

# Algorithm for determining $U(1)$ charges in free fermionic heterotic string models

William Hicks,\* Lesley Vestal,† Jared Greenwald, Douglas Moore, Timothy Renner, and Gerald Cleaver  
*CASPER, Department of Physics, Baylor University, Waco, TX, 76798-7316, USA*

(Dated: July 31, 2011)

To assist in the search for phenomenologically realistic models in the string landscape, we must develop tools for investigating all gauge charges, including  $U(1)$  charges, in string models. We introduce the model-building process and present an algorithm for determining the  $U(1)$  gauge states and  $U(1)$  charges in weakly-coupled free fermionic heterotic string (WCFHFS) models given their matter and non-Abelian gauge content. We determine the computational complexity of this algorithm and discuss a particular implementation that can be used in conjunction with a framework developed at Baylor University for building WCFHFS models. We conclude by considering various avenues of investigation to which this algorithm may be applied.

## I. INTRODUCTION

String theory models have the potential to completely describe both the particle content and field interactions of our universe, yet attempts to obtain precise, testable predictions based on string theory have been stymied by the sheer number of viable models. Work by a number of theorists in the early 2000's helped establish that the number of stable models, though finite, could be upwards of  $10^{100}$  [1–3]. Any predictions based on string theory will require a more complete understanding of this collection of models, known as the string theory landscape. In particular, systematic searches of the landscape can help us gain a statistical understanding of the range of available models. Such searches, however, require fast algorithms for determining properties of string models. In this paper, we discuss a new algorithm used to determine the  $U(1)$  charges of particles in weakly coupled free fermionic heterotic string (WCFHFS) models. Such models are of particular interest due to a number of studies which have discovered phenomenologically realistic models of this type [4–38].

### A. WCFHFS model building

For our investigation of  $U(1)$  charges, we built on research conducted at Baylor University to develop a fast, versatile framework for building WCFHFS models [39]. The process of model-building begins with a set of 64-component basis vectors which specify information about the shape of the world-sheet as well as a valid GSO (Gliozzi, Scherk, and Olive) projection matrix, which is used to eliminate non-physical particle states encountered during model-building. The 64 components of the

basis vectors are divided into two parts, the first of which specifies information about the left-movers on the world sheet and the second of which specifies information about the right-movers. The number of components in each part is determined by the number of large spacetime dimensions. For four large spacetime dimensions, the left-moving part consists of 20 components while the right-moving part consists of 44 components. The 64 components of the entire vector are usually labeled as shown:

$$\left( \psi^{1,2}(x, y, w)^{1,\dots,6} \parallel \bar{\psi}^{1,\dots,10} \bar{\eta}^{1,\dots,6} \bar{y}^{1,\dots,6} \bar{w}^{1,\dots,6} \bar{\phi}^{1,\dots,16} \right). \quad (1)$$

Additionally, each input vector has an order  $N$ , which specifies the denominator of all components in the basis vector. These components take on values between  $-1$  and  $1$ .

These vectors must obey a number of constraints to constitute valid basis vectors. One particularly useful constraint specifies that every component in the entire vector must be paired with an equal component in order for the vector to be valid. This allows us to represent these vectors in a complex basis using only half as many components, an important advantage for manipulating them in calculations. In the description of our algorithm, we will use a complex basis to describe all vectors.

The input basis vectors must also obey certain constraints to guarantee that the generated model satisfies the condition of modular invariance. This condition ensures that modular transformations on the world-sheet will not alter the physics of the model. The specific constraints on the input basis vectors  $\{\vec{\alpha}_i^B\}$  with orders  $\{N_i\}$  are given by

$$N_i (\vec{\alpha}_i^B)^2 = 0 \pmod{16} \text{ for even } N \quad (2)$$

$$N_i (\vec{\alpha}_i^B)^2 = 0 \pmod{8} \text{ for odd } N \quad (3)$$

$$N_{ij} \vec{\alpha}_i^B \cdot \vec{\alpha}_j^B = 0 \pmod{8}, \quad (4)$$

where  $N_{ij}$  is the least common multiple of  $N_i$  and  $N_j$ .

