ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

INTERPRETATION OF NUCLEAR MULTIFRAGMENTATION DATA IN THE FRAMEWORK OF PERCOLATION MODELS

By

Marko Kleine Berkenbusch

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Physics and Astronomy

2001

UMI Number: 1403876

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT

INTERPRETATION OF NUCLEAR MULTIFRAGMENTATION DATA IN THE FRAMEWORK OF PERCOLATION MODELS

By

Marko Kleine Berkenbusch

Phase transitions occur in a great variety of physical systems on all length and energy scales. One of the smallest systems which is believed to show critical behavior in the sense of phase transitions is the atomic nucleus. Due to the size of the nucleus, the experimental investigation of these phenomena is extremely difficult. One class of experiments employed in such an analysis is multifragmentation reactions.

This work deals with the application of a statistical percolation model of multifragmentation to the analysis of an experimental data set obtained by the ISiS collaboration.

We discuss the importance of considering physical limitations of the detection procedure in modeling the experimental results. Strong indications of the existence of a liquid-gas type phase transition in the data set are found. A method for the determination of two critical exponents, σ and τ, of the transition is introduced in the framework of the percolation model. This method is applied to the experimental data to obtain numerical values of the exponents. It is demonstrated that sequential decay processes have to be taken into account in this analysis.

ACKNOWLEDGMENTS

First of all, I would like to express my thanks to Prof. Wolfgang Bauer for guiding my research as my advisor and for giving me the opportunity to work in the nuclear theory group at the National Superconducting Cyclotron Laboratory at Michigan State University, East Lansing. I enjoyed the friendly and open atmosphere of this facility very much throughout this year.

I would also like to thank the German National Merit Foundation, the Studienstiftung des Deutschen Volkes, for the financial support I received during this year.

Finally, I want to thank my family for the support they gave me during my studies throughout the years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Critical phenomena and phase transitions are common features in a wide variety of physical systems. They carry extremely important functions on all time, length and energy scales in our world. Everyone is familiar with two prominent cases of a liquid-gas and a solid-liquid phase transition we encounter in everyday life: the boiling of water and the melting of ice. There are also less obvious examples like the boiling of an egg or other bio-chemical polymerization processes that are crucial for biological life and that do not just take place in highly specialized laboratory environments.

What these phenomena have in common is that the system under consideration undergoes a profound qualitative change when a freely adjustable parameter crosses a certain threshold; in the example of boiling water or melting ice, the adjustable parameter is the temperature.

Although the aforementioned phenomena seem to be significantly different from each other, they share deep connections on a more abstract level. A good theoretical understanding of these underlying similarities is not only of academic interest, but may also result in practical applications of great importance.

One system in a length and energy scale that is not as easily perceived in everyday experience is the nucleus of an atom. It is assumed that nuclear matter undergoes at least two distinct phase transitions. Even though practical applications of a theory of phase transitions in nuclear matter seem to be rare, such a theory might help to solve some of the persistent problems in understanding the structure of matter.

In this thesis, we will deal with a phase transition of nuclear matter. In particular, the framework of percolation theory will be used to analyze data that have been collected in high energy nuclear reactions. The easily adjustable environment of a model will help to interpret the experimental findings.

In Chapter 2, a short overview of percolation theory and critical phenomena in general will be given, and a model of nuclear fragmentation reactions based on percolation theory will be described. Chapter 3 will address the application of this theory to the interpretation of a set of data obtained in proton/pion - Au collision experiments at the AGS accelerator facility of the Brookhaven National Laboratory. Some emphasis will be put on the question of whether traces of a phase transition can be detected in the collected data. The appendix gives some parts of the source code used to implement the percolation model and to analyze the data.

A typographical convention will be that important terminology is typeset in *italic style* when it is introduced for the first time.

Chapter 2

Overview of Percolation Theory

Percolation models are some of the simplest physical models for systems with many degrees of freedom which exhibit a whole range of critical phenomena that are widely studied in statistical physics. In particular, percolation models provide an easy means to study the typical behavior associated with phase transitions. While the first model of this type was introduced by Flory [22] and Stockmayer [43] in 1941 and 1943, respectively, the term *Percolation Theory* was only coined in 1957 by Broadbent and Hammersley [8].

In the following sections, we will give a short introduction to the different types of percolation models and present the most important results in this field.

2.1 Basic Definitions

The main feature of most types of percolation models is a graph in d-dimensional space. In many cases – including the nuclear lattice model, which will be discussed in this thesis in further detail – the graph can be associated with $\mathbf{Z}^d$ (where $\mathbf{Z}$ denotes the set of integers). We will call the points of this graph *vertices* or *sites*. In the case of $\mathbf{Z}^d$, these vertices would be given by coordinate vectors $x = (x_1, x_2, \ldots, x_d)$ with integer components x_i. The connection between two vertices x and y will be denoted by $\langle x, y\rangle$ and called *edge* or *bond*. In standard percolation theory, only edges between nearest neighbors (with respect to the **1**-norm) are considered. The number z of interacting neighbors of a site is called the *coordination number* of the lattice. For $\mathbf{Z}^d$, for example, we have the coordination number $z = 2 \times d$.

2.1.1 Bond Percolation

For typographical convenience, we will introduce the term *percolation tuple* for (at the moment) a pair of two numbers (p, q) with the properties $p,\ q \in [0, 1]$ and $p + q = 1$. These numbers will be interpreted as probabilities.

In bond percolation, we start by introducing a percolation tuple (p, q). We then randomly assign the property of being *closed* (with probability p) or *open* (with the corresponding probability $q = 1 - p$) to all edges of the graph under consideration. In the first case we will call the two end-vertices of an edge *connected*; in the second case we will call them *not connected*. It is important to mention here that the states of all the edges of the graph are statistically independent.

Based on these conventions, we define a *bond-cluster* as a subset of vertices of the graph in which every pair $\{x, y\}$ of two vertices of the subset can be connected by a path of closed edges. We will write $x \leftrightarrow y$ for this situation. In the case $x \not\leftrightarrow y$, i.e., when there is no connecting path between the two vertices, they do not belong to the same bond-cluster. The *size* s of a bond-cluster is defined as, as for all the following definitions of clusters, the number of sites that belong to that specific cluster.

As an example of this rather abstract model one could think of the following:

Imagine an oil field. Typically the oil (or gas) is enclosed in porous rocks. In our simple model we want to assume that the pores in the rock which hold the oil can be represented by the vertices of our graph. The property of an edge between two vertices to be closed will then be interpreted as an open connection between the corresponding pores that allows the oil to flow between them. When one now starts to drill into this rock and pump oil from one of the pores (or a group of pores), the question arises: to how many other pores is this pore connected, i.e., how big is the cluster the pore belongs to? As we will see later, percolation theory provides us with information about the distributions and general properties of these clusters and thus could help to evaluate the chances of "hitting" a big oil field.

2.1.2 Site Percolation

We consider a situation similar to the one above, but this time we assume that all edges are closed a priori. Again, we choose a percolation tuple (p, q). Now we will not apply the probability p to the edges of the graph to determine if they are open or closed, but rather to the sites in order to assign the state of being *occupied* or *empty* to them. A *site-cluster* will now be defined as a subset of connected occupied sites of the graph. This means that every member vertex of a given cluster that consists of more than just one single vertex has at least on e occupied nearest neighbor belonging to the same cluster. A single-vertex site-cluster must therefore be surrounded by empty sites. This corresponds to an isolated black black square in Figure 2.1. For a system with a higher density of occupied sites see Figure 2.2.

To illustrate this idea, we could think of two types of balls, say metal and plastic balls, being tightly packed in one layer on a flat surface (thus forming a 2-dimensional *hexagonal* lattice). We distribute these two types randomly according to the probabilities p and q. In this situation, the question of electric conductivity of the lattice is apparently directly connected to the structure of the "metal-ball site-clusters".

Many results of percolation theory hold for both bond and site percolation. It has been shown by Stauffer in 1979 [42] that the two models belong to the same universality class, which means that they share the same set of critical exponents (see Sections 2.2.5 and 2.3.3). Although, historically, bond percolation was developed first, every bond percolation model is equivalent to a site percolation problem (possibly on a different lattice and with consideration of correlations).

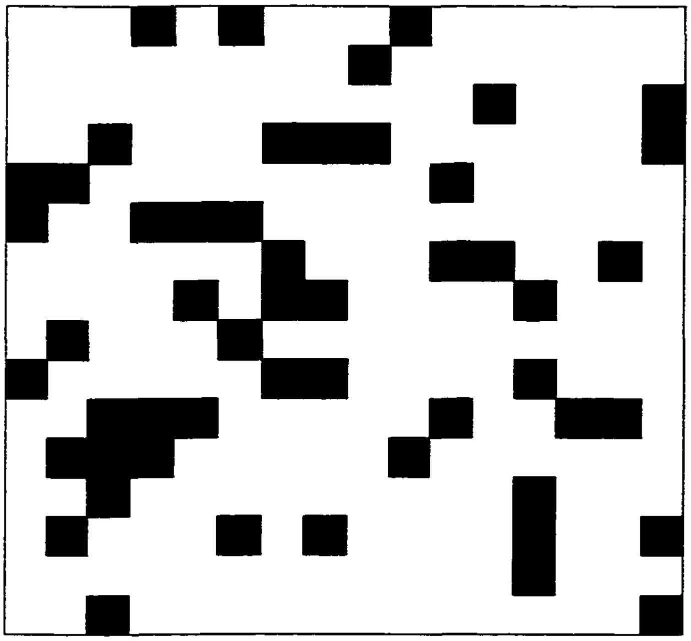

Figure 2.1: Example of a typical site percolation configuration with a density of p=0.2

2.1.3 Site-Bond Percolation

The two aforementioned concepts of percolation can be combined into the more general concept of site-bond percolation. In a model of this type, we choose two percolation tuples (p, q) and (p', q'). The probability p is used to determine the distribution of closed bonds; the probability p' is used to determine the distribution of occupied sites. A *site-bond-cluster*, in this model, is defined as a subset of connected (in the way explained in the context of bond percolation) and occupied sites of the lattice. In particular, with the choice $p' \equiv 1$ we find ourselves in the situation of bond percolation whereas the choice $p \equiv 1$ yields the case of pure site percolation. For further references see [39, 45, 9, 2].

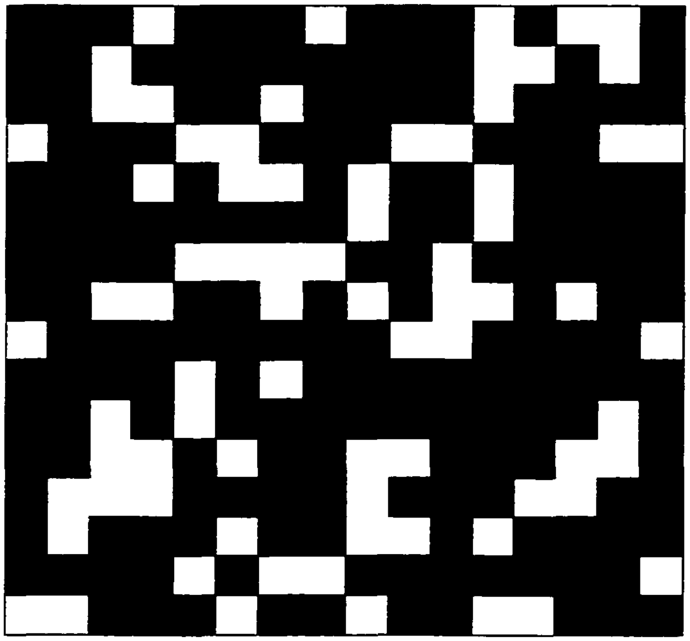

Figure 2.2: Example of a typical site percolation configuration with a density of p=0.7

2.1.4 Polychromatic Percolation

Polychromatic percolation can be considered as a generalization of the three other models. In polychromatic percolation we relax the condition that every site has to be in either one of two states (occupied or empty), but rather allow any (fixed) number n of states to choose from for the individual sites. The different states could be thought of as different colors, as the name of the model indicates. A percolation tuple does no longer consist of two numbers, but of n numbers $p_1, p_2, \ldots, p_n \in [0, 1]$ with $p_1 + \ldots + p_n = 1$ (also called a *simplex*). Now, clusters can be defined for each of the different colors (states), and their mutual dependencies can be studied. One example for a model of this type is the 'Three-Component Reactive Percolation Model' introduced by Halley and Holcomb [26]. They consider the situation of two types of atoms that can react and form dimer molecules. Simulations of this situation in a 3-state percolation model can reproduce the empirical results of resistivity measurements fairly accurately.

2.1.5 Continuous Percolation

Since a lot of processes in nature do not necessarily take place on a regular lattice, percolation models have been introduced that are not set on a lattice with a given structure, but that rather utilitize a description with continuous random variables. In 3 dimensions, numerical evidence indicates that the critical exponents are the same as in lattice percolation. Certain arguments from renormalization theory are also in favor of this view. It is therefore in most cases sufficient to stay in the computationally easier framework of lattice percolation.

2.1.6 Percolation with long-range Interaction

In our discussions so far, we only considered nearest neighbor interactions of lattice sites. One could also think of a generalization in which long-range interactions between lattice sites that are not nearest neighbors are considered. This would result in a higher coordination number and a redefinition of the term cluster. Taking into account this redefinition, numerical simulations again show no difference (concerning critical exponents) to standard percolation models.

2.1.7 Percolation on the Bethe Lattice

Percolation on the Bethe lattice, sometimes also referred to as percolation on the Cayley tree, takes place on a lattice with a special structure, the Bethe lattice. To construct this lattice, one starts with one point, the origin, and adds z neighbor vertices to it, out of which z neighbors emanate again, one of which is the connection to the origin, but $z - 1$ are new vertices. Therefore, no closed loops are possible on this type of lattice. It can be interpreted as equivalent to percolation on a hypercubic lattice in d dimensions in the limit $d \to \infty$. The Bethe lattice has attracted special attention because it is one of the few non-trivial cases in which the percolation problem can be solved analytically.

2.1.8 Percolation Threshold and the Percolating Cluster

We will explain these two central terms of percolation theory in the context of site percolation. A translation to the other types of percolation models will be obvious. One qualitative feature of the dependence of the structure of site-clusters on the probability p (concentration of occupied sites throughout the lattice) is the following: The closer the concentration p is to one, the more likely it is to find clusters consisting of a large number of sites in the lattice. In fact, it can be shown that in the case of a lattice of infinite extent, there exists exactly one number $p_c \leq 1$, the *percolation threshold*, for which a cluster exists that extends the whole system. Below this threshold, no such *percolating* cluster exists, above the threshold, at least one such cluster occurs. Table 2.1 shows some examples of percolation thresholds for different lattices. Thresholds for bond percolation are also shown in this table. In the case of a finite, two-dimensional system, for example, the presence of a percolating cluster is equivalent to the notion that a cluster that spans from one edge of the system to the other exists. However, it must be noted that the existence of a definite threshold holds only in systems of infinite extent (see also: Section 2.3.4).

2.2 Phase Transitions

One of the main objectives of studying percolation theory is, as mentioned above, the ability of percolation systems to model phase transitions. Before we proceed with our description of results of percolation theory, we give a short overview of phase transitions. For further reference, see also [37].

Lattice	Site	Bond
Hexagonal	0.6962	0.65271
Square	0.592746	0.5
Triangular	0.5	0.34729
Diamond	0.43	0.388
Simple cubic	0.3116	0.2488
BCC	0.246	0.1803
FCC	0.198	0.119
$d = 4$ hypercubic	0.197	0.1601
$d = 5$ hypercubic	0.141	0.1182
$d = 6$ hypercubic	0.107	0.0942
$d = 7$ hypercubic	0.089	0.0787

Table 2.1: Percolation thresholds for various lattices. 'Site' refers to site percolation, 'Bond' refers to bond percolation. Only nearest neighbor interactions are considered.

2.2.1 Definitions

One can categorize physical systems with many degrees of freedom, for which a temperature can be defined, in two different classes: If the constituents of the system do not interact with each other, it is always possible to apply the formalism of statistical mechanics in a way that the partition function of the system separates into a product of partition functions of the single constituents (eventually after a transformation to normal coordinates). It remains analytical at all values of the temperature T. If, on the other hand, interactions exist, the situation may occur that the thermodynamic functions possess singularities for certain values T_c of the temperature T. This phenomenon manifests in the undergoing of the system of a qualitative change in macroscopic behavior.

Before this change occurs, the system is in one possible state of thermal equilibrium, which is denoted as a *phase* of the system. Throughout the system, certain macroscopic observables assume constant values. After the change, the system will be in a different phase, with different values of some of the macroscopic observables.

Prominent examples of such phase transitions are the melting of solids, boiling of liquids or changes from ferromagnetic to paramagnetic states in magnetic systems.

2.2.2 Classification of Phase Transitions

In experimental observations, qualitatively different types of phase transitions are encountered. Ehrenfest introduced the following classification in 1933:

A phase transition is said to be of n^{th} order, if the first $n-1$ partial derivatives of the thermodynamic potentials with respect to their natural variables (e.g. $G(T, p, N)$ for fluid systems) are continuous at the phase transition point, whereas at least one of the n^{th} partial derivatives exhibits a discontinuity (jump of finite size).

A prominent feature of first-order phase transitions is the appearance of phenomena like latent heat (as seen when melting ice or boiling water).

The Ehrenfest scheme is no longer used today because of several reasons:

- Since the physical behavior of a system is mainly governed by the thermodynamic functions and their first derivatives, the differences between phase transitions of high orders start to vanish, which makes the differentiation between them uninteresting

- Although this property cannot be verified experimentally, newer experimental results seem to suggest that the observed discontinuities in the thermodynamic functions or their derivatives are rather singularities than finite jumps. This is also supported by the analytical Onsager solution for a phase transition in a two-dimensional spin-system.

Therefore, today one usually differentiates between two different classes of phase transitions: *discontinuous* phase transitions, which correspond to first-order phase transitions in the Ehrenfest sense, and *continuous* phase transitions for all other cases.

2.2.3 Order Parameter

Another typical feature of continuous phase transitions is the appearance of a quantity called the *order parameter*. An order parameter is an observable quantity of the system that can only be defined for one of the two phases.

As an example consider a fluid system (e.g. water). In the temperature range below the critical temperature T_c and above the melting temperature, two different phases can appear in the system: liquid and gas. Above the critical temperature, the liquid phase can no longer be assumed by the system. Therefore, the observable $\Delta\rho = \rho_{liquid} - \rho_{gas}$ can only be defined when the system is in the subcritical phase, but not when it is in the supercritical phase.

2.2.4 Correlation Length

The *correlation function* of a quantity X of the system (for example the magnetization) is defined as:

$$g(\mathbf{r}, \mathbf{r}') = \langle x(\mathbf{r})x(\mathbf{r}')\rangle - \langle x(\mathbf{r})\rangle \langle x(\mathbf{r}')\rangle \tag{2.1}$$

where $x(\mathbf{r})$ is the density of the quantity X:

$$X = \int \mathbf{d}^3\mathbf{r}\, x(\mathbf{r}) \tag{2.2}$$

$g(\mathbf{r}, \mathbf{r}')$ is a measure for the correlation between the system properties of X at the points $\mathbf{r}$ and $\mathbf{r}'$. In a spatially homogeneous system we have $g(\mathbf{r}, \mathbf{r}') = g(|\mathbf{r} - \mathbf{r}'|)$, i.e., the correlation is invariant under translations. In the vicinity of critical (phase transition) points, $g(\mathbf{r}, \mathbf{r}')$ is approximated by

$$g(\mathbf{r}, \mathbf{r}') = c_0 \frac{\exp\left(-\frac{|\mathbf{r}-\mathbf{r}'|}{\xi(T)}\right)}{|\mathbf{r}-\mathbf{r}'|^{d-2+\eta}} \tag{2.3}$$

Equation 2.3 implicitly introduces a new quantity, the *correlation length* $\xi(T)$. As can be seen from the expression for g, it defines a length scale for regions affecting each other and those that do not.

