

Universidade de São Paulo
Instituto de Física

O Comportamento das Flutuações dos Campos em Modelos de Campos Escalares

Francisco Dellatorre Borges

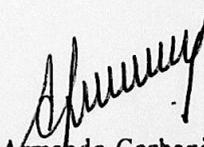
Banca Examinadora

Prof. Dr. Jorge L. deLyra (IFUSP) - Orientador
Prof. Carlos Eugenio I. Carneiro (IFUSP)
Prof. Nelson Augusto Alves (FFCLRP-USP)



Dissertação apresentada ao
IFUSP para a obtenção do
grau de mestre em ciências.

Fevereiro de 2002


Prof. Armando Corbani Ferraz
Presidente da Comissão de Pós Graduação

INSTITUTO DE FÍSICA

Serviço de Biblioteca e Informação

Tombo: _____

3637

ex. 2

Universidade de São Paulo
Instituto de Física

O Comportamento das Flutuações dos Campos em Modelos de Campos Escalares

Francisco Dellatorre Borges

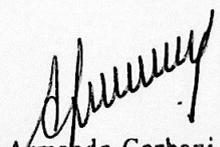
Banca Examinadora

Prof. Dr. Jorge L. deLyra (IFUSP) - Orientador
Prof. Carlos Eugenio I. Carneiro (IFUSP)
Prof. Nelson Augusto Alves (FFCLRP-USP)



Dissertação apresentada ao
IFUSP para a obtenção do
grau de mestre em ciências.

Fevereiro de 2002



Prof. Armando Corbani Ferraz
Presidente da Comissão de Pós Graduação

INSTITUTO DE FÍSICA

Serviço de Biblioteca e Informação

Tombo: _____

3637

ex. 2

FICHA CATALOGRÁFICA

Preparada pelo Serviço de Biblioteca e Informação
do Instituto de Física da Universidade de São Paulo

Borges, Francisco Dellatorre

O Comportamento das Flutuações dos Campos
em Modelos de Campos Escalares.
São Paulo, 2002.

Dissertação (Mestrado) - Universidade de São Paulo.
Instituto de Física - Departamento de Física Matemática

Orientador: Prof. Dr. Jorge L. deLyra
Área de Concentração: Física das Partículas
Elementares e Campos.

Unitermos: 1. Modelos Escalares
2. Modelo Sigma Não-Linear;
3. Modelo Gaussiano;
4. Modelo Lambda-Phi-4 ($\lambda\varphi^4$).

USP/IF/SBI-011/2002

Resumo

O objetivo deste trabalho é explorar uma possível relação de constância na largura local dos campos nos modelos Sigma Não-Linear, $\lambda\varphi^4$ e Gaussiano. Descobrimos que esta relação é realizada apenas de forma aproximada, não de forma exata.

Realizamos medidas da largura da distribuição local dos campos nos modelos $\lambda\varphi^4$ e Sigma Não-Linear tentando medir a variação desta no limite do contínuo ao longo da curva crítica desde $\lambda = 0$ (modelo Gaussiano), até $\lambda \rightarrow \infty$ quando alcança-se o modelo Sigma.

Apresentamos uma breve introdução à Teoria Quântica de Campos na Rede, demonstrando a conexão entre os modelos $\lambda\varphi^4$ e Sigma Não-Linear e cálculos de campo médio sugerindo constância no valor da largura dos campos nestes modelos.

Descrevemos o método de Monte Carlo e os algoritmos necessários para implementá-lo neste trabalho. Em seguida são apresentados os métodos utilizados para aproximação da linha crítica e os resultados correspondentes.

Sunabrio

Abstract

The goal of this work is to explore a possible relation between the critical points of the Non-Linear Sigma model and of the $\lambda\varphi^4$ model. We found that this relation is realized only in an approximate way.

We took measures of the local field width in both models, trying to measure the variation of it in the continuum limit through the critical curve since $\lambda = 0$, the Gaussian model, until $\lambda \rightarrow \infty$ when the Sigma model is reached.

We present a brief introduction to the Quantum Field Theory in the Lattice, showing the connection between the $\lambda\varphi^4$ model and the Non-Linear Sigma model and mean field results that indicate an equal value for the local field width in both models.