Given a set of valid basis vectors, we construct sectors  $\{\vec{\alpha}_i\}$  by taking all linear combinations of the form

$$\vec{\alpha} = \sum m_j \alpha_j^B \quad (5)$$

\*William.Hicks@Brown.edu; Department of Physics, Brown University, Providence, RI 02912, USA

†lvestal@ubc.interchange.ca; Department of Physics and Astronomy, University of British Columbia, Okanagan, Kelowna, British Columbia V1V 1V7, Canada

where the  $m_j$  obey

$$m_j \in \mathbb{Z} \cap [0, N_j]. \quad (6)$$

The sectors determine how the free world-sheet fermions  $f_i$  transform when transported around a non-contractible loop as given by

$$f \longrightarrow e^{-i\pi\alpha} f. \quad (7)$$

This information is determined by the topology of the world-sheet. All possible charges or particle states  $Q^{\vec{\alpha}}$  may then be constructed from these sectors as given by

$$Q^{\vec{\alpha}} = \frac{1}{2}\vec{\alpha} + \vec{F}, \quad (8)$$

where the components of the fermion number operator  $\vec{F}$  take on values given by

$$\vec{F}_p \in \{1, 0, -1\}. \quad (9)$$

We maintain only states that are massless at the string scale, because these are the states that exist at low energies and therefore represent observable particles. We exclude string-scale massive particles because they will not be observable. Additionally, we apply the GSO projection to remove non-physical states such as tachyons. This projection ensures the remaining gauge states fit together to form a group theoretic root structure, compatible with the remaining matter states [40]. The GSO coefficient matrix  $k_{ij}$  is subject to modular invariance constraints of its own, given by [39]

$$N_j k_{ij} = 0 \pmod{2} \quad (10)$$

$$k_{ij} + k_{ji} = \frac{1}{2}\vec{\alpha}_i^B \cdot \vec{\alpha}_j^B \pmod{2} \quad (11)$$

$$k_{ii} + k_{i1} = \frac{1}{4}\vec{\alpha}_i^B \cdot \vec{\alpha}_i^B - s_i \pmod{2}. \quad (12)$$

Given a valid GSO coefficient matrix, we reject any particle states failing the requirement

$$\vec{\alpha}_i^B \cdot \vec{Q}^\alpha = \sum m_j k_{ij} + s_i \pmod{2} \quad (13)$$

where  $s_i$  equals 0 if  $\alpha_i^B$  is a bosonic state, and 1 if it is fermionic, and  $m_j$  and  $\alpha$  are defined as in ??.

We can identify the type of particle that a particular state specifies by considering the components of the left-moving part for that state. Specifically, for fermions

$$\psi^1 = \psi^2 = \frac{1}{2}, \quad (14)$$

and for gauge bosons

$$\psi^1 = \psi^2 = 1. \quad (15)$$

Furthermore, all other left-moving components of gauge states are equal to zero. Thus, if the right-moving part of the  $U(1)$  gauge states can be generated, adding the left-moving part is straightforward. Once all non-physical states have been removed from the model, we can determine the  $U(1)$  charges of the remaining physical matter states.

## B. $U(1)$ charges in WCCFFHS models

In order to fully understand the gauge interactions in a particular string model, we must determine the  $U(1)$  charges of the matter states. This point is perhaps best illustrated by considering the role of  $U(1)$  charges in the familiar context of the Standard Model. In the Standard Model, the charge of a particle under  $U(1)$  determines that particle's hypercharge, which in turn determines its electric charge through the Gell-Mann-Nishijima formula [41]

$$Q = I_3 + \frac{Y}{2}, \quad (16)$$

where  $Q$  is the electric charge of the particle,  $I_3$  is an eigenvalue of its isospin, and  $Y$  is its hypercharge.

In addition to completing the picture of the gauge interactions for a model, analyzing  $U(1)$  charges can help eliminate unwanted exotic particles from a model - in particular fractionally charged exotic particles. If a certain type of  $U(1)$  charge known as an anomalous  $U(1)$  charge is present in a model, we can break that  $U(1)$ , providing a mass to the exotic particles at the string scale and thereby eliminating them from the low-energy effective field theory [24, 25]. This can help dramatically in searching for phenomenologically realistic models, but identifying models of this type requires an efficient method for calculating  $U(1)$  charges.

