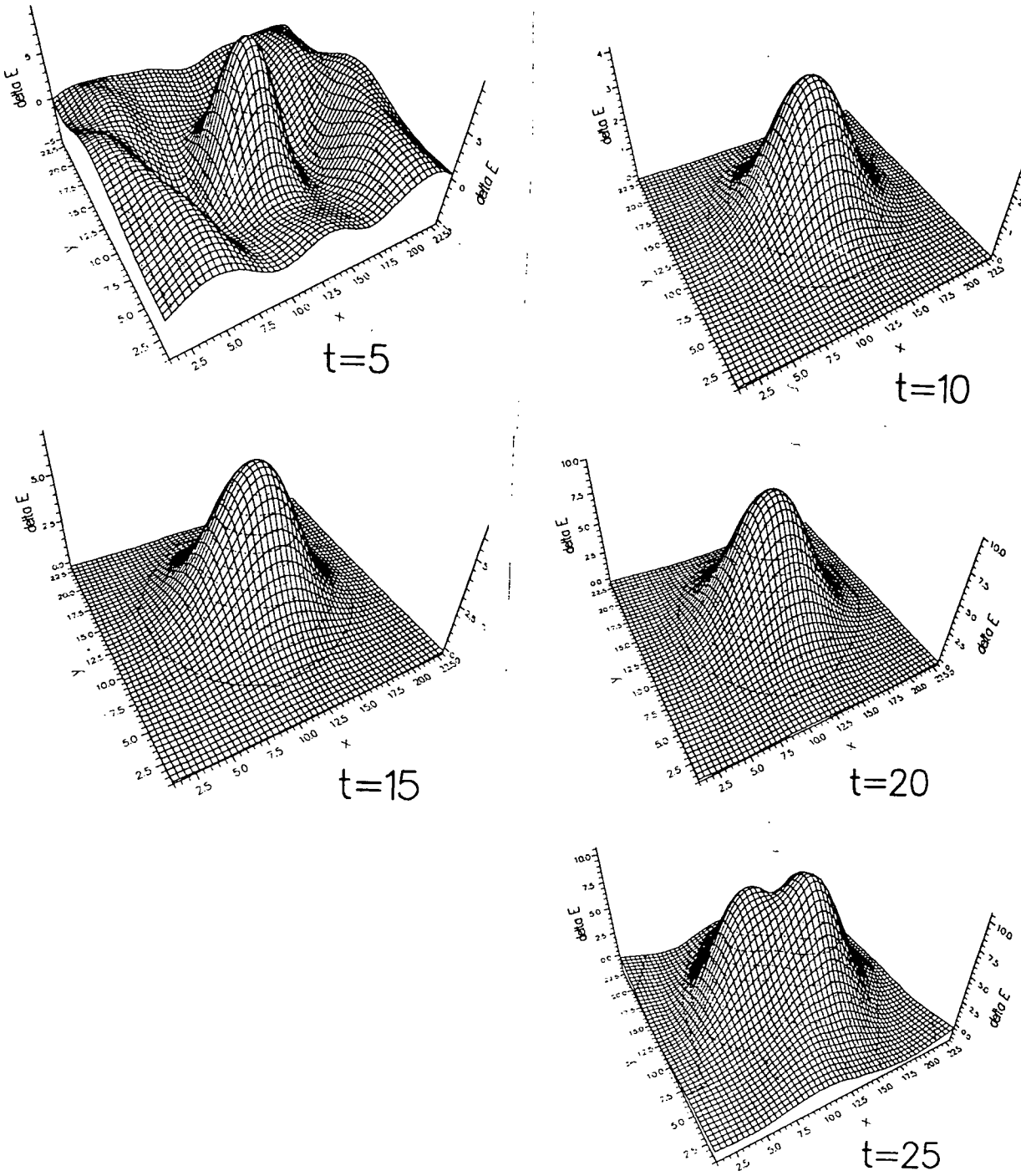

Figure 4.3: Evolution of Perturbed Lump

will result in x being evaluated to (the value of) the r.h.s., and being printed out into to body of the FORTRAN program in the form

$$X = \text{an expression in } \gamma,\ \theta, \text{ and parameters.}$$

An example of the output may be seen in the sample listing.

A second point is that, once expressions for x and y have been written out, we no longer want these variables to be bound to the expressions, as this would give rise to an error when we tried to perform the formal differentiations to find $\partial_x J_0$ and so on. That is the reason for the line

```
clear x, y$
```

Whilst REDUCE is appropriate for the calculation of the derivatives of J_0 and the inversion of J_0, the substitution of the values of x and y would generally result in a huge expression for Ξ, and it is more efficient (and usually unavoidable) to leave most of these manipulations for the FORTRAN stage. In addition, the variable I must be defined to be $\sqrt{(-1)}$ at the FORTRAN stage as it is built into REDUCE but would be taken to be an (undefined) integer by FORTRAN, and some editing of the FORTRAN code is necessary to ensure that intrinsic functions such as SIN are taken to be double precision (DSIN). It is also useful to allow REDUCE to take the expressions for the endpoints p and q at this stage and construct the correcting matrix, putting it into the body of a second FORTRAN program. The manipulation of the expressions for p and q allows quite general choices to be made.

4.6.2 The FORTRAN Codes.

The FORTRAN programs themselves are fairly straightforward, using double precision and double complex variables and parameters in order to achieve compatibility with the implementation of the NAG library at Durham. One point to notice is that the arguments GA and TH may be passed to the routine D02BAF as extra components of Y(10). This avoids the need to modify the NAG routine to accept two more parameters. The routine chosen here uses a backward difference formula method, which is useful for problems which are potentially stiff. The error analysis

for this method is not straightforward [36], but it has been possible to make direct comparison with analytic expressions in some cases, and the agreement has been very good.

The numbering of the array entries is influenced by the fact that some of the core routines work only with real arrays, and for these the odd-numbered components are the real parts, the even imaginary. In fact, each matrix is expressed as a 1-dimensional array in parts of the code. In some parts of the code this rearrangement of the matrix elements needs careful tracking. For example, the numbering of the elements of a 1-dimensional version $M(r)$ of the 2×2 matrix F_k^j would follow the pattern

$$M(2(k-1)+m) = F_m^k.$$

If the elements of M are required to be real we imagine an extra index on F taking the value 1 for real parts, 2 for imaginary parts, and put the elements into an 8-component vector $\hat{M}$, so that

$$\hat{M}(4(k-1)+2(m-1)+1) = \mathrm{Re}F_m^k = F_{m,1}^k$$

$$\hat{M}(4(k-1)+2(m-1)+2) = \mathrm{Im}F_m^k = F_{m,2}^k.$$

The linear splitting routine is presented in the second appendix, the most important feature being the choice by the user of the grid dimensions. It is clearly critical that this agrees with the choice in the other routines. Whilst it would be possible to make the choice automatic, the modular construction of the code makes it quite sensible for the user to do it by hand. The subroutine FINDZA allows the user to choose the degrees of the interpolating polynomials for the components of the matrix $\mathbf{F}$ and, as mentioned above, some experimentation with these degrees may be needed.

The size of the area in the (x, y) plane covered by the splitting procedure is determined by the final time t and the range of γ used initially: the parameter GTOP appearing in some of the codes accommodates the choice of t. The range of γ, as seen above, may affect the choice of parameters in the initial data (or *vice versa*) due to the possible occurrence of singularities.

The splitting problem itself is manifested as a problem in linear algebra: at each (x, y, t) point a (large) system of linear equations must be solved. In the code the 2 by 2 matrix F is incorporated into the matrix linear system simply by using a $2N \times 2N$ matrix, and solving this. It is possible that one could make some economies here by exploiting the properties of F, but this has not been pursued.

In the visualisation of the solutions the quantity used is related to the energy density of the solution, which is calculated from the space derivatives of J. The code in appendix B was used to perform the numerical calculations necessary to obtain the figures for this chapter.

4.7 Conclusions.

Undoubtedly, the code presented here is not the last word in the solution of this type of initial value problem by the twistor method: some of its shortcomings have been alluded to above: the construction of the twistor matrix may be affected by the presence of poles on the integration path as a consequence of the use of complex data, necessary in this approach, but possibly avoidable by the use of an alternative method; the interpolation of F may lead to inaccuracies; and there is some difficulty in taking large times.

The splitting procedure is in principle efficient for suitably well-behaved $\mathbf{F}$, although there are some difficulties when large values of γ are used, which is to say, when large times are needed. Some investigation of the errors arising in various stages of the method is an obvious next step.

However, the principles of the new method, exploiting the integrability of the field equation in constructing the solution, avoid some of the problems inherent in the application of standard techniques which destroy the integrability. The method presented therefore merits further attention and development.

Chapter 5

Darboux Transformations and factorisation.

5.1 Introduction

This chapter introduces the idea of a Darboux transformation by way of a factorisation problem for the Schrödinger equation with a potential obtained from a simple N-soliton solution to the KdV equation. The result is generalised and related to other systems.

5.1.1 A Simple Example.

The fundamental example of the Darboux transformation is in the study of solutions to the Schrödinger equation

$$-\psi_{xx} + u\psi = \lambda\psi. \tag{5.1}$$

It is easy to confirm that the transformed field $\psi[1]$ satisfies (5.1) with the new potential $u[1]$ and the same eigenvalue λ when

$$\psi[1] = \psi_x - \frac{\phi_x}{\phi}\psi,$$

$$u[1] = u - 2(\ln\phi)_{xx},$$

and ϕ is a solution of (5.1).

The process may be iterated to obtain towers of solutions, and more importantly for us, towers of new potentials $u[k]$ [32]. In the next section we will examine a scheme for generating such new potentials by factorising a Hamiltonian.

5.2 The Factorisation Method.

5.2.1 A Lax Pair for the KdV Equation.

The Korteweg-de Vries equation has been extensively studied as rich source of information about integrable systems, and has become a model testing ground for various methods and ideas. The property of particular interest here is the fact that the equation may be expressed as the consistency condition for a Lax pair of linear equations:

$$L\psi = \lambda\psi, \quad L = -\partial_x^2 + u, \tag{5.2}$$

$$A\psi = \partial_t\psi, \quad A = -\partial_x^3 + 3u\partial_x + 6u_x, \tag{5.3}$$

where u must be a solution to the KdV equation. This is dictated by the consistency condition

$$L_t + [L, A] = 0.$$

The equation (5.2) is a Schrödinger equation for ψ, and (5.3) will be treated as an auxilliary condition. The appearance of the Schrödinger equation allows us to examine the nature of the potentials u using well established methods: the existence of bound states, the scattering properties, all lead to information about u. This is the basis of the inverse scattering transform (IST), one of the earliest methods to be applied to the study of integrable systems.

In the next section we will choose those u corresponding to certain N-soliton solutions, which are in fact reflectionless potentials, and examine the properties of the Shrödinger equation, in particular the possibility of relating its factorisation to the Darboux transform for u.

5.2.2 The N-soliton Case.

In this section we will use as a potential in (5.2) a simple N-soliton solution to the KdV equation at time $t = 0$, and ignoring the phase. The problem

$$\mathrm{H}_N|\psi\rangle = \lambda|\psi\rangle$$

may then be written explicitly as

$$\partial_x^2\psi + (\lambda + N(N+1)\mathrm{sech}^2(x))\psi = 0,$$

which can be transformed into the defining equation for the associated Legendre functions P_N^m [14] by the substitution T=tanh x. A convenient representation of these functions is

$$\mathrm{P}_N^m = (-1)^m(1-T^2)^{m/2}(\frac{\mathrm{d}}{\mathrm{d}T})^m\mathrm{P}_N \tag{5.4}$$

where

$$\mathrm{P}_N(T) = \frac{1}{2^N N!}(\frac{\mathrm{d}}{\mathrm{dT}})^N(T^2-1)^N.$$

Clearly

$$\mathrm{P}_N^N = (\frac{-1}{2})^N(\frac{(2N)!}{N!})\mathrm{sech}^N(x), \tag{5.5}$$

so that, for example,

$$\mathrm{P}_2^2 = 3\mathrm{sech}^2 x.$$

This simply means that the problem

$$\mathrm{H}_N\psi = -m^2\psi$$

has solutions

$$\psi_m = \alpha\mathrm{P}_N^m(\mathrm{tanh}x)$$

which will be denoted $|\psi_N^m\rangle$. Here α is a normalisation constant which will not be of importance in what follows.

5.2.3 Highest-weight States.

We now show that it is possible to generate highest weight states, that is, the states $|\psi_N^N\rangle$, for the Hamiltonian H_N by a simple factorisation method. We begin by considering the state $|\psi_N^N\rangle$ for which

$$(\mathrm{H}_N - N^2)|\psi_N^N\rangle = 0.$$

The operator on the left can now be factorised into $B_N^- B_N^+$ where

$$B_N^{\pm} = \pm\partial_x + N\tanh x. \tag{5.6}$$

Then we have

$$B_N^-(B_N^+|\psi_N^N\rangle) = 0,$$

and we can use the explicit forms of $B_N^{\pm}$ and the integrating factor method to solve this explicitly for $|\psi_N^N\rangle$ as a function of x. The expressions for $B_N^{\pm}$ in this case are easy to work with, and we can quickly check that (5.5) are indeed proportional to the ψ_N^N a obtained by factorisation.

We have

$$B^-(f) = 0$$

for some $f = B^+\psi$. We use (5.6) to expand this expression and find

$$\partial_x f = (N\tanh x)f.$$

The integrating factor we need is

$$\exp\{-\int N\tanh x dx\} = \mathrm{sech}^N x,$$

which leads us to

$$f = A\cosh^N x.$$

Now, in the same way, we solve $B^+ g = f$. The new integrating factor is $\cosh^N x$, and we find

$$|\psi_N^N\rangle = \alpha\mathrm{sech}^N x$$

in agreement with our earlier observation.

5.2.4 Lower Weight States.

The next problem is to look for the states of lower weight. It is natural to make use of the factors of the Hamiltonian, and the fundamental relation

$$[B_N^-, B_N^+] = -2N\mathrm{sech}^2 x, \tag{5.7}$$

which may be verified by direct calculation.

Now suppose that $|\psi\rangle$ satisfies

$$\mathrm{H}_N|\psi\rangle = \lambda|\psi\rangle.$$

Then again using the notation $B_N^{\pm}$ to denote the factors of H_N we have

$$\mathrm{H}_N B_N^+|\psi\rangle = [\mathrm{H}_N, B_N^+]|\psi\rangle + \lambda B^+|\psi\rangle.$$

But it is easy to see that

$$[\mathrm{H}_N, B_N^+] = [B_N^-, B_N^+]B_N^+,$$

from which we find

$$\mathrm{H}_N B_N^+|\psi\rangle = (\lambda - 2N\mathrm{sech}^2 x)B_N^+|\psi\rangle.$$

This, however, is readily seen to be the condition

$$\mathrm{H}_{N-1} B_N^+|\psi\rangle = \lambda B_N^+|\psi\rangle,$$

which means that $B_N^+|\psi\rangle$ is an eigenstate of H_{N-1} belonging to the eigenvalue λ. It may be shown that λ is minus the square of an integer m, where $m = 1, 2, ...N$, and m is the weight of the state.

Thus, starting with the hypothesis that, at level N, there exists an eigenstate with eigenvalue λ, we see that at level $N-1$ there is also such an eigenstate obtained by applying B_N^+ to the state at level N. However, our intention was to find all states starting from those of highest weight: it is readily shown that B_N^+ annihilates such states, so the method must be extended.

The key to the extension of the above method lies again in the factorisation of H_N, for, given that $|\psi_{N-1}^m\rangle$ was obtained as $B_N^+|\psi_N^m\rangle$ we have

$$B_N^-|\psi_{N-1}^m\rangle = B_N^- B_N^+|\psi_N^m\rangle = (\mathrm{H}_N + N^2)|\psi_N^m\rangle = (N^2 - m^2)|\psi_N^m\rangle,$$

showing that a state of weight m at level N is generated by operating with B_N^- on the corresponding state at level $N-1$. In particular, if $m = N-1$ we can use our knowledge of the highest-weight state at level $N-1$ to build the weight $N-1$

state at level N. This process may clearly be used iteratively to generate a state of any weight less than N at level N. The process will not terminate, a point to which we will return.

As an example,

$$|\psi_1^1\rangle = A\text{sech}x,$$

$$B_2^- = -\partial_x + 2\text{tanh}x,$$

and

$$B_2^-|\psi_1^1\rangle = 3A\text{tanh}x\text{sech}x$$

which can be shown to be a $\lambda = -1$ eigenstate of H_2.

In general, for $|m| < N$ the mth state at level N is otained from the state $|\psi_m^m\rangle$ by the successive application of B_k^-: explicitly

$$|\psi_N^m\rangle = B_N^- B_{N-1}^- ... B_{m+1}^-|\psi_m^m\rangle,$$

up to normalisation. It is evident that we may continue with this chain of operations indefinitely, in contrast to the usual case of the angular momentum operators in quantum mechanics.

To conclude this subsection, let us examine the algebra generated by the B_N's for the $N(N+1)\text{sech}^2 x$ potential. To this end we note that $B_N^\pm = \pm\partial + NT$, using $T = \text{tanh}x$ and dropping the subscript x on the derivative. Now,

$$\partial T = (1+T)(1-T),$$

so all derivatives of T may be expressed entirely in terms of polynomials in T. Thus the only operators appearing in the full algebra are combinations of powers of ∂ and of T.

Then we have

$$[B_N^\pm, T] = \mp(T+1)(T-1),$$

and

$$[B_N^\mp, T^m] = \pm mT^{m-1}(T+1)(T-1) = \pm m(T^{m+1} - T^{m-1}),$$

and other relations can be worked out to develop the full algebra. There is an interesting relationship between this kind of factoristion method and supersymmetric quantum mechanics, which is outlined in [32] and [5].

It is clear that this scheme has wider applicability, in particular to factorisable Hamiltonians for which the $B\pm$'s enjoy a commutation relation which links H_N's for different N. Behind the specifics of this example is a general form for Darboux transformations which we will examine next.

5.3 A General Form for the Darboux Transformation.

The Schrödinger equation

$$\mathbf{H}|\psi\rangle = \lambda|\psi\rangle$$

is only exactly solvable for relatively few special potentials U, In these cases the method of Darboux transformations may be applied to known solutions to generate further solutions. Here we examine conditions for the existence of Darboux transformations of a particular type.

From $|\psi\rangle$ we wish to generate $\mathcal{O}|\psi\rangle$ such that

$$\mathbf{H}\mathcal{O}|\psi\rangle = \Theta\mathcal{O}|\psi\rangle.$$

Let $\Theta = \lambda + W(x)$ so that

$$\Theta\mathcal{O}|\psi\rangle = \lambda\mathcal{O}|\psi\rangle + W\mathcal{O}|\psi\rangle = \mathcal{O}\mathbf{H}|\psi\rangle + W\mathcal{O}|\psi\rangle.$$

This means that

$$[\mathbf{H}, \mathcal{O}] = W\mathcal{O}, \tag{5.8}$$

which will be our fundamental operator relation.

Writing $\mathbf{H}$ as $\mathbf{p}^2 + U(x)$ and the transformed potential as $U[1]$ we can formally write

$$\mathbf{p}^2 + U[1] = \mathcal{O}(\mathbf{p}^2 + U)\mathcal{O}^{-1},$$

which indicates that $\mathcal{O}$ acts as an intertwining operator, linking the two potentials. Of course, the definition of $\mathcal{O}^{-1}$ requires the imposition of suitable boundary conditions on the functions in our Hilbert space.