The definition of the correlation length leads us to another feature differentiating continuous from discontinuous phase transitions:

In the case of a discontinuous transition the correlation length stays finite, while in the case of a continuous phase transition it diverges when the temperature approaches the critical temperature T_c:

$$\xi(T) \overset{T \to T_c}{\longrightarrow} \infty \tag{2.4}$$

This means that close to the critical temperature we have fluctuations that extend much further than the usual range of interaction in the system. The term *critical fluctuations* is used for this situation.

2.2.5 Critical Exponents and Universality

In the critical region of continuous phase transitions, the behavior of many system quantities can be characterized by means of *critical exponents*. It can be observed very often that a physical property F depends on the *reduced temperature*

$$\epsilon = \frac{T - T_c}{T_c} \tag{2.5}$$

in the following way:

$$F(\epsilon) = a\epsilon^{\varphi}(1 + b\epsilon^{x} + \ldots); \qquad x > 0. \tag{2.6}$$

For $\epsilon \to 0$, i.e., $T \to T_c$, all terms in the parenthesis except for 1 vanish and $F(\epsilon)$ follows a *power law*. We write:

$$F(\epsilon) \propto \epsilon^{\varphi} \tag{2.7}$$

The number φ governing $F(\epsilon)$ in the vicinity of T_c is called the *critical exponent* of F. Critical exponents are chosen to be positive by definition. The critical exponent of a quantity may depend on the side from which the critical temperature is approached. The more general definition therefore is:

$$\epsilon = \lim_{\epsilon \searrow 0} \frac{\ln |F(\epsilon)|}{\ln \epsilon} \tag{2.8}$$

$$\epsilon' = \lim_{\epsilon \nearrow 0} \frac{\ln |F(\epsilon)|}{\ln(-\epsilon)} \tag{2.9}$$

There is a widely accepted naming convention for critical exponents in statistical physics. We will give some of the more important exponents in the following table:

Property	Exponent	Definition
Heat capacity	α	$C_V \propto \epsilon^{-\alpha}$
Order parameter	β	$\Delta\rho \propto (-\epsilon)^{\beta}$
Compressibility	γ	$\kappa_T \propto \epsilon^{-\gamma}$
Critical Isotherm	δ	$p - p_C \propto (\rho - \rho_C)^{\delta}$
Correlation Length	ν	$\xi \propto \epsilon^{-\nu}$
Correlation Function	η	$g(\mathbf{r}, \mathbf{r}') \propto \lvert\mathbf{r} - \mathbf{r}'\rvert^{-d+2-\eta}$

Table 2.2: Critical exponents in a liquid-gas system (d is the topological dimension of the system under consideration)

It is interesting to mention here that phase transitions contain a *universality* that is manifested in the critical exponents: the critical exponents are almost universal, which means they are the same for almost all thermodynamic systems. They depend only on

- The dimensionality of the system

- The range of the microscopic interaction
- The spin-dimensionality of the system

This *universality hypothesis* was first expressed by Griffiths in 1970 [24].

2.3 Theoretical Results of Percolation Theory

The results presented in the following sections do not depend, unless explicitly stated otherwise, on the type of percolation system, in other words, it does not matter if site, bond, or site-bond percolation is regarded. We will therefore just write "cluster" and leave out the further distinctions introduced above. For purposes of simplicity, we will restrict the discussion to a system represented by a $\mathbf{Z}^d$ lattice and use the notations of site percolation.

2.3.1 Phase Transition in Percolation Theory

As already explained in Section 2.1.8, the properties of the clusters of a percolation system undergo a significant qualitative change when the probability p crosses the threshold p_c. This phenomenon can be associated with a continuous phase transition. The fact that this phase transition is continuous will be indirectly motivated by the discussion of the existence of various critical exponents in the following sections.

2.3.2 Cluster numbers

We consider a percolation system of size L^d and a given percolation tuple (p, q) (or, to be more precise, an ensemble of such systems). One system quantity of interest is the number of clusters of a certain size s, denoted by $N_s(p)$. To be independent of the actual size of the system, we will rather consider the number of clusters of a given size per lattice site,

$$n_s(p) = \frac{N_s(p)}{L^d} \tag{2.10}$$

and call this quantity *cluster number*. The notation already indicates that the cluster numbers depend on the concentration p of occupied sites (or closed bonds in the case of bond percolation). Since we are interested in the critical aspects of percolation, the behavior of these cluster numbers for concentrations p close to the percolation threshold p_c is of special interest. From analytical solutions and numerical results, it can be inferred that for p close to p_c the cluster numbers behave as follows:

$$n_s(p) = s^{-\tau} f[(p - p_c)s^{\sigma}] \qquad (p \to p_c, s \to \infty) \tag{2.11}$$

We see the power law dependence of n_s at the critical point that has already been mentioned in the context of critical exponents. The *scaling function* or *cutoff function* f accounts for the fact that a power law dependence is only correct in the case of $p = p_c$. This must be the case because for $p < p_c$ no system spanning cluster exists and therefore $n_s(p)$ has to decay faster than the power law for high s. The cutoff function $f(z)$ has the general form that it approaches a constant value for $|z| \ll 1$ and decays quickly for $|z| \gg 1$.

Implicitly introduced by equation 2.11 are two critical exponents of percolation theory: σ and τ. With the definition $s_\xi = (p - p_c)^{-1/\sigma}$, we can rewrite equation 2.11 as:

$$n_s(p) = s^{-\tau} f\left[\left(\frac{s}{s_\xi}\right)^{\sigma}\right] \tag{2.12}$$

This leads to the interpretation of s_ξ as a crossover size for the clustersizes from power law abundance for $s \ll s_\xi$ to exponentially rare clusters of size $s \gg s_\xi$.

In the case of the Bethe lattice, we can give explicit terms for the scaling behavior of the cluster numbers:

$$n_s(p) \propto s^{-5/2} \exp[-((p - p_c)s^{1/2})^2] \tag{2.13}$$

We can immediately see the values of the critical exponents, $\sigma = 1/2$, $\tau = 5/2$ and the form of the scaling function: $f(z) = \exp(z^2)$. f obviously shows the asymptotic behavior mentioned before.

2.3.3 Critical Exponents in Percolation Theory

We have already introduced two critical exponents in percolation theory in Section 2.3.2: σ and τ. In particular, τ is believed to provide information about the nature of the phase transition examined (see [20]). This is motivated by the fact that in the phase transition observed in a Van-der-Waals gas the size of condensing droplets scales analogous to the cluster numbers as described by Equation 2.11 (with a critical exponent of $\tau = 7/3$).

It has been shown by Broadbent and Hammersley [8, 27, 28], as cited in [25], that for systems with a dimensionality higher than 2 the percolation threshold fulfills $0 < p_c < 1$, i.e., we have a critical phenomenon where the critical point can be approached from both sides. Despite this fact, σ and τ are the same for both sides of the critical probability: in our notation from Section 2.2.5, we can write $\sigma = \sigma'$ and $\tau = \tau'$. The correlation function $g(r)$ in percolation theory is defined as the probability that a site at a distance r from an occupied site is also occupied and belongs to the same cluster. The average number of sites to which an occupied site at the origin is connected is therefore $\sum g(r)$, the sum running over all lattice sites. We then define the correlation length or *connectivity length* ξ somewhat differently than the way it is introduced in statistical mechanics:

$$\xi^2 = \frac{\sum_r r^2 g(r)}{\sum_r g(r)} \tag{2.14}$$

$\xi(p)$ shows the typical critical behavior close to the critical probability p_c:

$$\xi \propto |p - p_c|^{-\nu} \tag{2.15}$$

with the critical exponent ν (see also Table 2.2).

Let us also consider the fraction P of sites of t he lattice belonging to the percolating cluster. This value can of course only be different from 0 for $p \geq p_c$. It therefore is an order parameter of the percolation phase transition. It can be shown that P shows critical behavior:

$$P \propto \Theta(p - p_c)(p - p_c)^{\beta} \tag{2.16}$$

with the critical exponent

$$\beta = \frac{\tau - 2}{\sigma} \tag{2.17}$$

Next we are going to consider the convergence of the mean cluster size S at the percolation threshold:

$$S \propto |p - p_c|^{-\gamma} \tag{2.18}$$

This introduces the critical exponent:

$$\gamma = \frac{3 - \tau}{\sigma} \tag{2.19}$$

Another important quantity in the analysis of percolation systems is the k^{th} moment of the cluster size distribution:

$$M_k = \sum_{s}^{\infty - 1} s^k n_s. \tag{2.20}$$

The notation "$\infty - 1$" is used to indicate that in the summation the infinite cluster (if existent) is excluded (otherwise the sum does not necessarily converge). For M_k, we find:

$$M_k \propto |p - p_c|^{(\tau - 1 - k)/\sigma} \tag{2.21}$$

See Table 2.3 for some values of the critical expoments in systems of different dimensionality (note that no lattice topology is given, since – as discussed – according to the principle of universality the critical exponents only depend on the dimension of the system in this case).

Exponent	d=2	d=3	d=4	d=5	Bethe
β	5/36	0.41	0.64	0.84	1
γ	43/18	1.80	1.44	1.18	1
ν	4/3	0.88	0.68	0.57	1/2
σ	36/91	0.45	0.48	0.49	1/2
τ	187/91	2.18	2.31	2.41	5/2

Table 2.3: Percolation critical exponents for $d = 2, 3, 4, 5$ and in the Bethe lattice. Rational numbers give (presumably) exact results whereas those with a decimal fraction are numerical estimates.

2.3.4 Finite Size Scaling

Let us consider a finite, d-dimensional lattice of size L^d. As pointed out in the preceding section, several quantities of the cluster structure (like the mean cluster size S or the higher moments of the cluster size distribution) diverge at the critical probability p_c. This can not happen in a lattice of finite extent. There are also some other changes that take place when we deal with finite lattices instead of infinite ones. The main quantity governing the qualitative and quantitative behavior of system properties on lattices of finite size is the correlation length ξ, or, to be more precise, ξ in comparison to the linear system extent L. As has been discussed in Section 2.3.2, the power law dependence for the cluster size distribution is only valid as long as the clusters stay smaller than a cutoff size s_ξ that is directly connected to the correlation length by the dimensionality D of the clusters: $s_\xi = \xi^D$ (D does not necessarily correspond to the topological dimension of the system, since the clusters can be fractal, so that $D \leq d$). The key point of this observation is that the functional form of system quantities is basically the same as in the infinite case when the system size L is larger than ξ, $L \gg \xi$, since the cluster structure is basically determined by clusters with $s \ll s_\xi$. If the system size L is smaller than ξ, $L \ll \xi$, correlations exist that are larger than the actual system. Therefore, we have to expect qualitatively different behavior in this case. One can infer the following generalized scaling laws for systems of finite size. It is assumed, that ξ scales with the percolation probability according to $\xi \propto |p - p_c|^{-\nu}$, and that a given observable X scales according to $X \propto |p - p_c|^{\chi}$ in the infinite system with the critical exponent χ. We then get the following general form of the scaling law:

$$X(L, \xi) = \xi^{\chi/\nu} x_1(L/\xi) \propto \begin{cases} \xi^{\chi/\nu} & L \gg \xi \\ L^{\chi/\nu} & L \ll \xi \end{cases} \tag{2.22}$$

or

$$X(L, p) = (p - p_c)^{-\chi} x_2 \left((p - p_c) L^{1/\nu} \right) \tag{2.23}$$

where x_1 and x_2, respectively, are "transition functions" for the quantity X that describe the change in behavior for the transition $L \gg \xi \rightarrow L \ll \xi$, with the asymptotic form given by the last expressions in equation 2.22.

One "non-critical" quantity in percolation theory that is also affected by a change from infinite to finite lattices is the percolation threshold p_c. If we denote the probability for the existence of a percolating cluster as Π, it is clear from the discussion in Section 2.1.8 that $\Pi_\infty(p)$ must be a step function for the infinite lattice:

$$\Pi_\infty(p) = \begin{cases} 0, & p < p_c \\ 1, & p > p_c \end{cases} \tag{2.24}$$

For the finite lattice, of linear extent L for example, there can exist a system-spanning cluster for the density p slightly lower than p_c; also, no system-spanning cluster can occur for p slightly greater than p_c. Instead of having a sharp step at p_c, $\Pi_L(p)$ will rather be given by a smooth function approximating a step function – the larger the lattice, the better the approximation (see Figure 2.3).

It can be inferred from the analytically solvable, one-dimensional case that $\Pi_L(p)$ should behave as follows:

$$\Pi_L(p) = g\left((p - p_c) L^{1/\nu} \right) \quad \text{for} \quad |p - p_c| \ll 1, L \gg 1 \tag{2.25}$$

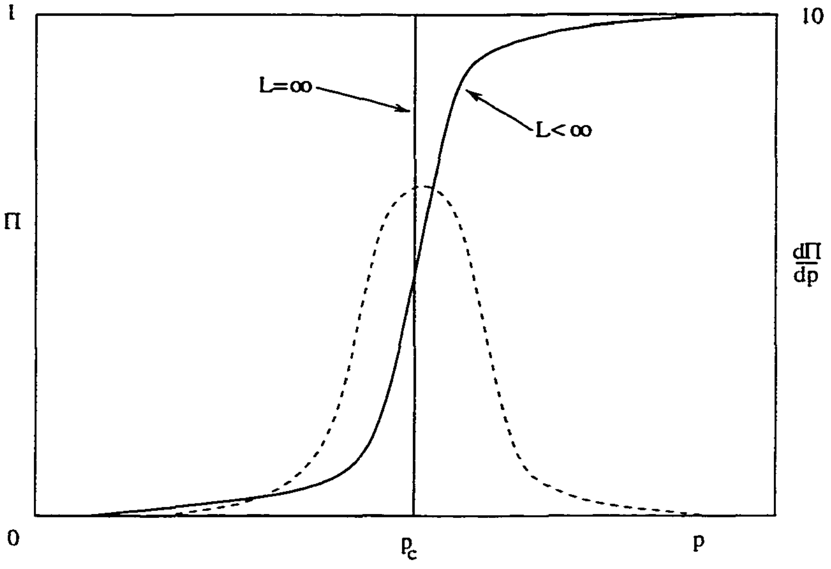

Figure 2.3: Variation of the probability Π (solid lines) that a cluster is spanning the whole system for medium and large system sizes. The dashed line gives $\frac{d\Pi}{dp}$, proportional to the probability that at concentration p a spanning cluster starts to appear. The width of the transition region or peak should vary according to $L^{1/\nu}$.

$g(x)$ is a scaling function increasing from 0 to 1 as its argument x increases from $-\infty$ (far below threshold) to $+\infty$ (far above threshold). The derivative of the expression in equation 2.25 then gives the probability density for the existence of a percolating cluster in the system:

$$\frac{\mathrm{d}\Pi}{\mathrm{d}p} = L^{1/\nu} g\prime\left((p - p_c)L^{1/\nu}\right) \tag{2.26}$$

The average concentration p_{av} at which, for the first time, a percolating cluster appears can then be defined as:

$$p_{av} \equiv \int p \left(\frac{\mathrm{d}\Pi}{\mathrm{d}p}\right) \mathrm{d}p \tag{2.27}$$

From equations 2.26 and 2.27 we find:

$$p_{av} - p_c \propto L^{-1/\nu} \tag{2.28}$$

With the probability density 2.26, we can also calculate the width Δ of the p-interval in which the transition takes place on average:

$$\Delta^2 \equiv \int (p - p_c)^2 \left(\frac{\mathrm{d}\Pi}{\mathrm{d}p}\right) \mathrm{d}p$$

$$= \langle p^2 \rangle - \langle p \rangle^2$$

The width scales according to:

$$\Delta \propto L^{-1/\nu} \tag{2.29}$$

Sometimes, the *fractional shift* $\epsilon(L)$ and the *fractional rounding* $\delta(L)$ are introduced as follows:

$$\epsilon(L) \equiv \frac{p_{av}(L) - p_c}{p_c} \tag{2.30}$$

$$\delta(L) \equiv \frac{\Delta(L)}{p_c} \tag{2.31}$$

They apparently also scale proportional to $L^{-1/\nu}$.

2.4 The Nuclear Lattice Model

The Nuclear Lattice Model (NLM) introduced by Bauer [6] (see also [7]) makes use of several of the aspects of percolation theory discussed so far. It was introduced as a model of the production of complex fragments in nuclear collisions at intermediate and high energies. Experiments carried out in this area have shown many seemingly different phenomena over the years. One class of results that has received special attention is multifragmentation reactions (MFR). The experimental data seem to show traces of critical behavior. In recent years, experiments concerned with the production of "Quark-Gluon-Plasmas" (QGP) and the corresponding phase transition (or quark deconfinement) have caught some public interest. However, there are indications that infinite nuclear matter also supposedly undergoes a "liquid-gas" phase transition, that is somewhat better understood than the QGP transition (see Figure 2.4). Since the assumed critical point for the liquid-gas transition lies in a region of the phase diagram that is physically well attainable in a variety of contemporary research facilities, it can examined quantitatively to some extent.

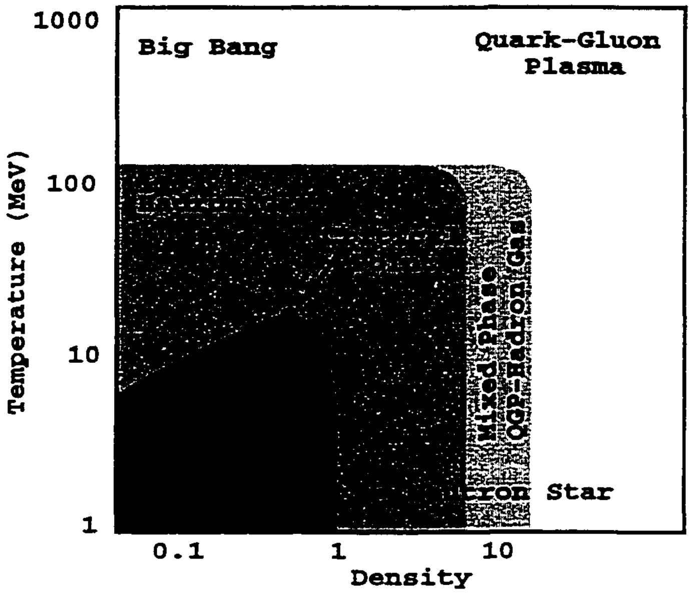

Figure 2.4: The Nuclear Phase Diagram (schematic)

It is one of the objectives of the NLM to provide a tool to study the region around this critical point in the nuclear phase diagram on a theoretical basis. It has to be mentioned, however, that it is by no means clear that MFR can be described correctly by a phase transition model at all. In particular, all theoretical models of phase transitions assume a system of infinite extent. Therefore, special attention has to be paid to the fact that we deal with systems of relatively small size (of the order of 10^2 constituents) in MFR experiments.