## II. ALGORITHM FOR DETERMINING $U(1)$ CHARGES

In order to determine the  $U(1)$  charges for the matter states in a string model, we must first construct the  $U(1)$  gauge states. These gauge states, by definition, have a number of convenient mathematical properties that simplify their construction. Firstly, because they are gauge states, their left moving parts are already determined, as noted in section IA. Secondly, we know that the right-moving part of the  $U(1)$  states must be orthogonal to the right-moving parts of the simple roots of the non-Abelian gauge groups in the model. If the state were not orthogonal to the simple roots of any one of the non-Abelian gauge groups, it would be a part of that gauge group and not the  $U(1)$  gauge group. Finally, since representations of  $U(1)$  have only a single generator, every  $U(1)$  gauge group in the model has exactly one gauge state and hence, the right-moving parts of every  $U(1)$  gauge state must be orthogonal to the right-moving part of every other  $U(1)$  gauge state. Thus, if we can construct a set of vectors which are orthogonal to the right-moving parts of the simple roots of the non-Abelian gauge groups in a model as well as to each other, we will have constructed the right-moving parts of all  $U(1)$  gauge states in the model. Once the gauge states have been constructed, we take the dot product of the right moving part of each of the matter states with the right-moving part of the  $U(1)$

gauge states to determine its charge under each  $U(1)$  in the model.

Throughout our discussion of the algorithm for generating  $U(1)$  gauge states, we will use a complex basis for all vectors, as described in section IA. So, we can represent every pair of equal components in the real basis, using a single component in the complex basis. To simplify our discussion of the problem, we denote the set of vectors given by the right-moving parts of the simple roots of the non-Abelian gauge groups by  $\mathbf{V}_{SR}$ . We denote the vectors whose components are given by the right-moving parts of the  $U(1)$  gauge states by  $\mathbf{V}_{U(1)}$ . Furthermore, we denote the cardinality of  $\mathbf{V}_{SR}$  by  $N_{SR}$  and the cardinality of  $\mathbf{V}_{U(1)}$  by  $N_{U(1)}$ .

### A. Outline of the algorithm

Our algorithm for generating  $U(1)$  gauge states begins with the set  $\mathbf{V}_{SR}$  for a particular model. The dimension  $D_{RM}$  of vectors in this set is given by

$$D_{RM} = 26 - D, \quad (17)$$

where  $D$  denotes the number of large spacetime dimensions for the model [39]. Since the vectors of  $\mathbf{V}_{SR}$  are formed from the simple roots of gauge groups, they are by definition linearly independent. Furthermore, since  $\mathbf{V}_{U(1)}$  contains all vectors orthogonal to the vectors of  $\mathbf{V}_{SR}$  and to each other,  $\mathbf{V}_{SR} \cup \mathbf{V}_{U(1)}$  forms a complete basis. Therefore,

$$N_{U(1)} = D_{RM} - N_{SR}. \quad (18)$$

If  $D_{RM} = N_{SR}$ , then the model in question contains no  $U(1)$  gauge groups.

We start by attempting to generate any vectors in  $\mathbf{V}_{U(1)}$  whose components all have value 0 except for one component which takes on the value 1. This can easily be done by constructing a matrix whose rows correspond to the components of the vectors in  $\mathbf{V}_{SR}$  and performing Gauss-Jordan elimination on that matrix. After elimination, we identify all columns containing only 0's, and we generate vectors in  $\mathbf{V}_{U(1)}$  which consist of all 0's except for the component corresponding to one of these columns. Gauge states of this form are known as external gauge states. We then proceed to generate any remaining vectors of  $\mathbf{V}_{U(1)}$ .

Let  $n_{U(1)}$  denote the number of vectors in  $\mathbf{V}_{U(1)}$  which have already been generated, where  $n_{U(1)}$  may range from 0 to  $N_{U(1)}$ . In order to construct an additional vector  $\mathbf{v}$  of  $\mathbf{V}_{U(1)}$  from  $\mathbf{V}_{SR}$  and the  $n_{U(1)}$  previously generated vectors, we must ensure that the dot product of  $\mathbf{v}$  with every vector in  $\mathbf{V}_{SR}$  as well as with every previously generated vector in  $\mathbf{V}_{U(1)}$  is equal to 0. Since there are  $D_{RM}$  components, this gives us a system of  $N_{SR} + n_{U(1)}$  linear constraint equations on  $D_{RM}$  unknowns. This means that the system has  $f$  free variables,

where  $f$  is given by

$$f = D_{RM} - N_{SR} - n_{U(1)}. \quad (19)$$

Thus we, can assign arbitrary values to  $f$  components of  $\mathbf{v}$  and then straightforwardly solve for the remaining components of  $\mathbf{v}$  by solving this system of equations. If we let  $F$  denote the value of  $f$  when  $n_{U(1)}$  equals 0, this further implies that the vectors of  $\mathbf{V}_{U(1)}$  will be unique only up to scaling and rotations in  $F$  dimensions.