The Monte Carlo method and the algorithms used in it's implementation will also be presented.

Sumário

1	Teoria Quântica de Campos na Rede	2
1.1	Espaço e Tempo Discretizados na Rede	2
1.1.1	A Rede Euclidiana	2
1.1.2	Limite do Contínuo	5
1.1.3	O Campo Escalar Livre	5
1.2	Modelo $\lambda\varphi^4$	7
1.3	Modelo Sigma	8
2	O Método de Campo Médio	12
2.1	O Método de Campo Médio	12
2.2	Resultados no Modelo Sigma	12
2.3	Resultados no Modelo $\lambda\varphi^4$	14
3	O Método de Monte Carlo	16
3.1	Sorteio de Configurações	17
3.1.1	Seleção Aleatória	18
3.2	Monte Carlo	18
3.2.1	Freiamento Crítico	19
3.2.2	Metropolis	20
3.2.3	Wolff	21
3.2.4	Calibrando os Algoritmos	22
3.3	Back Rotation	23
3.4	Auto-Correlações entre Configurações	24
4	A Proposta de Trabalho	25
4.1	Mínimo de α_R	25
4.2	Limite com α_R Constante	26
5	Resultados	27
5.1	Mínimo de α_R	27
5.1.1	Simulações para Obtenção de α e λ	27
5.1.2	Resultados Obtidos	28
5.2	Simulações Realizadas a α_R Constante	29
5.3	Modelo Sigma Não-Linear	30

	1
5.3.1	Simulações Realizadas 32
5.3.2	Resultados Obtidos 32
6	Conclusões 35
A	Tópicos em Computação Científica 36
A.1	Linguagens de Script e Computação Científica 36
A.1.1	Linguagens de Script 36
A.1.2	Aplicações na Área Científica 37
A.2	Paralelização: PVM/MPI 38
A.2.1	O Modelo de Paralelização Implementado 39
A.3	Python 39

Capítulo 1

Teoria Quântica de Campos na Rede

Descreveremos de maneira sucinta uma formulação não-perturbativa, na rede (*lattice*), da Teoria Quântica de Campos.

1.1 Espaço e Tempo Discretizados na Rede

Usualmente a Teoria Quântica de Campos (TQC) é discutida dentro do âmbito da integração funcional. No entanto, ainda não se encontrou uma formulação rigorosa para as integrais funcionais diretamente no contínuo.

Dentro deste contexto, a discretização do espaço-tempo contínuo numa rede euclidiana surge como uma formulação alternativa que nos permite realizar estudos de modelos da Teoria Quântica de Campos, pois nos provê de uma formulação adequada (rigorosa) para a integral funcional e um modelo apropriado para simulações computacionais.

1.1.1 A Rede Euclidiana

Nossa rede euclidiana, Λ , que descreverá o espaço-tempo, consiste de um número determinado de sítios com:

1. uma certa relação de vizinhança entre eles;
2. funções com valores reais associadas a cada sítio.

Cada sítio está ligado a $2d$ vizinhos, onde d é o número de dimensões da rede. Todos os sítios estão a uma mesma distância, a , de seus vizinhos. Assumimos que todas as redes têm o mesmo número N de sítios em cada uma das d dimensões o que implica em total de N^d sítios em toda a rede. Cada dimensão linear da rede terá associado a ela um comprimento L .

Neste trabalho as redes utilizadas serão tri-dimensionais e quadri-dimensionais e terão condições de contorno periódicas. Um problema associado a redes finitas com condições de contorno periódicas é que nestas não existe comportamento crítico [2], isto é, não há transições de fase em sistemas com um número finito de graus de liberdade.