2.4.1 The Model

The NLM describes, as mentioned earlier, reactions of the type

$$Y + A_\mathrm{T} \longrightarrow A_\mathrm{F} + X \qquad (2.32)$$

where Y is the projectile, which will, in our case, usually be a proton, A_T is the target nucleus, A_F is the remainder of the nucleus after the fast, first-stage "pre-equilibrium" particles X have been removed. A_F thermalizes, meaning that the energy deposited in the collision is distributed homogeneously over the system, and breaks up into fragments. The A_{T} nucleons of the target nucleus form, in this model, an approximately spherical distribution on a simple cubic lattice. Arbitrarily deformed nuclei can also be considered, but assuming that the target nucleus is reasonably compact, the results are essentially unchanged. The simple cubic lattice is chosen since it is particularly amenable to computation. The results of percolation theory show that the lattice topology should have no influence on the cluster properties. This is certainly only true for infinite lattices, but it is assumed that the changes caused by considering a finite system are small enough not to have a profound impact on the results derived from this model. For a given impact parameter b, those nucleons are removed from the lattice that lie within a cylindrical channel of radius r (radius of projectile) at impact parameter b (see Fig. 2.5 for the fireball geometry). For proton-induced reactions typically 6-8 nucleons are in the fireball, so the effect on the results are slight, i.e., one obtains very similar results without removing the nucleons from the fireball channel. Since the impact parameter introduces a physical length-scale in the model, a lattice spacing d has to be chosen. d can be computed approximately from the nuclear saturation density ρ_0 ($\rho_0 \approx 0.15$ nucleons/fm; see [6, 4])

$$d = \frac{1}{\rho_0^{1/3}} \approx 1.81\,\mathrm{fm}. \tag{2.33}$$

Using the breaking probability p_B (which would correspond to q in the terminology of our earlier discussion of bond percolation) as an input parameter, a Monte-Carlo algorithm decides for each bond individually whether it is broken or not. This procedure is followed by a counting algorithm that looks for clusters and evaluates their size. Finally, we integrate over all impact parameter b. This happens with the geometrical weights that different impact parameter intervals have:

$$\mathrm{d}N(b) \propto b\mathrm{d}b \tag{2.34}$$

(area of a ring of thickness db at radius b: $2\pi b\mathrm{d}b$). By this procedure, an inclusive

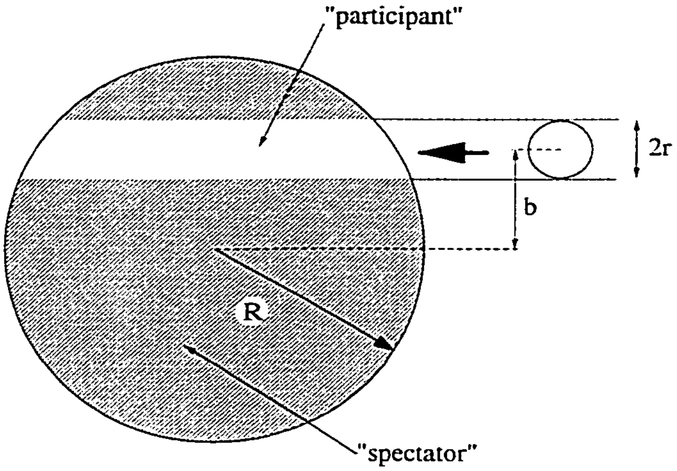

Figure 2.5: The fireball geometry: the lattice sites in the cylindrical channel with radius r at impact parameter b are left unoccupied.

mass yield distribution is obtained that can be compared to experimental data. At this stage, the only adjustable parameters that enter the model are the breaking probability p_B and the mass A_T of the target nucleus (which, of course, is given by the experimental setup).

2.4.2 Choice of $\mathrm{p_B}$

Next, we will discuss three different ways to choose the breaking probability p_B. This is crucial in order to reproduce the inclusive spectra obtained by typical experiments.

Constant $\mathrm{p_B}$

In this approach, a constant p_B is chosen for all events and all impact parameters b. Reconsidering the earlier discussion of the thermalized source with energy deposited all over the nucleus, it seems to be reasonable to assume that the individual bonds are more likely to break for higher excitation energies of the source. On the other hand, the fact that the excitation energy certainly depends on the impact parameter of a collision shows us that the excitation energy varies from event to event (which is also supported by experimental results). Therefore, the constant breaking probability approach seems to be too simple to reproduce inclusive experimental data.

The Woods-Saxon Approximation

In order to take into account the dependence of p_B on the excitation energy (and therefore on the impact parameter), p_B is assumed to be larger for central collisions than for grazing collisions. An ansatz analogous to the Woods-Saxon approximation of the nuclear density profile is chosen:

$$p_B(b) = \frac{p_{B0}}{1 + \exp[(b - R)/a]} \tag{2.35}$$

where R is the radius of the target nucleus and a parameterizes its "diffuseness". a should therefore be an experiment-independent fit parameter whereas p_{B0} is the adjustable parameter for the actual experimental situation.

The Glauber Approximation

This approach can be considered an extension of the Woods-Saxon approach. It is still assumed that p_B depends on the impact parameter or, equivalently, on the deposited energy. Now the deposition of energy in the nucleus is modeled by the ansatz that the breaking probability should be proportional to the integral over the nuclear density along the path traveled by the projectile. It is assumed to be spatially constant over the whole lattice, i.e., the situation of total thermalization is assumed.

$$p_B(b) = \frac{p_{B0} \int_{-\infty}^{+\infty} \rho[\mathbf{R}(b)]\mathrm{d}\mathbf{R}}{\int_{-\infty}^{+\infty} \rho[\mathbf{R}(0)]\mathrm{d}\mathbf{R}} \tag{2.36}$$

For the nuclear density $\rho(\mathbf{R})$, one of the standard parameterization (like Woods-Saxon) can be used. Again, p_{B0} (in addition to other parameters from the used model for the density) is left as the adjustable parameter for the experiment. $p_B(b)$ is a monotonically decreasing function with $p_B(0) = p_{B0}$. However, this model can only be correct for small projectiles, since for larger projectiles (like in heavy ion-ion collisions) the overlap integral of the two nuclear densities should be used instead.

$\mathbf{p_B}$ from experimental excitation energies (Hybrid Model)

If the results produced by the NLM are to be compared to experimental data on an event-by-event basis, and if the excitation energy for the single events in the experimental data is known, another approach to obtain p_B is possible (see [33]). We assume that the energy distributed into each bond of the lattice, ϵ_b, can be described by a Boltzmann distribution with mean energy $\langle\epsilon_b\rangle$. Each site of the lattice has an average of α bonds (on an infinite, 3-dimensional lattice, α would be 6/2, but we have to take into account that the lattice is finite which causes surface effects). The average deposited excitation energy per site then is $\langle E_s\rangle = \alpha\langle\epsilon_b\rangle$, and the binding energy of the initial nuclear system is $B = \alpha E_b$ (where E_b is the binding energy per bond). When the system undergoes the multifragmentation reaction, any bond which has an energy greater than E_b will break. Therefore, the bond-breaking probability is:

$$p_B = \frac{\int\limits_B^\infty \sqrt{E_s} e^{-E_s/T} \mathrm{d}E_s}{\int\limits_0^\infty \sqrt{E_s} e^{-E_s/T} \mathrm{d}E_s} \tag{2.37}$$

Here, T is the temperature of the nuclear system which has to be introduced in an appropriate way. It is through this temperature that the actual experimental excitation energy E^* enters the breaking probability: $T = T(E^*)$. The average binding energy per nucleon B is an adjustable parameter of the model (within the restrictions given by the possibilities to measure this quantity). With the help of the generalized, incomplete Gamma function $\Gamma(x, z_0, z_1)$, equation 2.37 can be rewritten as:

$$p_B(E^*) = 1 - 2\Gamma\left(\frac{3}{2}, 0, \frac{B}{T(E^*)}\right) / \sqrt{\pi} \tag{2.38}$$

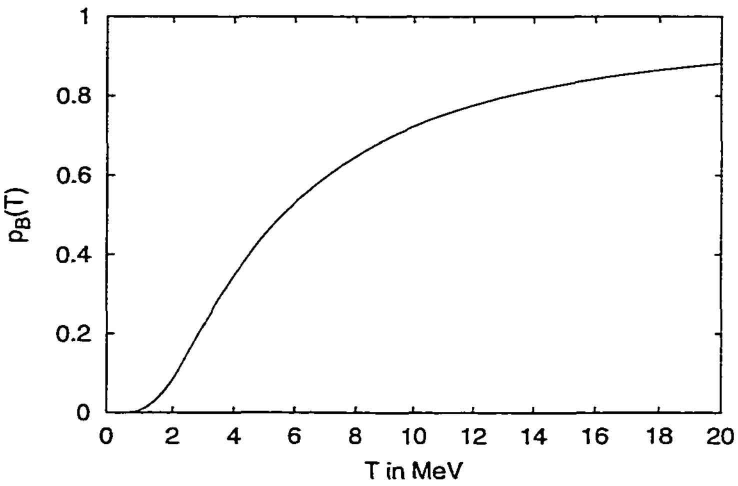

Figure 2.6: Relation between p_B and the temperature of the fragmenting nucleus as given by Equation 2.38 (B=6.6MeV)

In this approach, p_B no longer explicitly depends on the impact parameter b. Since it is not possible to measure the impact parameter of the reactions likely to be described by this model, one can no longer apply the removal of the fireball channel without further assumptions about the correlation between the impact parameter and the excitation energy. This will be discussed in further detail later (see 3.3.1).

Chapter 3

Application of the NLM to ISiS data

3.1 The ISiS Experiment

The experimental data analyzed in this thesis were obtained with the Indiana Silicon Sphere (ISiS) 4π charged-particle detector array [16] [32] in two experiments (E900 and E900a) at the Brookhaven National Laboratory AGS accelerator. In E900 untagged secondary positive beams at 5.0, 8.2 and 9.2 GeV/c incident on a ^{197}Au target were employed. In E900a a tagged negative beam of 8.0 GeV/c π^- and antiprotons incident on the same target was used. The Au target consisted of foils of 10^{-5} purity, which were prepared by vacuum evaporation onto a glass slide (KCl substrate).

The detector array consisted of 162 triple-detector telescopes arranged in a spherical geometry. The telescopes span the polar-angle range from 14° - 86.5° in five segments in the forward hemisphere and 93.5° - 166° in four backward-hemisphere segments. The detector telescopes consisted of a gas-ionization chamber operated at 16-18 Torr of C_3F_8 gas, a 500 μm passivated silicon detector, and a 28-mm CsI scintillator with photodiode readout.

In this thesis, we focused on the analysis of approximately 1.5×10^6 p $\rightarrow$ Au events at 10.2 GeV/c in the E900 experiment.

3.2 Implementation Details and Modifications of the NLM

3.2.1 General Setup

Since only the charges were detected in the inclusive mass yield spectra of the ISiS experiment (see Section3.1), we had to incorporate the fact that no information about the fragment masses was available in our model. We followed two approaches:

Lattice with 79 nucleons (protons): We just modeled the protons of the Au nucleus by distributing 79 sites approximately spherically on the lattice (as described in Section 2.4.1). Therefore, we could compare the sizes of the clusters obtained in the model calculation directly to the experimental data. However, as described in Section 2.3.4, finite-size effects have to be considered, and these effects certainly depend on the actual size of the system.

Lattice with 197 nucleons (full Au nucleus): This approach is certainly more likely to incorporate the finite-size effects correctly. On the other hand, since the model assumes isospin-symmetry, we had to derive the charges of the produced fragments (clusters) in order to be able to make a comparison with experimental data. We did this by downscaling the fragment size with the mass-charge 197/79 ratio of the Au nucleus. Since we are scaling from one discrete set of numbers ($\{1, \ldots, 197\}$) to another ($\{1, \ldots, 79\}$), Moiré effects occur in a simple multiplication of the fragment size by the scaling factor. Therefore, we implemented a Monte-Carlo algorithm to avoid these effects. This approach does not take into account binding energy effects, since the ratio of neutrons to protons in stable configurations changes from small fragments (1 : 1) to larger fragments (more neutrons than protons). Especially for small fragments, it is certainly not equal to the Au ratio 197/79.

A comparison of the two aforementioned approaches shows that the respective results vary only insignificantly. We used the approach that works with a smaller lattice because it is less time-consuming in numerical simulations. The part of the simulation that takes most time is the cluster-counting algorithm. Its time complexity is on the order of N^3, where N is the number of the lattice sites (however, algorithms with a time complexity on the order of N can be implemented).

3.2.2 The Filter Code

One peculiarity of the experimental data is the existence of "residual fragments" (in the data file, at most one per event), which contain the sum of all undetected fragments in a given event. The charge of these fragments can be determined by comparing the number of detected particles with the size of the thermalized source. Thus, the residual fragments do not describe fragments that physically appeared in an event. The reasons for the appearance of undetected residuals are as follows:

- The 4π geometry of the detector cannot be perfect, since parts of the surface of the detector-sphere cannot be "actively detecting" because of the area occupied by the beam pipe and "seams" between individual detectors;
- Some of the 162 detectors were found to be defective or operating improperly after the experiment was completely assembled;
- The detectors were only constructed for a certain mass and energy range, and therefore can not detect fragments that lie outside these boundaries.

To incorporate these effects in our model calculations, we "piped" all percolation fragments through a filter code that decides whether the fragment is detected or not. This code has been provided by the ISiS group. With the help of this filter, we can introduce artificial residual fragments in the model data in a controlled way.

The input that is required by the filter code is the charge, mass, energy and direction (in the laboratory frame of reference) of the fragment for which the successful detection has to be determined. Since the NLM in the way we implemented it (see above) only provides us with information about the charge of the fragments, the missing information has to be determined in the following ways (see A.2):

Mass of Fragment

Here we use the ratio of neutrons to protons in the Au nucleus to determine a preliminary mass A' from the charge Z by $A' = \alpha \times Z$. α is the ratio of protons to nucleons in the nucleus. In general, this will be a non-integer number. To choose an integer mass A of the fragment, we assign $\lfloor A' \rfloor$ or $\lceil A' \rceil$ to A with the probabilities $\lceil A' \rceil - A'$ or $A' - \lfloor A' \rfloor$, respectively.

Direction of Particle Emission

Since the energy of the protons is high enough to merely remove the fireball channel from the target nucleus (see Fig. 2.5) without transferring considerable linear momentum [32], the angular distribution of fragment emission in the multifragmentation reaction is assumed to be isotropic. Therefore, a random direction with respect to an isotropic distribution is chosen for each fragment.

Fragment Energy

The energy of the fragments is assumed to be distributed according to a Maxwell-Boltzmann distribution [30] in the rest frame of the target nucleus [44]. It can be written as:

$$\frac{d^2\sigma\left(A_F\right)}{dE^* d\Omega^*} = \frac{\sigma_0}{2(\pi T)^{3/2}} \sqrt{E^*} \exp\left(-\frac{E^*}{T}\right) \tag{3.1}$$

where E^* is the kinetic energy that is given by, after a correction for the Coulomb barrier and the recoil,

$$E^* = \frac{A_T}{A_T - A_F} \cdot E' - \kappa \cdot B \tag{3.2}$$

E' is the kinetic energy of the fragment in the moving frame of reference, κ is an adjustable parameter < 1, and B is the Coulomb barrier. The barrier B, for two fragments touching each other with charges Z_F and $Z_R = Z_T - Z_F$ and masses A_F and $A_R = A_T - A_F$, is:

$$B = \frac{e^2 Z_F Z_R}{r_0 \cdot \left(A_F^{1/3} + A_R^{1/3}\right)} \tag{3.3}$$

By transforming equation 3.1 to the laboratory frame one obtains the energy of the fragments in this frame:

$$\frac{d^2\sigma\left(A_F\right)}{dE d\Omega} = \sqrt{\frac{E}{E'}} \cdot \frac{d^2\sigma\left(A_F\right)}{dE' d\Omega'} \tag{3.4}$$

The connection between E' and the laboratory energy E is given by

$$E' = E + \frac{1}{2} A_F \beta^2 - \beta\sqrt{2 A_F E} \cdot \cos(\Theta_{lab}) \tag{3.5}$$

where β is the velocity v/c of the emitting system.

Since in our case high accuracy is not necessary and β is close to 0, we assume $\beta = 0$. This is supported by the findings described in [32]. Therefore, equation 3.5 is simplified to $E' = E$, and we can substitute E' by E in equation 3.2. The value of E^* to be used in equation 3.1 is therefore E shifted by $-\kappa \cdot B$. From earlier analyses conducted by Hirsch et al. [29] and Bauer et al. [3], we use their value of T=12 MeV. For κ we chose a value of 0.6 to reflect the lowered *freeze-out* density ρ_{f_0} of the nuclear matter before breaking up into fragments. Bauer in [3], for example, finds it to be

$$\rho_{f_0} \approx 0.36\rho_0 \tag{3.6}$$

for the reaction 300 GeV p+Xe → ^{12}C + X.

3.2.3 Charges 17-20

Another peculiarity of the experimental data is the fact that while the detectors were able to detect fragments with charges up to 20, it was only possible to resolve them elementally in the charge range 1-16. The fragments with charges between 16 and 21 were therefore assigned a charge number on a Monte-Carlo basis, extending the assumed power law of the mass yield curve. Since the provided filter code does not implement this feature – all fragments with charges of 17 and higher are declared undetected – we removed those fragments both in the experimental data and in the model calculations.

3.2.4 The Tcl/Tk Interface

The experimental data as well as the data derived from the model calculations, consist of a list containing the single events of the multifragmentation reactions. In order to simplify the fitting procedure, we implemented an interactive user interface that is able to display the different characteristics of the derived events (mass yield, multiplicity distribution, etc.) and that allows for immediate adjustment of the model parameters. As an easy and system-independent way to realize this, we implemented an interface in the script language Tcl/Tk. This interface serves as a front end to the percolation simulation and data acquisition routines that are written in C++. It displays the experimental data and the data from the model calculations simultaneously to make a visual comparison of the two data sets possible. With this setup, changes in the model can be evaluated quickly and easily.

3.3 Results

Unless stated otherwise, all diagrams in the following sections were created with 500,000 events produced with the C++ implementation of the NLM (see Section 3.2 and appendix A.2) and the same number of events from the experimental data (see Section 3.1). In particular, the bond-breaking probabilities for the percolation events have also been chosen from these events (in the final setup, see Section 3.3.1).

3.3.1 Choice of Parameters

In the beginning, we worked with the Glauber approach for the p_{break} - impact parameter dependence. As a fitting procedure we tried to reproduce the inclusive charge yield spectrum of the experimental data. In order to reach a reasonable correspondence of the two data sets, we used the following setup:

Projectile Mass $\approx$ 4 For good correspondence in the high mass region, a projectile mass of 4 (as opposed to one, which would be realistic for proton-induced reactions) had to be chosen. This might be due to the fact that the projectile energy of 10.2 GeV is comparably low, and therefore the cylindrical fireball-channel degenerates to a "trumpet-shape", which can be approximated by a cylindrical region of larger radius.

Impact Parameter Range restricted to 0-4 fm The range of impact parameters used to integrate the inclusive data had to be restricted to 0-4 fm (while the largest possible impact parameter for a grazing reaction would be 7 fm). The reason for this might be that the fall-off of the Glauber approximation is not steep enough. The approximation does not take into account the production of pions (rest mass of 140 MeV, which is certainly attainable in that projectile energy range) which decay within the nucleus and deposit their kinetic energy + rest energy.

Undercritical $\mathbf{p_{B0}}$ The adjustable model parameter p_{B0} had to be chosen to be $p_{B0} = 0.6$. Since this is the maximal breaking probability (realized in central collisions), no critical events appear in the model calculation (critical p_{break} for three dimensional bond-percolation on a cubic lattice: 0.7512, with finite size correction: ≈ 0.7).

Even though with this model setup a fair agreement of experimental and model data could be achieved (at least for the inclusive mass yield spectrum), some of the adjustments seem to be rather arbitrary and finally led us to a different approach:

$\mathbf{p_{break}}$ from Hybrid Model and experimental Source Size

In our final setup for the analysis, we used the determination of p_{break} by means of the hybrid model (see Section 2.4.2). In particular, p_{break} is given by Equation 2.38 with T determined from the experimental excitation energy. The value of the binding energy per nucleon, B, was adjusted to 6.6 MeV (to achieve best correspondence of the inclusive mass yield spectra). This binding energy matches well previous research (see, for example, [5]). It is assumed that the relation between the excitation energy E^* of the fragmenting nucleus and the temperature is given by $E^* = aT^2$ with $a = A_0/13$ (corresponding to the high temperature limit of a model for a degenerate Fermi-gas; A_0 is the mass number of the nucleus, see [17, 15, 14]). This is done on an event-by-event basis and the corresponding percolation events are generated. As already mentioned in Section 2.4.2, in this approach, we have no straightforward method to determine the impact parameter of the incident particle. Therefore, no controlled way of removing the fireball channel exists. Instead, we use the information about the size of the fragmenting thermalized source, which is obtained by adding the charges of the individual fragments and the charge of the residue (as given in the event file), and use this as the actual size of the percolation lattice (see Section 2.4.1). Thus, the size of the lattice generally changes from event to event, but matches exactly what is found in the experiment.

Working with a smaller spherical lattice representation of the thermalized source rather than with the fireball geometry described earlier obviously decreases the effective surface area of the system. Since nucleons at the surface have fewer bonds to adjacent neighbors, this method might introduce significant changes in the model data. However, nuclear transport theory calculations in the BUU (Boltzmann-Uehling-Uhlenbeck) model have shown that the channel tends to "heal" before the actual breaking up occurs. Therefore, no meaningful systematic errors should be introduced in the model by our method of using a compact thermalized source.