Although we must assign arbitrary values to  $f$  components of  $\mathbf{v}$ , we are not entirely free in choosing the components to which we will be assigning values. This is perhaps most easily illustrated by considering the problem of finding a vector  $\mathbf{w}$  orthogonal to the following (3-dimensional) vectors:

$$\begin{pmatrix} 1 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix} \quad \begin{pmatrix} 2 & 1 & 1 \\ 3 & 3 & 3 \end{pmatrix} \quad (20)$$

Since there are two vectors given, the system of constraint equations will consist of two equations in three unknowns. Thus, we may arbitrarily assign a value to one component of  $\mathbf{w}$ . Let  $w_i$  denote the  $i$ th component of  $\mathbf{w}$ , where  $i$  ranges from 1 to 3. If we choose to assign a value of 1 to  $w_1$ , then (taking the dot product of  $\mathbf{w}$  with the given vectors), the constraint equations are

$$\begin{aligned} \frac{2}{3}w_2 + \frac{2}{3}w_3 &= -\frac{1}{3} \\ \frac{w_2}{3} + \frac{w_3}{3} &= -\frac{2}{3} \end{aligned} \quad (21)$$

Attempting to solve for  $w_2$  and  $w_3$ , we see that the constraint equations reduce to:

$$\begin{aligned} 0 &= 3 \\ 0 &= -\frac{3}{2} \end{aligned} \quad (22)$$

The system is clearly inconsistent. A quick inspection of the original vectors shows that this occurred because they were not linearly independent in the subspace formed by the components for which we were solving.

In order to avoid this problem in generating a vector  $\mathbf{v}$  in  $\mathbf{V}_{U(1)}$ , we construct a matrix whose rows are determined by the components of the vectors in  $\mathbf{V}_{SR}$  as well as the vectors in  $\mathbf{V}_{U(1)}$  which have already been generated. Then, we perform Gaussian elimination on that matrix. After Gaussian elimination, we identify those columns which do not contain pivots and assign values to the corresponding components of  $\mathbf{v}$ . We can safely do this since the vectors are now guaranteed to form linearly independent subspaces in the remaining components.

To illustrate this method, consider the previous example of generating the 3-dimensional vector  $w$ . This time, we begin by creating a matrix from the given vectors:

$$\begin{pmatrix} \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \quad (23)$$

After performing Gaussian elimination, this gives

$$\begin{pmatrix} 2 & 1 & 1 \\ 0 & \frac{3}{2} & \frac{3}{2} \end{pmatrix} \quad (24)$$

Since the third column of this matrix does not contain a pivot, we can safely assign a value (arbitrarily chosen to be 1) to  $w_3$ . Proceeding as before, the system of equations for the remaining components is given by

$$\begin{aligned} w_1 + 2w_2 &= -2 \\ 2w_1 + w_2 &= -1. \end{aligned} \quad (25)$$

This system is consistent (as expected), and we get:

$$\mathbf{w} \equiv (0, -1, 1) \quad (26)$$

After assigning all necessary values to the components of  $\mathbf{v}$  and solving for the remaining components, we add  $\mathbf{v}$  to the set of generated vectors and repeat until we have produced the  $N_{U(1)}$  vectors in  $\mathbf{V}_{U(1)}$ . These vectors specify the components of the right-moving parts of the  $U(1)$  gauge states, and we can trivially prepend the necessary left-moving part to make the full state a gauge state. Then, taking the Lorentz dot product of each  $U(1)$  gauge state with each matter state in the model, we produce the charges of each matter state under each  $U(1)$  gauge group in the model.

### B. Implementation details and additional considerations

Our implementation of the algorithm was written as an extension of a framework developed at Baylor University to build WCFFHS models [39]. Given a set of valid input vectors as well as a valid GSO coefficient matrix, this framework generates the matter and non-Abelian gauge content for the model. Our algorithm then generates the  $U(1)$  charges for the generated matter states. The entire framework, including our implementation of the algorithm, is written in C++, and care is taken throughout to avoid floating numeric types in order to eliminate rounding errors. The framework was written with extensibility and readability in mind, since we hope that other researchers will eventually be able to use this software to further explore the string landscape. In order to guide future development, a few choices in the details of our implementation deserve particular consideration.