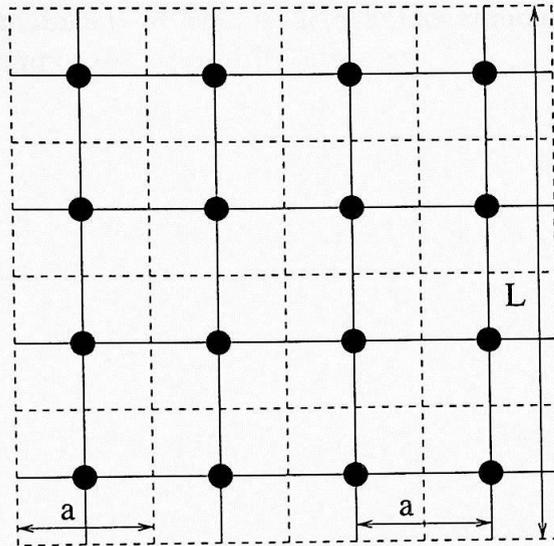


Figura 1.1: Os elementos básicos de uma rede bidimensional típica.

As variáveis elementares de nossas redes são campos, funções com valores reais que definimos em cada sítio da rede. Um campo escalar $\vec{\varphi}(\vec{n})$ associará, a cada sítio \vec{n} , um certo vetor $\vec{\varphi} = (\varphi_1, \varphi_2, \dots, \varphi_N)$ de valores reais, onde N é a ordem do grupo de simetria e portanto a dimensão do espaço interno dos $\vec{\varphi}$. Note que $\vec{\varphi}$ não representa um vetor do espaço-tempo, tratando-se apenas de uma notação para as componentes do vetor no espaço interno dos campos.

Precisamos então definir uma versão na rede do funcional de ação, que nos será dada por uma soma da densidade lagrangiana sobre todos os sítios da rede

$$S_N[\vec{\varphi}] = \sum_{\vec{n} \in \Lambda} a^d L_N(\vec{\varphi}).$$

Esta ação definida na rede determina por completo a teoria quântica através de integrais funcionais

$$Z = \int [d\varphi] e^{-S_N[\vec{\varphi}]},$$

onde usamos seguinte a notação para a medida da integral

$$[d\varphi] = \prod_{\forall i \in N, \vec{n}} d\varphi_i(\vec{n}).$$

O valor esperado de um observável $O[\vec{\varphi}]$ na rede será então calculado como

$$\langle O[\vec{\varphi}] \rangle_N = \frac{\int [d\varphi] O[\vec{\varphi}] e^{-S_N[\vec{\varphi}]} }{\int [d\varphi] e^{-S_N[\vec{\varphi}]} }.$$

Os observáveis normalmente utilizados são definidos como produtos de um número finito de fatores dos campos em sítios diferentes

$$O_{i,\dots,j}[\vec{\varphi}] = \varphi_i(\vec{n}_1) \dots \varphi_j(\vec{n}_m),$$

e seus valores esperados são chamados de *funções de m pontos* ou *funções de correlação*, e serão denotados por

$$\langle \varphi_i(\vec{n}_1) \dots \varphi_j(\vec{n}_m) \rangle.$$

Dois observáveis de grande interesse no nosso trabalho são $O[\vec{\varphi}] = \vec{\varphi}(\vec{n}_1)$, a função de um ponto, e $O[\vec{\varphi}] = \vec{\varphi}(\vec{n}_1)\vec{\varphi}(\vec{n}_2)$, a função de dois pontos.

A função de um ponto, de valor esperado $\langle \vec{\varphi} \rangle$, será usada como parâmetro de ordem de nosso modelo. Esta é calculada através do valor médio dos campos da rede,

$$\vec{\bar{\varphi}} = \frac{1}{N^d} \sum_{\vec{n}} \vec{\varphi}(\vec{n}),$$

equivalente à magnetização dos modelos de Ising. No entanto $\langle \vec{\varphi} \rangle \neq \vec{\bar{\varphi}}$, pois $\vec{\bar{\varphi}}$ é a média dos campos de uma particular configuração da rede. Para que se tenha o valor de $\langle \vec{\varphi} \rangle$ é preciso retirar as flutuações estatísticas de $\vec{\varphi}$, tomando sua média sobre várias configurações diferentes¹

$$\langle \vec{\varphi} \rangle = \langle \vec{\bar{\varphi}} \rangle.$$

Dizemos que o módulo da média de $\vec{\varphi}$, $|\langle \vec{\varphi} \rangle|$, é o parâmetro de ordem do sistema, pois seu comportamento nos modelos $\lambda\varphi^4$ e Sigma Não-Linear (que serão apresentados adiante) demarca as duas fases que estes sistemas apresentam. A fase *desordenada*, caracterizada pelo valor $|\langle \vec{\varphi} \rangle| = 0$ e a fase *ordenada* caracterizada por $|\langle \vec{\varphi} \rangle| \neq 0$.