Now take $\mathcal{O}$ to be of the form $-i\mathbf{p} + v$, (higher order forms are possible- see [5]), and expand the operator relation:

$$[\mathbf{p}^2 + U, -i\mathbf{p} + v] = -iW\mathbf{p} + vW.$$

On setting $\mathbf{p} = -i\partial_x$, this gives $W = 2v_x$ and

$$U_x = v_{xx} + (v^2)_x.$$

This in turn gives

$$U = v_x + v^2 + k,$$

(where k is a constant) which is a Ricatti equation for v. The usual linearisation of this equation, setting $v = \frac{\psi_x}{\psi}$ tells us that ψ satisfies the original Schrödinger equation with eigenvalue k. This is familiar from the usual theory of Darboux transformations.

As an example, take $U(x) = x^2$. It is clear that $v = x$ satisfies the Ricatti equation for $k = -1$,(and in fact the Hamiltonian factorises in this case). Starting from the solution $\psi = \exp(-\frac{x^2}{2})$ we find

$$\mathcal{O} = -\partial_x + x, \quad W = 2,$$

and

$$\mathcal{O}\psi = 2x\exp(-\frac{x^2}{2}).$$

Continuing in this way it is possible to generate all the Hermite polynomials, which correspond to solutions of the Schrödinger equation associated to the negative integer eigenvalues. The Darboux transformation in this case is related to creation and annihilation operators for the harmonic oscillator.

Higher order operators may arise as compositions of operators of lower order, in which case they are called reducible, or may be irreducible. The analysis of (5.8) in this case follows a similar pattern, but is more involved.

5.4 A Comment on the General Form.

The operator relation (5.8) is reminiscent of commutation relations for general Lie algebras, a fact which leads us to ask what extensions of this method may be

possible. Consider the $SU(2)$ relations

$$[J_3, J_\pm] = \pm J_\pm,$$

$$[J_+, J_-] = 2J_3.$$

Since J^2 is the $SU(2)$ casimir operator, we could take H to be some combination of J^2 and J_3. Quadratic casimirs exist, of course, for other Lie groups and this will be taken up later.

5.5 A Two Soliton Solution.

The general scheme underlies our particular example, and leads us to consider the factorisation of more general problems associated to solutions to the KdV equation.

In particular we shall examine a general 2-soliton solution at $t = 0$,

$$u_2 = -2(\kappa_2^2 - \kappa_1^2)\frac{\kappa_2^2 \cosh^2(\theta_1) + \kappa_1^2 \mathrm{sech}^2(\theta_2)}{\kappa_2 \cosh\theta_1 \cosh\theta_2 - \kappa_1 \sinh\theta_1 \sinh\theta_2}. \tag{5.9}$$

This solution is obtained from $u_0 = 0$ by a D.T.

$$u_0 \mapsto u_0 - 2\partial_x^2 \ln W(\psi_1, \psi_2),$$

where

$$\psi_1 = \cosh\theta_1,$$

$$\psi_2 = \sinh\theta_2, \quad \theta_i = \kappa_i x.$$

The Ricatti equation for v is solved by

$$v = \frac{-\psi_x}{\psi}, \quad \psi = \frac{W(\psi_1, \psi_2, f)}{W(\psi_1, \psi_2)},$$

where f is an eigenfunction associated to λ, with $u_0 = 0$, and ψ_i is associated to λ_i.

As a specific example for which we have some results, consider the 2-soliton potential

$$u_2 = \frac{-6}{\cosh^2 x}.$$

Let $\psi = 2\cosh^2 x$, so

$$v = \frac{2\cosh x \sinh x}{\cosh^2 x} = \frac{\sinh 2x}{\cosh^2 x},$$

$$v = \frac{\sinh 2x}{\cosh 2x}\frac{(\cosh^2 x + \sinh^2 x)}{\cosh^2 x} = \tanh 2x(1 + \tanh^2 x),$$

so that

$$v = 2\tanh x.$$

Now,

$$v_x - v^2 = \frac{2}{\cosh^2 x} - 4(1 - \frac{1}{\cosh^2 x})$$

$$= \frac{6}{\cosh^2 x} - 4 = -u + \lambda.$$

ψ is a solution of the Schrödinger equation for the potential

$$u_1 = -\frac{2}{\cosh^2 x}$$

belonging to the eigenvalue 2, and indeed

$$u_2 + 2v_x = u_1,$$

showing that in this case the action of the Darboux transformation 'steps down' u. ψ_3 is given by

$$\psi_3 = \frac{W(\cosh x, \sinh 2x)}{\cosh x}.$$

There is a well-known algebraic relation between the v's for Darboux- transformed potentials: namely, if v_i is a 'potential' for u_i, and u_1 and u_2 are 1-soliton solutions obtained from $u_0 = 0$ by Darboux transformations for eigenvalues λ_1 and λ_2 respectively, then the function u_3 obtained from

$$v_3 = \frac{\lambda_2 - \lambda_1}{v_1 - v_2}$$

is a 2- soliton solution. Now let $\psi_3 = W(\psi_1, \psi_2)$, so that

$$v_3 = \frac{-W(\psi_1, \psi_2)_x}{W(\psi_1, \psi_2)}.$$

We have $W(\psi_1, \psi_2)_x = (\lambda_2 - \lambda_1)\psi_1\psi_2$, giving

$$v_3 = \frac{(\lambda_2 - \lambda_1)\psi_1\psi_2}{(\psi_1\psi_{2x} - \psi_2\psi_{1x})}$$

leading to (5.9). The Wronkskian form of the solutions leads directly to a particular example of the superposition rule.

We expect that starting from a solution $\psi[\kappa_1, ..., \kappa_m]$ we may apply an operator $B^+_{\kappa_1 ... \kappa_{m+1}}$ to generate a solution of the same weight at a new level. We could indeed verify that this is the case, developing the methods considered above.

5.6 A Connection with Lie Algebras.

If we make the substitution $\cos\theta = \tanh x$ above, as indeed we could have chosen to do from the start, we find that

$$B^{\pm} = \mp \sin\theta \partial_\theta + N\cos\theta.$$

and that this is essentially $L_{\mp}$ when we apply this operator to states for which $L_z|\psi\rangle = N|\psi\rangle$, with the usual definitions of the L's (see, for example, [26].)

The algebra of the B's is then related to that of the $L_{\pm}$ in a direct way. The Hamiltonians given above, however, do not directly correspond to the angular part of the Laplacian in three dimensions as obtained from the usual separation of variables, as the following argument shows.

The relevant operators are

$$L_{\pm} = \mathrm{e}^{\pm i\phi}(\pm\frac{\partial}{\partial\theta} + i\cot\ \theta\frac{\partial}{\partial\phi}).$$

When applied to an N-eigenstate of $-i\partial/\partial\phi$, which we recognise as L_z, the operator L_+L_- becomes

$$\frac{1}{\sin^2\theta}((\sin\theta\frac{\partial}{\partial\theta})^2 + N\sin^2\theta + N(N+1)).$$

The action of L_+ and L_- is less mysterious when we realise that the $N(N+1)$ derives from the $z-$ component of angular momentum, and not the total: the transformation maps states of fixed z angular momentum to states having different total angular momentum, and so the chain of operations need not terminate, as we saw above. For each N we have these raising and lowering operators, and it is the totality of these operators for all N which constitutes the relevant algebraic structure.

We have established a link between the algebra of angular momentum and the Darboux transformation of the KdV potentials. It is natural to ask whether we could expand on the analogy by examining the casimirs of other groups as potential hamiltonians. An obvious choice is the group $SU(3)$. For this group there are three sets of raising and lowering operators, some of which we anticipate to be interpretable as Darboux operators. As mentioned above, it is possible to define supersymmetry algebras starting from sets of raising and lowering operators: in the case of the simple KdV potentials we could construct a superalgebra at each level. Perhaps the best way to extend the method to other Lie algebras would be through these supersymmetric systems. Although we will not pursue this further, it seems possible that higher dimensional integrable systems could be investigated using this approach. In the following chapters alternative approaches to higher dimensions will be considered.

Chapter 6

Twistor-inspired Darboux Transformations.

6.1 Introduction.

The sine-Gordon equation is a frequently encountered and intensively studied equation, about which a lot is known. In that sense the current chapter can say little that is new. However, the analysis given here has a quite natural generalisation to systems in higher dimensions, and the structure of certain ansatze for these higher dimensional systems may be understood in the light of this simple case.

We will see that a Darboux transformation for a linearisation of the sine-Gordon equation induces a transformation of the solution of the original equation which may be related to the well known Bäcklund transformation.

We will write the sine-Gordon equation for the field φ in lightcone coordinates as

$$\varphi_{pq} = \sin\varphi, \tag{6.1}$$

where the notation f_p is used to indicate $\partial f/\partial p$, and $p \equiv (y+t)$, $q \equiv (y-t)$.

In the next section we introduce the particular form of the Darboux transformation which is relevant to our study of the sine-Gordon equation in terms of a linear system of equations. We go on to consider the occurrence of the sine-Gordon equation as a special case of the Bogomolnyi equations, and apply the twistor method, introduced in chapter 4, to its solution.

The main topic of the chapter is the extraction of a link between the Darboux transformation for a certain linear equation and the twistor method, and a possible extension to the construction of monopoles. This is developed in the later sections, when the well-known ansatze developed by various authors ([56, 10, 43, 42]) are considered in the light of the results for the sine-Gordon case.

6.2 The Darboux Transformation.

Our starting point will be the linear differential equation

$$\Psi_{pq} = \Psi, \tag{6.2}$$

which is a particular case of

$$\Psi_{pq} + u\Psi = 0, \tag{6.3}$$

where u is a function of p and q. This equation and its elliptic counterpart are examined in [32]. We will consider later how it arises in the study of the sine-Gordon equation. Take particular solutions g and f to (6.2) and define the transformed field $\Psi[1]$ through

$$\begin{aligned} (g\Psi[1])_p &= fg_p - f_p g, \\ (g\Psi[1])_q &= gf_q - g_q f. \end{aligned} \tag{6.4}$$

The consistency condition on (6.4) is

$$g_{pq} f = f_{pq} g, \tag{6.5}$$

which holds since f and g satisfy (6.2).

By straightforward calculation, using (6.4) to expand the derivatives of f and g we find

$$(g\Psi[1])_{pq} = f_q g_p - f_p g_q, \tag{6.6}$$

which gives

$$\Psi[1]_{pq} = (\frac{2g_p g_q}{g^2} - \frac{g_{pq}}{g})\Psi[1]. \tag{6.7}$$

This may be rewritten, using (6.2), as

$$\Psi[1]_{pq} = (1 - 2\partial_p\partial_q \ln g)\Psi[1]. \tag{6.8}$$

The Darboux transformation for the system (6.2) is the set of transformations

$$\Psi \mapsto \Psi[1],$$

$$u \mapsto u[1],$$

where Ψ satisfies (6.4) for some f and g, and

$$u[1] = u + 2\partial_p\partial_q \ln g. \tag{6.9}$$

A particular instance of this transformation will be described in the next section.

6.3 Twistor Solution of the sine-Gordon Equation.

6.3.1 Reduction of the Bogomolnyi Equations.

To approach the solution of the sine-Gordon equation by means of established twistor methods we examine the sine-Gordon equation as a special case of the Bogomolnyi equations, here considered as equations on a 2+1 dimensional spacetime,

$$D_i\Phi = \frac{1}{2}\epsilon_{ijk}F^{jk}, \tag{6.10}$$

where

$$D_i\Phi \equiv \partial_i\Phi + [A_i, \Phi], \tag{6.11}$$

and

$$F_{ij} \equiv \partial_i A_j - \partial_j A_i + [A_i, A_j]. \tag{6.12}$$

In these equations $\mathbf{A}$ is an $su(2)$-valued connection field, Φ a scalar Higgs field in the adjoint representation (which accounts for the definition of the covariant derivative of Φ,) ϵ_{ijk} the totally antisymmetric symbol in 2+1 dimensions, and the roman indices are raised and lowered using the flat metric η^{ij} of signature (+ - -).

We will see in chapter 7 that these equations arise as a reduction of the full ASDYM equations in 2+2 dimensions. The twistor description of solutions to the full equations has a natural reduction to a so-called minitwistor description of solutions to the Bogomolnyi equations, although the Bogomolnyi and related equations can be related directly to bundles over minitwistor space [30]. Putting A_i and Φ into the form $J^{-1}\partial_z J$ for some z enables us to write the Bogomolnyi equation as an integrable chiral equation which in turn effectively yields the sine-Gordon equation when J is chosen appropriately. An example of one possible choice appears in [27], where extended wave solutions to the chiral equation for J are examined. In our case, choosing

$$J = \begin{pmatrix} \cos\frac{1}{2}\varphi & e^{ix}\sin\frac{1}{2}\varphi \\ -e^{-ix}\sin\frac{1}{2}\varphi & \cos\frac{1}{2}\varphi \end{pmatrix},$$

in

$$(J^{-1}J_x)_x + (J^{-1}J_p)_q, \tag{6.13}$$

which is a reduction of the Bogomolnyi equations from chapter 4, or taking

$$A_y = -A_t = \begin{pmatrix} 0 & ig(y,t) \\ -ig(y,t) & 0 \end{pmatrix},$$

$$A_x = -\frac{1}{2}\cos(\frac{1}{2}\varphi)\begin{pmatrix} \cos\frac{1}{2}\varphi & \sin\frac{1}{2}\varphi \\ \sin\frac{1}{2}\varphi & -\cos\frac{1}{2}\varphi \end{pmatrix},$$

and

$$\Phi = \frac{1}{2}\sin(\frac{1}{2}\varphi)\begin{pmatrix} \sin\frac{1}{2}\varphi & -\cos\frac{1}{2}\varphi \\ -\cos\frac{1}{2}\varphi & -\sin\frac{1}{2}\varphi \end{pmatrix}$$

gives rise to the sine-Gordon equation for φ.

6.3.2 The Twistor Construction.

Ward [59] describes the construction of solutions to the sine-Gordon equation from essentially a special case of the n-monopole ansatze developed in [43, 42, 56] and

references therein. The following is a brief outline of the method, with certain simplifications. A more complete account may be found in [59]. The starting-point is a patching matrix F for a vector bundle over minitwistor space. The class of bundles determined by this patching matrix is known to correspond to solutions of the Bogomolnyi equations: these solutions are, in principle, obtained by a splitting method. For some patching matrices the splitting may be performed analytically, and this is the case for those giving rise to the sine-Gordon ansatze considered here. These ansatze arise from patching matrices of the form

$$F = \begin{pmatrix} \lambda^n & \rho \\ 0 & \lambda^{-n} \end{pmatrix}.$$

The splitting involves the introduction of fields $\psi_{AB\ldots C}$ which are solutions to the 2+1 dimensional equation

$$\partial^{AB}\psi_{BC\ldots D} = 0, \tag{6.14}$$

where the spinor indices are contracted with the skew symbol ϵ^{AB}. (The indices may take the values 0 and 1.)

Explicitly, if ρ is an element of $H^1(\mathcal{M}, \mathcal{O}(-2n-2))$ [22], with $\mathcal{M}$ some suitable region of minitwistor space,

$$\psi_{A\ldots B} = \frac{1}{2\pi i}\int_\Gamma \pi_A \ldots \pi_B \rho(\pi_D \pi_E x^{DE}, \pi_D)\pi_C \mathrm{d}\pi^C \tag{6.15}$$

is a field satisfying (6.14). Here $[\pi_0, \pi_1]$ are homogeneous coordinates on $\mathbf{P}^1$, and Γ is taken to be the equator. These fields then also satisfy

$$\partial_{AB}\partial^{AB}\psi_{CD\ldots E} = 0, \tag{6.16}$$

and lead to the ansatze

$$\cos\varphi = 1 - 2\partial_p\partial_q \ln \det M, \tag{6.17}$$

where M is constructed from $\psi_{AB\ldots C}$ in a manner to be described. In fact, these fields $\psi_{A\ldots B}$ can also yield ansatze for the modified chiral model, and other soliton reductions, but here we will examine only the sine-Gordon equation.

An element of $\mathbf{H}^1(\mathcal{M}, \mathcal{O}(-2n-2))$ here corresponds to a spin-n field ψ, meaning that there are $2n$ spinor indices on ψ.

In order to make the reduction to 1+1 dimensions we impose the following x-dependence on all components:

$$\partial_x \psi = \pm i\psi. \tag{6.18}$$

It follows that

$$\partial_p \partial_q \psi = \psi.$$

The occurrence of this equation leads to consideration of the Darboux transformation outlined above.

6.3.3 An Application of the Darboux Transformation.

Given a solution to the Bogomolnyi equations (6.10) the quantity $\mathrm{tr}(\Phi^2)$ is gauge invariant. For the sine-Gordon equation this quantity is given by

$$-2\mathrm{tr}(\Phi^2) = \frac{1}{2}(1 - \cos\varphi), \tag{6.19}$$

whilst Prasad has shown that, for the ansatz given above

$$-2\mathrm{tr}(\Phi^2) = \partial_p \partial_q \ln\psi, \tag{6.20}$$

since $\partial_x^2 \ln\psi = 0$ (see, e.g. [10]). This may be interpreted as a link between the Darboux transformation for the linear problem generating the sine-Gordon ansatz, and the resulting 1-soliton solution. We take u to be given by

$$u = -4\mathrm{tr}(\Phi^2) - 1 = -\cos\varphi. \tag{6.21}$$

When $\cos\varphi$ is 1, this is equal to -1, which gives the original equation (6.2). The Darboux transformed field is then

$$\cos\varphi[1] = \cos\varphi - 2\partial_p \partial_q \ln\psi. \tag{6.22}$$

An example given in [56] uses $\psi = \cosh(\alpha p + \alpha^{-1} q)$, and $u = -1$, giving a simple 1-soliton solution.

A 2-soliton solution requires as input a solution ψ_1 to the wave equation, depending on two independent parameters, α and β. The transformed potential is

$$u[2] = u + 2\partial_p\partial_q \ln \det M \tag{6.23}$$

with

$$M = \begin{pmatrix} \psi_1 & \psi_2 \\ \psi_2 & \psi_3 \end{pmatrix}.$$

The ψ_i are determined from

$$\psi_{1p} = i\psi_2,$$

$$\psi_{2p} = i\psi_3,$$

$$-i\psi_1 = \psi_{2q},$$

$$-i\psi_2 = \psi_{3q},$$

where we are imposing the condition

$$\partial_x\psi_k = -i\psi_k. \tag{6.24}$$

(In this case, ψ_1 cannot be, for example, a single hyperbolic cosine term, as the determinant will then vanish. The higher k-soliton ansatze require that ψ_1 be a sum of at least k independent terms.) In fact det M corresponds to the contraction

$$\psi^{AB}\psi_{AB}, \tag{6.25}$$

where ψ^{00} is proportional to our ψ_1, and this correspondence persists for higher-spin fields (this is discussed in [63, 57]). In the Darboux transform picture, this system arises from a solution ψ_1 of (2.1), and a solution $\psi_1[1]$ of the transformed equation

$$\psi_{pq} + u[1]\psi = 0. \tag{6.26}$$

Under the given conditions it can be shown that

$$\psi_1[1] = \psi_3 - \frac{\psi_2^2}{\psi_1} \tag{6.27}$$

satisfies (6.27) with $u[1] = u + 2\partial_p\partial_q \ln\psi_1$. We may relate this directly to the sine-Gordon ansatz by factorising the determinant $\psi_1\psi_3 - \psi_2^2$ into $\psi_1\psi_1[1]$, and using the elementary identity

$$\ln \det M = \ln\psi_1 + \ln\psi_1[1]. \tag{6.28}$$

Taking

$$a = \cosh(\alpha p + \alpha^{-1}q),\ b = \sinh(\beta p + \beta^{-1}q),\ \psi_1 = a + b,$$

gives us a prospective 2-soliton solution;

$$\cos\varphi = 1 - 2\partial_p\partial_q \ln\det M. \tag{6.29}$$

6.4 The Bäcklund Form of the Darboux Transformation.

6.4.1 Recovering the Bäcklund Transformation.

We need to consider whether (6.29) does indeed satisfy the sine-Gordon equation. Direct calculation is prohibitive, and the calculation for higher ansatze will be less and less tractable, so an indirect and more widely applicable approach is needed. In our attempt to solve this problem we will find, as a byproduct, an approach to finding Bäcklund transformations for equations like the sine-Gordon equation, and will furnish another example of this method.