In our implementation, Gauss-Jordan elimination is used to solve for the unassigned components in the generated vectors. Since Gaussian or Gauss-Jordan elimination is required for two other steps in the algorithm, this allows for significant code re-use, but a few downsides to Gauss-Jordan elimination must also be considered. Firstly, Gaussian elimination is not a numerically stable algorithm, so we must consider the possibility that the program would overflow the C++ long integer type

used to store the coefficients of the systems of equations. In order to minimize this possibility, we implemented partial pivoting, which substantially improves the stability of Gaussian elimination [42]. We then tested the program by asking it to produce a complete orthogonal basis from a single randomly generated 22-dimensional vector. This goes beyond the worst case that our implementation would attempt to solve, since all non-Abelian gauge groups that could appear in WCFFHS models have at least two generators. After running several hundred thousand trials, we determined that so long as the absolute value of the components in the randomly generated vector remained below about 250 (a very reasonable assumption), the program never overflowed the long integer type. Furthermore, since this test went well beyond the worst case that our implementation will reasonably encounter, the absolute value of the components of the input vectors can likely range substantially higher without difficulty.

The second downside to Gauss-Jordan elimination is that it is typically somewhat slower than other common methods for solving systems of linear equations [42]. This downside is balanced by a number of factors, including the fact that Gauss-Jordan elimination with pivoting is usually slightly more stable than most other general solution methods, and the fact that Gaussian or Gauss-Jordan elimination is required in other steps of the algorithm. Furthermore, since major goals of our implementation included readability and extensibility, the use of a single easily-understood algorithm was deemed preferable to several faster, more esoteric algorithms. Nevertheless, if future advances improve the speed of generating the matter and non-Abelian gauge content of string models to the point that the process of generating  $U(1)$  charges becomes a bottleneck, other solution methods should be considered.

### III. ANALYSIS OF COMPUTATIONAL COMPLEXITY

In order to facilitate analysis of the computational complexity of the algorithm for generating all  $U(1)$  gauge states, we consider each significant step in the process. Using the notation of section II, this process begins with Gauss-Jordan elimination on an  $N_{SR} \times D_{RM}$  matrix to find any external  $U(1)$  gauge states. Gauss-Jordan elimination is generally  $\mathcal{O}(n^2m)$  on the number of rows ( $n$ ) and columns ( $m$ ) in a matrix, but our implementation includes a sorting step after each elimination step in order to maintain optimal pivoting. At worst, this sort will be an  $\mathcal{O}(D_{RM}N_{SR}^2)$  operation, but is likely to be closer to or better than the average case, which is an  $\mathcal{O}(D_{RM}N_{SR} \log N_{SR})$  operation. Thus, the entire operation of Gauss-Jordan elimination with pivoting will have complexity

$$\mathcal{O}(D_{RM}N_{SR}^3) \quad (27)$$

for the worst case.

Next, we must generate all internal  $U(1)$  gauge states. If we let  $N_{\text{ext}}$  denote the number of external  $U(1)$  gauge states, then there will be  $D_{RM} - N_{SR} - N_{\text{ext}}$  such states. For each state, we must perform Gaussian elimination to determine to which components we may assign arbitrary values. We must also perform Gauss-Jordan elimination to determine the values of the remaining components. In the first of these steps, the sorting process is simplified by the fact that the sorted results from the previous iteration can be stored and used to analyze the next vector to be constructed. Thus, the worst case for the Gaussian elimination step will be:

$$\mathcal{O}\left(D_{RM} (N_{SR} + n_{U(1)})^2\right), \quad (28)$$

where  $n_{U(1)}$  is the number of previously generated  $U(1)$  gauge states, including the external  $U(1)$  gauge states. The Gauss-Jordan elimination step will have no such advantage. Thus, using the same analysis as before, the worst case complexity for this step will be

$$\mathcal{O}\left((D_{RM} - (N_{SR} + n_{U(1)})) (N_{SR} + n_{U(1)})^3\right). \quad (29)$$

Both the Gaussian and Gauss-Jordan elimination steps will be repeated until  $n_{U(1)} = D_{RM} - N_{SR}$ . The complexity of this step will be at a maximum when

$$N_{SR} + n_{U(1)} = \frac{3}{4}D_{RM}. \quad (30)$$

Thus the overall worst case complexity of the algorithm for generating  $U(1)$  gauge states will be

$$\mathcal{O}(D_{RM}^4). \quad (31)$$

Therefore, the overall worst case complexity depends solely on the number of right-moving components in the states of the model, which is itself determined by the number of large spacetime dimensions.