Na fase desordenada, os campos $\vec{\varphi}$ apontam em direções aleatórias. As chances de dois sítios próximos apontarem numa mesma direção ou em direções contrárias são praticamente idênticas. Dizemos que esta fase é descorrelacionada, que tem o *comprimento de correlação curto*. Na fase ordenada, os sítios estarão todos alinhados uns com os outros, daí $|\langle \vec{\varphi} \rangle|$ ser diferente de zero. Dizemos que na fase ordenada os campos estão correlacionados ou que têm *comprimento de correlação longo*.

¹As técnicas necessárias para fazer isso serão expostas no capítulo 3

A função de dois pontos, $\langle \vec{\varphi}(\vec{n}_1) \vec{\varphi}(\vec{n}_2) \rangle$, também chamada de *propagador* da teoria, é a função mais simples que nos permite medir correlações entre os valores dos campos ao longo da rede. Para que possamos ter uma escala de distância na rede é preciso que $\langle \vec{\varphi}(\vec{n}_1) \vec{\varphi}(\vec{n}_2) \rangle$ decaia com a distância (discreta) ao longo da rede e, para que esta escala seja finita, será preciso que o decaimento seja exponencial.

Este decaimento irá definir uma escala de distâncias intrínseca ao modelo que é estabelecida pela distância máxima na qual os valores dos campos de dois sítios ainda estão significativamente correlacionados. Chamaremos esta distância de *comprimento de correlação*, representado por ξ quando este comprimento for adimensional.

1.1.2 Limite do Contínuo

Definimos a TQC contínua como o limite da TQC na rede quando a , o espaçamento entre sítios, vai a zero e N , o número de sítios em cada dimensão linear, vai a infinito. Os valores dos observáveis na TQC no limite do contínuo são obtidos através de

$$\langle O[\vec{\varphi}] \rangle = \lim_{\substack{N \rightarrow \infty \\ a \rightarrow 0}} \langle O[\vec{\varphi}] \rangle_{a,N}.$$

Para que a teoria quântica esteja bem definida é preciso que este limite exista e seja finito para todos observáveis de interesse.

Algumas simetrias como as invariâncias por translação ou rotação do espaço-tempo têm de ser substituídas pelos seus subgrupos da rede e sua recuperação no limite do contínuo deve ser verificada.

Uma das principais condições sobre este tipo de limite é que o comprimento de correlação dimensional seja não nulo, pois caso contrário a teoria ficaria desprovida de significado físico, pois isso corresponderia a partículas de massa física m infinita. Sendo ξ o comprimento de correlação adimensional, teremos $\chi = a\xi = 1/m$ como comprimento de correlação dimensional. A imposição que aplicaremos ao limite do contínuo é que χ seja não nulo, de forma que $\chi = a\xi = \xi L/N$ seja não-nulo, o que implica que $\xi \rightarrow \infty$ no limite.

Desta forma podemos desde já notar a importância dos pontos críticos dos modelos, uma vez que apenas nestes pontos o comprimento de correlação adimensional ξ diverge e a função de correlação decai polinomialmente, permitindo que χ seja não-nulo no limite do contínuo, quando $a \rightarrow 0$.

1.1.3 O Campo Escalar Livre

Podemos então definir a ação para o *modelo Gaussiano* ou de *campo escalar livre*

$$S_0[\vec{\varphi}] = \frac{1}{2} \sum_{\ell \in \Lambda} (\Delta_\ell \vec{\varphi})^2 + \frac{\alpha_0}{2} \sum_{\vec{n}} \varphi^2(\vec{n}), \quad (1.1)$$