3.3.2 Excitation Energy Spectrum and Distribution of pre-equilibrium emitted Particles

In Figure 3.1, the spectrum of the excitation energy $e^* = E^*/A_0$ per nucleon of the thermalized source (circles) and the distribution of pre-equilibrium emitted charges (lines) are plotted. In this diagram, all available ISiS events have been plotted. As already explained (see Sections 2.4.2 and 3.3.1), the first set of data is used to determine a bond-breaking probability for our model, while the second set gives the size of the lattice used in the calculations. The distribution of the pre-equilibrium emitted charges is fairly wide and could not be produced in this form by using the

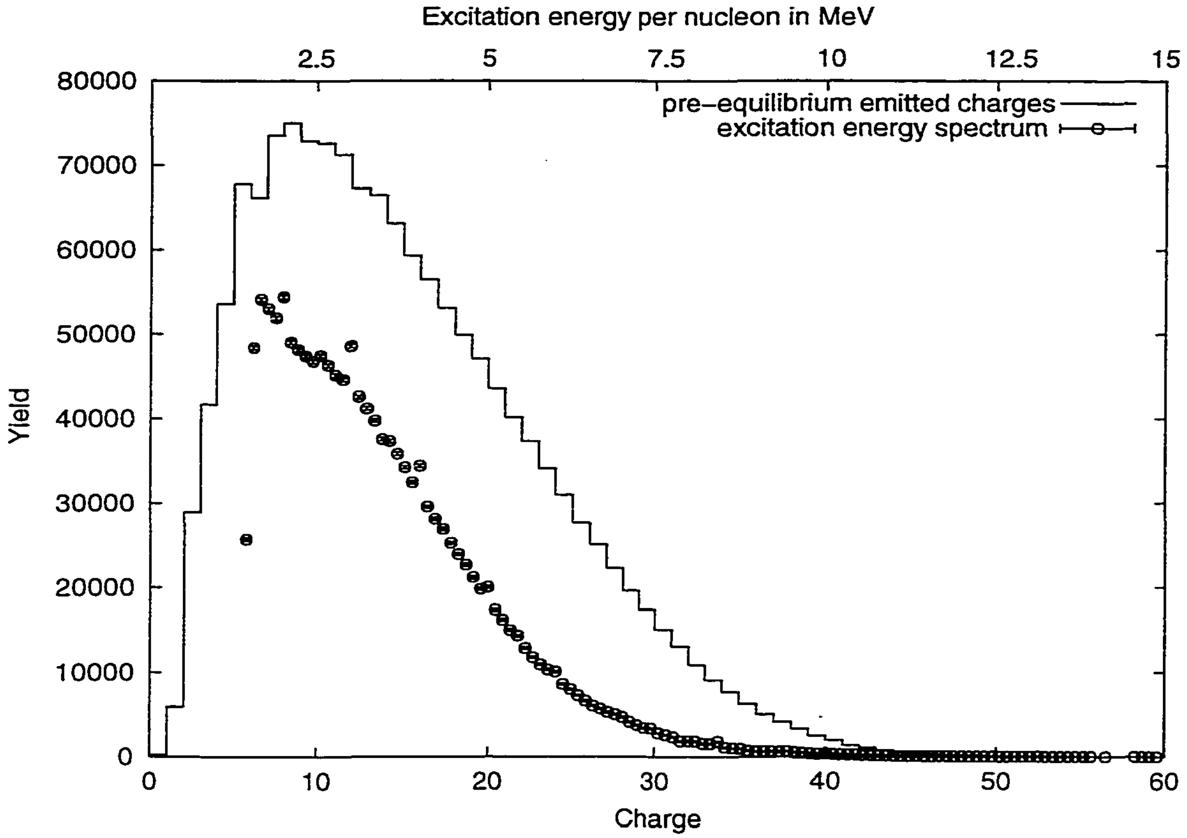

Figure 3.1: Spectrum of excitation energy per nucleon and frequency of pre-equilbrium emitted charges in experimental data.

simple fireball geometry model. Using this distribution we could calculate an average thermalized source size of approximately 65 charges, i.e., on average about 14 "fast" charges are emitted before the source equilibrates. However, it should be mentioned that the distinction between pre-equilibrium and equilibrium emitted particles is by no means exact and unambiguous [32].

3.3.3 Charge Yield

Figure 3.2 shows the resulting inclusive charge yield spectra. In this figure, the experimental yields are plotted as circles, the filtered model data are given by the solid line and the unfiltered by the dotted line.

The model calculation and the experimental data are in close agreement. In the

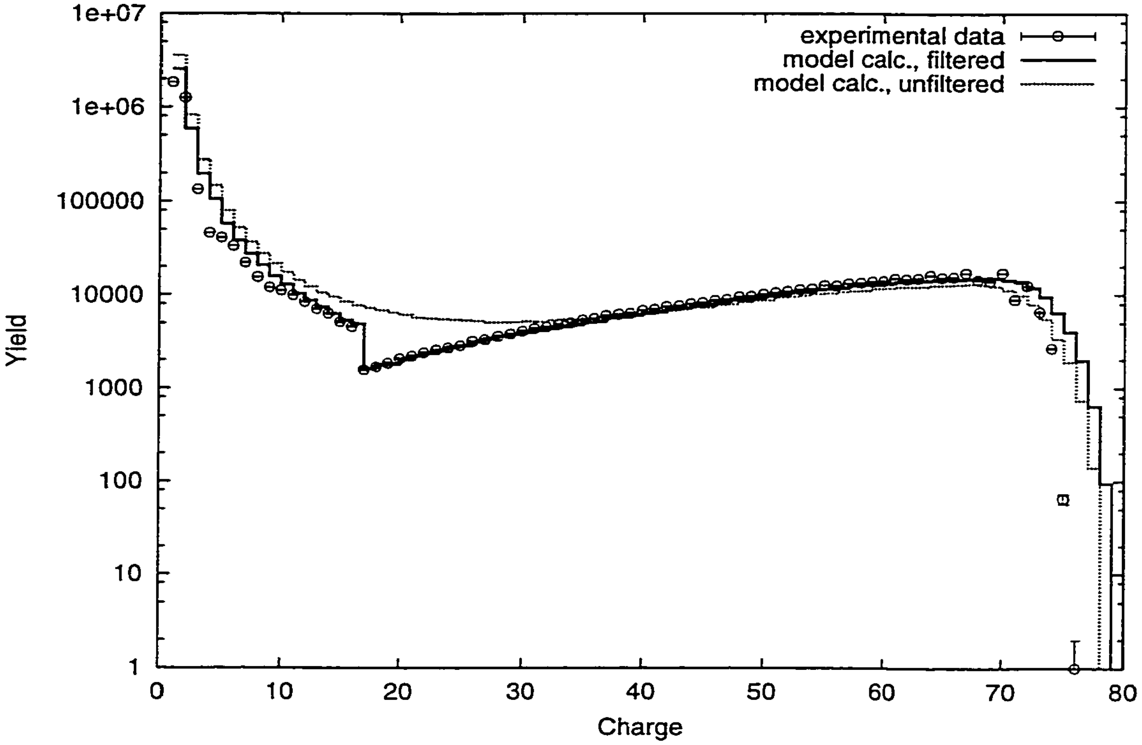

Figure 3.2: Inclusive charge yield spectrum. The experimental data (corrected by the "17-20" events, see Section 3.2.3) is given by the circles, the filtered and unfiltered model data by the solid and dotted lines, respectively. The charge is given, as in all following diagrams, in multiples of the charge of a proton.

experimental data, the fragments with charges between 17 and 20 have been omitted, as described earlier in Section 3.2.3. Therefore, the yield in the range 17-79 is given by residual fragments. One can clearly see the "gap" that is produced by this method at $Z = 16$. This gap is not present in the unfiltered calculations. Besides this artificially introduced difference, the filtered and unfiltered model data differ significantly for small and large charges.

That the experimental data do not follow an exact power law for small charges is probably due to the fact that the primary yield of fragments is altered by sequential decay of unstable fragments before their detection. For instance, the small jump at $Z = 4$ is most probably caused by the lack of a stable $^8_4$B nucleus (see also Section 3.3.8).

3.3.4 Multiplicity Distribution

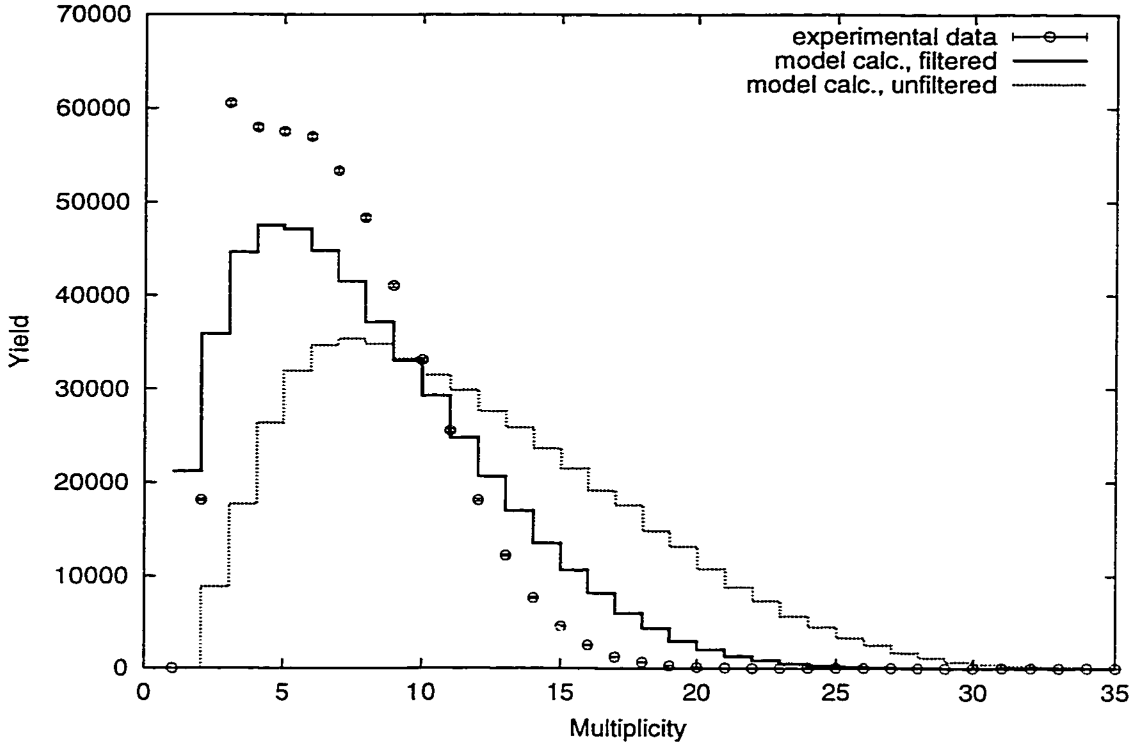

Figure 3.3: Multiplicity distribution. The experimental data are given by the circles, the filtered and unfiltered model data by the solid and dotted line, respectively.

Figure 3.3 shows a plot of the multiplicity distribution for the experimental data, filtered model data and unfiltered model data. Again, the experimental data is plotted with circles, and the filtered and unfiltered model data with solid and dotted lines, respectively. Although the correspondence between model calculation and the experimental results is not as good as in the case of the charge yield spectrum, one can clearly see that the filtered data are considerably closer to the experimental data than the unfiltered data. Since the filter basically "eliminates" fragments by assigning them to the residual fragment, the filtered multiplicity spectrum is to be shifted towards smaller multiplicities in comparison to the unfiltered spectrum. This can be clearly seen in the diagram.

3.3.5 Z_{residue} Distribution

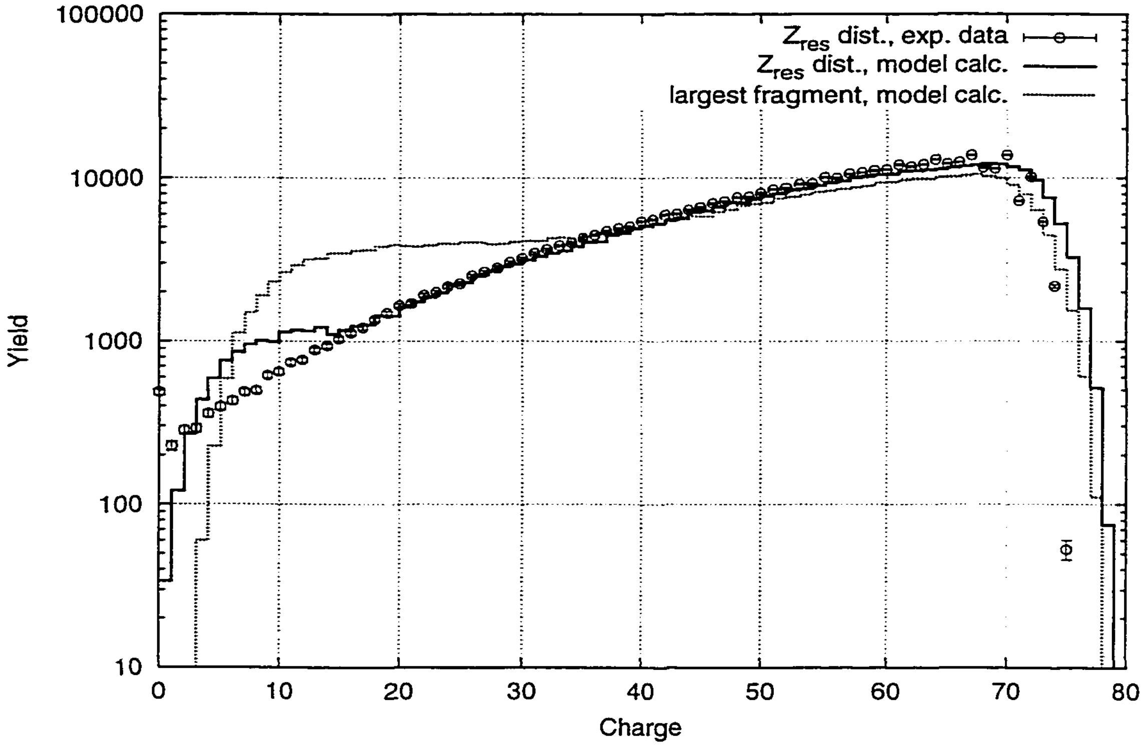

Figure 3.4: Distribution of $Z_{residue}$ (experimental data and filtered model data, circles and solid lines) and of the largest fragment (unfiltered model data, dotted lines).

The distribution of $Z_{residue}$ is shown in Figure 3.4. Since for the unfiltered model data no residual fragment is created, the distribution of the largest fragments is plotted for this data set instead. For the experimental data and the filtered model data, the yield of fragments with a charge greater than 16 corresponds exactly to the charge yield spectrum, because no regular fragments with a charge > 16 can pass the filter. A comparison with the largest fragment of the unfiltered data shows that the yield for the high charge region is dominated by single fragments per event. The effect of the filtering can mostly be seen in the range of 0-30 charges.

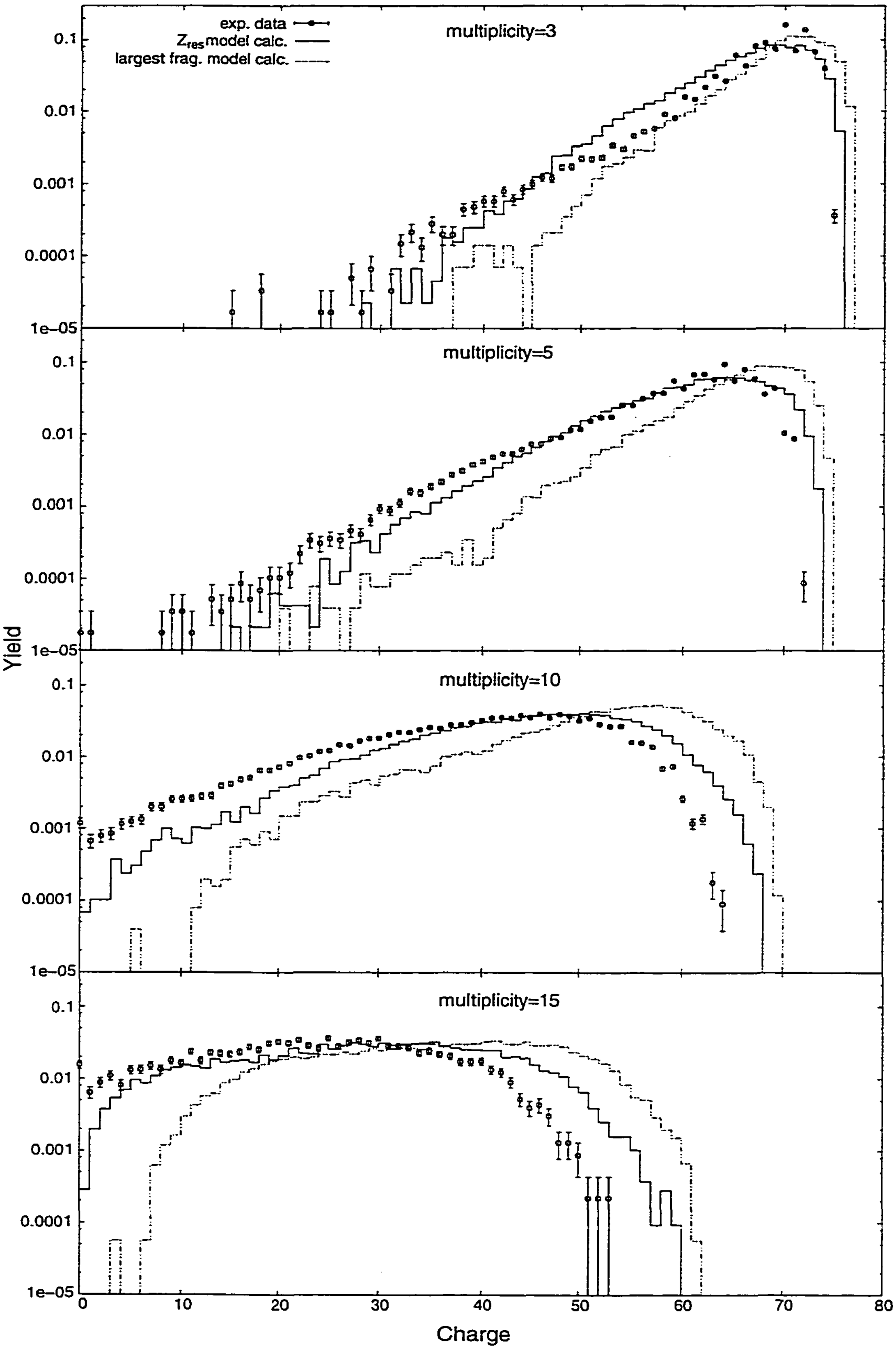

Figure 3.5: Z_{res} distribution for different multiplicities. In the case of the unfiltered data the distribution of the largest fragment is plotted instead.

3.3.6 $Z_{residue}$ Distribution gated on Multiplicity

In Figure 3.5, the $Z_{residue}$ (or, in the case of the unfiltered model data, the largest fragment) distribution for different values of the event multiplicity is plotted. It is obvious that the use of the filter has a much larger overall impact in these plots than for the inclusive $Z_{residue}$ plot (Figure 3.4). This has to be seen in connection with the shift in the fragment multiplicities caused by the employment of the filter code (Section 3.3.4). This fact can be of some importance for a critical exponent analysis using higher moments of the fragment distribution.

3.3.7 Second Moment Distribution

Figure 3.6 shows the second moments of the fragment distribution (see also Section 2.3.3) as a function of the event multiplicity. In the upper branch of the diagram, the largest fragment is included in the calculation of the second moment for each event, whereas the lower branch is obtained by excluding the largest fragment. In the case of the experimental data and the filtered model data, the largest fragment is usually (but not necessarily) given by the residual fragment. The correspondence between the experimental data and the model data is not as good as for the charge yield spectrum; however, it is apparent that the filtered model data match the experimental results considerably better than the unfiltered data.