Our assertion is that when φ_1 is a k-soliton solution,

$$\varphi_3 = \arccos(\cos\varphi_1 - 2\partial_p\partial_q \ln U) \tag{6.30}$$

is a $(k+1)$-soliton solution, where

$$U_{pq} = U\cos\varphi_1.$$

This means that

$$\cos\varphi_3 = \cos\varphi_1 - 2\frac{U_{pq}}{U} + 2\frac{U_pU_q}{U^2} = -\cos\varphi_1 + 2\frac{U_pU_q}{U^2}. \tag{6.31}$$

Straightforward use of trigonometric identities then leads to

$$\cos(\frac{\varphi_3+\varphi_1}{2})\cos(\frac{\varphi_3-\varphi_1}{2}) = \frac{U_p}{U}\frac{U_q}{U}. \tag{6.32}$$

A fruitful approach to (6.32) is to require

$$U_p = \beta\cos(\frac{\varphi_3+\varphi_1}{2})U, \tag{6.33}$$

$$U_q = \frac{1}{\beta}\cos(\frac{\varphi_3-\varphi_1}{2})U. \tag{6.34}$$

We now take this to *define* ϕ_3: that is, out of the possible ϕ_3 satisfying (6.31) we are selecting those defined by (6.33, 6.34). The remaining ambiguities amount to choosing $\beta \mapsto \beta^{-1}$ and (or) $\phi_3 \mapsto -\phi_3$, which still lead to solutions of the sine-Gordon equation. The consistency condition on (6.33,6.34) is

$$\beta(\frac{\varphi_{3q}+\varphi_{1q}}{2})\sin(\frac{\varphi_3+\varphi_1}{2}) = \frac{1}{\beta}(\frac{\varphi_{3p}-\varphi_{1p}}{2})\sin(\frac{\varphi_3-\varphi_1}{2}), \tag{6.35}$$

which reduces to the usual form of the Bäcklund transformation for the sine-Gordon equation

$$\varphi_{3q} = -\varphi_{1q} - \frac{2}{\beta}\sin(\frac{\varphi_3-\varphi_1}{2}), \tag{6.36}$$

$$\varphi_{3p} = \varphi_{1p} - 2\beta\sin(\frac{\varphi_3+\varphi_1}{2}) \tag{6.37}$$

when (6.2) is used. Thus φ_3 is a Bäcklund transform of φ_1, and therefore satisfies the sine-Gordon equation. The two soliton ansatz leads to an iterated Bäcklund transformation in the same way.

6.4.2 Finding a Linear Problem.

Before looking at a second example of the derivation of the Bäcklund transformation, we consider how we might arrive at the appropriate linear problem. Such an occurrence of the linear formulation leading to the Darboux transformation is to be found in the following analysis.

Let $\Psi = i\varphi_p/2, \eta = \exp(i\varphi)$, and note that the sine-Gordon equation becomes equivalent to the system

$$\Psi_q = \frac{1}{4}(\eta - \frac{1}{\eta}), \tag{6.38}$$

$$\eta_p = 2\eta\Psi. \tag{6.39}$$

Differentiating the first of these equations with respect to p we find that

$$\Psi_{pq} = \frac{1}{2}(\eta + \frac{1}{\eta})\Psi \tag{6.40}$$

$$= \cos(\varphi)\Psi. \tag{6.41}$$

For further development, the linear system may be considered as the condition for a generalised vertical vector field (see, e.g. [46]) to describe a symmetry of the sine-Gordon equation. Let such a field be given by

$$X = \Psi\partial_u. \tag{6.42}$$

Then the second prolongation of X, $X^{(2)}$ is simply

$$\Psi_{xx}\partial_{u_{xx}} + \Psi_{xt}\partial_{u_{xt}} + \ldots + \Psi\partial_u, \tag{6.43}$$

where the subscripts on Ψ indicate total derivatives. Now the symmetry condition reads

$$0 = X^{(2)}[u_{xt} - \sin u] = \Psi_{xt} - \Psi\cos u. \tag{6.44}$$

We can check that, for example, $u_x\partial_u$ and $u_t\partial_u$ are symmetry generating fields, but (6.44) is the more general expression. The linear system then amounts to a linearisation of the original equation. This is a natural way to perform the linearisation upon which we wish to act with the Darboux transform. In fact, this construction is a particular case of a 'universal linearisation' technique [53].

As our second example, the Liouville equation

$$u_{xt} = \mathrm{e}^u \tag{6.45}$$

may be linearised using a generalised vertical vector field $\psi\partial_u$:

$$\psi_{xt} - \mathrm{e}^u\psi = 0. \tag{6.46}$$

We may carry out the analysis leading to the Bäcklund transformation as we did for the sine-Gordon equation. Let U be a solution of (6.46) associated to a given u, and let $\tilde{u}$ be a Darboux transform of u using the seed U.

Then we find

$$e^{\tilde{u}} = e^{u} - 2\partial_x\partial_t \ln U,$$

which gives, on using (6.46),

$$\frac{1}{2}(e^{\tilde{u}} + e^{u}) = \frac{U_x}{U}\frac{U_t}{U}. \tag{6.47}$$

If we could factorise the l.h.s. of (6.47) we could choose to set each of the factors on the r.h.s. equal to one of the factors on the l.h.s.

Now use

$$\frac{1}{2}(e^{\tilde{u}} + e^{u}) = e^{\frac{1}{2}(\tilde{u}+u)}\cosh(\frac{1}{2}(\tilde{u} - u)).$$

We will write

$$U_x = \beta e^{(\tilde{u}+u)/2}U, \tag{6.48}$$

$$U_t = \beta^{-1}\cosh((\tilde{u} - u)/2)U. \tag{6.49}$$

In order for (6.48) to be consistent we require that

$$\frac{\beta}{2}(\tilde{u} + u)_t e^{\frac{1}{2}(\tilde{u}+u)} = \frac{1}{2\beta}(\tilde{u} - u)_x \sinh\frac{1}{2}(\tilde{u} - u).$$

This leads us to the auto Bäcklund transformation for the Liouville equation

$$(\tilde{u} + u)_t = \frac{2}{\beta}\sinh\frac{1}{2}(\tilde{u} - u), \tag{6.50}$$

$$(\tilde{u} - u)_x = 2\beta e^{\frac{1}{2}(\tilde{u}+u)}, \tag{6.51}$$

which can easily be shown to yield Liouville equations for $\tilde{u}$ and u independently. This completes our examination of the Bäcklund transformations.

6.4.3 A Remark on the Linear Problem.

Darboux transformations for the system (6.38) have been studied, and shown to give rise to a Bäcklund transformation like (6.36). Of some interest is the fact that a Bäcklund transformation may be derived from (6.4). Note that (6.4) implies that

$$(\Psi[1] + \psi)_p = -\frac{g_p}{g}(\Psi[1] - \psi), \tag{6.52}$$

$$(\Psi[1] - \psi)_q = -\frac{g_q}{g}(\Psi[1] + \psi), \tag{6.53}$$

where we have taken $f = \psi$ and g to be a solution of (6.2). This is indeed in the form of a B.T. for (6.1), and more importantly, when $g = \cosh(\alpha p + \alpha^{-1}q)$, leads to the expression $\cos\varphi[1] = 1 - 2\mathrm{sech}^2(\vartheta)$ as we saw using the equivalent Darboux transform.

6.5 Remarks and Extensions.

6.5.1 The n-monopole Ansatz.

Prasad [42] defines n-monopole ansatze to the self-duality equations taken in Yang's gauge by applying a combination of 'Bäcklund transformations' and gauge transformations to Yang's equations [64]. It is interesting to note that the iterative step in Prasad's method generates a version of the well-known form of the sine-Gordon Bäcklund transformation when applied to the appropriate ansatz. The relationship between the Bäcklund transformation and what we would wish to identify as a Darboux transform is established in a way which is indirect but should admit an explicit link. We bear in mind that the monopole equations are reductions from 4 dimensions to 3, whilst our Bogomolnyi equations arise as reductions from 2+2 to 2+1, and the sine-Gordon equation is obtained by a further specialisation to 1+1.

Let us first take up a remark made by Yang [64] concerning a simple solution to the SDYM in the R gauge: take ϕ to be a solution of the 4 dimensional Laplace equation, and ρ and $\tilde{\rho}$ to be related to ϕ by the Bäcklund transformation

$$\phi_z = \rho_{\bar{y}}, \quad \phi_y = -\rho_{\bar{z}},$$

$$\phi_{\bar{y}} = \tilde{\rho}_z, \quad \phi_{\bar{z}} = -\tilde{\rho}_y. \tag{6.54}$$

These functions then constitute a solution to the SDYM in the R gauge:

$$\nabla_4^2 \ln\phi + \frac{1}{\phi^2}\left(\rho_{\bar{y}}\tilde{\rho}_y + \rho_{\bar{z}}\tilde{\rho}_z\right) = 0,$$

$$\left(\frac{\rho_{\bar{z}}}{\phi^2}\right)_z + \left(\frac{\rho_{\bar{y}}}{\phi^2}\right)_y = 0,$$

$$\left(\frac{\tilde{\rho}_z}{\phi^2}\right)_{\bar{z}} + \left(\frac{\tilde{\rho}_y}{\phi^2}\right)_{\bar{y}} = 0. \tag{6.55}$$

Here ∇_4^2 is the four dimensional Laplace operator.

In chapter 7 we will show that a simple generalised symmetry of Yang's form of SDYM allows us to replace ϕ by $\hat{\phi} = b/\phi$, where b is a constant (ρ and $\tilde{\rho}$ are also transformed, but we do not consider this for the moment.) The full transformation can actually almost be read off from (6.55), but a systematic derivation for a related equation is presented in chapter 7. Prasad includes a gauge transformation in his full transformation, but we will examine only the simplest form of the symmetry transformation. The following result in terms of $\hat{\phi}$ holds in both cases, as may be readily checked by a direct calculation. Now, of course, $\hat{\phi}$ does not satisfy the Laplace equation, but instead is a solution of

$$\nabla_4^2\hat{\phi} + u\hat{\phi} = 0, \tag{6.56}$$

where $u = -2\nabla_4^2 \ln\phi$. This has the form of a Darboux transformation of the Laplace equation: compare (6.52) with $\psi = 0$. For 'Yang's ansatz' the general results of Prasad are readily made explicit.

6.5.2 The Darboux Method and Monopoles.

An adaptation of the methods used above to the monopole solutions to ASYDM arises from the fact that (6.20) holds for the ansatze developed by Atiyah and Ward and by Corrigan *et al.* The ansatze thus indicate explicit forms for the u-part of the Darboux transformations for the equation

$$\nabla_3^2\psi = -u\psi \tag{6.57}$$

where ∇_3^2 is the three-dimensional Laplace operator. An explicit link between this and the classical Darboux transformation for the Schrödinger equation in three spatial dimensions is the condition, imposed for the monopole ansatze, and mirrored in the above reduction to 1+1 dimensions,

$$\psi(\mathbf{x}, t) = \exp(it)\hat{\psi}(\mathbf{x}). \tag{6.58}$$

The wave equation and the Schrödinger equation with $u = -1$ are equivalent for ψ of this particular form.

6.5.3 Conclusions.

The Darboux transformation for certain systems derived from Helmholtz equations may be used to obtain multisoliton solutions to specialisations of the Bogomolnyi equations. In the case of the sine-Gordon equation the results are well known solutions, and the link between the twistor method and the Darboux transform can be made explicit.

The multi-monopole solutions may be derived systematically from suitable solutions to the Helmholz equation, and the corresponding energy density is simply related to the potential in the iterated Darboux transformation of this Helmholz equation. The advantage of the twistor formulation is that it exploits the fact that the ansatze are all based on fields satisfying Dirac-like equations, which in turn are explicit Bäcklund transformations for the Laplace (or wave) equation [11].

Chapter 7

A Jet Bundle Approach to Bäcklund Transformations.

7.1 Introduction.

In the next section, after establishing some motivation for our work, we introduce several of the central ideas of the chapter, including jet bundles, pullback bundles and coordinate-free descriptions of partial differential equations. These ideas are then used to define Bäcklund transformations of partial differential equations in a coordinate-free way. Examples of Bäcklund transformations are then given in coordinates with the jet bundle notation exhibited for comparison. The development follows that of [41], avoiding discussion of the contact structure. All structures are taken to be smooth. The geometric picture is then used to examine so-called generalised symmetries of an integrable chiral equation in 2+1 dimensions. This equation is closely related to the equation treated numerically in chapter 4, and to the SDYM equations examined in 6.

7.2 Motivation

Jet bundles provide a natural geometric setting for the study of partial differential equations, and the introduction of geometric ideas allows one to exploit many of the highly developed tools available to geometers [20]. One might at first think it unlikely that geometry could shed very much light on the behaviour of differential equations and their solutions, but brief consideration of some of the methods which

have been applied soon counters this impression.

Firstly, as we will see below, a differential equation may be described as a submanifold of a certain large manifold, the total space of a jet bundle. It may be that the submanifold is invariant under certain deformations of the total space. If these deformations are bundle automorphisms, a term which will be defined in the next section, then we are justified in calling them symmetries of the submanifold, and therefore of the equation. The power of this description of symmetries is that the methods applied to the study of symmetries of submanifolds may be applied with equal ease to differential equations [48]. It should be remarked that certain extensions of the methods presented below are possible- and necessary- to describe, for example, breaking or multi-valued solutions to some equations [38]. It is also prudent to note that the identification of differential equations and their symmetries may be a little more subtle than our treatment would seem to suggest [53], although it is quite adequate for our discussion.

An interesting aspect of the treatment of partial differential equations is the occurrence of symmetries depending not only on the space and fibre coordinates, but on 'derivative coordinates'. An example of this will be presented in a later section.

Our main interest in this chapter is in integrable equations for which a different type of transformation exists. These transformations are known as Bäcklund transformations, and, as mentioned in 2 are a characteristic feature of integrable equations. It is a remarkable fact that there exists a quite simple geometric description of Bäcklund transformations, and this description forms the main topic of this chapter. It will be seen that the final form of the Bäcklund transformation in some cases admits a more symmetric interpretation than would at first appear.

7.3 An Introduction to Jet Bundles.

7.3.1 A Note on Fibre Bundles.

The definition of fibre bundle which is appropriate here is: a pair of manifolds E and M together with a map π, the projection, from E to M. The map must be

surjective and a submersion, which means that the jacobian has maximal rank. Locally, the bundle decomposes into a cartesian product with one of the factors locally diffeomorphic to M and the other to the typical fibre. A very schematic impression of the situation is presented in Fig 7.1.

Although everything constructed below makes sense for any fibre bundle, it is useful to have the following picture in mind. The bundle we consider consists of: a product $\mathbf{R}^n \times \mathbf{R}^m$, the total space; a projection π onto the first n factors; and the base space $\mathbf{R}^n$. In this picture, we can put everything into coordinates (x^i, u^α) with i running from 1 to n, and α running from 1 to m. Thus a point (a, b) in the total space will be projected onto a, a point in the base space. A general bundle will be written (E, π, M), so our bundle is the trivial $(\mathbf{R}^n \times \mathbf{R}^m, pr_1, \mathbf{R}^n)$. We will use the more general version since it is notationally just as simple, and in any case we have seen that all fibre bundles do indeed locally have this product structure.

Whenever a bundle is mentioned it will be helpful to have this trivial bundle as a model. Following [46], a bundle will be labelled by its projection in cases where the base and total spaces are clearly identified. In much of what follows we make no mention of any symmetry group acting on the fibres of the bundle, but we should be aware that the existence of such a group is a requirement of some definitions of a fibre bundle.

7.3.2 Sections of π, Jets and Bundle Morphisms.

Given the above bundle, we can consider maps $\phi : M \to E$ satisfying $\pi \circ \phi = \mathrm{id}_M$. Such maps are called sections of π, sending a point p in the base space to a point 'directly overhead' in the total space. Since we have stated that all our spaces and maps are smooth it makes sense to consider objects such as $\partial\phi/\partial x^i$ where x^i is a coordinate for M.

The 1-jet at p of a section ϕ is the equivalence class of sections whose values and first derivatives at p coincide with those of ϕ. It is denoted $j^1_p\phi$. An instructive way of looking at the 1-jet is to consider it as the equivalence class of sections having the same linear approximation as ϕ at p.

In the same way, 2-jets and so on can be defined. See [46] for a detailed

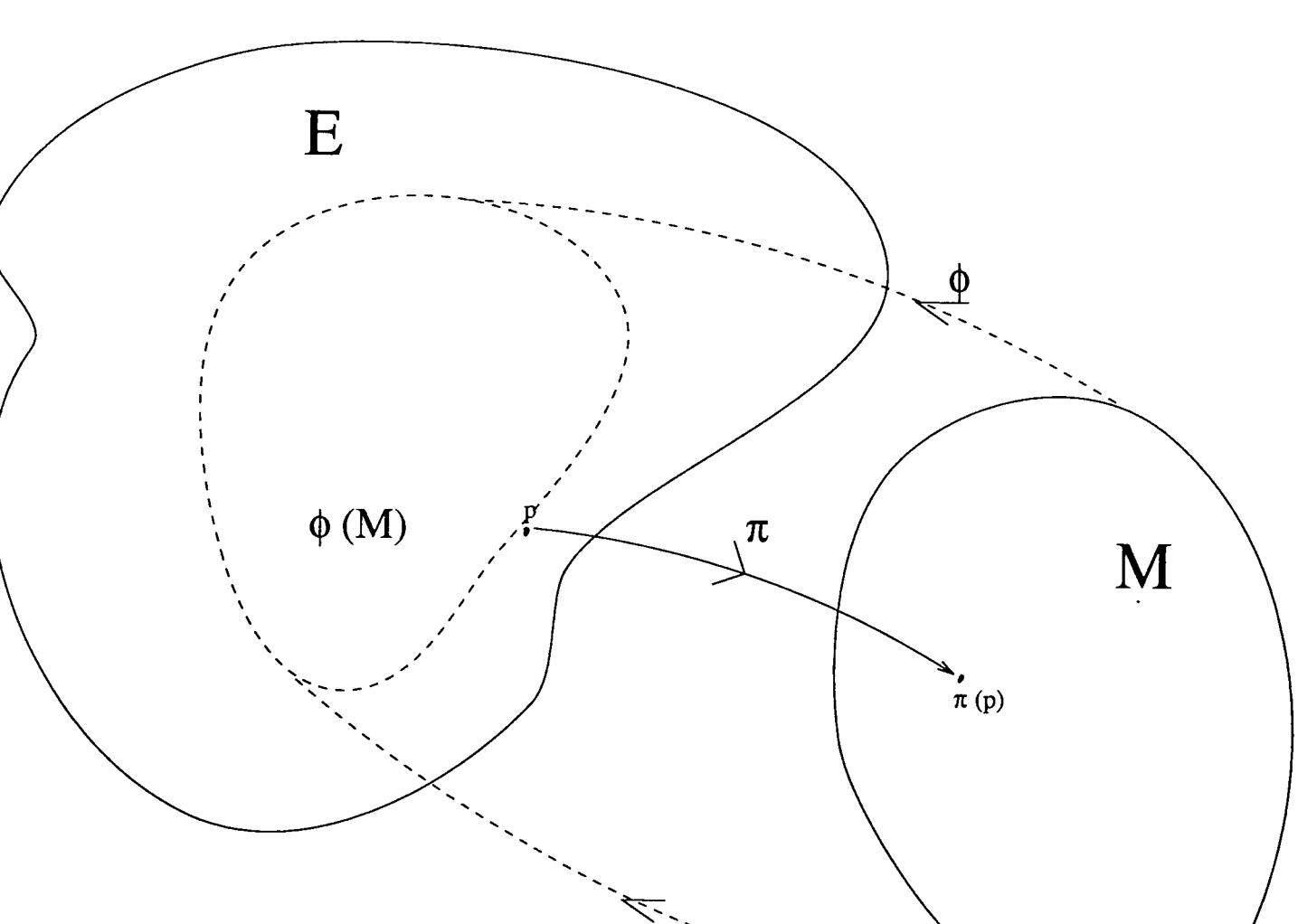

Figure 7.1: Sketch of a fibre bundle π.

treatment. In general, the k-jet of ϕ at p will be denoted $j^k_p\phi$. Extending our analogy with linear approximations, we can identify the k-jet of ϕ at p with its kth order Taylor expansion about p: the collection of sections sharing the same kth order expansion forms the equivalence class defining $j^k\phi$. This definition is independent of the particular coordinate system, as may be demonstrated by an application of the chain rule.