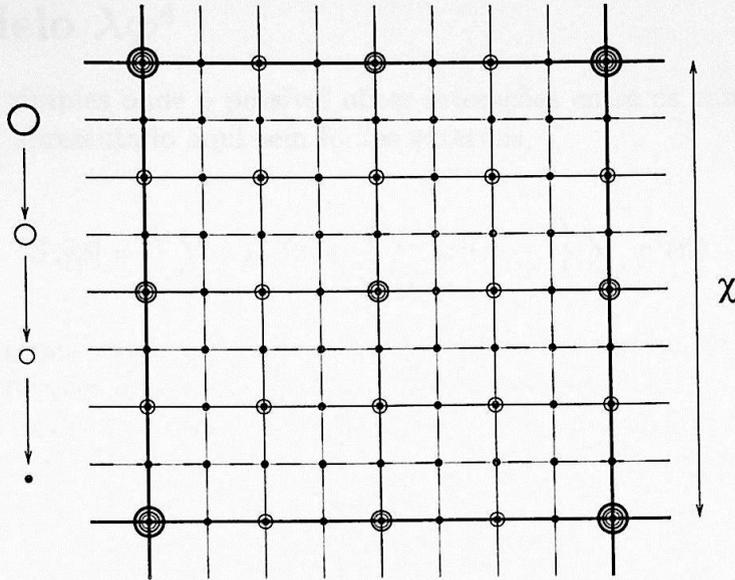


Figura 1.2: Limite do contínuo: espaçamento entre os sítios diminui e número total de sítios aumenta.

onde $\ell \in \Lambda$ são *links* ligando sítios vizinhos e $\Delta_\ell \vec{\varphi}$ a diferença do valor dos campos de sítios conectados por ℓ . Exigiremos que $\alpha_0 \geq 0$ para que a ação tenha um limite inferior. Como todas as integrais envolvidas no cálculo de observáveis do modelo do campo escalar livre são gaussianas, elas são todas solúveis. O modelo Gaussiano é o único que conseguimos resolver analiticamente de forma completa. Apesar de não apresentar transição de fase, a única fase existente apresenta ordem de longo alcance em um ponto crítico, o ponto Gaussiano ($\alpha = 0$). Sendo solúvel, este modelo serve como ponto de partida para a análise de modelos mais complexos. Estudados através de métodos perturbativos ou de Monte Carlo.

Um dos principais resultados que pode ser obtido é o *propagador da teoria livre* ou *função de dois-pontos* no espaço de momentos,

$$\langle |\tilde{\varphi}(\vec{k})|^2 \rangle = \frac{1}{N^d [\rho^2(\vec{k}) + \alpha_0]}, \quad \text{onde } \rho^2(\vec{k}) \equiv 4 \sum_{\mu} \text{sen}^2 \left(\frac{\pi k_{\mu}}{N} \right), \quad (1.2)$$

e $\tilde{\varphi}^2$ é a transformada de Fourier discreta de $\vec{\varphi}$ dada por

$$\tilde{\varphi}(\vec{k}) = \frac{1}{N^d} \sum_{\vec{n} \in \Lambda} e^{i \frac{2\pi}{N} \vec{k} \cdot \vec{n}} \vec{\varphi}(\vec{n}).$$

Há duas outras ações de interesse para o nosso trabalho, as do modelo $\lambda\varphi^4$ e Sigma Não-Linear, ambas possuem termos não-lineares em suas ações, incluídos para que possamos obter interações entre campos e, possivelmente, entre partículas.

²Por consistência deveríamos escrever $\vec{\tilde{\varphi}}$, mas para não sobrecarregar a notação a seta vetorial será omitida.

1.2 Modelo $\lambda\varphi^4$

O modelo mais simples onde é possível obter interações entre os campos é o *modelo polinomial* $\lambda\varphi^4$, apresentado aqui sem fontes externas,

$$S_\lambda[\vec{\varphi}] = \frac{1}{2} \sum_{\ell} (\Delta_{\ell} \vec{\varphi})^2 + \frac{\alpha}{2} \sum_{\vec{n}} \varphi^2(\vec{n}) + \frac{\lambda}{4} \sum_{\vec{n}} \varphi^4(\vec{n}). \quad (1.3)$$

Neste modelo, iremos exigir que $\lambda > 0$ para que tenhamos um limite inferior da ação, não havendo restrições quanto ao valor de α exceto quando $\lambda = 0$.