It has been proposed by Bauer [4] that the existence of a maximum in a M_2/M_1 versus M_0 plot can be interpreted as a trace of a phase transition in the percolating system (i.e., that events with critical and overcritical percolation probability are present). This is a manifestation of the fact that the value of M_2/M_1 diverges according to $|p-p_c|^{-1/\sigma}$. This holds only for infinite systems. Also, as described in Section 2.3.3, the infinite cluster has to be excluded in the calculation of the moments. In [4],

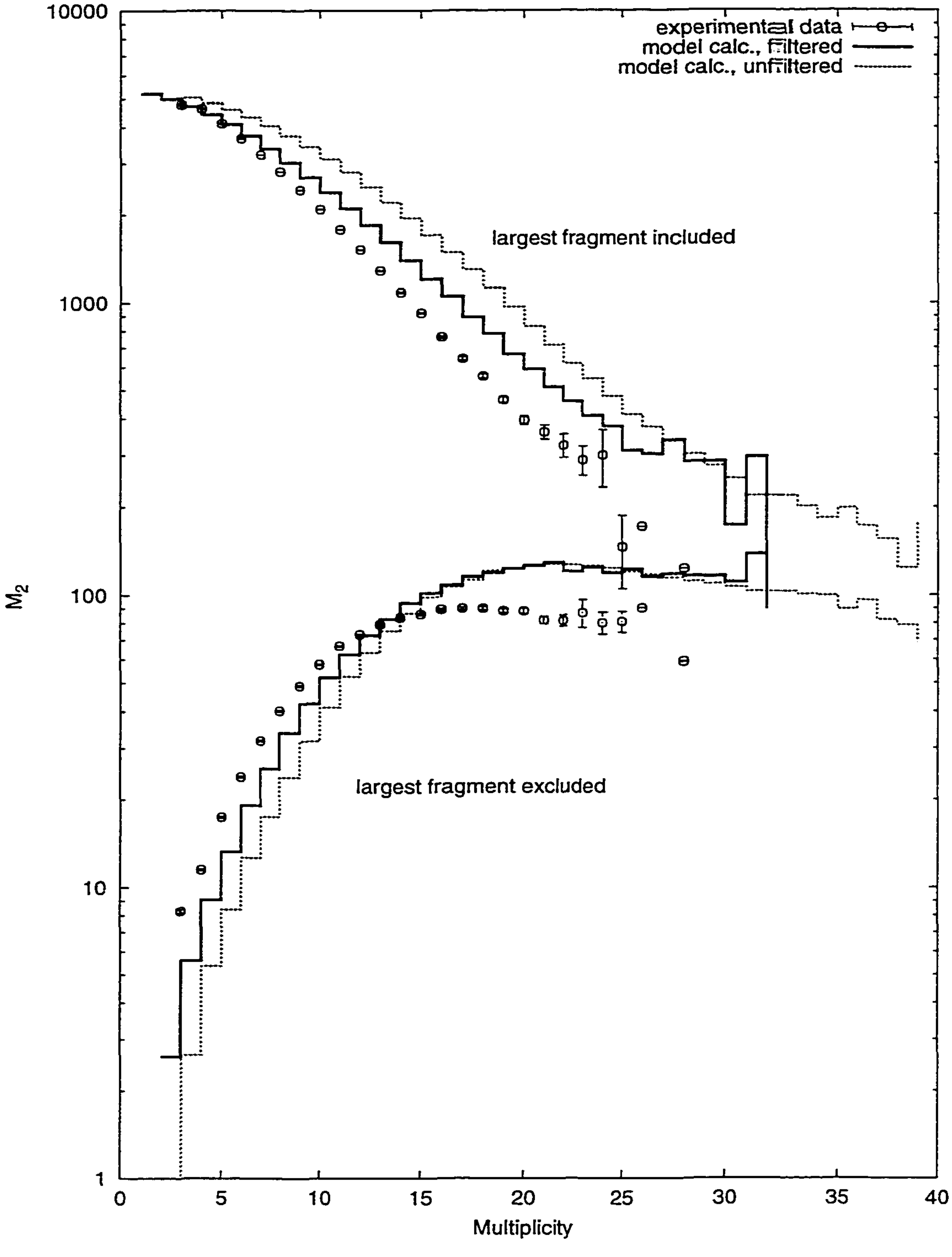

Figure 3.6: Second moment (M_2) versus multiplicity. In the upper branch, the second moments have been calculated including the largest fragment, in the lower branch the largest fragment has been excluded.

it is shown that for finite systems, excluding the infinite cluster translates into excluding the largest cluster (or, in our case, the largest fragment). Since, on average, the multiplicity M_0 of a percolation event increases with increasing bond-breaking probability monotonously, a maximum can be expected in a M_2/M_1 versus M_0 diagram if events with undercritical, critical and overcritical percolation probability are present in the data. For practical purposes, a cutoff at about 1/3 of the system size should be used when evaluating the moments of the distribution.

It cannot be inferred unambiguously that a phase transition is present either from the experimental data or from the filtered model data in Figure 3.6 (lower branch).

3.3.8 Fisher Droplet Model (FDM) Scaling

In 1967 Fisher introduced a model of droplet formation and condensation [21], which, since then, has been applied to nuclear multifragmentation in different ways. It is based on the assumption that the non-ideal gas can be approximated by an ideal gas of clusters. Elliot et al. [12] applied the basic scaling ideas from the Fisher droplet model both to percolation theory (a connection that can be based on theoretical reasoning) and to nuclear multifragmentation data from the EOS collaboration [10, 15, 17].

One of the basic features of the droplet model and its forerunners [23, 35] is that the mean number of droplets of a given size A can be written as:

$$\langle N_A \rangle \propto \left[\frac{A\Delta\mu}{T}\right] \tag{3.7}$$

where $\Delta\mu = \mu - \mu_l$ and μ and μ_l are the actual and liquid chemical potential, respectively. In order to describe the distribution for intermediate values of A, Equation 3.7 was modified to include the influence of the surface of the droplets:

$$\langle N_A \rangle \propto \left[\frac{A\Delta\mu}{T} - \frac{c(T)A^{2/3}}{T}\right] \tag{3.8}$$

where $c(T)$ is the surface free-energy density. To account for the properties near criticality, Fisher introduced an explicit term for $c(T)$ and a topological factor resulting in an expression for the normalized droplet distribution:

$$\langle n_A \rangle = \left\langle \frac{N_A}{A_0} \right\rangle = q_0 A^{-\tau} \exp\left[\frac{A\Delta\mu}{T} - \frac{c_0 \epsilon A^{\sigma}}{T}\right] \tag{3.9}$$

This scaling behavior will be referred to as "FDM scaling". In Equation 3.9, A_0 is the size of the system and q_0 is a normalization constant depending only on the value of τ, $q_0 = 1/\zeta(\tau - 1)$ (with Riemann's Zeta function ζ) [36]; τ, the topological critical exponent, depends on the dimensionality of the system with origins that lie in considerations of a n-dimensional random walk on a surface closing on itself. In percolation theory, τ depends on the scaling behavior of the normalized cluster distribution close to the critical percolation probability (see Section 2.3.2 and Equation 2.11). For three dimensions $2 \leq \tau \leq 3$. $c_0 \epsilon A^{\sigma}$ is the surface free energy of a droplet of size A; c_0 is the surface energy coefficient; σ is the critical exponent related to the ratio of the dimensionality of the surface to that of the volume (for percolation, see Section 2.3.2); and $\epsilon = (T_c - T)/T_c$ is the scaled control parameter that measures the distance from the critical point, T_c.

It becomes clear that, when we substitute $\epsilon = (p_c - p_{break})/p_c$ and $T = p_{break}$, Equation 3.9 takes on the form of the cluster scaling relation 2.11 in Section 2.3.2 with the special cutoff function

$$f\left[(p_{break} - p_c)A^{\sigma}\right] = \exp\left[\frac{A\Delta\mu}{p_{break}} - \frac{c_0 \epsilon A^{\sigma}}{p_{break}}\right] \tag{3.10}$$

This once again shows that percolation phenomena, with a geometrical phase transition, share with thermal critical phenomena the same scaling behavior (as well as the same renormalization group and features of universality [13]).

It has been found by Elliot et al. [12] that the bulk factor $\exp[A\Delta\mu/p_{break}]$ for percolation and $\exp[A\Delta\mu/T]$ for the droplet model is very close to unity, i.e. that

$\Delta\mu \approx 0$ in the vicinity of the critical point. With this information, we can rewrite Equation 3.9 as:

$$\langle n_A \rangle / q_0 A^{-\tau} = \exp\left[-\frac{c_0 \epsilon A^{\sigma}}{T}\right] = \exp\left[-c_0 A^{\sigma}\left(\frac{T_c - T}{TT_c}\right)\right] \tag{3.11}$$

Therefore, a logarithmic plot of $\langle n_A \rangle / q_0 A^{-\tau}$ versus the scaled control parameter, $A^{\sigma}\epsilon/T_c$ or $A^{\sigma}\epsilon/p_{break}$, respectively, for a given A, should show a straight line crossing $(0, 1)$ for critical events. Furthermore, all these graphs for different A should collapse. This, of course, only holds for events close to the critical temperature or probability, i.e., for data points with the scaled control parameter close to 0.

Percolation Model

Figure 3.7 shows a plot of the scaled fragment distribution versus the scaled control parameter for the percolation (model) events. Instead of the control parameter in Equation 3.9, we use the scaled parameter multiplied by p_c in order to work with a dimensionless quantity: $Z^{\sigma}(p_c - p_{break})/p_{break}$. Part (a) shows the filtered model data and part (b), the unfiltered model data. The diagrams include plots for Z values ranging from 3 to 8. A critical breaking probability of $p_c = 0.7$ has been chosen. This value differs from the theoretical value of 0.7512 (see Table 2.1, page 10) since we have to take finite-size scaling effects into account. For this purpose, we used the parameterization for the fractional shift given by Bauer [6] for simple cubic lattices:

$$(p_{av} - p_c)/p_c = \epsilon(n) \approx 0.20/n^{0.96} \tag{3.12}$$

Here, n is the linear extent of the system, for which we chose $\sqrt[3]{79}$. This yields a p_{av} value of approximately 0.7.

As critical exponents in this plot we used the values $\sigma = 0.45$ and $\tau = 2.18$ (independent of the lattice topology, see Section 2.2.5), which are published, for example, in [34].

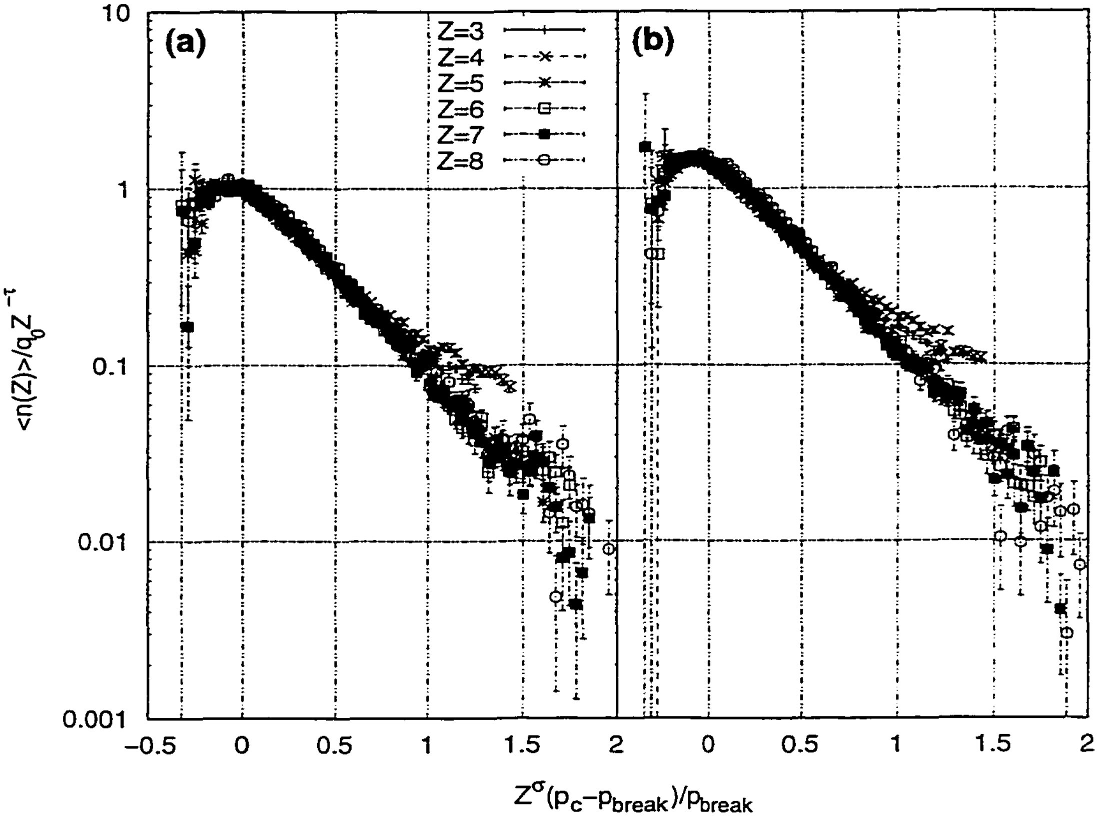

Figure 3.7: Plot of the scaled fragment distribution as a function of the scaled control parameter (multiplied by p_c) for fragments of charge Z. (a) shows the filtered data, (b) the unfiltered model data. In this plot, a critical percolation probability $p_c = 0.7$, $\sigma = 0.45$ and τ=2.18 have been chosen.

It can be seen that the graphs for different Z values both for the filtered data and the unfiltered data collapse onto single curves in the plotted Z range. This is not surprising, since FDM scaling behavior is predicted by standard percolation theory. This feature is obviously not profoundly affected by applying the filter code. The fact that the unfiltered data curve does not cross the "critical point" (0,1) might be due to an imprecise estimate of p_c and the necessity to adjust the values of σ and τ for the filtered data (to account for the filter effects). Besides these effects, the "bend-over" of the curves for small values of the scaled control parameter is a strong indication of the fact that overcritical events are produced in the simulation – independent of the explicit choice of p_c. This deviation from a pure exponential dependence is due to limitations of the FDM to describe droplet formation correctly far beyond the critical point.

To sum up, it becomes clear that only with the right choice of the critical exponents σ and τ and the critical bond-breaking probability p_c, the curves for different Z values will collapse and follow an exponential curve in the vicinity of the critical point. This circumstance can be used to introduce a new method of finding the critical exponents σ and τ, and p_c of the phase transition.

For a given choice of p_c, the "quality" of the FDM scaling is calculated (in terms of a scalar value decreasing with increasing correspondence with the assumed scaling behavior) for points in the $\sigma - \tau$ plane. The result of this calculation can be visualized in a contour plot indicating the agreement of the data with the assumed scaling behavior. The best values for σ and τ are then given by the minima of the profile.

To evaluate the quality of the scaling for a given pair of values for σ and τ, we analyzed the (supposedly) exponential part of the scaled distribution close to the critical point for the individual charges Z. The logarithm of the distributions should have a linear functional form crossing the point (0,ln(1)):

$$\ln\left(\langle n(Z)\rangle q_0 Z^{-\tau}\right) = a_Z \cdot [Z^{\sigma}(p_c - p_{break})/p_{break})] + b_Z \tag{3.13}$$

with the slope a_Z and the offset b_Z. Ideally, b_Z should be 0 and a_Z should equal $a_{Z'}$ for all pairs Z and Z' for which data are available.

Instead of considering theg data points for different Z individually, we combined them into one set of points $\{(x_i, y_i \pm \sigma_i)|i = 1 \ldots N\}$ and determine the degree of correspondence with a linear dependence by applying a "least χ^2" fit. In this method, the function

$$\chi^2(b,a) \equiv \sum_{i=1}^{N} \left(\frac{y_i - b - ax_i}{\sigma_i}\right)^2 \tag{3.14}$$

with two free parameters a and b is minimized. For details, refer to [41]. Since we already know that the data should run through the critical point (0,1) (or, in our consideration, (0,ln(1)=0)), we used

$$\chi^2(0,a) = \sum_{i=1}^{N} \left(\frac{y_i - ax_i}{\sigma_i}\right)^2 \tag{3.15}$$

as a measure for the quality of the fit.

In Figure 3.8, this analysis has been conducted for the unfiltered model data. A critical breaking probability of $p_c = 0.65 \pm 0.2$ has been found to yield the lowest χ^2 values. This value is lower than the value of 0.7 estimated earlier. One reason for this discrepancy could be that the size of the lattice varies from event to event (see Section 3.3.1). For τ and σ, we find the values $\tau = 2.18 \pm 0.01$ and $\sigma = 0.5 \pm 0.1$. These values are in good agreement with the theoretical values of 2.18 and 0.45, respectively. It is quite apparent that the determination of τ is more precise than the determination of σ. This is probably due to the fact that the Z values in this analysis only range from 3 to 8 so that the variation of Z^{σ} with σ is smaller than the variation of Z^{τ} with τ. Also, the normalization constant q_0 depends on τ, but not on σ.

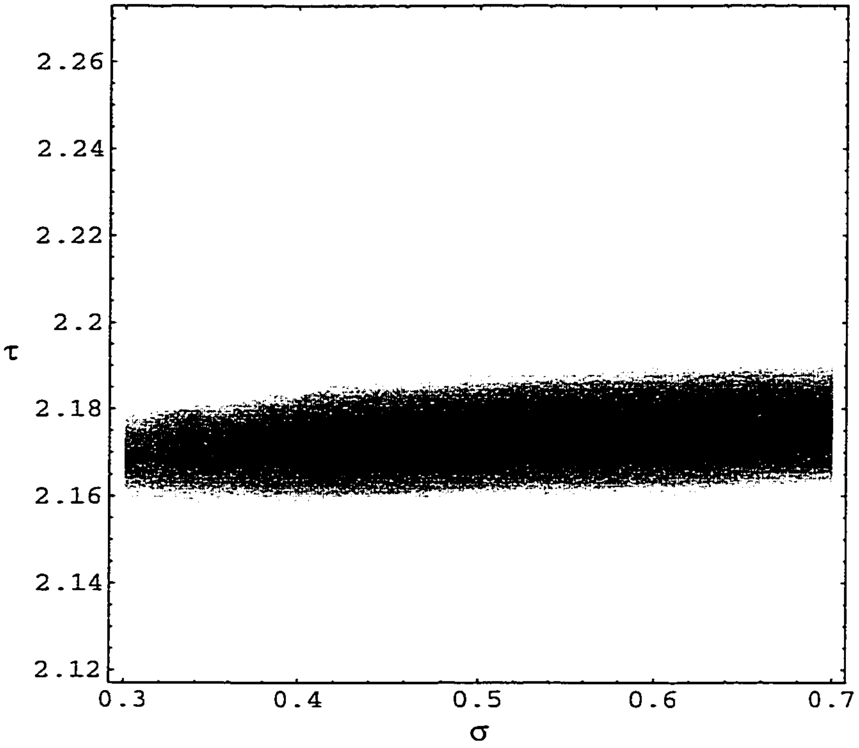

Figure 3.8: Density plot of χ^2 as a function of σ and τ for the unfiltered percolation events. The 4 innermost contours represent $[\chi^2/(\text{degrees of freedom})]$-values of 3.2, 3.5, 3.8 and 4.1, respectively. A critical breaking probability of 0.65 has been used in this plot.