The collection of all 1-jets forms a bundle over M, with local coordinates $(x^i, u^\alpha, u^\alpha_i)$ where

$$u^\alpha_i(j^1_p\phi) = \frac{\partial\phi^\alpha}{\partial x^i}\Big|_p.$$

This first jet bundle will be denoted $(J^1\pi, \pi_1, M)$. Similarly, the k^{th} jet bundle will be written

$$(J^k\pi, \pi_k, M)$$

or simply π_k. We remark that the object $j^k\phi$ is a section of the bundle π_k, by which we understand that

$$j^k\phi : p \mapsto j^k_p\phi,$$

and

$$\pi_k \circ j^k\phi = \mathrm{id}_M.$$

Here, and elsewhere, id_X denotes the identity map on X.

In addition, the pair $(J^1\pi, E)$ can be given the structure of a bundle with projection $\pi_{1,0}$ such that

$$\pi_{1,0}(j^1_p\phi) = \phi(p)$$

A jet field is defined to be a section of $\pi_{1,0}$. Jet fields will form the basis of our characterisation of Bäcklund transformations later in this chapter. We will see that there is a close relationship between jet fields and connections.

Later we will need the idea of a bundle automorphism in the description of symmetries. A bundle automorphism is a pair of maps $f : E \to E$ and $\tilde{f} : M \to M$ such that

$$\pi \circ f = \tilde{f} \circ \pi.$$

This implies that fibres move to stay over their basepoints when a bundle automorphism acts. The *first prolongation* of a bundle automorphism is the map $j^1(f, \tilde{f})$

acting on 1-jets as

$$j^1(f,\tilde{f})j^1_p\phi = j^1_{\tilde{f}(p)} f \circ \phi \circ \tilde{f}^{-1}.$$

The prolongation is used to find symmetries of differential equations.

For completeness we mention that the objects $j^k_p\phi$ form a dense subset of the space of m-dimensional submanifolds of E (m is the dimension of the base space M) in contact to order k at p.

7.3.3 Pullback Bundles and Bäcklund Transformations.

Given a bundle (E,π,M) and a map $\rho : H \to M$, the pullback of π by ρ is the bundle

$$(\rho^*(E),\rho^*(\pi),H)$$

where

$$\rho^*(E) = \{(a,b) \in E \times H : \pi(a) = \rho(b)\}$$

and

$$\rho^*(\pi)(a,b) = b.$$

We can think of the pullback bundle as fixing the fibres of π onto the total space of ρ, with this total space becoming the base space of the new bundle. Certain structures on the old bundles give rise to interesting structures on the new.

Consider the jet bundle π_k, and a second bundle (H,ρ,M) . We can use ρ to define the pullback bundle $\rho^*(\pi_k)$. Suppose now we are given a kth-order partial differential equation, which may be nonlinear, and that we wish to determine a Bäcklund transformation to a second equation. We express the given equation as a submanifold S of $J^k\pi$, with the condition that sections ϕ of π are solutions only when $j^k\phi$ takes values in S. The value of this coordinate-free description is that it will allow us to construct the Bäcklund transformation in a coordinate-free way.

As a simple linear example, suppose we wish to examine the heat equation $u_t = u_{xx}$ in a coordinate-free way. The base space of our bundle is $\mathbf{R}^2$, and the fibre is $\mathbf{R}$. Since the equation is second-order we are dealing with $J^2\pi$ which has local coordinates $(t,x,u;\eta_1,\ldots,\eta_5)$, where $\eta_1 \equiv u_t$, $\eta_2 \equiv u_x$, $\eta_3 \equiv u_{tt}$, $\eta_4 \equiv u_{xt}$, and $\eta_5 \equiv u_{xx}$. We avoid writing u_i for the moment to emphasise that the

coordinates are *a priori* independent in the sense that they can act on objects which are not the 2-jets of some section. The heat equation is an algebraically determined submanifold of $J^2\pi$ given by

$$\eta_1 = \eta_5.$$

We could consider this submanifold as the zero set of a function f from $J^2\pi$ to $\mathbf{R}$. The section ϕ is a solution if $j^2\phi$ lies in this submanifold, which means that $f \circ j^2\phi = 0$. That is, writing out the ηs explicitly in terms of the us,

$$u_t(j_p^2\phi) - u_{xx}(j_p^2\phi) = 0, \quad p \text{ in} \mathbf{R}^2,$$

which is just

$$\frac{\partial\phi}{\partial t} = \frac{\partial^2\phi}{\partial x^2}.$$

This is the heat equation as it is usually written, and we see that the descriptions are equivalent.

7.4 Bäcklund Transformations and Jet Fields.

7.4.1 Definitions.

Having established this we can move on to the description of Bäcklund transformations. A Bäcklund transformation will be a map

$$\Psi : \rho^*(\pi_k) \to J^1\rho$$

satisfying certain integrability conditions. These conditions are simply that the jet field Γ given by

$$\Gamma(b) = \Psi \circ \rho^*(j^k\gamma)(b)$$

be integrable in the sense that

$$\Gamma \circ \phi(p) = j_p^1\phi \tag{7.1}$$

for some sections ϕ. Here γ is a section of π , and $\rho^*(j^k\gamma)$ is the induced section of $\rho^*(\pi_k)$ given locally by

$$\rho^*(j^k\gamma)(b) = (j^k_{\rho(b)}\gamma, b).$$

We can see that there is a one to one correspondence between jet fields for ρ (that is, sections of $\rho_{1,0}$) and connections on ρ, since the latter are idempotent horizontal vector-valued 1-forms whilst the former may be used to determine a unique such one-form as may be seen in a coordinate representation. It is also of interest to note that the image of H (the total space of ρ) may be viewed as a differential equation in coordinate free form. It is a straightforward matter to demonstrate that integrability of a jet field is equivalent to the vanishing of the curvature of the associated connection. This follows from (7.1) by application of u_i:

$$u_i^\alpha(\Gamma \circ \phi) = \Gamma_i^\alpha = \frac{\partial \phi^\alpha}{\partial x^i}, \tag{7.2}$$

and the cross-differentiability condition on (7.2) is

$$\partial_j \Gamma_i^\alpha + \partial_j \phi^\beta \partial_\beta \Gamma_i^\alpha = \partial_i \Gamma_j^\alpha + \partial_i \phi^\beta \partial_\beta \Gamma_j^\alpha,$$

which may be expressed as

$$R_{\tilde{\Gamma}} = 0,$$

where $R_{\tilde{\Gamma}}$ is just the curvature of the connection $\tilde{\Gamma}$ associated to the jet field Γ. We require that Γ be integrable whenever $j^k\gamma$ takes values in S.

7.4.2 Examples.

A well-known example of this approach is the auto-Bäcklund transformation for the sine-Gordon equation. Here the bundle ρ is identified with π, so the Bäcklund transformation is defined by a map

$$\Psi : \pi^*(J^1\pi) \times \mathcal{S} \to J^1\pi,$$

and is given in coordinates by

$$\tilde{u}_x(\Psi) = u_x - 2\beta \sin(\frac{u+\tilde{u}}{2}),$$

$$\tilde{u}_t(\Psi) = -u_t + \frac{2}{\beta} \sin(\frac{u-\tilde{u}}{2}). \tag{7.3}$$

Here we understand that

$$u(j^1_p\gamma) = \gamma(p)$$

and

$$u_x(j^1_p\gamma) = \frac{\partial\gamma}{\partial x}\Big|_{\gamma(p)}.$$

The space $\mathcal{S}$ is a parameter space from which the element β comes. The appearance of this parameter is important for the production of new nontrivial solutions.

The appearance of the Bäcklund transformation suggests that in this case it could be considered as a pair of maps from $J^1(\pi \times_M \pi) \times \mathcal{S} \to \mathbf{R}$, which is a pair of differential equations on $\pi \times_M \pi$. This is a more symmetric interpretation of the Bäcklund transformation which, after all, is supposed to work both ways. A derivation of this Bäcklund transformation was presented in chapter 6. The bundle π here is $\mathbf{R}^2 \times \mathbf{R}$, with $\pi = \mathrm{pr}_1$, with coordinates $(x, t, u, u_x, u_t; \tilde{u})$ on $\pi^*(\pi_1)$, and $(x, t, \tilde{u}, \tilde{u}_x, \tilde{u}_t)$ on π_1. We could equally say that there is a 1-parameter family of Bäcklund transformations for the sine-Gordon equation, with the parameter coming from $\mathcal{S}$. Of course since the Bäcklund transformation takes two 'arguments', one from the space of sections of each of the bundles, we would be justified in writing it as $\hat{\Gamma}(\gamma, \zeta) = 0$, where γ is any solution of the sine-Gordon equation, and we apply $\Gamma = \Psi \circ \pi^*(j^1\gamma)$ to ζ to obtain $\hat{\Gamma}$. As we stated above Ψ is a B.T. if for each γ there exists a section ζ of π such that

$$\Gamma \circ \zeta(p) = j^1_p\zeta.$$

Let

$$\Psi(x, t, u, u_x, u_t; \tilde{u}) = (x, t, \tilde{u}, \Psi_x, \Psi_t),$$

i.e $\tilde{u}_x(\Psi) = \Psi_x$ and so on.

The integrability condition here translates to

$$D_t\Psi_x - D_x\Psi_t = 0,$$

where

$$D_i = \frac{\partial}{\partial x^i} + u_i\frac{\partial}{\partial u} + \ldots + u_{ij_1\ldots j_n}\frac{\partial}{\partial u_{j_1\ldots j_n}} + \Psi_i\frac{\partial}{\partial\tilde{u}}$$

in the general case. In this example only terms up to $u_{ij}\partial/\partial_{u_i}$ are needed.

The coordinate form of the integrability condition gives $\tilde{u}_{xt} = \sin\tilde{u}$ and $u_{xt} = \sin u$, which is why this is called an auto-Bäcklund transformation. The fact that the two equations are satisfied independently makes this what is called a strong Bäcklund transformation. Strong Bäcklund transformations do not exist for all equations possessing a Bäcklund transformation, it usually being the case that the transformed field satisfies its equation only when the original field solves its own equation.

A Bäcklund transformation linking the Liouville equation and the two dimensional wave equation may be constructed as follows [25]. Let u satisfy the Liouville equation, and let

$$u_t = v_t - 2\beta e^{\frac{u+v}{2}},$$
$$u_x = -v_x - \frac{2}{\beta}e^{\frac{u-v}{2}}.$$

Then v satisfies the wave equation. An auto Bäcklund transformation for the Liouville equation was considered in chapter 6.

7.4.3 Applications.

The above ideas can be developed quite extensively in the jet bundle formulation by the use of the natural contact structure [41] but this is not essential for our needs. Here we simply note that the transformations above allow us to construct interesting solutions to the sine-Gordon equation from trivial ones such as $u = 0$ simply by choosing the parameter β, making the substitution $u = u_x = u_t = 0$, and solving for $\tilde{u}$, and to obtain the general solution to the Liouville equation from that of the wave equation. The Bäcklund transformation also allows us to construct a hierarchy of conserved currents: from (7.3) we can identify the conservation law

$$\lambda^{-1}\partial_x \cos(\frac{1}{2}(\phi + \tilde{\phi})) = \lambda\partial_t \cos(\frac{1}{2}(\phi - \tilde{\phi})). \tag{7.4}$$

The idea now is to expand $\tilde{\phi}$ as a Taylor-Laurent series in λ

$$\tilde{\phi}(x,t;\lambda) = \sum_{n=-\infty}^{\infty} \lambda^n \tilde{\phi}^{(n)}(x,t).$$

and equate powers of λ. This does indeed produce the required currents.

7.5 Generalised Symmetries of a Chiral Equation.

7.5.1 The Chiral Equation in Yang's Gauge.

Generalised symmetries, which are an extension of ordinary point symmetries, are also known as Lie-Bäcklund symmetries, and may be used to generate new solutions from old in just the same way as point symmetries.

As we mentioned in chapter 6 the Self dual Yang-Mills equations possess such symmetries, and as an illustration of the techniques involved we will in this section find generalised symmetries of the chiral model equation in 2+1 dimensions. These are not, strictly speaking, Bäcklund transformations as we defined them above, although they can be combined with gauge transformations to similar effect. The geometrical approach, using the ideas of jet theory, will be useful in the search for these generalised symmetries. The essential step is the use of derivative coordinates, which in this problem allows a reduction in the number of variables considered, and considerable reduction of the amount of work required. We follow the method of [33].

We work with the equations in Yang's R gauge [64], as this leads to the simplifications mentioned above. Let

$$g = \frac{1}{\phi}\begin{pmatrix} 1 & \rho \\ \tilde{\rho} & \phi^2 + \rho\tilde{\rho} \end{pmatrix}. \tag{7.5}$$

Then consider the chiral equation with $SO(2)$ symmetry

$$(\eta^{ij} + i\epsilon^{ijk}V_k)\partial_i(g^{-1}\partial_j g) = 0,$$

where V is the vector $(1,0,0)$ which breaks Lorentz invariance and ϵ^{ijk} is the totally antisymmetric symbol on $\mathbf{R}^{2+1}$, with metric tensor η^{ij}.

With g given by (7.5) these equations become

$$\phi(\rho_{z\bar{z}} - \rho_{tt}) - 2(\rho_{\bar{z}}\ \phi_z - \rho_t\ \phi_t) = 0,$$

$$\phi(\tilde{\rho}_{z\bar{z}} - \tilde{\rho}_{tt}) - 2(\tilde{\rho}_z\ \phi_{\bar{z}} - \tilde{\rho}_t\ \phi_t) = 0,$$

$$\phi(\phi_{z\bar{z}} - \phi_{tt}) - \phi_z\phi_{\bar{z}} + \phi_t\phi_t + \rho_{\bar{z}}\tilde{\rho}_z - \rho_t\tilde{\rho}_t = 0. \tag{7.6}$$

The explicit reference to i is absorbed into z and $\tilde{z}$. We now use the fact that ρ appears only as $\rho_{\tilde{z}}, \rho_t$, and $\tilde{\rho}$ as $\tilde{\rho}_z, \tilde{\rho}_t$ to introduce the set of variables

$$\vartheta^1 = \phi,\ \vartheta^2 = \rho_t,\ \vartheta^3 = \rho_{\tilde{z}},\ \vartheta^4 = \tilde{\rho}_t,\ \vartheta^5 = \tilde{\rho}_z,$$

$$x^1 = t,\ x^2 = z,\ x^3 = \tilde{z}.$$

This restricted set of coordinates on $J^1\pi$ allows us to formulate the symmetry problem in terms of point symmetries of the solution manifold more readily. Then we have

$$\vartheta^1(\vartheta^3_2 - \vartheta^2_1) - 2(\vartheta^3\vartheta^1_2 - \vartheta^2\vartheta^1_1) = \alpha_1,$$

$$\vartheta^1(\vartheta^5_3 - \vartheta^4_1) - 2(\vartheta^5\vartheta^1_3 - \vartheta^4\vartheta^1_1) = \alpha_2,$$

$$\vartheta^1(\vartheta^1_{23} - \vartheta^1_{11}) + (\vartheta^3\vartheta^5 - \vartheta^2\vartheta^4 - \vartheta^1_2\vartheta^1_3 + \vartheta^1_1\vartheta^1_1) = \alpha_3,$$

$$\vartheta^4_2 - \vartheta^5_1 = \alpha_4,$$

$$\vartheta^2_3 - \vartheta^3_1 = \alpha_7.$$

Here the αs are a set of functions which vanish on the solution manifold of the system (recall the coordinate-free description of a set of differential equations.) α_4 and α_7 are cross differentiability conditions. The ϑ^j_i are evidently coordinate functions on the repeated jet bundle $J^1\pi_1$, which contains the second jet bundle π_2.

We seek point transformations of the jet bundle which may depend on all the coordinates, including the derivative coordinates. To do this we look at automorphisms (B, id_M), $B : H \to H$, where H is the fibre of $J^1\pi$, such that

$$\alpha_r \circ B = \sum_s u^s_r \alpha_s, \tag{7.7}$$

where u^s_r are functions of ϑ^α. With this condition we see that the transformed αs vanish when the original αs do, so that we get the same differential equation back.

Now let

$$\eta^\alpha = B^\alpha(\vartheta^1, \ldots, \vartheta^5). \tag{7.8}$$

We do not allow η to depend explicitly on x^i here, for simplicity.

We then have

$$\eta_j^\alpha = \vartheta_j^\beta \frac{\partial \eta^\alpha}{\partial \vartheta^\beta} \equiv \vartheta_j^\beta \eta_{,\beta}^\alpha. \tag{7.9}$$

This will be used to satisfy (7.7) by demanding that the coefficients of the ϑ_k^α vanish independently.

Take the expression for $\alpha_4 \circ B$;

$$\eta_2^4 - \eta_1^5 = \sum_k u_4^k \alpha_k.$$

We expand this in terms of the ϑ_i^α, and examine the terms, as in table 1. As an example of the reasoning, we will examine the coefficient of ϑ_1^1. On the left, from (7.9), we see that only

$$\vartheta_1^1 \frac{\partial \eta^5}{\partial \vartheta^1}$$

contributes, whilst on the right terms from α_1 and α_2 will occur. Since η does not depend on derivatives of ϑ we will not have a contribution from α_3. The rest of the entries can be deduced in a similar way.

Term	coefficient in $\sum u_4^k \alpha_k$	coefficient in $\alpha_4 \circ B$
ϑ_1^1	$2u_4^1\vartheta^2 + 2u_4^2\vartheta^4$	$-\eta_{,1}^5$
ϑ_1^2	$u_4^1\vartheta^1$	$\eta_{,2}^5$
ϑ_1^3	u_4^7	$\eta_{,3}^5$
ϑ_1^4	$u_2^1\vartheta^1$	$\eta_{,4}^5$
ϑ_1^5	u_4^4	$\eta_{,5}^5$
ϑ_2^1	$-2u_4^1\vartheta^3$	$\eta_{,1}^4$
ϑ_2^2	0	$\eta_{,2}^4$
ϑ_2^3	$u_4^1\vartheta^1$	$\eta_{,3}^4$
ϑ_2^4	u_4^4	$\eta_{,4}^4$
ϑ_2^5	0	$\eta_{,5}^4$
ϑ_3^1	$-2u_4^2\vartheta^5$	0
ϑ_3^2	u_4^7	0
ϑ_3^5	u_4^2	0

Table 1.

We see at once that $u_4^2 = u_4^3 = u_4^7 = 0$ (notice that u_k^3 must vanish since we do not allow η to depend on ϑ_{23}^1 and so on), and that

$$\eta^4 = a_4^4\vartheta^4 + a_3^4\frac{\vartheta^3}{(\vartheta^1)^2} + c^4,$$

$$\eta^5 = a_4^4\vartheta^5 + a_3^4\frac{\vartheta^2}{(\vartheta^1)^2} + c^5,$$

where the a_k^j and c^k are now constants. Proceeding in the same way with $\alpha_7 \circ B$ we find

$$\eta^2 = a_3^3\vartheta^2 + a_4^3\frac{\vartheta^5}{(\vartheta^1)^2} + c^2,$$

$$\eta^3 = a_3^3\vartheta^3 + a_4^3\frac{\vartheta^4}{(\vartheta^1)^2} + c^3.$$

We are now left with the task of solving the expressions for $\alpha_1 \circ B$, $\alpha_2 \circ B$ and $\alpha_3 \circ B$ to find η_1 and conditions on the a_j^i and c^j.