Este modelo apresenta uma transição de fase, uma maneira intuitiva e qualitativamente correta de se analisar a maneira pela qual ela ocorre é através da análise separada da parte potencial de sua ação

$$V(\vec{\varphi}) = \frac{\alpha}{2} \varphi^2 + \frac{\lambda}{4} \varphi^4, \quad (1.4)$$

onde $\varphi = \sqrt{\vec{\varphi} \cdot \vec{\varphi}}$. Além da exigência de que $\lambda > 0$, para que existam limites do contínuo não triviais³, α deve ser negativo, pois para $\alpha \geq 0$ o potencial na equação 1.4 será apenas um poço potencial simétrico não sendo possível, portanto, obter a quebra de simetria e nem o comportamento crítico. Para $\alpha < 0$, o modelo pode apresentar uma transição de fase pois o potencial apresentará um poço duplo (figura 1.3). O mínimo do potencial se localiza em $\varphi = \pm \sqrt{-\alpha/\lambda}$ e a profundidade dos poços é $\alpha^2/(4\lambda)$.

A parte cinética, dada pelos termos derivativos

$$S_0[\vec{\varphi}] = \frac{1}{2} \sum_{\ell \in \Lambda} (\Delta_{\ell} \vec{\varphi}) \cdot (\Delta_{\ell} \vec{\varphi}), \quad (1.5)$$

produzirá, na ausência de potencial, uma distribuição gaussiana dos campos centrada no valor de $\langle \vec{\varphi} \rangle$ com largura σ_0 dada por

$$\sigma_0^2 = \langle \varphi_i^2(\vec{n}) \rangle - \langle \vec{\varphi}_i(\vec{n}) \rangle^2 = \langle \varphi_i^2 \rangle - \langle \vec{\varphi}_i \rangle^2, \quad (1.6)$$

uma vez que o modelo tem simetria discreta de translação.

Assim como visto na figura 1.3, podemos entender as condições para quebra de simetria do modelo como uma relação entre a largura do poço de potencial e a largura da distribuição local dos campos 1.6. Se a largura da gaussiana for muito grande quando comparada com a do poço duplo, o sistema não deve sofrer transição de fase. No entanto, conforme α aumenta em relação a λ , tanto a largura dos poços, $\sqrt{-\alpha/\lambda}$, quanto sua profundidade, $\alpha^2/(4\lambda)$, aumentam, contribuindo para que a distribuição

³Limites *triviais* são aqueles que nos levam para o ponto gaussiano.

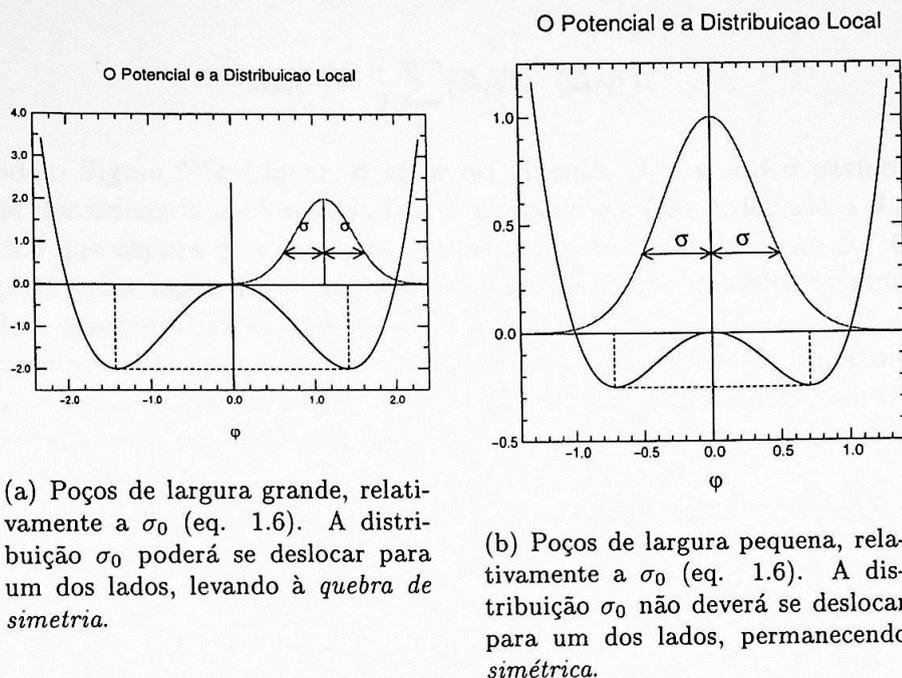


Fig. 1.3: Representação do potencial 1.4 e da parte cinética 1.5 do modelo $\lambda\varphi^4$.

se desloque para algum dos lados. Como os poços são idênticos a direção da quebra de simetria será sempre aleatória (na ausência de fontes externas).