#### IV. CONCLUSIONS

In this paper, we presented a new algorithm for calculating  $U(1)$  charges of particles in WCCFH models

given the matter and non-Abelian gauge content for those models. We analyzed the computational complexity, and found it to be  $\mathcal{O}(D_{RM}^4)$  in the worst case, where  $D_{RM}$  refers to the number of right-moving components of the particle states in a complex basis. We also discussed an implementation of this algorithm developed as an extension of a model-building framework developed at Baylor University which emphasizes readability, extensibility, and speed [39].

Determining the  $U(1)$  charges for a particular model is important in understanding its gauge content, but it can also help in constructing phenomenologically realistic models by identifying anomalous  $U(1)$  charges. Additionally, constructing  $U(1)$  charges may shed some light on the problem of identifying unique models, since multiple inputs may all lead to models with the same matter and non-Abelian gauge content. While some studies have been conducted to analyze patterns in the occurrence of unique models, it is possible that by studying the  $U(1)$  gauge content, we will find that some models with the same matter and non-Abelian gauge content are actually distinct due to their varying  $U(1)$  charges [39]. Indeed, in a preliminary study of one thousand layer-1 gauge models, we found that almost all contained states which were unique only in their  $U(1)$  charges. As of this writing, we are running a search of approximately one million layer-1 gauge models in order to gain a more complete picture of trends in  $U(1)$  uniqueness.

Like many aspects of the string landscape, we cannot predict everything that the  $U(1)$  gauge content of string models will tell us. Only through systematic exploration of the landscape can we completely understand how  $U(1)$  charges affect the search for phenomenologically realistic models.

#### ACKNOWLEDGMENTS

We would like to thank the CASPER group at Baylor University for their sponsorship of the REU program under which this research was conducted. We would also like to thank the NSF for funding this research through grant PHY-1002637.

- 
- [1] R. Bousso and J. Polchinski, *JHEP* **06** (2000), arXiv:hep-th/0004134v3.
  - [2] S. K. Ashok and M. R. Douglas, *JHEP* **01** (2004), arXiv:hep-th/0307049v3.
  - [3] M. R. Douglas, *JHEP* **05** (2003), arXiv:hep-th/0303194v4.
  - [4] G. B. Cleaver, A. E. Faraggi, D. V. Nanopoulos, and J. W. Walker, *Nucl. Phys.* **B593**, 471 (2001), arXiv:hep-ph/9910230.
  - [5] J. L. Lopez, D. V. Nanopoulos, and K.-j. Yuan, *Nucl. Phys.* **B399**, 654 (1993), arXiv:hep-th/9203025.
  - [6] A. E. Faraggi, D. V. Nanopoulos, and K.-j. Yuan, *Nucl. Phys.* **B335**, 347 (1990).
  - [7] A. E. Faraggi, *Nucl. Phys.* **B387**, 239 (1992), arXiv:hep-th/9208024.
  - [8] I. Antoniadis, G. K. Leontaris, and J. Rizos, *Phys. Lett.* **B245**, 161 (1990).
  - [9] G. K. Leontaris and J. Rizos, *Nucl. Phys.* **B554**, 3 (1999),