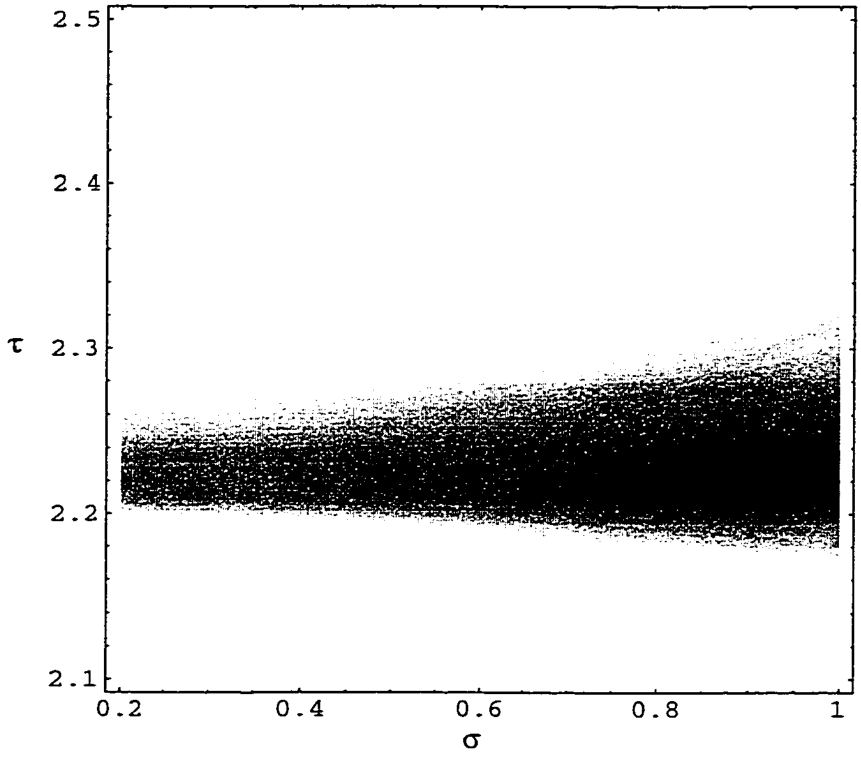

Figure 3.9: Density plot of χ^2 as a function of σ and τ for the experimental data (uncorrected). The innermost contour corresponds to a $[\chi^2/(\text{degrees of freedom})]$-value of ≈ 12.

Experimental data

The true potential of the method described above is to provide a way to determine the critical exponents in the case of experimental data. Unlike other methods, as the analysis of higher moments of the fragment distribution for example, this methods allows to find T_c, σ and τ at the same time. As discussed in Section 2.3.3, the two other critical exponents β and γ can be expressed as functions of σ and τ.

Figure 3.9 shows the result of the determination of σ and τ using the "least χ^2" method. It is apparent that the minimization procedure fails to produce meaningful results. Figure 3.10 (using the "best fit" determined in the analysis visualized in Fig. 3.9) shows that the yields for Z=4 are lower than the yields for Z=3. Since these differences should be compensated for by the factor Z^τ in the scaling of the cluster

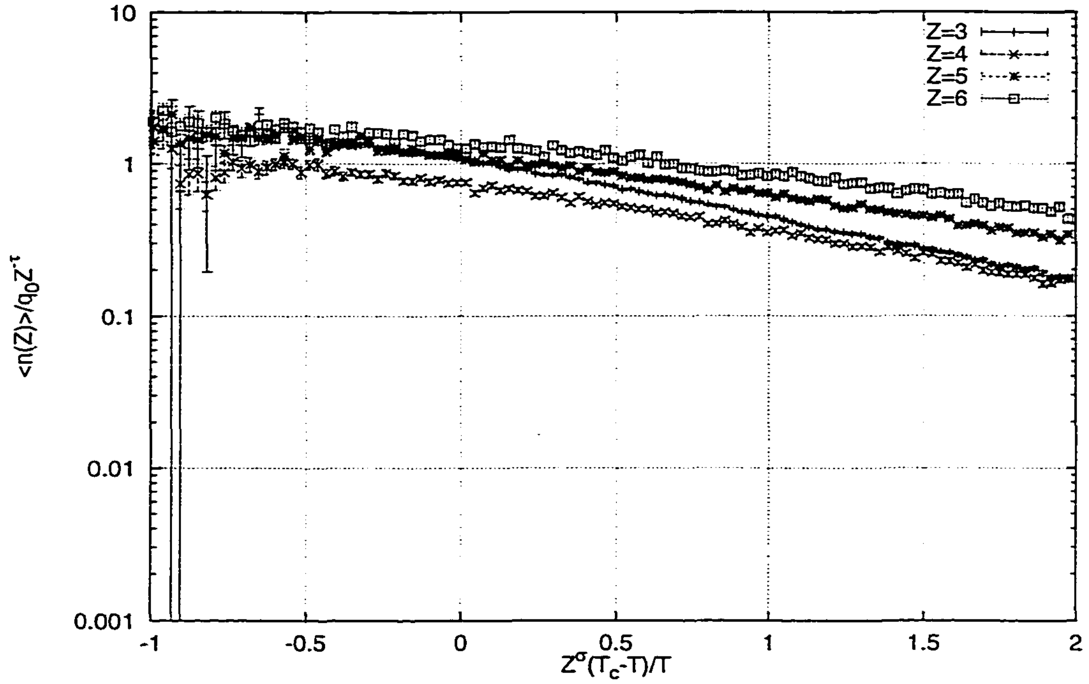

Figure 3.10: Plot of the scaled fragment distribution as a function of the scaled control parameter (multiplied by T_c) for the experimental data (uncorrected). Values of $\sigma = 0.99$, $\tau = 2.13$ and $T_c = 9.5$ MeV have been used.

distribution, it becomes clear that the order of $Z = 3$ and $Z = 4$ cannot be changed without increasing the discrepancy between the yields of $Z = 5$ and $Z = 6$.

The reason for this problem is that the fragments of the nuclei that are produced in the experiment can undergo sequential decays before they are detected. The energy of the incident particle in the experimental setup described is high enough to induce isospin-symmetric fragmentation reactions; that is, at the time a nucleus breaks up, the actual composition of a fragment (in terms of isospin) barely affects the probability for its emission. Thus, both fragments with stable and unstable neutron/proton ratios will be produced. Therefore, the primary yield of fragments, which would carry the signatures of the phase transition, i.e., which would exhibit FDM scaling behavior, is altered (through decays) before its detection and is not directly attainable from the experimental data. For a more detailed discussion, see for example [19].

Furthermore, the "non-perfect" detection process introduces changes that are not related to the phenomenon we want to study, and should be compensated for. The latter problem is overcome by extracting correction factors for individual charges Z accounting for detection effects from the model data before and after filtering (see Figure 3.2). From a comparison of part (a) and part (b) of Figure 3.7, it can be seen that the shift in the data for different Z is mostly coherent, i. e., all curves are approximately shifted by the same amount. However, this shift impacts the determination of the critical breaking probability or temperature, respectively, and should therefore be considered in the analysis.

In order to obtain correction factors for the sequential decay effect, we used the "EASY" (Exact Nuclear Statistical Yields) code written by Pratt [40]. The simulation requires the size of the nucleus and the excitation temperature as inputs. We used the average value of 65 for the number of protons in the decaying nucleus (see Section 3.3.2) and a number of neutrons corresponding to the neutron/proton ratio of an Au nucleus (assuming iso-spin symmetric emission of pre-equilibrium particles). We found that a temperature of 7 MeV yields the best results. The EASY code produces data for primary (after fragmentation) and final (after sequential decay) yields for different (p,n) configurations. We summed over n in order to calculate yields and ratios depending only on the charge p. Unfortunately, since the EASY simulation was originally not intended for analyses of the low mass region and because it is still considered as "work in progress", we could only obtain useful correction factors for charges up to 6.

Figure 3.11 shows the results of the χ^2 fitting procedure for the critical exponents σ and τ (and the critical temperature T_c) using the sequential decay correction. A critical temperature of 8.3 ± 0.2 Mev has been used in this plot. This value is comparable to theoretical estimates [18]. Again, it is difficult to determine the value

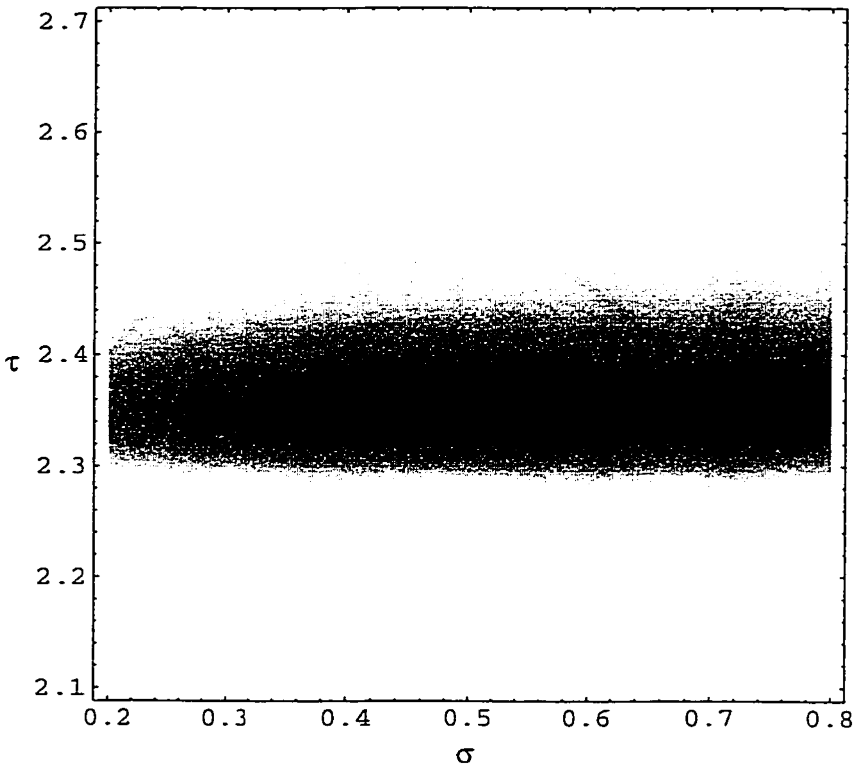

Figure 3.11: Density plot of χ^2 as a function of σ and τ for the experimental data (with filter- and sequential decay correction. The 4 innermost contours represent $[\chi^2/($degrees of freedom]-values of 3.3, 3.7, 4.1 and 4.6, respectively. A critical temperature of 8.3 MeV has been used in this plot.

of σ exactly. In this case, the range of Z values is even smaller than for the percolation calculation. We derive the numerical values $\sigma = 0.5 \pm 0.1$ and $\tau = 2.35 \pm 0.05$.

However, it should be mentioned here that the errors are estimated from the fitting procedure. The actual values of T_c, σ and τ to a large degree also depend on the correction factors for sequential decays. Therefore, a precise determination of these factors is crucial in order to obtain quantitative results and certainly also adds systematic uncertainties to the given values.

In this analysis, we are only interested in the critical temperature and the critical exponents of the phase transition. In order to derive values for other parameters in the Fisher Droplet Model, like the surface free energy coefficient c_0 (see Equation 3.9), it is necessary to use the masses A of the fragments instead of the charges (see [11]). One possibility to do this (since no information about masses is available in the data set) is to scale all charges Z by the mass/charge ratio of the Au nucleus. Under this redefinition of the fragment sizes, one also has to rescale the normalized droplet distribution $\langle n_Z \rangle$ to $\langle n_A \rangle$ by dividing through the mass/charge ratio. Otherwise, coherent shifts in the FDM scaling plots yield different values of the critical temperature.

In Figure 3.12, an FDM scaling plot is given with the optimized parameters T_c, σ and τ found above. It can be seen that the data collapse very well onto a single curve. Again, this agreement with Fisher's model must be interpreted as a strong indication at a liquid-gas type phase transition in excited nuclei.

As for future analyses, it would be desirable to determine sequential decay correction factors for particles with charges greater than 6. A broader Z range should help to alleviate the significant uncertainties in the σ determination.

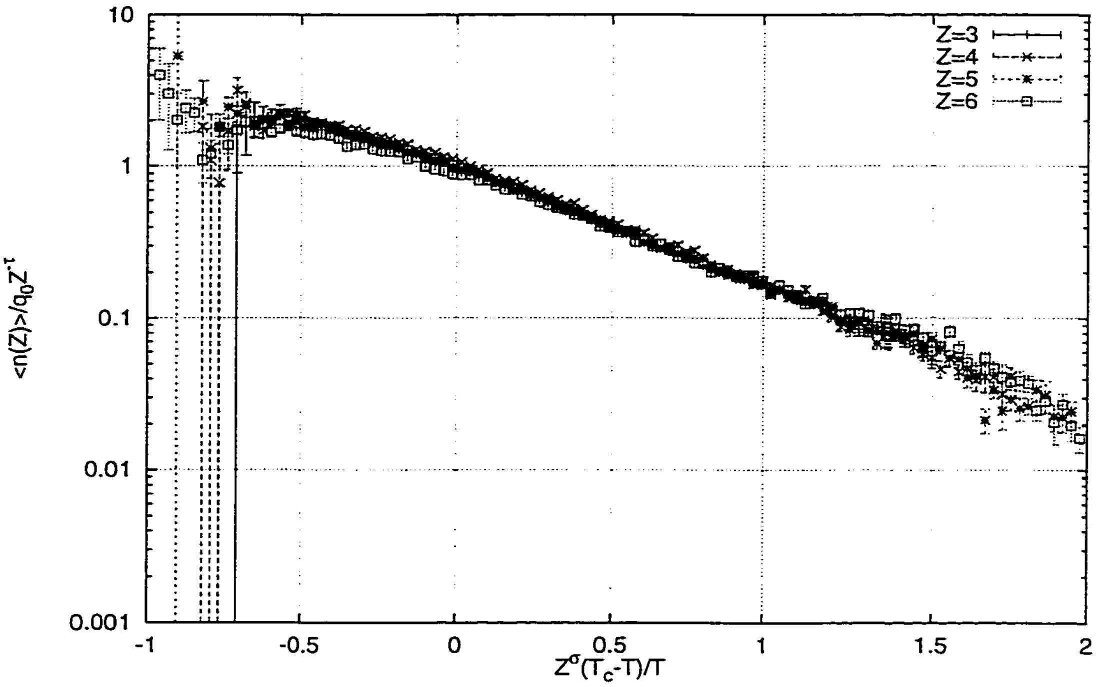

Figure 3.12: Plot of the scaled fragment distribution as a function of the scaled control parameter (multiplied by T_c) for the experimental data (with filter- and sequential decay corrections) Values of $\sigma = 0.5$, $\tau = 2.35$ and $T_c = 8.3$ MeV^{σ} have been used.

Chapter 4

Summary and Conclusion

Percolation theory represents a simple statistical model for systems undergoing a phase transition. Despite the simplicity of the model, it shows all the features of critical behavior, universality and critical exponents. Unlike most other (thermal) models in statistical physics, the phase transition dealt with in percolation theory is of a purely geometrical nature. The role of the adjustable parameter "temperature" is played by a probability (the bond-breaking probability in bond percolation or the probability for a site to be occupied in site percolation, for example), and almost all properties and functions that mark the critical behavior of the system are geometrical. One of the most important quantities describing all crucial features of the system, for instance, is the (normalized) distribution of cluster sizes. As in thermal phase transitions, simple power laws govern the system behavior in the vicinity of the phase transition point. The critical exponents introduced in this context have been shown to be independent of topological details of the model and to depend only on the dimensionality of the system. This can be interpreted as a strong indication that results derived from (certainly simplifying) models based on percolation theory can also be of practical relevance for "real world" applications.

Consequently, percolation theory has been successfully applied to all kinds of different areas inside and outside of physics (for examples, see [1, 38, 31, 46]). Even though hardly any analytical results can be derived, the percolation models' simplicity makes them ideally qualified to be analyzed in Monte Carlo simulations with rather moderate demands in computational performance.

To summarize, Chapter 2 of this thesis presented a short introduction to the basic ideas of percolation theory. Furthermore, the general concepts and features of phase transitions were explained in this chapter, and the atomic nucleus as a system which is believed to undergo a liquid-gas type phase transition was introduced. Traces for this transition are expected to be found in nuclear multifragmentation reactions. A percolation model for nuclear multifragmentation reactions presented by Bauer [6] was described.

Chapter 3 dealt with the application of this model to the interpretation of a data set of proton and pion induced multifragmentation reactions that has been collected by the ISiS collaboration at the AGS accelerator facility at the Brookhaven National Laboratory. In trying to reproduce the experimental data by model calculations, it was shown that some modifications of the model have to be made. In particular, the comparably low energy (for this kind of reactions) of the fragmentation-inducing projectiles made it necessary to carefully analyze the size of the thermalized source, which could no longer be determined by simple geometric considerations (see Sections 2.4.1, 3.3.1 and Figure 2.5). Also, the fact that the ISiS data set provided information about the excitation energy of the source allowed us to determine bond-breaking probabilities that are directly correlated to the experiment (Section 2.4.2) by employing a $p_{break}(T)$ relation of the hybrid model. In a more detailed analysis of the data, we found that it is important to take into account effects that arise from inevitable experimental constraints in the detection process. In the model, this was done by introducing a filter that simulates the non-detection of fragments (Section

3.2.2). Especially quantities depending on the multiplicity of events and moments of the cluster size distribution were shown to be sensitive to the changes induced by the filter.

Finally, the data were analyzed with respect to their scaling behavior according to the Fisher Droplet Model. Since this model has been introduced to describe phase transition behavior in liquid-gas systems, the actual analysis allowed us to conclude that traces of a phase transition are present in the experimental data. Also, the FDM scaling behavior can be described in terms of two critical exponents σ and τ, which characterize this type of transition. A method for the determination of these two exponents, together with the critical temperature T_c of the transition, based on FDM scaling was explained in this work. In conducting this analysis, we also tried to compensate for the filter effects discovered in the model calculations and for sequential decay processes inherent in this type of reactions. It has been shown that the latter have a great impact on the quality of the analysis. We derived values of $T_c = 8.3 \pm 0.2$ MeV, $\sigma = 0.5 \pm 0.1$ and $\tau = 2.35 \pm 0.05$.

We believe that this method can also be applied to other data sets. Its advantage over other methods is that it allows for the determination of three important quantities characterizing the phase transition at the same time. It would be interesting to compare the results achieved with this method with numerical values derived from other methods for other sets of data.

To conclude, it is obvious that with recent experimental data, new information about the nuclear phase diagram can obtained. The transition is still far from being understood. Therefore, studies of the nuclear phase diagram provide numerous opportunities for future research, both in trying to interpret experimental results and in trying to develop new theoretical models or refine existing ones.

APPENDIX

Appendix A

Source Code

In the appendix, we will present the most important parts of the source code used for the simulations discussed in this thesis. All programs were written in C++ and were compiled with different C++-compilers on different architectures.

A.1 isisevent.h and isisevent.cpp

The C++-class isisEvent is used to interface the data file provided by the ISiS corporation. Since the results of our percolation simulations are stored in the same format, they are also accessed using this class. The class contains routines for input/output of events and basic analyses.