Quando $\lambda \rightarrow 0$ nos aproximamos do modelo Gaussiano e, no caso das fontes externas serem nulas, ao tomarmos $\lambda \rightarrow \infty$ nos aproximamos do modelo Sigma Não-Linear, como será visto na próxima seção.

1.3 Modelo Sigma

No modelo Sigma Não-Linear, ao invés de tentarmos obter acoplamentos não-lineares através da adição de termos quárticos (ou de potências ainda maiores), adicionaremos um vínculo à ação dos campos do modelo Gaussiano. Assim, impondo aos campos o vínculo invariante por $SO(N)$

$$\vec{\varphi} \cdot \vec{\varphi} = \beta, \quad (1.7)$$

onde β é um número real e positivo, obtemos

$$S[\vec{\varphi}] = \frac{1}{2} \sum_{\ell} (\Delta_{\ell} \vec{\varphi}) \cdot (\Delta_{\ell} \vec{\varphi}) + \frac{\alpha_0}{2} \beta N^d.$$

Por simplicidade, redefinimos a ação, retirando os termos que não variam com os campos

$$S_\sigma[\vec{\varphi}] = \frac{1}{2} \sum_\ell (\Delta_\ell \vec{\varphi}) \cdot (\Delta_\ell \vec{\varphi}).$$

No modelo Sigma Não-Linear, o valor do vínculo, β , é o único parâmetro livre. Nele, a fase desordenada está associada a β pequeno e a fase ordenada a β grande. O ponto crítico que separa as duas fases é, normalmente, referido como β_c . O valor de β , pode, para guiar nossa intuição, ser associado ao β dos modelos termodinâmicos, relacionado à temperatura através de $\beta = 1/(k_b T)$.

Este mesmo modelo pode ser obtido a partir do modelo $\lambda\varphi^4$ ao tomarmos o limite $\lambda \rightarrow \infty$ na direção da curva crítica ou de qualquer reta partindo do ponto gaussiano com declividade negativa, estabelecendo uma conexão entre este modelo e os modelos polinomiais $SO(N)$.

Para demonstrar esta relação, começaremos decompondo a distribuição polinomial do $\lambda\varphi^4$

$$\begin{aligned} \exp(-S_\lambda[\vec{\varphi}]) &= \exp\left(\frac{1}{2} \sum_\ell (\Delta_\ell \vec{\varphi})^2 + \frac{\alpha}{2} \sum_s \varphi^2(s) + \frac{\lambda}{4} \sum_s \varphi^4(s)\right) \\ \exp(-S_\lambda[\vec{\varphi}]) &= \exp\left[\sum_s \left(\frac{\alpha}{2} \varphi^2(s) + \frac{\lambda}{4} \varphi^4(s)\right)\right] \exp\left(\frac{1}{2} \sum_\ell (\Delta_\ell \vec{\varphi})^2\right). \end{aligned}$$

Para analisar a parte local da distribuição é conveniente definir uma versão normalizada desta. Definimos então $g(\alpha, \lambda, \varphi)$, a distribuição normalizada do potencial ultra-local $V(\varphi) = \sum_s \left(\frac{\alpha}{2} \varphi^2(s) + \frac{\lambda}{4} \varphi^4(s)\right)$ em cada sítio de Λ como

$$g(\alpha, \lambda, \varphi) = \frac{e^{-V(\varphi)}}{\int_0^\infty [d\varphi] e^{-V(\varphi)}}. \quad (1.8)$$

Os pontos de máximo da distribuição estarão localizados nos pontos de mínimo do potencial ou seja $\varphi = \sqrt{-\alpha/\lambda}$ caso $\varphi \in O(1)$.