For $\alpha_1 \circ B$ we have

$$\eta^1(a_3^3\vartheta_2^3 + a_4^3\frac{\vartheta_2^4}{(\vartheta^1)^2} - 2a_4^3\frac{\vartheta^4\vartheta_2^1}{(\vartheta^1)^3} - a_3^3\vartheta_1^2 - a_4^3\frac{\vartheta_1^5}{(\vartheta^1)^2} + 2a_4^3\frac{\vartheta^5\vartheta_1^1}{(\vartheta^1)^3})$$

$$-2(\eta^3\vartheta_2^\alpha\eta_{,\alpha}^1 - \eta^2\vartheta_1^\alpha\eta_{,\alpha}^1) = \sum_k u_1^k\alpha_k.$$

No terms in $\vartheta_3^1, \vartheta_3^2$ or ϑ_3^5 occur on the l.h.s so $u_1^7 = u_1^2 = u_1^3 = 0$. For the rest of the terms we have

$$2u_1^1\vartheta^2 = 2a_4^3\eta^1\frac{\vartheta^5}{(\vartheta^1)^3} + 2\eta^2\eta_{,1}^1,$$

$$2\eta^2\eta_{,2}^1 - a_3^3\eta^1 = -u_1^1\vartheta^1,$$

$$\eta_{,3}^1 = 0,$$

$$\eta_{,4}^1 = 0,$$

$$2\eta^2\eta_{,5}^1 - a_4^3\frac{\eta^1}{(\vartheta^1)^2} = -u_1^4,$$

$$\eta^3\eta_{,1}^1 + a_4^3\frac{\eta^1\vartheta^4}{(\vartheta^1)^3} = u_1^1\vartheta^3,$$

$$2\eta^3\eta_{,3}^1 - a_3^3\eta^1 = -u_1^1\vartheta^1,$$

$$2\eta^3\eta_{,4}^1 - a_4^3\frac{\eta^1}{(\vartheta^1)^2} = -u_1^4,$$

$$\eta^1_{,5} = 0.$$

Let us take $a^3_3 \neq 0$. Then

$$\eta^1 = u^1_1 \frac{\vartheta^1}{a^3_3}.$$

Expanding the ϑ^1_2 term we find

$$-2\frac{a^3_4}{a^3_3}u^1_1\frac{\vartheta^4}{(\vartheta^1)^2} - 2(a^3_3\vartheta^3 + a^3_4\frac{\vartheta^4}{(\vartheta^1)^2} + c_3)(u^1_{1,1}\vartheta^1 + u^1_1) = -2u^1_1\vartheta^3.$$

This leads to the assignment $u^1_1 = 0$, and to the trivial result $\eta^1 = 0$, so we take $a^3_3 = 0$ and find

$$\eta^1 = \frac{b}{\vartheta^1}.$$

This also solves $\alpha_2 \circ B$, so we examine $\alpha_3 \circ B$. This simply gives, after a little manipulation,

$$a^3_4 a^4_3 = b^2.$$

The action of the symmetries so defined, where we have set the various cs to zero, is

$$(\vartheta^1, \vartheta^2, \vartheta^3, \vartheta^4, \vartheta^5) \mapsto (\frac{\sqrt{a}c}{\vartheta^1}, \frac{a\vartheta^5}{(\vartheta^1)^2}, \frac{a\vartheta^4}{(\vartheta^1)^2}, \frac{c\vartheta^3}{(\vartheta^1)^2}, \frac{c\vartheta^2}{(\vartheta^1)^2}).$$

Written in our old variables $\phi, \rho_t, \ldots$ and transforming to $\hat{\phi}, \hat{\rho}_t, \ldots$ we have

$$\hat{\phi} = \frac{b}{\phi},$$

$$\hat{\rho}_t = a\frac{\tilde{\rho}_z}{\phi^2},$$

$$\hat{\rho}_{\tilde{z}} = a\frac{\tilde{\rho}_t}{\phi^2},$$

$$\hat{\tilde{\rho}}_t = c\frac{\rho_{\tilde{z}}}{\phi^2},$$

$$\hat{\tilde{\rho}}_z = c\frac{\rho_t}{\phi^2}.$$

and the the consistency conditions on each of the last two pairs of equations are each proportional to one of the original equations. This is perhaps easier to see from the form given in (6.55.)

7.5.2 Conclusions

We have demonstrated that the chiral equation (7.6) admits a family of generalised symmetries, and that uncovering these symmetries can appear quite laborious. However, variants of the method can be put into the form of an 'algorithm' suitable for algebraic computing, so the manipulations above need not necessarily be carried out explicitly by hand. Transformations of this form, applied to the full SDYM equations appear in [42], as discussed in chapter 6. There the effect, when combined with a suitable gauge transformation, was to build the $n+1$ monopole ansatz from the n monopole ansatz. This stepping-up of the monopole number is indeed reminiscent of the effect of Bäcklund transformations. As an application of the geometric jet bundle approach it is quite instructive. The method applied here may be used to find generalised symmetries for other equations including the closely related SDYM equations [33]. The generalised symmetries are sometimes referred to as Bäcklund transformations, but this is a little misleading, as the nature of the transformations involved is rather different.

7.6 Remarks.

As we noted above, Bäcklund transformations have many applications. Their uses are discussed extensively in [45], partly from a jet theoretic viewpoint. Whilst we have avoided a technical discussion of jet bundles, many of these aspects, together with applications to variational problems and a useful discussion of many aspects of bundle theory in general, are covered in [46]. The geometrical picture of B.T.s given here is based on [41], although the use of contact forms has been avoided here in favour of a more immediate and, hopefully, transparent jet field formulation. The interpretation of certain Bäcklund transformations in the context of the twistor treatment of ASDYM appears in [29], whilst a more standard treatment of Bäcklund transformations for the principal chiral equation is given by [37]. A fairly general examination of the type of equation possessing Bäcklund transformations appears in [11], where the jet formulation is again exploited.

Clearly, Bäcklund transformations are of great interest in the study of integrable

systems, and their applications are widespread. The geometrical picture presented here has been motivated by the philosophy that an effective setting for studies of these and other transformations is that of jet theory.

Appendix A

Multinomial Coefficients

In this appendix we briefly review the use of the multinomial coefficients in chapter 3, and derive the identities used there.

The multinomial coefficients arise in many combinatorial problems. The use which will concern us is the expansion

$$(x_1 + x_2 + \ldots + x_k)^n = \sum_{l_1+l_2+\ldots l_k=n} \binom{n}{l_1 l_2 \ldots l_k} x_1^{l_1} x_2^{l_2} \ldots x_k^{l_k}. \tag{A.1}$$

The object

$$\binom{n}{l_1 l_2 \ldots l_k} \equiv \frac{n!}{l_1! l_2! \ldots l_k!}$$

is called a multinomial coefficient.

From (A.1) we can derive a large number of identities: for our purposes there are just two essential ones. First, set $x_1 = x_2 = \ldots = x_k = 1$. This gives

$$k^n = \sum_{l_1+l_2+\ldots l_k=n} \binom{n}{l_1 l_2 \ldots l_k},$$

which is used to find the $Q^{(m)}$s for $\exp * \alpha x$.

If k=3, put $x_1 = 1,\ x_2 = 1,\ x_3 = -1$. Then

$$1 = \sum_{l_1+l_2+l_3=n} \binom{n}{l_1 l_2 l_3} (-1)^{l_3}.$$

This identity is used in 3.5.2.

The special case $k = 2$ yields the binomial coefficients, which are usually written $\binom{n}{l}$ instead of $\binom{n}{l_1 l_2}$.

We adopt the convention that all coefficients vanish when any of the components is non-integral, when n is less than 1, and when any $l_r > n$, although other conventions may be consistently defined.

Appendix B

Code for Discrete Systems.

```
      PROGRAM MNPQS
C This program generates the symbols M
C                                      ijk
C
C        as defined in the text.
C
      INTEGER I,J,K,N,P,Q
      DOUBLE PRECISION U,V,W,S,TOT
C
      INTRINSIC ABS
      EXTERNAL TRINOM,BINOM
C
      OPEN(10,FILE="mnp.dat")
      OPEN(11,FILE="mnp1.dat")
      DO 10,N=0,10
       DO 20,P=0,N
        TOT=0.0D0
        DO 30,Q=0,N-P
         Z=0.0D0
         DO 40,K=(N+1)/2,N
          DO 50,I=0,2*K-N
           DO 60,J=0,2*K-N
           IF (((I-P)/2)*2.EQ.(I-P)) THEN
           IF (((J-Q)/2)*2.EQ.(J-Q)) THEN
           CALL TRINOM(2*K-N,I,J,U)
           CALL BINOM(I,(I-P)/2,V)
           CALL BINOM(J,(J-Q)/2,W)
           CALL BINOM(K,2*K-N,S)
           Z=S*V*W*U*(2**(2*K-N-I-J))*((-1)**(K-I-J))+Z
           END IF
           END IF
  60       CONTINUE
  50      CONTINUE
  40     CONTINUE
C        IF (ABS(Z).GE.(0.9D0)) THEN
         WRITE(10,900)Z
```

```
C         END IF
          IF (Q.EQ.0) THEN
          TOT=TOT+Z
          ELSE
          TOT=TOT+2*Z
          END IF
  30     CONTINUE
         WRITE(11,990)N,P,TOT
  20    CONTINUE
  10   CONTINUE
       STOP
 900   FORMAT(F10.1)
 990   FORMAT(2I4,F10.1)
       END
*******************************************************
       SUBROUTINE BINOM(A,B,C)
C  Generates the binomial coefficient a!/b!(a-b)!
       INTEGER A,B,TOP,B1,B2
       DOUBLE PRECISION C
       IF ((B.GT.A).OR.(B.LT.0)) THEN
        C=0
       ELSE
        CALL FACT(A,TOP)
        CALL FACT(B,B1)
        CALL FACT(A-B,B2)
        C=(1.0D0*TOP)/(B1*B2)
       END IF
       RETURN
       END
*******************************************************
       SUBROUTINE TRINOM(A,B,D,C)
C  Generates the trinomial coefficient a!/b!d!(a-b-d)!
       INTEGER A,B,TOP,B1,B2,B3,D
       DOUBLE PRECISION C
       IF (((B+D).GT.A).OR.(B.LT.0).OR.(D.LT.0)) THEN
        C=0
       ELSE
        CALL FACT(A,TOP)
        CALL FACT(B,B1)
        CALL FACT(D,B2)
        CALL FACT(A-B-D,B3)
        C=(1.0D0*TOP)/(B1*B2*B3)
       END IF
       RETURN
       END
*******************************************************
       SUBROUTINE FACT(A,B)
C   Find the factorial of a (small !) integer.
       INTEGER A,B,J
       IF (A.EQ.0) THEN
        B=1
```

```
      ELSE
       B=1
       DO 10,J=1,A
        B=B*J
10     CONTINUE
      END IF
      RETURN
      END
```

Appendix C

REDUCE code.

```
 %      Code For Generation Of O.D.E For Twistor Matrix      $
 %                                                           $
 %     this version determines an o.d.e for the twistor      $
 %  matrix F, and also the form of the correcting matrix to  $
 %            be applied to the output of Flin2.f            $
 %                                                           $
ON FORT$               % turns on FORTRAN output mode. This  $
OUT "nZpert.f"$        % section simply writes out a FORTRAN $
WRITE"       PROGRAM NINETYM"; % program $
WRITE"C   ..........variables.............";
WRITE"       INTEGER IW,N,GAS,THEB,TMAX,GMAX  ";
WRITE"       PARAMETER(TMAX=23,GMAX=1*TMAX) ";
WRITE"       DOUBLE PRECISION X,XEND,ATOL(1),RTOL(1),GRANG ";
WRITE"       DOUBLE PRECISION Y(8),GA,TH,COMM(5),TWO, ";
WRITE"      .CONST(6),COUT(16),CIN(7),W(220),PI,GG,TT       ";
WRITE"C   ........parameters.....................";
WRITE"       PARAMETER (PI=3.141592653589793,GG=1.0D0*GMAX,";
WRITE"      . TT=1.0D0*TMAX,TWO=2.0D0,XEND=1.0D0, ";
WRITE"      . GRANG=8.0D0)";
WRITE"       INTEGER MBANDS(2),J,IFAIL,IWK(8)";
WRITE"C   ......external elements.................";
WRITE"       EXTERNAL D02EJY,FCN,NAGDIF";
WRITE"       INTRINSIC DCOS,DSIN";
WRITE"C   ..........file..........................";
WRITE"       OPEN(10,FILE=""ftn10"")";
WRITE"C ";
WRITE"C   the points GA are chosen to be the zeros of";
WRITE"C        a suitable Tchebychev polynomial ";
WRITE"C ";
WRITE"       DO 10,GAS=1,GMAX";
WRITE"        DO 20,THEB=1,TMAX";
WRITE"C ";                          % advice on parameter choice $
WRITE"C  GRANG is the range of gamma, and its choice ";
WRITE"C     is important as singularities in the ";
WRITE"C  twistor matrix may occur for GRANG too large. ";
WRITE"C   The value of GRANG used here must be carried ";
```

```
WRITE"C          through Flin2.f and cor.f. ";
WRITE"C ";
WRITE"        GA=GRANG*(DCOS((TWO*GAS-1.0D0)*PI/(TWO*GG)))";
WRITE"     .+GRANG";
WRITE"        TH=THEB*TWO*PI/TT ";
WRITE"        N=8 ";              % number of equations for $
WRITE"        IW=220";            % each gamma-theta point. $
WRITE"        X=0.0D0";
WRITE"        CIN(1)=0.0D0";
WRITE"        RTOL(1)=1.0D-10";
WRITE"        ATOL(1)=1.0D-11";
WRITE"C   .....set initial values.....................";
WRITE"        Y(1)=1.0D0";
WRITE"        Y(2)=0.0D0";
WRITE"        Y(3)=0.0D0  ";
WRITE"        Y(4)=0.0D0  ";
WRITE"        Y(5)=0.0D0  ";
WRITE"        Y(6)=0.0D0  ";
WRITE"        Y(7)=1.0D0  ";
WRITE"        Y(8)=0.0D0 ";
WRITE"        IFAIL=-1       ";
WRITE"C   .......call NAG routine.....................";
WRITE"        CALL NAGDIF(X,XEND,N,Y,CIN,RTOL,ATOL,FCN,COMM,";
WRITE" CONST,COUT,""Full"",MBANDS,D02EJY,W,IW,IWK,IFAIL,GA,TH)";
WRITE"C   ...........................................";
WRITE"C     WRITE(10,*)""CIN""";
WRITE"C     DO 5,J=1,6     ";      % These quantities are only $
WRITE"C     WRITE(10,140)J,CIN(J) ";% relevant for bug-chasing.$
WRITE"C  5  CONTINUE       ";
WRITE"C     WRITE(10,*)""COMM""    ";
WRITE"C     DO 6,J=1,4     ";
WRITE"C     WRITE(10,140)J,COMM(J) ";
WRITE"C  6  CONTINUE       ";
WRITE"C     WRITE(10,*)""CONST""";
WRITE"C     DO 7,J=1,6     ";
WRITE"C     WRITE(10,140)J,CONST(J)";
WRITE"C  7  CONTINUE       ";
WRITE"        DO 30,J=1,8  ";
WRITE"          WRITE(10,120)Y(J) ";
WRITE" 30     CONTINUE        ";
WRITE"C   ......if non-zero,error possible...";
WRITE"C       WRITE(10,130)""IFAIL VALUE"",IFAIL  ";
WRITE" 20    CONTINUE";
WRITE" 10   CONTINUE";
WRITE"      CLOSE(10)      ";
WRITE"      STOP           ";
WRITE" 120  FORMAT(F22.15,X)";
WRITE" 130  FORMAT(I1)     ";
WRITE" 140  FORMAT(I1,F22.15)  ";
WRITE"      END            ";
WRITE"*****************************************";
```

```
WRITE"      SUBROUTINE FCN(SIG,P,F,GA,TH)";
WRITE"      DOUBLE PRECISION SIG,GA,TH,  ";
WRITE"     .P(8),F(8),KB(2,4)   ";
WRITE"      DOUBLE COMPLEX X,Y,I,JOX(2,2),JOY(2,2),";
WRITE"     .JOT(2,2),AMAT(2,2),INVJ(2,2),XPROD(2,2), ";
WRITE"     .YPROD(2,2),DTPROD(2,2),INNER(2,2)";
WRITE"      INTRINSIC DCOS,DSIN,DREAL,DIMAG,DCMPLX  ";
WRITE"      EXTERNAL PRODCOM ";
WRITE"      I=DCMPLX(0.0D0,1.0D0) ";
 %                                                  $
 %                 *  NOTE  *                       $
 %                                                  $
 % the points p and phat used here are relevant     $
 % to the generation of the correcting matrix.      $
 %                                                  $
 %                                                  $
 zp:=i*ga*la$
 zbp:=0$
 zph:=0$
 zbph:=-i*ga*(la**(-1))$
 xp:=(zp+zbp)/2$
 yp:=(zp-zbp)/(2*i)$
 xph:=(zph+zbph)/2$
 yph:=(zph-zbph)/(2*i)$
 la:=cos(th)-i*sin(th)$
 x:=xp*(1-sig)+xph*sig;
 y:=yp*(1-sig)+yph*sig;
 clear x,y$
OFF FORT$    % this part generates the coefficient matrix $
OUT T$        %       to be fed into the NAG routine     $
    MATRIX J,KX,KY,KT,INV$
    MATRIX JX,JY,JT,INVJ$
    MATRIX JOX,JOY,JOT $
    FF:=(X+I*Y)$
    FB:=(X-I*Y)$
    RSQ:=FF*FB$
    J:=MAT((I*(1-RSQ)/(1+RSQ),I*FF*2/(1+RSQ)),
           (2*I*FB/(1+RSQ),I*(RSQ-1)/(1+RSQ)))$
ON FORT$             % this is where the correcting $
OUT "corpert.f"$          %    matrix is built $
WRITE"      PROGRAM CORRECTJ      ";
WRITE"C     ***************       ";
WRITE"C  Reads in J and premultiplies by  ";
WRITE"C      a correcting matrix.   ";
WRITE"C    ";
WRITE"      INTEGER J,K,INDEX  ";
WRITE"      DOUBLE PRECISION JMAT(2,4),WY,GGH,PI,XMAX,TZ, ";
WRITE"     . GRANG,GMZ,TT,GAMAX,GAMIN ";
WRITE"C   ";
WRITE"C TT is the value of TO used in Flin2.f ";
WRITE"C   ";
```

```
WRITE"      DOUBLE COMPLEX JJJ(2,2),COR(2,2),   ";
WRITE"     .I,NEWJ(2,2)    ";
WRITE"      PARAMETER (GGH=46.0D0,PI=3.1415926535898,   ";
WRITE"     . GRANG=8.0D0,TT=4.05D0) ";
WRITE"      INTRINSIC DCMPLX,DREAL,DIMAG,DCOS,DSQRT   ";
WRITE"      EXTERNAL PRODCOM    ";
WRITE"      OPEN(27,FILE=""ftn20"")  ";
WRITE"      OPEN(34,FILE=""ftn34"")  ";
WRITE"      OPEN(33,FILE=""ftn33"")  ";
WRITE"C  ";
WRITE"      I=DCMPLX(0.0D0,1.0D0)   ";
WRITE"      TZ=0.0D0    ";
WRITE"      GAMAX=GRANG*(DCOS(PI/GGH)+1.0D0) ";
WRITE"      GAMIN=GRANG*(DCOS((GGH-1.0D0)*PI/GGH)+1.0D0) ";
WRITE"      GMZ=MAX(ABS(GAMAX),ABS(GAMIN)) ";
WRITE"      XMAX=0.25D0*DSQRT(2.0D0)*(GMZ-2.0D0*TT) ";
WRITE"      DO 20,J=1,23   ";
WRITE"       DO 25,K=1,23   ";
WRITE"        WY=XMAX*K/GGH   ";
WRITE"        DO 40,L=1,2        ";
WRITE"         DO 50,M=1,2       ";
WRITE"          READ(27,950)JMAT(L,2*M-1)  ";
WRITE"          READ(27,950)JMAT(L,2*M)    ";
WRITE"          JJJ(L,M)=DCMPLX(JMAT(L,2*M-1),JMAT(L,2*M)) ";
WRITE"  50     CONTINUE    ";
WRITE"  40    CONTINUE    ";
WRITE"C     ";
    CORZ:=sub(X=XP,Y=YP,T=0,J)$
    COR:=SUB(GA=2*(TZ-WY),TH=PI,CORZ);
WRITE"C     ";
WRITE"C     ";
WRITE"        CALL PRODCOM(COR,JJJ,NEWJ)  ";
WRITE"        DO 80,L=1,2   ";
WRITE"         DO 90,M=1,2   ";
WRITE"          WRITE(33,990)COR(L,M)  ";
WRITE"          WRITE(34,950)DREAL(NEWJ(L,M))  ";
WRITE"          WRITE(34,950)DIMAG(NEWJ(L,M))  ";
WRITE"  90     CONTINUE    ";
WRITE"  80    CONTINUE    ";
WRITE"  25   CONTINUE    ";
WRITE"  20  CONTINUE    ";
WRITE"      CLOSE(27)   ";
WRITE"      CLOSE(34)   ";
WRITE"      STOP        ";
WRITE" 950  FORMAT(F22.15)  ";
WRITE" 990  FORMAT(2(F22.15,X))  ";
WRITE"      END     ";
OFF FORT$
OUT T$
    SIGMA3:=MAT((1,0),(0,-1))$
    KX:=MAT((DF(J(1,1),X),DF(J(1,2),X)),
```

```
            (DF(J(2,1),X),DF(J(2,2),X)))$
    KY:=MAT((DF(J(1,1),Y),DF(J(1,2),Y)),
            (DF(J(2,1),Y),DF(J(2,2),Y)))$
    JX:=SUB(T=0,KX)$
    JY:=SUB(T=0,KY)$
  % JT:=(FF*(JX+I*JY)+FB*(JX-I*JY))*1/(4*(1+RSQ))$
  % JT:=I*(1.0/(1+RSQ))*J*SIGMA3$
    JT:=MAT((0,0),(0,0))$
    INV:=SUB(T=0,J**(-1))$
ON FORT$
OUT "nZpert.f"$            % the output is largely left $
    JOX:=JX;              % for FORTRAN to manipulate $
    JOY:=JY;            % to avoid overlong expressions $
    JOT:=JT;                    % in nZ.f $
    INVJ:=INV;
WRITE"      CALL PRODCOM(INVJ,JOX,XPROD)";
WRITE"      CALL PRODCOM(INVJ,JOY,YPROD)";
WRITE"      CALL PRODCOM(INVJ,JOT,DTPROD)";
WRITE"      DO 10,L=1,2 ";
WRITE"       DO 20,J=1,2 ";
 %                                          $
 %                * NOTE   *                 $
 %                                          $
 %    In AMAT the value of b is *****       $
 %   AMAT must be determined separately     $
 %      and inserted here if needed.        $
 %    Changes will be needed above in this  $
 %                case                      $
 %                                          $
 %                                          $
WRITE"        INNER(L,J)=((1.0-DCOS(TH))*XPROD(L,J) ";
WRITE"     .   +DSIN(TH)*(DTPROD(L,J)+YPROD(L,J)))/2.0D0";
WRITE"        AMAT(L,J)=-GA*I*INNER(L,J)";
WRITE"        KB(L,2*J-1)=DREAL(AMAT(L,J)) ";
WRITE"        KB(L,2*J)=DIMAG(AMAT(L,J)) ";
WRITE"  20   CONTINUE ";
WRITE"  10  CONTINUE ";
WRITE"         F(1)=KB(1,1)*P(1)"; % This is the linear$
WRITE"     .          -KB(1,2)*P(2) "; %system to be $
WRITE"     .          +KB(1,3)*P(5) "; %solved by the$
WRITE"     .          -KB(1,4)*P(6)";  %NAG routine.$
WRITE"         F(2)=KB(1,1)*P(2)";
WRITE"     .          +KB(1,2)*P(1) ";
WRITE"     .          +KB(1,3)*P(6) ";
WRITE"     .          +KB(1,4)*P(5)";
WRITE"         F(3)=KB(1,1)*P(3)";
WRITE"     .          -KB(1,2)*P(4) ";
WRITE"     .          +KB(1,3)*P(7) ";
WRITE"     .          -KB(1,4)*P(8)";
WRITE"         F(4)=KB(1,1)*P(4)";
WRITE"     .          +KB(1,2)*P(3) ";
```

```
WRITE"      .           +KB(1,3)*P(8)";
WRITE"      .           +KB(1,4)*P(7)";
WRITE"          F(5)=KB(2,1)*P(1)  ";
WRITE"      .           -KB(2,2)*P(2)";
WRITE"      .           +KB(2,3)*P(5)";
WRITE"      .           -KB(2,4)*P(6)";
WRITE"          F(6)=KB(2,1)*P(2)  ";
WRITE"      .           +KB(2,2)*P(1)";
WRITE"      .           +KB(2,3)*P(6)";
WRITE"      .           +KB(2,4)*P(5)";
WRITE"          F(7)=KB(2,1)*P(3)  ";
WRITE"      .           -KB(2,2)*P(4)";
WRITE"      .           +KB(2,3)*P(7)";
WRITE"      .           -KB(2,4)*P(8)";
WRITE"          F(8)=KB(2,1)*P(4)";
WRITE"      .           +KB(2,2)*P(3)";
WRITE"      .           +KB(2,3)*P(8)";
WRITE"      .           +KB(2,4)*P(7)";
WRITE"  70  CONTINUE    ";
WRITE"      RETURN ";
WRITE"      END ";
END$
```