Headerfile:

```
#ifndef ISISEVENT_H
#define ISISEVENT_H

#include <slist>
#include <iostream>
#include <fstream>
#include <string>

#define isisnum         float
#define max_frag_num    79
#define buffersize_b    (32768+4)*2
```

```
class isisEvent
{
public:
  isisEvent (string Typ = "") ;
  ~isisEvent () ;
  isisnum               mult ;
  isisnum       ExA ;
  isisnum               Zsrc ;
  isisnum               Zres ;
  isisnum               moment(int m) ;
  isisnum               moment(int m, int cut) ;
  isisnum               moment_zres (int m) ;
  isisnum               moment_zres (int m, int cut) ;
  isisnum         moment_zres_ex16_20 (int m) ;
  isisnum         moment_zres_ex16_20 (int m, int cut) ;
  isisnum         sourcesize() ;
  bool          fission() ;
  void            rearrange() ;
  isisnum         realmult() ;
  bool            tagged() ;
  isisnum               Ztherm[max_frag_num] ;

  string                eventtype ;
private:
  friend istream& operator>>(istream& is, isisEvent &iE) ;
  friend ostream& operator<<(ostream& os, isisEvent &iE) ;
  friend ifstream& operator>>(ifstream& ifs, isisEvent &iE) ;
  friend ofstream& operator<<(ofstream& ofs, isisEvent &iE) ;
} ;

#endif
```

Implemenation:

```
#include "isisevent.h"
#include <cmath>

#define min(a,b)        (((a) < (b)) ? (a) : (b))

isisEvent::isisEvent (string Typ)
{
  eventtype = Typ ;
}

isisEvent::~isisEvent ()
{
}

isisnum isisEvent::moment (int m)
{
  isisnum mom = 0 ;

  if (m <= 0) return mult ;
  else {
    for (int i=0; i<mult; i++)
      {
        mom += pow(Ztherm[i], m) ;
      }
    return mom ;
  }
}

isisnum isisEvent::moment (int m, int cut)
{
  isisnum mom = 0 ;

  if (m <= 0) return mult ;
  else {
    for (int i=0; i<mult; i++) if (Ztherm[i] <= cut) mom += pow(Ztherm[i], m) ;
    return mom ;
  }
}


isisnum isisEvent::moment_zres (int m)
{
  isisnum mom=0 ;

  if (m <= 0) return mult+1 ;           // +1 because of zres
  else {
    for (int i=0; i<mult; i++) mom += pow(Ztherm[i], m) ;
    mom += pow(Zres, m) ;
    return mom ;
  }
}

isisnum isisEvent::moment_zres (int m, int cut)
{
  isisnum mom=0 ;

  if (m <= 0) return mult+1 ;           // +1 because of zres
  else {
    for (int i=0; i<mult; i++) if (Ztherm[i] <= cut)  mom += pow(Ztherm[i], m) ;
    if (Zres <= cut) mom += pow(Zres, m) ;
    return mom ;
  }
}

isisnum isisEvent::moment_zres_ex16_20 (int m)
{
  isisnum mom=0 ;

  if (m <= 0) return mult+1 ;           // +1 because of zres
  else {
    for (int i=0; i<mult; i++) if (Ztherm[i] <= 16)  mom += pow(Ztherm[i], m) ;
    mom += pow(Zres, m) ;
    return mom ;
  }
}

isisnum isisEvent::moment_zres_ex16_20 (int m, int cut)
{
  isisnum mom=0 ;

  if (m <= 0) return mult+1 ;           // +1 because of zres
  else {
    for (int i=0; i<mult; i++)
      if (Ztherm[i] <= min(cut, 16)) mom += pow(Ztherm[i], m) ;
    if (Zres <= cut) mom += pow(Zres, m) ;
    return mom ;
  }
}

isisnum isisEvent::sourcesize ()
{
  return moment_zres(1) ;
}

bool isisEvent::fission ()
{
  bool temp = false ;

  for (int i=0; i<mult; i++) temp = temp | (Ztherm[i] == 25) ;

  return temp ;
}

void isisEvent::rearrange ()
{
  int i, j ;
  isisnum temp ;

  if (mult > 1)
    for (i=0; i<mult-1; i++)
      for (j=i; j<mult; j++)
        if (Ztherm[i] > Ztherm[j]) {
          temp = Ztherm[j] ;
          Ztherm[j] = Ztherm[i] ;
          Ztherm[i] = temp ;
        }

  Zres = 0 ;
  for (i=0; i<mult; i++)
    if (Ztherm[i] > 1000) Zres += Ztherm[i]-1000 ;
}

isisnum isisEvent::realmult ()
{
  int i ;
  isisnum mul=0 ;
  for (i=0; i<mult; i++)
    if (Ztherm[i] < 1000) mul++ ;

  return mul ;
}

bool isisEvent::tagged ()
{
  int i ;
  bool temp = false ;

  for (i=0; i<mult; i++) temp = temp | (Ztherm[i] > 1000) ;

  return temp ;
}

istream& operator>>(istream& is, isisEvent &iE)
{
  short dummy, i ;

  is >> dummy ; iE.mult = (isisnum)dummy ;
  is >> dummy ; iE.ExA = (isisnum)dummy ;
  is >> dummy ; iE.Zsrc = (isisnum)dummy ;
  is >> dummy ; iE.Zres = (isisnum)dummy ;

  if (iE.mult < max_frag_num)
    for (i=0; i<iE.mult; i++) {
      is >> dummy ;
      iE.Ztherm[i] = (isisnum)dummy ;
    }

  return is ;
}

ifstream& operator>>(ifstream& ifs, isisEvent &iE)
{
  short dummy[max_frag_num] ;

  ifs.read(&dummy, 4*2) ;
  iE.mult = (isisnum)dummy[0] ;
  iE.ExA = (isisnum)dummy[1]/100.0 ;
  iE.Zsrc = (isisnum)dummy[2] ;
  iE.Zres = (isisnum)dummy[3] ;

  if (iE.mult < max_frag_num)
    {
      ifs.read(&dummy, (int)iE.mult*2) ;
      for (int i=0; i<iE.mult; i++) iE.Ztherm[i] = (isisnum)dummy[i] ;
    }
  return ifs ;
}

ofstream& operator<<(ofstream &ofs, isisEvent &iE)
{
  short dummy[max_frag_num] ;

  dummy[0] = (short)iE.mult ;
  dummy[1] = (short)(100*iE.ExA) ;
  dummy[2] = (short)iE.Zsrc ;
  dummy[3] = (short)iE.Zres ;
  ofs.write(dummy, 4*2) ;

  if (iE.mult < max_frag_num) {
    for (int i=0; i<iE.mult; i++) dummy[i] = (short)iE.Ztherm[i] ;
  }
  ofs.write(dummy, (int)iE.mult*2) ;

  return ofs ;
}

ostream& operator<<(ostream& os, isisEvent &iE)
{
  os << iE.eventtype << endl ;
  os << iE.mult << endl ;
  os << iE.ExA << endl ;
  os << iE.Zsrc << endl ;
  os << iE.Zres << endl ;

  for (int i=0; i<iE.mult; i++) {
    os << iE.Ztherm[i] << endl ;
  }

  return os ;
}
```

A.2 percolate.cpp

The program percolate.cpp contains the actual simulation implementing the NLM. It writes its results to a datafile in the same format in which the data of the ISiS group was provided. The parameters are read from the standard input, which allows us to write batch files via redirection of the standard input. percolate.cpp makes use of the external routines Gamma(double a, double z0, double z1) and filter(double theta, double phi, double z, double &a, double e, double &ring, double &det, double &cod) implemented in the files gamma.cpp and filter.cpp respectively.

```
/***************************************************************************
                          percolate.cpp  -  description
                             -------------------
    begin                : Tue, Oct 10, 2000
 ***************************************************************************/

#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
#include <cmath>
#include "isisevent.h"
#include "isisreadbuffer.h"
#include "filter.h"
#include "gamma.h"

#define PI      3.141592654

#define min(a,b)        (((a) < (b)) ? (a) : (b))
#define max(a,b)        (((a) > (b)) ? (a) : (b))

int     startbuffer = 1 ;
int     endbuffer = 726 ;

ifstream datafile ;
ofstream outfile ;
char     outfilename[256] ;


isisEvent eventlist[eventbuffersize] ;
isisEvent percevents[eventbuffersize] ;
int       evcount, aevent=0, abuffer=startbuffer ;

bool    clstmb[12][12][12] ;            // Clustermembers
bool    con[10][10][10][3] ;            // Bonds
bool    nonucl[10][10][10] ;            // sites

long int        newpts[1000][3] ;
long int        nucnum[10][10][10] ;
long int        mevent[1000] ;


int     A1 ;            // target charge
int     Ar ;            // target mass
int     A1r ;           // backup for target charge
int     Mass ;
int     nx, ny, nz ;
int     multip ;
int     noruns ;
bool    fixed ;
bool    tag_undetected ;
```

```
double  pbreak ;

double  r1, r2 ;                // target / projectile radius
unsigned long int irand0 ;

bool    Filter ;

int             mlow, mhigh ;           // cutoff masses

double          isisT, isisE ;
double          B = 6.7 ;               // Binding energy per nucleon

double ran_ (unsigned long int *i)
{
  return (double)rand()/RAND_MAX ;
}

void    input_params ()
{
  int h ;

  cin >> A1 ;                           //  Target Mass
  A1r = A1 ;
  cin >> Ar ;
  cin >> B ;                            //  Binding energy per nucleon [MeV]
  cin >> irand0 ;
  cin >> noruns ;                       //  ov &= (onoruns == noruns) ;
  cin >> mlow ;
  cin >> mhigh ;
  cin >> h ;
  fixed = (h != 0) ;                    //   break fixed number of bonds
  cin >> h ;                            //   Model detector efficiency
  Filter = (h != 0) ;
  cin >> h ;
  tag_undetected = (h != 0) ;           // just mark undetected,
  cin >> startbuffer ;                  // without creating zres...
  abuffer = startbuffer ;
  cin >> endbuffer ;
  cin >> outfilename ;
}

inline double sqr (double x)
{       return x*x ; }

void shape ()
{
  const double  xs = 0.05 ;
  const double  ys = 0.07 ;
  const double  zs = 0.13 ;

  double        D[11][11][11] ;
// upper array bound 11 -> 1..10 can be used (no messing with index shift)

  double        x0, y0, z0 ;
  int           ix, iy, iz, ixdmin, iydmin, izdmin ;
  int           n1 ;
  double        dmin, dxyz ;

  for (ix=0; ix<10; ix++)
    for (iy=0; iy<10; iy++)
      for (iz=0; iz<10; iz++)   nonucl[ix-1][iy-1][iz-1] = true ;

  r1 = pow((0.75*(double)A1/PI), 1.0/3.0) ;
  nx = (ny = (nz = 2*((int)r1+1))) ;

  x0 = ((double)nx+1.0)/2.0 + xs ;
  y0 = ((double)ny+1.0)/2.0 + ys ;
  z0 = ((double)nz+1.0)/2.0 + zs ;

  // Distribute nucleons approx. spherical in cubic lattice
  for (ix=1; ix<=nx; ix++)
    for (iy=1; iy<=ny; iy++)
      for (iz=1; iz<=nz; iz++)
        D[ix][iy][iz] = sqrt(sqr(ix-x0)+sqr(iy-y0)+sqr(iz-z0)) ;

  ixdmin = iydmin = izdmin = 0 ;
  for (n1=1; n1<=A1; n1++) {
    dmin = 999 ;
    for (ix=1; ix<=nx; ix++)
      for (iy=1; iy<=ny; iy++)
        for (iz=1; iz<=nz; iz++) {
          dxyz = D[ix][iy][iz] ;
          if (dxyz < dmin) {
            ixdmin = ix ;
            iydmin = iy ;
            izdmin = iz ;
            dmin = dxyz ;
          }
        }
    nonucl[ixdmin-1][iydmin-1][izdmin-1] = false ;
    D[ixdmin][iydmin][izdmin] = 9999 ;
  }

  Mass = A1 ;
}

void break_bonds ()
{
  //    extern double ran_ (unsigned long int *) ;

  int   nbreak, nbonds = 0 ;
  int   idir ;
  int   ix, iy, iz, ixnext, iynext, iznext, i, n ;
  int   mix[2000], miy[2000], miz[2000], mid[2000] ;

  for (idir=0; idir<3; idir++) {
    ixnext = 0 ;
    iynext = 0 ;
  iznext = 0 ;

  if (idir == 0) ixnext = 1 ;
  else if (idir == 1) iynext = 1 ;
  else iznext = 1 ;

  for (iz=0; iz<nz; iz++)
   for (iy=0; iy<ny; iy++)
    for (ix=0; ix<nx; ix++) {

     if (nonucl[ix][iy][iz]) {
      con[ix][iy][iz][idir] = false ;
      continue ;
     }

     if (nonucl[ix+ixnext][iy+iynext][iz+iznext])
      con[ix][iy][iz][idir] = false ;
     else {
      if (fixed) {
       con[ix][iy][iz][idir] = true ;
       nbonds++ ;
       mix[nbonds-1] = ix ;
       miy[nbonds-1] = iy ;
       miz[nbonds-1] = iz ;
       mid[nbonds-1] = idir ;
      }
      else {
       if (ran_(&irand0) < pbreak) con[ix][iy][iz][idir] = false ;
       else {
        con[ix][iy][iz][idir] = true ;
        nbonds++ ;
       }
      }
     }
    }

 }

 nbreak = (int)(pbreak*nbonds+0.5) ; // <- "round"
 if (fixed) {
  for (n=1; n<=nbreak; n++) {
   do {
    i = (int)(0.5+ran_(&irand0)*nbonds+0.5) ; // "round"
   } while ((i <= 0) || (i > nbonds)) ;
   if (con[mix[i-1]][miy[i-1]][miz[i-1]][mid[i-1]])
    con[mix[i-1]][miy[i-1]][miz[i-1]][mid[i-1]] = false ;
   else {
    i++ ;
    if (i > nbonds) i = 1 ;
   }
  }
 }

}
```

```
void newtst (int ipx, ipy, ipz, icx, icy, icz, idir, int& imax)
{
  if (!clstmb[ipx][ipy][ipz])
    if (con[icx-1][icy-1][icz-1][idir-1]) {
      newpts[imax-1][0] = ipx ;
      newpts[imax-1][1] = ipy ;
      newpts[imax-1][2] = ipz ;
      imax++ ;
      clstmb[ipx][ipy][ipz] = true ;
    }
}

bool not_detected (int z)
{
#define epsilon0 8.85419e-12
#define e0       1.602e-19*1e15*1e-6
// 1e15 correction due to fact that r given in fm, 1e-6 for MeV !

  double theta, phi ;
  double r, E ;
  double bmax ;
  double a, ring, det, cod ;
  double kappa = 0.6;
  double T = 12.0       ;       // Temperature in MeV
  double Barrier ;


  if (!Filter) return false ;

  // determine ejection angle...
  phi = ran_(&irand0)*360.0 ;

  do {
    theta = ran_(&irand0)*PI ;
  } while (ran_(&irand0) > sin(theta)) ;   // <- rejection method !?
  theta *= 360.0/(2.0*PI) ;

  // determine Energy-distribution plus Coulomb-barrier
  bmax = sqrt(0.5*T)*exp(-0.5) ;


  r = 1.2*(pow(z*Ar/A1r, 1.0/3.0)+pow(Mass*Ar/A1r, 1.0/3.0)) ;
  Barrier = e0*z*(Mass-z)/(4*PI*epsilon0*r) ;

  if ((Ar*z/A1r-(int)(Ar*z/A1r)) > ran_(&iran0)) a = (int)(Ar*z/A1r)+1.0 ;
      else a = (int)(Ar*z/A1r);

  if (a == 0) a == 1 ;

  do {
    do {
      E = ran_(&irand0)*20*T ;
    } while (ran_(&irand0) > sqrt(E)*exp(-E/T)/bmax) ;
    E = (E + kappa*Barrier)*(Ar-a)/Ar ;
  } while (E < 0) ;

  filter(theta, phi, z, a, E, ring, det, cod) ;

  if (a == 0) return true ;
  else return false ;
}

bool process_isisevent ()
{
  isisEvent     ev ;

  if (aevent < evcount) {
    ev = eventlist[aevent++] ;

    if (ev.fission()) return false ;

    isisT = sqrt(13.0*ev.ExA) ;
    isisE = ev.ExA ;
    A1 = (int)ev.sourcesize() ;
    if (A1 > 79) return false ;
    return true ;
  }
  else {
    if (abuffer < endbuffer) {
      evcount = isisreadbuffer(datafile, &(eventlist[0])) ;
      abuffer++ ;
      aevent = 0 ;
    }
    else {
      datafile.seekg((startbuffer-1)*buffersize_b, ios::beg) ;
      evcount = isisreadbuffer(datafile, &(eventlist[0])) ;
      abuffer = startbuffer ;
      aevent = 0 ;
    }
    return false ;
  }
}

void percolate ()
{
  int           ix, iy, iz, iix, iiy, iiz ;
  int           imaxo, imino, imax, imin ;
  int           nrun, i ;
  int           nxp1, nyp1, nzp1 ;
  int           ifrg ;
  double                zres ;
  short           iword ;
  int             events, overallevents, buffers ;

  events = 0 ;
  iword = 1 ;  // 1 for iword
```

```
overallevents = 0 ;
buffers = 0 ;
// ++++++++++ Main loop over events   ***********
for (nrun=1; nrun<=noruns; nrun++)
  {
    //        Experimental data
    do {} while (!process_isisevent()) ;
    pbreak = 1.0-2.0*Gamma(1.5, 0, B/isisT)/sqrt(PI) ;
  L10:

    // Determination of nucleon distribution on the lattice
    shape() ;

    nxp1 = nx + 1 ;
    nyp1 = ny + 1 ;
    nzp1 = nz + 1 ;

    for (ix=1; ix<=nx; ix++)
      for (iy=1; iy<=ny; iy++) {
        clstmb[ix][iy][0] = true ;
        clstmb[ix][iy][nzp1] = true ;
      }

    for (ix=1; ix<=nx; ix++)
      for (iz=1; iz<=nz; iz++) {
        clstmb[ix][0][iz] = true ;
        clstmb[ix][nyp1][iz] = true ;
      }

    for (iy=1; iy<=ny; iy++)
      for (iz=1; iz<=nz; iz++) {
        clstmb[0][iy][iz] = true ;
        clstmb[nxp1][iy][iz] = true ;
      }

    // breaking of bonds
    break_bonds() ;
    zres = 0 ;

    // Cluster recognition algorithm:

    //        1) Initialization of array clstmb

    for (iz=1; iz<=nz; iz++)
      for (iy=1; iy<=ny; iy++)
        for (ix=1; ix<=nx; ix++)
          clstmb[ix][iy][iz] = nonucl[ix-1][iy-1][iz-1] ;

    multip = 0 ;

    //        2) Cluster recognition and size determination:

    for (iz=1; iz<=nz; iz++)
      for (iy=1; iy<=ny; iy++)
    for (ix=1; ix<=nx; ix++)
      if (!clstmb[ix][iy][iz]) {
        //        new cluster, --- start ---
        clstmb[ix][iy][iz] = true ;
        newpts[0][0] = ix ;
        newpts[0][1] = iy ;
        newpts[0][2] = iz ;
        imino = 1 ;
        imaxo = 1 ;
      L1400:
        imin = imaxo + 1 ;
        imax = imin ;
        for (i=imino; i<=imaxo; i++) {
          iix = newpts[i-1][0] ;
          iiy = newpts[i-1][1] ;
          iiz = newpts[i-1][2] ;
          newtst(iix-1, iiy, iiz, iix-1, iiy, iiz, 1, imax) ;
          newtst(iix+1, iiy, iiz, iix, iiy, iiz, 1, imax) ;
          newtst(iix, iiy-1, iiz, iix, iiy-1, iiz, 2, imax) ;
          newtst(iix, iiy+1, iiz, iix, iiy, iiz, 2, imax) ;
          newtst(iix, iiy, iiz-1, iix, iiy, iiz-1, 3, imax) ;
          newtst(iix, iiy, iiz+1, iix, iiy, iiz, 3, imax) ;
        }

        if (imax != imin) {
          imino = imin ;
          imaxo = imax-1 ;
          goto L1400 ;
        }
        if (imaxo == A1) goto L10 ;

        if (imaxo > 79) goto L10 ;

        //        Determine if fragment was detected
        if (not_detected(imaxo)) {
          if (tag_undetected) {
            multip++ ;
            mevent[multip-1] = imaxo + 1000 ;
          }
          else {
            zres += imaxo ;
          }
        }
        else {
          if ((imaxo >= mlow) && (imaxo <= mhigh)) {
            multip++ ;
            mevent[multip-1] = imaxo ;
          }
        }
        //        New cluster, -- end --
      }

percevents[events].mult = (isisnum)multip ;
percevents[events].ExA = (isisnum)isisE ;
```

```
      percevents[events].Zsrc = (isisnum)A1 ;
      percevents[events].Zres = (isisnum)zres ;

      for (ifrg=0; ifrg<multip; ifrg++)
        percevents[events].Ztherm[ifrg] = (isisnum)mevent[ifrg] ;
      if (tag_undetected) percevents[events].rearrange() ;

      iword += multip + 4 ;
      events++ ;
      overallevents++ ;

      if (iword > 32768-(79+4)) {
        outfile.write(&iword, 2) ; // Fortran?
        outfile.write(&iword, 2) ; // Fortran?
        outfile.write(&iword, 2) ;
        for (i=0; i<events; i++) {
          outfile << percevents[i] ;
        }
        outfile.write(&(percevents[0]), (32768+2-iword)*2) ;
        buffers++ ;
        iword = 1 ;
        events = 0 ;
      }
    }
  if (iword > 0) { // write last buffer, if necessary
    outfile.write(&iword, 2) ;
    outfile.write(&iword, 2) ;
    outfile.write(&iword, 2) ;
    for (i=0; i<events; i++) outfile << percevents[i] ;
    outfile.write(&(percevents[0]), (32768+2-iword)*2) ;
    buffers++ ;
  }
  cout << "Written " << overallevents << " events in " ;
  cout << buffers << " buffers." << endl ;
}

int main (int argc, char *argv[])
{
  input_params() ;
  datafile.open("pAu102.bin") ;
  datafile.seekg((startbuffer-1)*buffersize_b, ios::beg) ;
  evcount = isisreadbuffer(datafile, &(eventlist[0])) ;

  outfile.open(outfilename) ;
  srand(irand0) ;
  percolate() ;

  datafile.close() ;
  outfile.close() ;
}
```