- arXiv:hep-th/9901098.
- [10] A. E. Faraggi, Phys. Lett. **B278**, 131 (1992).
  - [11] A. E. Faraggi, Nucl. Phys. **B403**, 101 (1993), arXiv:hep-th/9208023.
  - [12] A. E. Faraggi, Nucl. Phys. **B407**, 57 (1993), arXiv:hep-ph/9210256.
  - [13] A. E. Faraggi, Phys. Lett. **B274**, 47 (1992).
  - [14] A. E. Faraggi, Phys. Rev. **D47**, 5021 (1993).
  - [15] A. E. Faraggi, Phys. Lett. **B377**, 43 (1996), arXiv:hep-ph/9506388.
  - [16] A. E. Faraggi, Nucl. Phys. **B487**, 55 (1997), arXiv:hep-ph/9601332.
  - [17] G. B. Cleaver, Nucl. Phys. Proc. Suppl. **62**, 161 (1998), arXiv:hep-th/9708023.
  - [18] G. B. Cleaver and A. E. Faraggi, Int. J. Mod. Phys. **A14**, 2335 (1999), arXiv:hep-ph/9711339.
  - [19] G. Cleaver, M. Cvetič, J. R. Espinosa, L. L. Everett, and P. Langacker, Nucl. Phys. **B525**, 3 (1998), arXiv:hep-th/9711178.
  - [20] G. Cleaver, M. Cvetič, J. R. Espinosa, L. L. Everett, and P. Langacker, Nucl. Phys. **B545**, 47 (1999), arXiv:hep-th/9805133.
  - [21] G. Cleaver *et al.*, Phys. Rev. **D59**, 055005 (1999), arXiv:hep-ph/9807479.
  - [22] G. Cleaver *et al.*, Phys. Rev. **D59**, 115003 (1999), arXiv:hep-ph/9811355.
  - [23] G. B. Cleaver(1998), arXiv:hep-ph/9812262.
  - [24] G. B. Cleaver, A. E. Faraggi, and D. V. Nanopoulos, Phys. Lett. **B455**, 135 (1999), arXiv:hep-ph/9811427.
  - [25] G. B. Cleaver, A. E. Faraggi, and D. V. Nanopoulos, Int. J. Mod. Phys. **A16**, 425 (2001), arXiv:hep-ph/9904301.
  - [26] G. B. Cleaver, A. E. Faraggi, D. V. Nanopoulos, and J. W. Walker, Nucl. Phys. **B593**, 471 (2001), arXiv:hep-ph/9910230.
  - [27] G. B. Cleaver, A. E. Faraggi, D. V. Nanopoulos, and J. W. Walker, Mod. Phys. Lett. **A15**, 1191 (2000), arXiv:hep-ph/0002060.
  - [28] G. B. Cleaver, A. E. Faraggi, and C. Savage, Phys. Rev. **D63**, 066001 (2001), arXiv:hep-ph/0006331.
  - [29] G. B. Cleaver, A. E. Faraggi, D. V. Nanopoulos, and J. W. Walker, Nucl. Phys. **B620**, 259 (2002), arXiv:hep-ph/0104091.
  - [30] G. B. Cleaver, D. J. Clements, and A. E. Faraggi, Phys. Rev. **D65**, 106003 (2002), arXiv:hep-ph/0106060.
  - [31] G. B. Cleaver, A. E. Faraggi, and S. Nooij, Nucl. Phys. **B672**, 64 (2003), arXiv:hep-ph/0301037.
  - [32] G. B. Cleaver(2002), arXiv:hep-ph/0210093.
  - [33] G. Cleaver *et al.*, Phys. Rev. **D67**, 026009 (2003), arXiv:hep-ph/0209050.
  - [34] J. Perkins *et al.*(2003), arXiv:hep-ph/0310155.
  - [35] J. Perkins *et al.*, Phys. Rev. **D75**, 026007 (2007), arXiv:hep-ph/0510141.
  - [36] G. B. Cleaver, A. E. Faraggi, E. Manno, and C. Timirgaziu, Phys. Rev. **D78**, 046009 (2008), arXiv:0802.0470 [hep-th].
  - [37] J. Greenwald, D. Moore, T. R. Kristen Pechan, T. Ali, and G. Cleaver(2009), arXiv:0912.5207 [hep-ph].
  - [38] G. Cleaver *et al.*(2011), arXiv:1105.0447 [hep-ph].
  - [39] T. Renner, *Initial Systematic Investigations of the Weakly Coupled Free Fermionic Heterotic String Landscape Statistics*, PhD dissertation, Baylor University, Department of Physics, Baylor University, Waco, TX, 76798-7316, USA (July 2011), pending publication as of July 27, 2011.
  - [40] G. C. M. Robinson and M. Hunziker, Mod. Phys. Lett. A **24** (2009), arXiv:hep-th/0809.5094.
  - [41] K. Gottfried and V. F. Weisskopf, *Concepts of Particle Physics*, Vol. 1 (Oxford University Press, Inc., New York, 1984).
  - [42] W. T. V. William H. Press, Saul A. Teukolsky and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. (Cambridge University Press, New York, 1992).