Appendix D

Twistor Matrix Generating Code.

```
      PROGRAM NINETYM
C   ..........variables.......................
      INTEGER IW,N,GAS,THEB,TMAX,GMAX
      PARAMETER(TMAX=11,GMAX=1*TMAX)
      DOUBLE PRECISION X,XEND,ATOL(1),RTOL(1),GRANG
      DOUBLE PRECISION Y(8),GA,TH,COMM(5),TWO,GTOP,
     .CONST(6),COUT(16),CIN(7),W(220),PI,GG,TT
C   ........parameters.......................
      PARAMETER (PI=3.141592653589793,GG=1.0D0*GMAX,
     . TT=1.0D0*TMAX,TWO=2.0D0,XEND=1.0D0,GTOP=100.0D0,
     . GRANG=8.0D0)
      INTEGER MBANDS(2),J,IFAIL,IWK(8)
C   ......external elements...................
      EXTERNAL D02EJY,FCN,NAGDIF
      INTRINSIC DCOS,DSIN
C   ..........file............................
      OPEN(10,FILE="ftn10")
C
C   the points GA are chosen to be the zeros of
C        a suitable Tchebychev polynomial
C
      DO 10,GAS=1,GMAX
       DO 20,THEB=1,TMAX
C
C  GRANG is the range of gamma, and its choice
C     is important as singularities in the
C  twistor matrix may occur for GRANG too large.
C   The value of GRANG used here must be carried
C          through Flin2.f and cor.f.
C
        GA=GRANG*(DCOS((TWO*GAS-1.0D0)*PI/(TWO*GG)))
     .+GRANG+GTOP
        TH=THEB*TWO*PI/TT
        N=8
```

```
        IW=220
        X=0.0D0
        CIN(1)=0.0D0
        RTOL(1)=1.0D-9
        ATOL(1)=1.0D-8
C   .....set initial values.....................
        Y(1)=1.0D0
        Y(2)=0.0D0
        Y(3)=0.0D0
        Y(4)=0.0D0
        Y(5)=0.0D0
        Y(6)=0.0D0
        Y(7)=1.0D0
        Y(8)=0.0D0
        IFAIL=-1
C   .......call NAG routine.....................
        CALL NAGDIF(X,XEND,N,Y,CIN,RTOL,ATOL,FCN,COMM,
     .CONST,COUT,"Full",MBANDS,D02EJY,W,IW,IWK,IFAIL,GA,TH)
C   ..........................................
C     WRITE(10,*)"CIN"
C     DO 5,J=1,6
C     WRITE(10,140)J,CIN(J)
C  5  CONTINUE
C     WRITE(10,*)"COMM"
C     DO 6,J=1,4
C     WRITE(10,140)J,COMM(J)
C  6  CONTINUE
C     WRITE(10,*)"CONST"
C     DO 7,J=1,6
C     WRITE(10,140)J,CONST(J)
C  7  CONTINUE
        DO 30,J=1,8
          WRITE(10,120)Y(J)
 30     CONTINUE
C   ......if non-zero,error possible...
C       WRITE(10,130)"IFAIL VALUE",IFAIL
 20    CONTINUE
 10   CONTINUE
      CLOSE(10)
      STOP
 120  FORMAT(F30.15,X)
 130  FORMAT(I1)
 140  FORMAT(I1,F30.15)
      END
*****************************************
      SUBROUTINE FCN(SIG,P,F,GA,TH)
      DOUBLE PRECISION SIG,GA,TH,
     .P(8),F(8),KB(2,4)
      DOUBLE COMPLEX X,Y,I,JOX(2,2),JOY(2,2),
     .JOT(2,2),AMAT(2,2),INVJ(2,2),XPROD(2,2),
     .YPROD(2,2),DTPROD(2,2),INNER(2,2)
```

```
      INTRINSIC DCOS,DSIN,DREAL,DIMAG,DCMPLX
      EXTERNAL PRODCOM
      I=DCMPLX(0.0D0,1.0D0)
      X=(GA*(-SIN(TH)**2*I*SIG+SIN(TH)**2*I+2.*SIN(TH)*COS
     . (TH)*SIG-2.*SIN(TH)*COS(TH)+COS(TH)**2*I*SIG-COS(TH
     . )**2*I+I*SIG))/(2.*(SIN(TH)*I-COS(TH)))
      Y=(GA*(-SIN(TH)**2*SIG+SIN(TH)**2-2.*SIN(TH)*COS(TH)
     . *I*SIG+2.*SIN(TH)*COS(TH)*I+COS(TH)**2*SIG-COS(TH)
     . **2-SIG))/(2.*(SIN(TH)*I-COS(TH)))
      JOX(1,1)=(-4.*I*X)/(X**4+2.*X**2*Y**2+2.*X**2+Y**4+
     . 2.*Y**2+1.)
      JOX(1,2)=(2.*(-I*X**2+I*Y**2+I+2.*X*Y))/(X**4+2.*X**
     . 2*Y**2+2.*X**2+Y**4+2.*Y**2+1.)
      JOX(2,1)=(2.*(-I*X**2+I*Y**2+I-2.*X*Y))/(X**4+2.*X**
     . 2*Y**2+2.*X**2+Y**4+2.*Y**2+1.)
      JOX(2,2)=(4.*I*X)/(X**4+2.*X**2*Y**2+2.*X**2+Y**4+2.
     . *Y**2+1.)
      JOY(1,1)=(-4.*I*Y)/(X**4+2.*X**2*Y**2+2.*X**2+Y**4+
     . 2.*Y**2+1.)
      JOY(1,2)=(2.*(-2.*I*X*Y-X**2+Y**2-1.))/(X**4+2.*X**2
     . *Y**2+2.*X**2+Y**4+2.*Y**2+1.)
      JOY(2,1)=(2.*(-2.*I*X*Y+X**2-Y**2+1.))/(X**4+2.*X**2
     . *Y**2+2.*X**2+Y**4+2.*Y**2+1.)
      JOY(2,2)=(4.*I*Y)/(X**4+2.*X**2*Y**2+2.*X**2+Y**4+2.
     . *Y**2+1.)
      JOT(1,1)=0.
      JOT(1,2)=0.
      JOT(2,1)=0.
      JOT(2,2)=0.
      INVJ(1,1)=(I*(X**4+2.*X**2*Y**2+Y**4-1.))/(X**4+2.*X
     . **2*Y**2+2.*X**2+Y**4+2.*Y**2+1.)
      INVJ(1,2)=(2.*(-I*X+Y))/(X**2+Y**2+1.)
      INVJ(2,1)=(-2.*(I*X+Y))/(X**2+Y**2+1.)
      INVJ(2,2)=(I*(-X**4-2.*X**2*Y**2-Y**4+1.))/(X**4+2.*
     . X**2*Y**2+2.*X**2+Y**4+2.*Y**2+1.)
      CALL PRODCOM(INVJ,JOX,XPROD)
      CALL PRODCOM(INVJ,JOY,YPROD)
      CALL PRODCOM(INVJ,JOT,DTPROD)
      DO 10,L=1,2
       DO 20,J=1,2
        INNER(L,J)=((1.0-DCOS(TH))*XPROD(L,J)
     .    +DSIN(TH)*(DTPROD(L,J)+YPROD(L,J)))/2.0D0
        AMAT(L,J)=-GA*I*INNER(L,J)
        KB(L,2*J-1)=DREAL(AMAT(L,J))
        KB(L,2*J)=DIMAG(AMAT(L,J))
20     CONTINUE
10    CONTINUE
        F(1)=KB(1,1)*P(1)
     .          -KB(1,2)*P(2)
     .          +KB(1,3)*P(5)
     .          -KB(1,4)*P(6)
```

```
      F(2)=KB(1,1)*P(2)
     .          +KB(1,2)*P(1)
     .          +KB(1,3)*P(6)
     .          +KB(1,4)*P(5)
      F(3)=KB(1,1)*P(3)
     .          -KB(1,2)*P(4)
     .          +KB(1,3)*P(7)
     .          -KB(1,4)*P(8)
      F(4)=KB(1,1)*P(4)
     .          +KB(1,2)*P(3)
     .          +KB(1,3)*P(8)
     .          +KB(1,4)*P(7)
      F(5)=KB(2,1)*P(1)
     .          -KB(2,2)*P(2)
     .          +KB(2,3)*P(5)
     .          -KB(2,4)*P(6)
      F(6)=KB(2,1)*P(2)
     .          +KB(2,2)*P(1)
     .          +KB(2,3)*P(6)
     .          +KB(2,4)*P(5)
      F(7)=KB(2,1)*P(3)
     .          -KB(2,2)*P(4)
     .          +KB(2,3)*P(7)
     .          -KB(2,4)*P(8)
      F(8)=KB(2,1)*P(4)
     .          +KB(2,2)*P(3)
     .          +KB(2,3)*P(8)
     .          +KB(2,4)*P(7)
70    CONTINUE
      RETURN
      END
```