A.3 filter.cpp

This source code contains the filter used to model the detector efficiency of the ISiS detector. It is a C++-port of the Fortran version provided by the ISiS group.

```
#include "filter.h"
#include "baddet.h"

bool first = true ;

extern unsigned long irand0 ;
extern double ran_ (unsigned long int *) ;

double seuilsim[16] = {3., 1., 0.85, 0.89, 0.9, 0.83, 0.85, 0.93, 0.94,
                       1.0, 0.96, 1.08, 1.04, 1.17, 1.22, 1.31 } ;
double Esimax[5]    = {12.5, 38.6, 65.9, 100.9, 136. } ;
double lim_tet[11]  = {14, 22, 33, 52, 69, 86.4, 93.6, 111, 128, 147, 166} ;
double thepp[9]     = {18.0, 27.5, 42.5, 60.5, 77.5, 102.5, 119.5, 137.5, 156.5} ;
double limtet[18]   = {14.0, 21.5, 22.5, 32., 34., 51., 53., 68., 70., 86.4,
                       93.6, 110., 112., 127., 129., 146., 148., 166.} ;
double e_th[9][20] ;


void filter (double theta, double phi, double z, double &a, double e,
             double &ring, double &det, double &cod)
{
  double       ir, iz, et, ea ;
  double       phimin, phimax, ip ;
  int   j, ic ;
  bool  out ;

  if (first) {
    ifstream eth("eth.dat") ;
    while (!eth.eof()) {
      eth >> ir ;
      eth >> iz ;
      eth >> et ;
      eth >> ea ;
      e_th[(int)ir-1][(int)iz-1] = ea ;
    }
    eth.close() ;
    first = false ;
  }
  det = 0 ;

  if (theta < lim_tet[0]) det = -100 ;
  if (theta > lim_tet[10]) det = -100 ;
  if ((theta >= lim_tet[5]) && (theta < lim_tet[6])) det = -100 ;

  if ((int)det == -100) {
    ring = 0 ;
  det = 0 ;
  a = 0 ;
  return ;
}

out = true ;
for (j=1; (j<=10)&&(out); j++) {
  if (j <= 5) {
    if ((theta >= lim_tet[j-1]) && (theta < lim_tet[j])) {
      ring = j ;
      out = false ;
    }
  }
  else if (j >= 7) {
    if ((theta >= lim_tet[j-1]) && (theta < lim_tet[j])) {
      ring = j-1 ;
      out = false ;
    }
  }
}

ir = ring ;
iz = (int)z ;

if ((theta >= limtet[1]) && (theta < limtet[2])) det = -100 ;
if ((theta >= limtet[3]) && (theta < limtet[4])) det = -100 ;
if ((theta >= limtet[5]) && (theta < limtet[6])) det = -100 ;
if ((theta >= limtet[7]) && (theta < limtet[8])) det = -100 ;
if ((theta >= limtet[9]) && (theta < limtet[10])) det = -100 ;
if ((theta >= limtet[11]) && (theta < limtet[12])) det = -100 ;
if ((theta >= limtet[13]) && (theta < limtet[14])) det = -100 ;
if ((theta >= limtet[15]) && (theta < limtet[16])) det = -100 ;

for (j=1; j<=18; j++) {
  phimin = 20.0*j - 10.0 - 1.0 ;
  phimax = 20.0*j - 10.0 + 1.0 ;
  if ((phi >= phimin) && (phi < phimax)) det = -100 ;
}

if ((int)det == -100) {
  ring = 0 ;
  det = 0 ;
  a = 0 ;
  return ;
}

if ((ring >= 1) && (ring <= 9)) theta = thepp[(int)ring-1] ;
else {
  ring = 0 ;
  det = 0 ;
  a = 0 ;
  return ;
}
```

```
out = true ;
for (j=0; (j<=17)&&(out); j++) {
  phimin = 20.0*j - 10.0 ;
  phimax = 20.0*(j+1) - 10.0 ;
  if (phi > 350) phi -= 360 ;
  if ((phi >= phimin) && (phi < phimax))      {
    det = j+1+(ring-1)*18 ;
    out = false ;
  }
}

ip = (int)det - (18*(ring-1)) ;
phi = ((double)ip-1)*20.0 ;

ic = baddet_e900[(int)det-1] ;

if (ic == -1) a = 0 ;
if ((ic == -2) && (z >= 3)) a = 0 ;

if ((int)z == 1) {
  if ((ic==-2) && (e <= Esimax[(int)z-1])) a = 0 ;
  if ((ic==-3) && (e > Esimax[(int)z-1])) a = 0 ;
}

if ((int)z == 2) {
  if ((ic==-2) && (e <= Esimax[(int)z-1])) a = 0 ;
  if ((ic==-3) && (e > Esimax[(int)z-1])) a = 0 ;
}

if ((int)z == 3)
  if ((ic==-3) && (e > Esimax[(int)z-1])) a= 0 ;

if ((int)z==0) a = 0 ;

if (z > 16) a = 0 ;

if ((int)z==1) {
  if (((int)a==2) && (e <= 10.6)) a = 1 ;
  if (((int)a==3) && (e <= 12.5)) a = 1 ;
  if ((int)a==4) a = 0 ;
}

if ((int)z==2) if (((int)a==3) && (e<=28)) a=4 ;

if ((int)a != 0)
  if (z <= 16) {
    if ((e/a) <= e_th[(int)ir-1][(int)iz-1]) a = 0 ;
  }

if ((int)a != 0)
  if (z >= 4) if ((e/a) > 12) a = 0 ;


if (((int)z==1) || ((int)z==2) || ((int)z==3)) {
    if (e <= Esimax[(int)z-1]) cod=10 ;
    if (e >= Esimax[(int)z-1]) cod=20 ;
  }

  if (((int)z==4) || ((int)z==5)) {
    if (e <= Esimax[(int)z-1]) cod=10 ;
    if (e >= Esimax[(int)z-1]) a=0 ;
  }

  return ;
}
```

A.4 varfdmparams.cpp

This section contains the C++ source code of the routine that varies the two critical exponents σ and τ for a given p_c to evaluates the quality of the "FDM-scaling" behavior.

```
#include <iostream.h>
#include <fstream.h>
#include <cmath>
#include "isisevent.h"
#include "isisreadbuffer.h"
#include "gamma.h"

#define min(a,b)        (((a) < (b)) ? (a) : (b))
#define max(a,b)        (((a) > (b)) ? (a) : (b))

#define PI      3.141592654
#define Euler   0.577

ifstream datafile ;
ifstream ratiofile ;

isisEvent events[eventbuffersize], ev ;

int i, j, k, x, evcount, evts ;

double  scaled_low = -0.3 ;
double  scaled_high = 0.7 ;
double  scaled_range = scaled_high-scaled_low ;
#define Alow            3
#define Ahigh           10
#define Arange          Ahigh-Alow+1
#define bins            50
double sigma = 0.65 ;
double tau = 2.1 ;
double xl[bins*Arange], yl[bins*Arange], yerr[bins*Arange] ;
int points ;

#define runs  500000

double B = 6.6 ;

double Tc = 9.5 ;
double pc = 0.65 ;
double q0 = 1.0/7.730171158 ;

double A1 = 65 ;

double scaled_dist[Arange][bins+2] ;
double scaled_err[Arange][bins+2] ;
double scaling[Arange][bins+2] ;
double chi2 ;

double scaled_start = -0.1 ;
double scaled_end = 0.4 ;

double sigma_low = 0.2  ;
double sigma_high = 0.8 ;
double tau_low = 2.01 ;
double tau_high = 2.61 ;
int    sigma_steps = 61 ;
int    tau_steps = 61 ;
int    s, t ;

double ratios[11] ;

int    startbuffer = 1 ;
int    endbuffer = 409 ;

double minchi2=1E15, minsigma, mintau ;

void init_stuff ()
{
  int i, j ;

  datafile.seekg((startbuffer-1)*buffersize_b, ios::beg) ;

  evts = 0 ;

  for (i=0; i<Arange; i++)
    for (j=0; j<bins; j++) {
      scaled_dist[i][j] = 0 ;
      scaling[i][j] = 0 ;
    }
}

void process_event (isisEvent ev)
{
  double scp, T, pbreak ;
  T = sqrt(ev.ExA*13.0) ;
  //  pbreak = 1.0 - 2.0*Gamma(1.5, 0, B/T)/sqrt(PI) ;

  for (i=Alow; i<=Ahigh; i++)  {
    scp = pow((double)i, sigma)*(Tc-T)/T ;
    if ((scp > scaled_low) && (scp < scaled_high)) {
      scp = (int)((double)(scp-scaled_low)*(double)bins/(double)scaled_range) ;
      scaling[i-Alow][(int)scp]++ ;
    }
  }

  for (i=0; i<ev.mult; i++) {
    if ((ev.Ztherm[i] >= Alow) && (ev.Ztherm[i] <= Ahigh)) {
      scp = pow((double)ev.Ztherm[i], sigma)*(Tc-T)/T ;
      if ((scp > scaled_low) && (scp < scaled_high)) {
        scp = (int)((double)(scp-scaled_low)*(double)bins/(double)scaled_range) ;
        scaled_dist[(int)ev.Ztherm[i]-Alow][(int)scp]++ ;
      }
    }
  }
}

inline double sqr (double x)
{       return x*x ; }

void chi_square ()
{
  int A, i, lowbin, highbin ;
  double S, Sxx, Sxy, Sx, Sy ;
  double b, temp ;

  lowbin = (int)((scaled_start-scaled_low)*bins/scaled_range) ;
  highbin = (int)((scaled_end-scaled_low)*bins/scaled_range) ;

  points = 0 ;

  for (A=Alow; A<=6; A++) {

    for (i=lowbin; i<=highbin; i++) {
      if (scaled_dist[A-Alow][i] > 0) {
        temp = scaled_dist[A-Alow][i] ;
        xl[points] = (double)scaled_low+(i*scaled_range)/(double)bins ;
        yl[points] =
         log10(temp*ratios[A]/(A1*max(scaling[A-Alow][i],1)*q0*pow(A,-tau))) ;
        yerr[points] = yl[points]
         - log10((temp-sqrt(temp))*ratios[A]/
                 (A1*max(scaling[A-Alow][i], 1)*q0*pow(A, -tau))) ;
        points++ ;
      }
    }
  }

  S = 0 ;
  S = 0 ;
  Sx = 0 ;
  Sy = 0 ;
  Sxx = 0 ;
  Sxy = 0 ;
  for (i=0; i<points; i++) {
    S += 1/sqr(yerr[i]) ;
    Sx += xl[i]/sqr(yerr[i]) ;
    Sy += yl[i]/sqr(yerr[i]) ;
    Sxx += sqr(xl[i])/sqr(yerr[i]) ;
    Sxy += xl[i]*yl[i]/sqr(yerr[i]) ;
  }

  b = (S*Sxy - Sx*Sy)/(S*Sxx-sqr(Sx)) ;

  chi2 = 0 ;
  for (i=0; i<points; i++) chi2 += sqr((yl[i]-b*xl[i])/yerr[i]) ;
}

void determine_q0 ()
{
  q0 = 1.0/(1.0/((tau-1)-1)+Euler) ;
}

int main (int argc, char *arg[])
{
  int i ;

  cin >> Tc ;

  ratiofile.open("ratiosT7_f.dat") ;
  i = 0 ;
  while (i < 10) {
    ratiofile >> i ;
    ratiofile >> ratios[i] ;
   } ;
  ratiofile.close() ;


  datafile.open("pAu102.bin") ;

  for (s=0; s<sigma_steps; s++) {
    for (t=0; t<tau_steps; t++) {

      sigma = sigma_low + (sigma_high-sigma_low)*s/(sigma_steps-1) ;
      tau = tau_low + (tau_high-tau_low)*t/(tau_steps-1) ;

      determine_q0() ;
      init_stuff() ;


      for (i=0; i<(endbuffer-startbuffer)+1; i++) {
        evcount = isisreadbuffer(datafile, &(events[0])) ;
        for (j=0; j<evcount; j++) {
          ev = events[j] ;
          evts++ ;
          if (evts == runs) break ;
          process_event(ev) ;
        }

        if (evts == runs) break ;

      }

      chi_square() ;
      cout << sigma << "\t" << tau << "\t" << chi2 << endl ;
      if (chi2 < minchi2) {
        minchi2 = chi2 ;
        minsigma = sigma ;
        mintau = tau ;
      }
    }
    cout << endl ;
  }
  cout <<"#\t"<<Tc<<"\t"<<minsigma<<"\t"<<mintau<<"\t"<<minchi2 ;
  cout << endl ;
  datafile.close() ;

}
```

BIBLIOGRAPHY

Bibliography

[1] J. Adler, Y. Aharony, A. B. Harris, and L. Klein. *J. Stat. Phys.*, 58:511, 1990.

[2] P. Agrawal, S. Redner, P. J. Reynolds, and H. E. Stanley. *J. Phys. C*, 12, 1979.

[3] W. Bauer. *Zerfallskanäle hochangeregter Schwerionen*. Ph.D. thesis, Justus-Liebig-Universität Gießen, 1987.

[4] W. Bauer. Extraction of signals of a phase transition from nuclear multifragmentation. *Phys. Rev. C*, 38:1297–1303, 1988.

[5] W. Bauer and A. Botvina. Pre-equilibrium emission and critical exponent analysis. *Phys. Rev. C*, 52:1760–1763, 1995.

[6] W. Bauer, U. Post, D. R. Dean, and U. Mosel. *Phys. Lett.*, page 53, 1985.

[7] W. Bauer, U. Post, D. R. Dean, and U. Mosel. The nuclear lattice model of proton-induced multi-fragmentation reactions. *Nuclear Physics*, A452:699–722, 1986.

[8] S. R. Broadbent and J. M. Hammersley. *Percolation processes in crystals and mazes.*, volume 53. Proceedings of the Cambridge Philosophical Society, 1957.

[9] A. Coniglio, H. E. Stanley, and W. Klein. *Phys. Rev. Lett*, 42:518, 1979.

[10] J. B. Elliot and et al. *Phys. Lett. B*, 418:35, 1998.

[11] J. B. Elliot, L. G. Moretto, L. Phair, and G. J. Wozniak. The liquid to vapor phase transition in excited nuclei. ArXiv:nucl-ex/0104013 12 Apr 2001.

[12] J. B. Elliot, L. G. Moretto, L. Phair, G. J. Wozniak, S. Albergo, F. Bieser, F. P. Brady, Z. Caccia, and et al. Nuclear Multifragmentation, Percolation and the Fisher Droplet Model: Common Features of Reducibility and Thermal Scaling. *Phys. Rev. Lett.*, 85:1194, 2000.

[13] H. E. Stanley et al. *Physica A*, 266:5, 1999.

[14] J. A. Hauger et al. Dynamics of the multifragmentation of 1A GeV Gold on Carbon. *Phys. Rev. Lett.*, 77:235, 1996.

[15] J. B. Elliot et al. *Phys. Lett. B*, 381:35, 1996.

[16] K. Kwiatkowski et al. *Nucl. Instrum. Meth. Phys. Res. Sec.A*, 360:571, 1995.

[17] M. L. Gilkes et al. *Phys. Rev. Lett*, 73:1590, 1994.

[18] P. Bonche et al. *Nucl. Phys. A*, 436:265, 1985.

[19] D. J. Fields, C. K. Gelbke, W. G. Lynch, and J. Pochodzalla. Nuclear Fragmentation and Sequential Decay. *Phys. Lett. B*, 187(3,4):257–260, 1987.

[20] J. E. Finch, S. Agarwal, A. Bujak, J. Chuang, L. J. Gutay, A. S. Hirsch, R. W. Minich, N. T. Porile, R. P. Scharenberg, B. C. Stringfellow, and F. Turkot. *Phys. Rev. Lett.*, 49:1321, 1982.

[21] M. E. Fisher. The Theory of Condensation and the Critical Point. *Physics*, 3(5):255–283, 1967.

[22] P. J. Flory. *J. Am. Chem. Soc.*, 63:3083, 1941.

[23] J. Frenkel. *Kinetic Theory of Liquids*. Oxford University Press, 1946.

[24] R. B. Griffiths. *Phys. Rev. Lett.*, 24:1949, 1970.

[25] G. Grimmett. *Percolation*. Springer Verlag, New York, 1989.

[26] J. W. Halley and W. K. Holcomb. *Phys. Rev. Lett.*, 40:1670, 1978.

[27] J. M. Hammersley. Percolation processes. lower bounds for critical probability. *Annals of the Mathematical Statistics*, 28:790–795, 1957.

[28] J. M. Hammersley. *Bornes supérieueres de la probabilité critique dans un processus de filtration*. Le Caclul des Probabilité et ses Applications, CRNS, 1959.

[29] A. S. Hirsch, A. Bujak, J. E. Finn, L. J. Gutay, R. W. Minich, N. T. Porile, R. P. Scharenberg, B. C. Stringfellow, and F. Turkot. *Phys. Rev. C*, 29:508, 1984.

[30] L. Landau and E. M. Lifshitz. *Lehrbuch der theoretischen Physik*, volume 5. Akademie-Verlag, Berlin, 1979.

[31] B. J. Last and D. J. Thouless. *Phys. Rev. Lett.*, 27:1719, 1971.

[32] T. Lefort, L. Beaulieu, K. Kwiatkowski, W. c. Hsi, and V. E. Viola. Thermal excitation-energy deposition in 5-15 GeV/c hadron-induced reactions with ^{197}Au. *Phys. Rev. C*, 63, 2001.

[33] T. Li, W. Bauer, D. Craig, E. Gualtieri, S. Hannuschke, R. Pak, A. M. Vander Molen, G. D. Westfall, J. S. Winfield, J. Yee, and S. J. Yennello. Mass dependence of critical behavior in nucleus-nucleus collisions. *Phys. Rev. C*, 49:1630–1634, 1994.

[34] A. Margolina, Z. Djordjevic, H. E. Stanley, and D. Stauffer. *Phys. Rev. B*, 28:1652, 1983.

[35] J. E. Mayer and M. G. Mayer. *Statistical Mechanics*. John Wiley, 1940.

[36] H. Nakanishi and H. E. Stanley. Scaling studies of percolation phenomena in systems of dimensionality two to seven: Cluster numbers. *Phys. Rev. B*, 22:2466, 1980.

[37] Wolfgang Nolting. *Grundkurs Theoretische Physik - 6 Statistische Physik*. Vieweg, 1998.

[38] J. M. Normand, H. J. Herrmann, and M. Hajjar. *J. Stat. Phys.*, 51:441, 1988.

[39] H. Ottavi, J. Clerc, and et al. C. Giraud. *J. Phys. C*, 11:1311, 1978.

[40] S. Pratt. EASY 1.0-beta. http://www.nscl.msu.edu/ pratt/freecodes/ easy/home.html, April 2001.

[41] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P Flannery. *Numerical Recipes in FORTRAN*. Cambridge University Press, 2nd edition edition, 1992.

[42] D. Stauffer. *Phys. Rep.*, 54:3, 1979.

[43] W. H. Stockmayer. *J. Chem. Phys.*, 11:45, 1943.

[44] G. D. Westfall, R. G. Sexto, A. M. Poskanzer, A. M. Zebelman, G. W. Butler, and E. K. Hyde. *Phys. Rev. C*, 17:1368, 1978.

[45] H. J. Wintle and P. H. Puhach. *J. Statist. Phys.*, 18:557, 1978.

[46] J. G. Zabolizky, D. J. Bergman, and D. Stauffer. *J. Stat. Phys.*, 44:211, 1986.