Appendix E

Linear Splitting Code.

```
      PROGRAM LINEAR
C     **************
C                  ********************
C                 ***TT = ELEVEN***
C           TT is the number of theta points
C    used to generate the data in the input file, ftn10
C                  ********************
C
C                Linear combinations of
C       low-order Tchebychev polynomials are used
C                     to interpolate
C the twistor matrix F in gamma and find F as a function
C  of x,y and theta,then solves a linear system of eqns
C                      to find H.
C
C  Many of the parameters below are descibed in the
C       NAG manuals. Others are described in the
C              body of the program.
C
      INTEGER I,J,K,NROWS,KPLUS1,ZA,R,L,IJ,SQR,
C     -------
     .IWK1,IWK2,IK,TZH,IA,IB,IC,MT,INFO,TWOTT,FOTT,
     .NPLUS1,IFAILD,IFAILE,O,MM,DG,GG,TT,DIMN,IFAIL
C
C           GG is the number of gamma points, DG is
C            the maximum degree of the polynomial
C                       interpolant.
C
      PARAMETER (TT=11,GG=1*TT,DG=GG,SQR=GG*TT,DIMN=8*SQR)
C     ---------
      PARAMETER (IWK1=3*DG,IWK2=2*GG,TZH=(TT-1)/2,TWOTT=2*TT)
C     ---------
      INTEGER IPIV(TWOTT)
C     -------
      PARAMETER (IA=TWOTT,IB=TWOTT,IC=TWOTT,MT=2,FOTT=4*TT)
C     ---------
```

```
C.....................................................
      DOUBLE PRECISION X(DG),Y(DG),PI,W(DG),AAA(DG,DG),F(DIMN),
C     ---------------
     .TH(TT),GA,WORK1(IWK1),WORK2(IWK2),AB(DG),S(DG),Q,XX,YY,
     .FH(SQR,TT,2,4),PP(TT,2,4,DG),FF(TT,GG,2,4),TO,GAP,GAMAX,
     .GAMIN,GGH,TTH,RWORK(FOTT),ALPHA,BETA,TTS,
     .ANORM,RCOND,F06UAF,GRANG,GMZ,GTOP
      PARAMETER (GGH=2.0D0*GG,TTH=2.0D0*TT,GTOP=100.0D0,
C     ---------
     .ALPHA=1.0D0,BETA=0.0D0,GRANG=8.0D0,TTS=1.0D0*TT)
C.....................................................
      DOUBLE COMPLEX FFH(TT,2,2),OM,EE(TT,TT),BMAT(TWOTT,2),
C     --------------
     .ID(2,2),BIGMATRIX(TWOTT,TWOTT),EM(TT,TT),H(TT,2,2),
     .ZWORK(FOTT),JMAT(2,2)
C.....intrinsic functions.............................
      INTRINSIC DCOS,DSIN,DSQRT,DREAL,DCMPLX,DIMAG,
C     ---------
     . ABS,MIN
C.....external functions..............................
      EXTERNAL E02ADF,E02AEF,FINDZA,INVNA,
C     --------
C
C     The following routines are LAPACK linear algebra
C       programs and functions, used in the linear
C                     splitting.
C
     .F06UAF,F07ARF,F07AUF,F07ASF,F06ZAF
C
C     EXECUTABLES
C     -----------
      OPEN(10,FILE="ftn10")
      OPEN(20,FILE="ftn20")
      OPEN(22,FILE="ftn22")
C
C  TO is the time after t = 0, and, together with
C     the range of gamma used to generate the
C     input file ftn10, determines the ranges
C                  of x and y.
C
C     TO involves a constant multiple of GTOP which is
C            intended to keep GA in range.
C
      TO=4.05D0+GTOP*0.50625D0
      PI=3.141592653589793D0
      ID(1,1)=(1.0D0,0.0D0)
      ID(1,2)=(0.0D0,0.0D0)
      ID(2,1)=(0.0D0,0.0D0)
      ID(2,2)=(1.0D0,0.0D0)
C
C  The input file ftn10 is read in here.
```

```
C
      DO 10,IJ=1,DIMN
          READ(10,997)F(IJ)
   10 CONTINUE
C
C   See NAG manual for these parameters. They
C         are used in E02ADF, E02AEF.
C
      IFAILD=0
      IFAILE=0
      KPLUS1=DG
      NROWS=DG
      NPLUS1=DG
      MM=DG
C
C  This part interpolates the Fk data so that each Fk
C         is a function of x, y, and theta.
C
      DO 20,K=1,2
       DO 30,L=1,4
        DO 40,R=1,TT
C
C  The gamma values must be presented in non-decreasing
C         order,so the sequence is reversed.
C       Note that X(JJ) represents gamma here.
C
         DO 50,JJ=1,GG
          J=GG-JJ+1
          FF(R,J,K,L)=F(8*TT*(J-1)+8*(R-1)+4*(K-1)+L)
          X(JJ)=GRANG*(DCOS((2.0D0*J-1.0D0)*PI/GGH)+1.0D0)+GTOP
          Y(JJ)=FF(R,J,K,L)
          W(JJ)=0.1D1
   50     CONTINUE
C
C  A polynomial approximation to FF is found.This is
C                the PP() array.
C
         CALL E02ADF(MM,KPLUS1,NROWS,X,Y,W,
     .               WORK1,WORK2,AAA,S,IFAILD)
C
C  Here the degree n of the interpolating polynomial
C           is chosen:if n is too large
C         then the polynomial may fluctuate
C     too wildly between the data points.FINDZA
C           allows the user to choose n.
C
         CALL FINDZA(K,L,ZA)
         DO 60,O=1,DG
          PP(R,K,L,O)=AAA(ZA,O)
   60     CONTINUE
   40    CONTINUE
```

```
C
C  The Chebyshev polynomials are defined for
C   the interval (-1,1) and the code below
C         puts gamma into this range.
C
C        XMAX=GRANG*DSQRT(2.0D0)*DCOS(PI/GGH)
C        GAMAX=2.0D0*DCOS(PI/GGH)
C        GAMIN=2.0D0*DCOS((GGH-1.0D0)*PI/GGH)
         GAMAX=GRANG*DCOS(PI/GGH)+GRANG+GTOP
         GAMIN=GRANG*DCOS((GGH-1.0D0)*PI/GGH)+GRANG+GTOP
C
C  The following formulae for XMAX,XX and YY follow
C      directly from the null-plane equation.
C   They optimise the ranges of x and y given that
C                    of gamma.
C      However, if the fields are needed at two
C   times but at the same points, the smaller of
C        the two ranges is appropriate.
C
         GMZ=MAX(ABS(GAMAX),ABS(GAMIN))
         XMAX=0.25D0*DSQRT(2.0D0)*(GMZ-2.0D0*TO)
         DO 70,I=1,TT
          DO 80,J=1,TT
           IJ=TT*(I-1)+J
           I2=2*(I-1)-11
           J2=2*(J-1)-11
           XX=XMAX*(I2/TTH)
           YY=XMAX*(J2/TTH)
           DO 90,R=1,TT
            TH(R)=4.0D0*PI*R/TTH
C
C  the polynomial PP() is now used to interpolate
C  to find FH().Note that the x and y indices are
C             "lumped" into IJ.
C
           DO 100,IK=1,DG
            AB(IK)=PP(R,K,L,IK)
 100       CONTINUE
           GA=2*(TO+XX*DSIN(TH(R))+YY*DCOS(TH(R)))
           GAP=((GA-GAMAX)+(GA-GAMIN))/(GAMAX-GAMIN)
           CALL E02AEF(NPLUS1,AB,GAP,Q,IFAILE)
           FH(IJ,R,K,L)=Q
C
  90      CONTINUE
  80     CONTINUE
  70    CONTINUE
  30   CONTINUE
  20  CONTINUE
C
C  the array FH is changed here to the complex version FFH
C
```

```
      DO 200,IJ=1,SQR
       DO 210,R=1,TT
        DO 220,K=1,2
         DO 230,L=1,2
          FFH(R,K,L)=DCMPLX(FH(IJ,R,K,2*L-1),FH(IJ,R,K,2*L))
C         WRITE(22,998)FFH(R,K,L)
 230     CONTINUE
 220    CONTINUE
 210   CONTINUE
C
C  Here OM and EE are needed to determine the
C       linear splitting equations.
C    The next step is to set everything to
C       zero at the start of each loop.
C
       OM=DCMPLX(DCOS(4*PI/TTH),DSIN(4*PI/TTH))
       DO 300,K=1,TT
        IF  (K.EQ.TT) THEN
         BMAT(2*K-1,1)=ID(1,1)
         BMAT(2*K-1,2)=ID(1,2)
         BMAT(2*K,1)=ID(2,1)
         BMAT(2*K,2)=ID(2,2)
        ELSE
         BMAT(2*K-1,1)=DCMPLX(0.0D0,0.0D0)
         BMAT(2*K-1,2)=DCMPLX(0.0D0,0.0D0)
         BMAT(2*K,1)=DCMPLX(0.0D0,0.0D0)
         BMAT(2*K,2)=DCMPLX(0.0D0,0.0D0)
        END IF
        DO 310,L=1,TT
         BIGMATRIX(2*K-1,2*L-1)=DCMPLX(0.0D0,0.0D0)
         BIGMATRIX(2*K-1,2*L)=DCMPLX(0.0D0,0.0D0)
         BIGMATRIX(2*K,2*L-1)=DCMPLX(0.0D0,0.0D0)
         BIGMATRIX(2*K,2*L)=DCMPLX(0.0D0,0.0D0)
C
         EE(K,L)=OM**(K*L)
         EM(K,L)=1.0D0/EE(K,L)
 310    CONTINUE
 300   CONTINUE
C
C  Now a matrix suitable for expressing the splitting
C         problem in linear terms is needed.
C   The first (TT-1)/2 rows are the holomorphicity
C    conditions on H; the next (TT-1)/2 rows are
C   the holomorphicity conditions on H^ combined
C    with the splitting formula H^=FH; and the
C  final row is the gauge condition F(0)H(0)=id.
C    BIGMATRIX is the matrix into which these
C                rows are fitted.
C
       DO 500,I=1,TZH
        DO 510,J=1,TT
```

```
          BIGMATRIX(2*I-1,2*J-1)=ID(1,1)*EM(I,J)
          BIGMATRIX(2*I-1,2*J)=ID(1,2)*EM(I,J)
          BIGMATRIX(2*I,2*J-1)=ID(2,1)*EM(I,J)
          BIGMATRIX(2*I,2*J)=ID(2,2)*EM(I,J)
          BIGMATRIX(2*(TZH+I)-1,2*J-1)=EE(I,J)*FFH(J,1,1)
          BIGMATRIX(2*(TZH+I)-1,2*J)=EE(I,J)*FFH(J,1,2)
          BIGMATRIX(2*(TZH+I),2*J-1)=EE(I,J)*FFH(J,2,1)
          BIGMATRIX(2*(TZH+I),2*J)=EE(I,J)*FFH(J,2,2)
 510     CONTINUE
 500    CONTINUE
        BIGMATRIX(2*TT-1,2*TT-1)=FFH(TT,1,1)
        BIGMATRIX(2*TT-1,2*TT)=FFH(TT,1,2)
        BIGMATRIX(2*TT,2*TT-1)=FFH(TT,2,1)
        BIGMATRIX(2*TT,2*TT)=FFH(TT,2,2)
        IFAIL=0
C
C    It is useful to check the condition number of the
C      system developed above: the following LAPACK
C              routines find this number.
C     The condition number, RCOND is written to ftn22.
C
        ANORM=F06UAF('O',TWOTT,TWOTT,BIGMATRIX,TWOTT,RWORK)
        CALL F07ARF(TWOTT,TWOTT,BIGMATRIX,TWOTT,IPIV,INFO)
        CALL F07AUF('O',TWOTT,BIGMATRIX,TWOTT,ANORM,
     . RCOND,ZWORK,RWORK,INFO)
        WRITE(22,995)RCOND,INFO
C
        CALL F07ASF('N',TWOTT,MT,BIGMATRIX,IA,IPIV,
     . BMAT,IB,INFO)
        WRITE(22,997)INFO*1.0D0
        DO 600,J=1,2
         DO 650,K=1,2
          DO 700,I=1,TT
           H(I,J,K)=BMAT(2*(I-1)+J,K)
C          WRITE(20,998)H(I,J,K)
 700      CONTINUE
C
C     In order to find H at pi we form an alternating
C            sum over the existing Hi,s.
C
          JMAT(J,K)=DCMPLX(0.0D0,0.0D0)
          DO 720,I=1,TT
           DO 730,L=1,TT
            JMAT(J,K)=JMAT(J,K)+((-1)**I)*H(L,J,K)*EM(I,L)
 730       CONTINUE
 720      CONTINUE
 650     CONTINUE
 600    CONTINUE
C
C     The (uncorrected) J is written to ftn20.
C
```

```
       DO 750,J=1,2
        DO 800,K=1,2
         WRITE(20,997)DREAL(JMAT(J,K))/TTS
         WRITE(20,997)DIMAG(JMAT(J,K))/TTS
 800    CONTINUE
 750   CONTINUE
 200  CONTINUE
      CLOSE(10)
      CLOSE(20)
      CLOSE(22)
      STOP
 995  FORMAT(F30.15,X,I5)
 997  FORMAT(F30.15,X)
 998  FORMAT(2(F30.15,X))
C999  FORMAT(4I4,X,F30.15)
      END
*************************************************
      SUBROUTINE FINDZA(K,L,ZA)
C
C  See calling program for details.ZA is the degree
C       of an interpolating polynomial.
C     K and L refer to matrix components.
C
      INTEGER K,L,ZA
      IF (K.EQ.1) THEN
       IF (L.EQ.2) THEN
C*********3*************
        ZA=3
       ELSE
C*********5*************
        ZA=5
       END IF
      ELSE
       IF (L.GE.3) THEN
C*********5*************
        ZA=5
       ELSE
C*********10************
        ZA=8
       END IF
      END IF
      RETURN
      END
```

Appendix F

Final Stage Code.

```
      PROGRAM CORRECTJ
C     ****************
C  Reads in J and premultiplies by
C      a correcting matrix.
C
      INTEGER J,K,INDEX
      DOUBLE PRECISION JMAT(2,4),WY,GGH,PI,XMAX,TZ,
     . GRANG,GMZ,TT,GAMAX,GAMIN,GTOP
C
C TT is the value of T0 used in Flin2.f
C
      DOUBLE COMPLEX JJJ(2,2),COR(2,2),
     .I,NEWJ(2,2)
      PARAMETER (GGH=22.0D0,PI=3.1415926535898,
     . GRANG=8.0D0,GTOP=100.0D0,TT=4.05D0+GTOP*0.50625D0)
      INTRINSIC DCMPLX,DREAL,DIMAG,DCOS,DSQRT
      EXTERNAL PRODCOM
      OPEN(27,FILE="ftn20")
      OPEN(34,FILE="ftn34")
      OPEN(33,FILE="ftn33")
C
      I=DCMPLX(0.0D0,1.0D0)
      TZ=TT
      GAMAX=GRANG*(DCOS(PI/GGH)+1.0D0)+GTOP
      GAMIN=GRANG*(DCOS((GGH-1.0D0)*PI/GGH)+1.0D0)+GTOP
      GMZ=MAX(ABS(GAMAX),ABS(GAMIN))
      XMAX=0.25D0*DSQRT(2.0D0)*(GMZ-2.0D0*TT)
      DO 20,J=1,11
       DO 25,K=1,11
        K2=2*(K-1)-11
        WY=XMAX*K2/GGH
        DO 40,L=1,2
         DO 50,M=1,2
          READ(27,950)JMAT(L,2*M-1)
          READ(27,950)JMAT(L,2*M)
          JJJ(L,M)=DCMPLX(JMAT(L,2*M-1),JMAT(L,2*M))
```

```
  50      CONTINUE
  40     CONTINUE
C
       COR(1,1)=I
       COR(1,2)=4.*(TZ-WY)
       COR(2,1)=0.
       COR(2,2)=-I
C
C
         CALL PRODCOM(COR,JJJ,NEWJ)
         DO 80,L=1,2
          DO 90,M=1,2
           WRITE(33,990)COR(L,M)
           WRITE(34,950)DREAL(NEWJ(L,M))
           WRITE(34,950)DIMAG(NEWJ(L,M))
  90      CONTINUE
  80     CONTINUE
  25    CONTINUE
  20   CONTINUE
       CLOSE(27)
       CLOSE(34)
       STOP
 950   FORMAT(F30.15)
 990   FORMAT(2(F30.15,X))
       END
```

Appendix G

Potential Energy Density Code.

```
      PROGRAM ENERGY
C     **************
      INTEGER I,J,K,L,N,NN,KK,LL
C     PARAMETER (NN=26,KK=2,LL=4,N=8)
      PARAMETER (NN=11,KK=2,LL=4,N=8)
      DOUBLE PRECISION FF(NN,NN,KK,LL),XX(NN),
     .F(NN),DX(NN),RANGE,DY(NN),G(NN),
     .DERX(NN,NN,KK,LL),DERY(NN,NN,KK,LL),
     .POLX(N),POLY(N),RETX(N),RETY(N),SUX(N),
     .SUY(N),ENRG,JMA(N),JIN(N)
      PARAMETER (RANGE=10.0D0)
      EXTERNAL E01BEF,PRODCOM,INVNA,TRACE
C     **************
      OPEN(34,FILE="ftn34")
      OPEN(35,FILE="ftn35")
C     **************
C
C     **** read in data***
C
      DO 10,I=1,NN
       DO 20,J=1,NN
        DO 30,K=1,KK
         DO 40,L=1,LL
          READ(34,900)FF(I,J,K,L)
 40      CONTINUE
 30     CONTINUE
 20    CONTINUE
 10   CONTINUE
C
C     ***interpolate to find ***
C     ******derivatives*********
C
      DO 50 J=1,NN
       DO 60,K=1,KK
        DO 70,L=1,LL
         DO 80 I=1,NN
```

```
          XX(I)=(I/6.0D0)*DSQRT(2.0D0)*6.0D0
          F(I)=FF(I,J,K,L)
          G(I)=FF(J,I,K,L)
 80      CONTINUE
          CALL E01BEF(NN,XX,F,DX,IFAIL)
          CALL E01BEF(NN,XX,G,DY,IFAIL)
         DO 90,I=1,NN
          DERX(I,J,K,L)=DX(I)
          DERY(J,I,K,L)=DY(I)
C         WRITE(35,950)DX(I),DY(I)
 90      CONTINUE
 70     CONTINUE
 60    CONTINUE
 50   CONTINUE
      DO 100,I=1,NN
       DO 110,J=1,NN
        DO 120,K=1,KK
         DO 130,L=1,LL
          INDEX=LL*(K-1)+L
          POLX(INDEX)=DERX(I,J,K,L)
          POLY(INDEX)=DERY(I,J,K,L)
          JMA(INDEX)=FF(I,J,K,L)
C
C     ***call product,[J(-1)Jx]**2**
C     ******and take trace**********
C
 130     CONTINUE
 120    CONTINUE
        CALL INVNA(JMA,JIN)
        CALL PRODCOM(JIN,POLX,RETX)
        CALL PRODCOM(RETX,RETX,SUX)
        CALL PRODCOM(JIN,POLY,RETY)
        CALL PRODCOM(RETY,RETY,SUY)
        CALL TRACE(SUX,SUY,ENRG)
        WRITE(35,950)ENRG
 110   CONTINUE
 100  CONTINUE
      CLOSE(34)
      CLOSE(35)
      STOP
 900  FORMAT(F30.15)
 950  FORMAT(F30.15,X)
      END
****************************************
      SUBROUTINE TRACE(A,B,C)
C     ********************
C     ***takes A and B ,adds,and****
C     *****takes the trace**********
C
      INTEGER NN,I
      PARAMETER (NN=8)
```

```
      DOUBLE PRECISION A(NN),B(NN),C,
     .D(NN)
      DOUBLE COMPLEX CC
      INTRINSIC CMPLX,ABS
      DO 10,I=1,NN
       D(I)=A(I)+B(I)
 10   CONTINUE
      CC=CMPLX(D(1)+D(7),D(2)+D(8))
      C=ABS(CC)
      RETURN
      END
*****************************************
      SUBROUTINE PRODCOM(P,R,Q)
      INTEGER IZ,N,PP,M,IFAIL,OPT
      PARAMETER (M=2,PP=M,N=M,IZ=1)
      DOUBLE PRECISION A(M,N),B(M,M),
     .AB(M,M),C(M,M),D(M,M),AD(M,M),
     .CB(M,M),CD(N,N),Z(IZ)
      DOUBLE PRECISION P(8),Q(8),R(8)
      EXTERNAL F01CKF
C
      IFAIL=0
        A(1,1)=P(1)
        A(1,2)=P(3)
        A(2,1)=P(5)
        A(2,2)=P(7)
C
        C(1,1)=P(2)
        C(1,2)=P(4)
        C(2,1)=P(6)
        C(2,2)=P(8)
C
C
        B(1,1)=R(1)
        B(1,2)=R(3)
        B(2,1)=R(5)
        B(2,2)=R(7)
C
        D(1,1)=R(2)
        D(1,2)=R(4)
        D(2,1)=R(6)
        D(2,2)=R(8)
C
      OPT=1
      CALL F01CKF(AB,A,B,N,PP,M,Z,IZ,OPT,IFAIL)
      CALL F01CKF(CD,C,D,N,PP,M,Z,IZ,OPT,IFAIL)
      CALL F01CKF(AD,A,D,N,PP,M,Z,IZ,OPT,IFAIL)
      CALL F01CKF(CB,C,B,N,PP,M,Z,IZ,OPT,IFAIL)
C
        Q(1)=AB(1,1)-CD(1,1)
        Q(2)=AD(1,1)+CB(1,1)
```

```
        Q(3)=AB(1,2)-CD(1,2)
        Q(4)=AD(1,2)+CB(1,2)
        Q(5)=AB(2,1)-CD(2,1)
        Q(6)=AD(2,1)+CB(2,1)
        Q(7)=AB(2,2)-CD(2,2)
        Q(8)=AD(2,2)+CB(2,2)
C
      RETURN
      END
*********************************************
      SUBROUTINE INVNA(P,Q)
      INTEGER IA,IB,IC,N,M,IFAIL
      PARAMETER (M=2,IA=M,IB=M,IC=M,N=M)
      DOUBLE COMPLEX A(IA,N),B(IB,M),
     .WRK(2),C(IC,M)
      DOUBLE PRECISION P(8),Q(8)
      INTRINSIC DCMPLX,DREAL,DIMAG
      EXTERNAL F04ADF
        A(1,1)=DCMPLX(P(1),P(2))
        A(1,2)=DCMPLX(P(3),P(4))
        A(2,1)=DCMPLX(P(5),P(6))
        A(2,2)=DCMPLX(P(7),P(8))
C
        B(1,1)=DCMPLX(1.0D0,0.0D0)
        B(1,2)=DCMPLX(0.0D0,0.0D0)
        B(2,1)=DCMPLX(0.0D0,0.0D0)
        B(2,2)=DCMPLX(1.0D0,0.0D0)
C
      CALL F04ADF(A,IA,B,IB,N,M,C,IC,WRK,IFAIL)
C
        Q(1)=DREAL(C(1,1))
        Q(2)=DIMAG(C(1,1))
        Q(3)=DREAL(C(1,2))
        Q(4)=DIMAG(C(1,2))
        Q(5)=DREAL(C(2,1))
        Q(6)=DIMAG(C(2,1))
        Q(7)=DREAL(C(2,2))
        Q(8)=DIMAG(C(2,2))
C
      RETURN
      END
```

Appendix H

Nonlinear Splitting Code.

```
      PROGRAM FACTORL
C
C             This routine takes output
C    from the program Fgen2.f or a related program.
C    This output is in the form of a collection of
C   Fourier coefficients. Using these coefficients,
C          a splitting is performed.
C
      INTEGER N,IFAIL,LWA,I,TT,NN,SQAR,GG
C
C      As in Fgen2, the parameters TT and GG must
C                     be set.
C    The parameter TT is the number of theta points
C   used in the routine generating the twistor matrix,
C       GG is the number of gamma points.
C   Where asterisks appear in comment lines, look for
C     these parameters and set them as required.
C
C   ************************************************
      PARAMETER (N=8,TT=29,GG=1*TT,NN=N*TT,SQAR=TT*GG)
C   ************************************************
      PARAMETER (LWA=NN*(3*NN+13)/2)
      DOUBLE PRECISION C(NN),XTOL,WA(LWA),FVEC(NN),
     .SS(TT,2,4,SQAR),SST(2,4,TT),EPS,H(29,29,NN),
     .JMATRIX(8)
C
      PARAMETER (EPS=1.0D-4)
C   ........external functions.............
      EXTERNAL NONLIN,FCN,JMAKE
C   ......................................
      OPEN(20,FILE="ftn20")
      OPEN(21,FILE="ftn21")
      OPEN(15,FILE="CDATA")
C   ......read in data...................
      DO 10,IJ=1,SQAR
       DO 20,J=1,2
```

```
        DO 30,K=1,2
         DO 40,I=1,TT
          READ(20,950)SS(I,J,2*K-1,IJ),SS(I,J,2*K,IJ)
  40     CONTINUE
  30    CONTINUE
  20   CONTINUE
  10  CONTINUE
C
      DO 50,I=1,NN
       C(I)=EPS
  50  CONTINUE
C
C     Here the data is put into the SST matrices
C  for use in the NAG routine solving the equations.
C   The upper limit on II, JJ is the square root of
C    TT*GG so that a square x-y grid is obtained.
C     The index IJ will need modification if TT
C                 has been altered.
C
C   ******************************************
      DO 90,II=1,29
      DO 95,JJ=1,29
       IJ=29*(II-1)+JJ
C   ******************************************
       DO 100,J=1,2
        DO 110,K=1,4
         DO 120,I=1,TT
          IND=4*TT*(J-1)+TT*(K-1)+I
          SST(J,K,I)=SS(I,J,K,IJ)
C
C     The starting values of C(.) may be
C              altered here.
C
C         IF  (II.EQ.13) THEN
C           C(IND)=H(II-1,JJ,IND)
C          END IF
C         ELSE
            C(IND)=EPS
C         END IF
 120     CONTINUE
 110    CONTINUE
 100   CONTINUE
C
C  The following parameters can be found in the
C    NAG manual. The subroutine NONLIN is the
C       modified version of C05NBF
C
       IFAIL=0
       XTOL=1.0D-8
C
       CALL NONLIN(FCN,NN,C,FVEC,XTOL,WA,LWA,IFAIL,SST)
```

```
C
C       WRITE(15,930)IJ,IFAIL
C
C    The output returned by NONLIN is used to construct
C                the final matrix J.
C
        CALL JMAKE(C,JMATRIX)
        DO 250,I=1,8
         WRITE(21,900)JMATRIX(I)
 250    CONTINUE
        DO 230,I=1,NN
         H(II,JJ,I)=C(I)
 230    CONTINUE
        WRITE(15,930)II,JJ
  95   CONTINUE
  90   CONTINUE
       CLOSE(20)
       CLOSE(21)
       CLOSE(15)
       STOP
 900   FORMAT(F22.15)
 930   FORMAT(3I4)
 950   FORMAT(2(F22.15,X))
       END
*************************************************************
       SUBROUTINE FCN(NN,C,FVEC,IFLAG,SST)
C
C  This subroutine is used to define the relevant
C    equation to be solved by the NAG routine.
C
       INTEGER NN,IFLAG,I,J,K,TT,R,SZ,P,HTT
C
C  TT should be set to the number of theta points
C  ***********************************************
       PARAMETER (TT=29,HTT=(TT+1)/2)
C  ***********************************************
       DOUBLE PRECISION C(NN),FVEC(NN),SST(2,4,TT),
      .A(8,HTT),B(8,HTT),AB(HTT,HTT,8),S(TT,8),SMM(TT,8)
       EXTERNAL CALPRO
       IFLAG=1
       DO 100,J=1,2
        DO 110,K=1,4
         INDEX=4*(J-1)+K
         DO 130,I=1,TT
          P=I+HTT
          IF (P.GT.TT) THEN
           P=P-TT
          END IF
          S(P,INDEX)=SST(J,K,I)
C
C  A(.,.) and B(.,.) are the sets of matrices which
```

```
C  are multiplied to give equations for H and H^.
C  The S(.,.) are the required coefficients in the
C                  product.
C
          IF (I.LE.HTT) THEN
           B(INDEX,I)=C(INDEX+8*(I-1))
          ELSE
           A(INDEX,TT+1-I)=C(INDEX+8*(I-1))
          END IF
  130    CONTINUE
         IF ((INDEX.EQ.1).OR.(INDEX.EQ.7)) THEN
          A(INDEX,HTT)=1.0D0
         ELSE
          A(INDEX,HTT)=0.0D0
         END IF
  110    CONTINUE
  100   CONTINUE
C
C  CALPRO calculates the "matrix product" of its
C           first two arguments.
C
      CALL CALPRO(HTT,A,B,AB)
      DO 150,J=1,8
C
C  FVEC is to be pushed to zero by the modified
C               NAG routine.
C
       DO 160,I=1,TT
        SMM(I,J)=0.0D0
        DO 190,R=1,HTT
         DO 180,SZ=1,HTT
          IF ((R+SZ).EQ.(I+1)) THEN
           SMM(I,J)=SMM(I,J)+AB(R,SZ,J)
          END IF
 180     CONTINUE
 190    CONTINUE
 160   CONTINUE
 150  CONTINUE
      DO 200,I=1,TT
       DO 250,J=1,8
        FVEC(J+8*(I-1))=SMM(I,J)-S(I,J)
 250   CONTINUE
 200  CONTINUE
      RETURN
      END
*************************************************************
      SUBROUTINE CALPRO(HTT,A,B,AB)
C
      INTEGER HTT
      DOUBLE PRECISION A(8,HTT),B(8,HTT),AB(HTT,HTT,8)
      DO 10,I=1,HTT
```

```
        DO 20,J=1,HTT
       AB(I,J,1)=A(1,I)*B(1,J)-A(2,I)*B(2,J)+A(3,I)*B(5,J)-
      .A(4,I)*B(6,J)
       AB(I,J,2)=A(1,I)*B(2,J)+A(2,I)*B(1,J)+A(3,I)*B(6,J)+
      .A(4,I)*B(5,J)
       AB(I,J,3)=A(1,I)*B(3,J)-A(2,I)*B(4,J)+A(3,I)*B(7,J)-
      .A(4,I)*B(8,J)
       AB(I,J,4)=A(1,I)*B(4,J)+A(2,I)*B(3,J)+A(3,I)*B(8,J)+
      .A(4,I)*B(7,J)
       AB(I,J,5)=A(5,I)*B(1,J)-A(6,I)*B(2,J)+A(7,I)*B(5,J)-
      .A(8,I)*B(6,J)
       AB(I,J,6)=A(5,I)*B(2,J)+A(6,I)*B(1,J)+A(7,I)*B(6,J)+
      .A(8,I)*B(5,J)
       AB(I,J,7)=A(5,I)*B(3,J)-A(6,I)*B(4,J)+A(7,I)*B(7,J)-
      .A(8,I)*B(8,J)
       AB(I,J,8)=A(5,I)*B(4,J)+A(6,I)*B(3,J)+A(7,I)*B(8,J)+
      .A(8,I)*B(7,J)
  20    CONTINUE
  10   CONTINUE
       RETURN
       END
*************************************************************
       SUBROUTINE JMAKE(C,JMATRIX)
C
C  This program takes the output of NONLIN and
C          forms the matrix h(-1)h~(1).
C  The general structure of this program is based on
C        the splitting used in FCN above.
C
       INTEGER I,J,K,L,INDEX,HTT,TT,TOPSQ,
      .ZY,ZZ
C
C             TT appears below
C    ****************************************
       PARAMETER (TT=29,HTT=(TT+1)/2,TOPSQ=TT*8)
C    ****************************************
       DOUBLE PRECISION A(8,HTT),B(8,HTT),
      .ID(8),C(TOPSQ),HMIN(8),
      .HHAT(8),FIN(8),JMATRIX(8)
C
       EXTERNAL PRODCOM,INVNA
C
       DO 10,I=1,8
        IF ((I.EQ.1).OR.(I.EQ.7)) THEN
         ID(I)=1.0D0
        ELSE
         ID(I)=0.0D0
        END IF
  10   CONTINUE
C
       DO 50,J=1,2
```

```
        DO 60,K=1,4
         INDEX=4*(J-1)+K
         DO 70,L=1,TT
          IF (L.LE.HTT) THEN
           B(INDEX,L)=C(INDEX+8*(L-1))
          ELSE
           A(INDEX,TT+1-L)=C(INDEX+8*(L-1))
          END IF
  70     CONTINUE
         A(INDEX,HTT)=ID(INDEX)
         HMIN(INDEX)=0.0D0
         HHAT(INDEX)=0.0D0
         DO 80,L=1,HTT
          ZY=1
          ZZ=(L/2)*2
          IF (ZZ.EQ.L) THEN
           ZY=-1
          END IF
          HMIN(INDEX)=HMIN(INDEX)+B(INDEX,L)*ZY
          HHAT(INDEX)=HHAT(INDEX)+A(INDEX,L)
  80     CONTINUE
  60    CONTINUE
  50   CONTINUE
C
C  the inverses are computed by INV() and the
C     matrix product JMATRIX then found
C
       CALL PRODCOM(HHAT,HMIN,FIN)
       CALL INVNA(FIN,JMATRIX)
       RETURN
       END
*************************************************************
       SUBROUTINE PRODCOM(P,R,Q)
       INTEGER IZ,N,PP,M,IFAIL,OPT
       PARAMETER (M=2,PP=M,N=M,IZ=1)
       DOUBLE PRECISION A(M,N),B(M,M),
      .AB(M,M),C(M,M),D(M,M),AD(M,M),
      .CB(M,M),CD(N,N),Z(IZ)
       DOUBLE PRECISION P(8),Q(8),R(8)
       EXTERNAL F01CKF
C
       IFAIL=0
         A(1,1)=P(1)
         A(1,2)=P(3)
         A(2,1)=P(5)
         A(2,2)=P(7)
C
         C(1,1)=P(2)
         C(1,2)=P(4)
         C(2,1)=P(6)
         C(2,2)=P(8)
```

```
C
C
        B(1,1)=R(1)
        B(1,2)=R(3)
        B(2,1)=R(5)
        B(2,2)=R(7)
C
        D(1,1)=R(2)
        D(1,2)=R(4)
        D(2,1)=R(6)
        D(2,2)=R(8)
C
      OPT=1
      CALL F01CKF(AB,A,B,N,PP,M,Z,IZ,OPT,IFAIL)
      CALL F01CKF(CD,C,D,N,PP,M,Z,IZ,OPT,IFAIL)
      CALL F01CKF(AD,A,D,N,PP,M,Z,IZ,OPT,IFAIL)
      CALL F01CKF(CB,C,B,N,PP,M,Z,IZ,OPT,IFAIL)
C
        Q(1)=AB(1,1)-CD(1,1)
        Q(2)=AD(1,1)+CB(1,1)
        Q(3)=AB(1,2)-CD(1,2)
        Q(4)=AD(1,2)+CB(1,2)
        Q(5)=AB(2,1)-CD(2,1)
        Q(6)=AD(2,1)+CB(2,1)
        Q(7)=AB(2,2)-CD(2,2)
        Q(8)=AD(2,2)+CB(2,2)
C
      RETURN
      END
*****************************************************
      SUBROUTINE INVNA(P,Q)
      INTEGER IA,IB,IC,N,M,IFAIL
      PARAMETER (M=2,IA=M,IB=M,IC=M,N=M)
      DOUBLE COMPLEX A(IA,N),B(IB,M),
     .WRK(2),C(IC,M)
      DOUBLE PRECISION P(8),Q(8)
      INTRINSIC DCMPLX,DREAL,DIMAG
      EXTERNAL F04ADF
C
        IFAIL=0
C
        A(1,1)=DCMPLX(P(1),P(2))
        A(1,2)=DCMPLX(P(3),P(4))
        A(2,1)=DCMPLX(P(5),P(6))
        A(2,2)=DCMPLX(P(7),P(8))
C
        B(1,1)=DCMPLX(1.0D0,0.0D0)
        B(1,2)=DCMPLX(0.0D0,0.0D0)
        B(2,1)=DCMPLX(0.0D0,0.0D0)
        B(2,2)=DCMPLX(1.0D0,0.0D0)
C
```

```
      CALL F04ADF(A,IA,B,IB,N,M,C,IC,WRK,IFAIL)
C
        Q(1)=DREAL(C(1,1))
        Q(2)=DIMAG(C(1,1))
        Q(3)=DREAL(C(1,2))
        Q(4)=DIMAG(C(1,2))
        Q(5)=DREAL(C(2,1))
        Q(6)=DIMAG(C(2,1))
        Q(7)=DREAL(C(2,2))
        Q(8)=DIMAG(C(2,2))
C
      RETURN
      END
```

References

[1] M.F. Ablowitz, S. Chakravarty, L.A. Takhtajan, Comm. Math. Phys. 158 (1993) 289

[2] M.F. Ablowitz, P.A. Clarkson, *Solitons, Nonlinear Evolution Equations and the Inverse Scattering Transform*, C.U.P. (1991)

[3] M.J. Ablowitz, A. Ladik, J. Math. Phys. 16 (1975) 598

[4] M.J. Ablowitz, A. Ladik, J. Math. Phys. 17 (1976) 1011

[5] A.A. Andrianov, M.V. Ioffe, D.N. Nishianidze, *Polynomial SUSY in Quantum Mechanics and second derivative Darboux transformations* Preprint SPbU-IP-94-05 (1994)

[6] T.N. Bailey, R.J. Baston, *Twistors in Mathematics and Physics*, C.U.P. (1990)

[7] L. Bonora, G. Mussardo, A. Schwimmer, L.Girardello, M. Martinelli, (Eds), *Integrable Quantum Field Theories*, Plenum Press (New York) (1993)

[8] Y. Bouguenaya, D.B. Fairlie, J. Phys. A 19 (1986) 1049

[9] S. Chakravarty, M.J. Ablowitz, P. Clarkson, Phys. Rev. Lett. 65 (1990) 1085

[10] E.F. Corrigan, P. Goddard, Comm. Math. Phys 80 (1981) 575

[11] P. Denes, J.D. Finley III, Physica 9D (1983) 236

[12] R. Dodd, A. Fordy, Proc. R. Soc. Lond. A 385 (1983) 389

[13] J.F. Donoghue, E. Golowich, B.R. Holstein, Dynamics of the Standard Model, C.U.P. (1992)

[14] P.G. Drazin, R.S. Johnson, *Solitons: an Introduction*, Cambridge University Press (1989)

[15] S.D. Drell, M. Weinstein, P. Yankielowicz, Phys. Rev. D 14 (1976) 1627

[16] A.P. Fordy, (Editor) *Soliton Theory- a Survey of Results*, Manchester University Press (1990)

[17] P. Forgacs, Z. Horvath, L. Palla, Phys. Rev. D. 23 8 (1981) 1876

[18] C.S. Gardner, J.M. Greene, M.D. Kruskal, R.M. Miura, Phys. Rev. Lett, 19 (1967) 1095

[19] D. Han, Y.S. Kim, W.W. Zachary (Eds), *Workshop on Harmonic Oscillators* NASA conference publication 3197 (1993)

[20] R.J. Herman, Math. Phys. 24 (3) (1983) 510

[21] N.J. Hitchen, Comm. Math. Phys. 83 (1982) 579

[22] S.A. Huggett, K.P. Tod, *An Introduction to Twistor Theory*, C.U.P. (1985)

[23] E. Infeld, G.Rowlands, *Nonlinear waves, solitons and chaos*, C.U.P. (1990)

[24] B. Kuperschmidt, Geometry of Jet Bundles and the Structure of Lagrangian and Hamiltonian Formalisms, *in Geometric Methods in Mathematical Physics*, Springer Lecture Notes in Mathematics 775, Springer-verlag, Berlin (1980).

[25] G.L. Lamb, *Elements of Soliton Theory*, (1980)

[26] L.D. Landau, E.M. Lifshitz, *Quantum Mechanics: nonrelativistic theory*, Oxford Pergamon Press (1977)

[27] R.J. Leese, Math. Phys. 30 (1989) 2072

[28] A.N. Leznov,*Backlund transformation for integrable systems* Preprint IHEP 92-87 (1992)

[29] L. Mason, S. Chakravarty, E.T. Newman, J. Math. Phys. 29 (1988) 4

[30] L.J. Mason, G.A.J. Sparling, J. Geom. Phys 8 (1992) 243

[31] L.J. Mason, N.M.J. Woodhouse, Nonlinearity 6 (1993) 569

[32] V.B. Mateev, M.A. Salle, *Darboux Transformations and Solitons*, Springer-Verlag (Berlin) (1991)

[33] H.C. Morris, J.Math. Phys. 21 (1980) 256

[34] H.C. Morris, J.Math. Phys. 21 (1980) 327

[35] M.K. Murray, Math. Ann. 272 (1985) 99

[36] NAG Fortran Library Mark 15, The Numerical Algorithms Group Ltd (1991)

[37] A.T. Ogielski, M.K. Prasad, A. Sinha, L-L. Chau Wang, Phys. Lett. 91B 3,4 (1980) 387

[38] P.J. Olver, *Application of Lie Groups to Differential Equations*, Springer-Verlag, Berlin (1986)

[39] R. Penrose, J. Math. Phys 8 (1967) 345

[40] R. Penrose, W. Rindler, *Spinors and Spacetime* Vol.2 C.U.P. (1986)

[41] F.A.E Pirani, D.C. Robinson, W.F. Shadwick, Math. Phys. Stud. 1 (1979)

[42] M.K. Prasad, Comm. Math. Phys. 80 (1981) 137

[43] M.K. Prasad, Physica 1D (1980) 167

[44] G. Rayna, *REDUCE: software for algebraic computation* Springer- Verlag (New York) (1987)

[45] C. Rogers, W.F. Shadwick, *Backlund Transformations and their Applications*, (1982)

[46] D.J. Saunders, *The Geometry of Jet Bundles*, C.U.P. (1989)

[47] J. Schiff *Self Dual Yang-Mills and the Hamiltonian Structures of Integrable Systems* Preprint IASSNS-HEP-92/34 (1992)

[48] H. Stephani, *Differential Equations, their Solution using Symmetries*, C.U.P. (1989)

[49] P.M. Sutcliffe, Phys. Rev. D 47 (1993) 5470

[50] P.M. Sutcliffe, J. Math. Phys 33 (1992) 2269

[51] P.M. Sutcliffe, Nonlinearity 4 (1991) 1109

[52] L.A.Takhtajan in O. Babelon, P. Cartier, Y. Kosmann-Schwarzbach (Eds), *Integrable Systems, The Verdier Memorial Conference*, Birkhauser (Boston) (1993)

[53] A.M. Vinogradov, J. Geom. Phys. 14 (1994) 146

[54] H.D. Wahlquist, F.B. Estabrook, J. Math. Phys. 16 (1975) 1

[55] R.S. Ward, Phs. Lett. 61 A (1977) 81

[56] R.S. Ward, Comm. Math. Phys. 80 (1981) 563

[57] R.S. Ward, R.O. Wells Jnr, *Twistor Geometry and Field Theory*, C.U.P. (1990)

[58] R.S. Ward, Comm. Math. Phys. 128 (1990) 319

[59] R.S. Ward, J. Math. Phys. 30 (10) (1989) 2246

[60] R.S. Ward, J. Math. Phys 29 (2) (1988) 386

[61] R. S. Ward, Nonlinearity 1 (1988) 671

[62] R.S. Ward, Phys. Lett. A 165 (1992) 325

[63] N.J.M. Woodhouse, Class. Quantum Grav. 2 (1985) 257

[64] C.N. Yang, Phys. Rev. Lett. 38 (1977) 1377

[65] B.N. Zakhariev, A.A. Suzko, *Direct and Inverse Problems*, Springer-Verlag (Berlin) (1990)