No limite $\lambda \rightarrow \infty$, definiremos $\alpha = -\beta\lambda$ onde $\beta \in \mathbb{R}^+$, desta forma os valores de α e λ passam a estar acoplados e o nosso limite estará sendo tomado por uma reta de declividade negativa. A motivação para esta substituição é que, com ela, iremos obter o vínculo que nos levará ao modelo Sigma.

O cálculo da integral de normalização de 1.8 pode ser realizado em termos de Funções Parabólicas Cilíndricas D_ν . De [8] temos

$$\int_0^\infty x^{\nu-1} \exp(-\sigma x^2 - \gamma x) dx = (2\sigma)^{-\frac{\nu}{2}} \Gamma(\nu) \exp\left(\frac{\gamma^2}{8\sigma}\right) D_{-\nu}\left(\frac{\gamma}{\sqrt{2\sigma}}\right). \quad (1.9)$$

Fazendo a transformação de variáveis $x = \varphi^2 \rightarrow dx = 2\varphi d\varphi$ e tomando $\nu = 1/2$, $\sigma = \lambda/4$ e $\gamma = \alpha/2 = -\beta\lambda/2$ obtemos,

$$\int_0^\infty [d\varphi] e^{-V(\varphi)} = \frac{\sqrt{\pi}}{2} \left(\frac{\lambda}{2}\right)^{-\frac{1}{4}} \exp\left(\frac{\beta^2 \lambda}{8}\right) \mathbf{D}_{-\frac{1}{2}}\left(\beta \sqrt{\frac{\lambda}{2}}\right).$$

Como $\lambda \rightarrow \infty$, podemos usar a forma assintótica de \mathbf{D}_ν ,

$$\begin{aligned} \mathbf{D}_\nu(z) \sim & \exp\left(-\frac{z^2}{4}\right) z^\nu \left(1 - \frac{\nu(\nu-1)}{2z^2} + \dots\right) \\ & - \frac{\sqrt{2\pi}}{\Gamma(-\nu)} e^{i\nu\pi} \exp\left(\frac{z^2}{4}\right) z^{-(\nu+1)} \left(1 + \frac{(\nu+1)(\nu+2)}{2z^2} + \dots\right), \end{aligned}$$

de forma que para $\lambda \gg 1$ teremos

$$\int_0^\infty [d\varphi] e^{-V(\varphi)} \approx \sqrt{\frac{\pi}{\beta\lambda}} \exp\left(\frac{\beta^2 \lambda}{4}\right).$$

Substituindo este resultado em 1.8, temos a distribuição do potencial para $\lambda \gg 1$,

$$g(-\beta\lambda, \lambda, \varphi) \approx \sqrt{\frac{\beta\lambda}{\pi}} \exp\left[-\frac{\lambda}{4} (\varphi^2 - \beta)^2\right]. \quad (1.10)$$

Esta distribuição se aproxima muito rapidamente de zero para qualquer φ exceto $\varphi = \sqrt{\beta}$, onde diverge com $\sqrt{\lambda}$.

Assim, para uma função $f(\varphi)$ limitada e contínua temos a relação

$$\int_0^\infty [d\varphi] f(\varphi) g(-\beta\lambda, \lambda, \varphi) = f(\sqrt{\beta}) \int_0^\infty [d\varphi] g(-\beta\lambda, \lambda, \varphi) = f(\sqrt{\beta}).$$

Como a distribuição $g(-\beta\lambda, \lambda, \varphi)$ é normalizada, no limite $\lambda \rightarrow \infty$ ela convergirá para uma função-delta,

$$\lim_{\lambda \rightarrow \infty} g(-\beta\lambda, \lambda, \varphi) = \delta(\varphi - \sqrt{\beta}) = 2\sqrt{\beta} \delta(\varphi^2 - \beta).$$

Neste limite $\lambda \rightarrow \infty$, $\alpha = -\beta\lambda$, o valor esperado de um observável O será

$$\langle O \rangle_N = \frac{\int [d\varphi] \delta(\vec{\varphi} \cdot \vec{\varphi} - \beta) O[\varphi] e^{-S_0[\varphi]}}{\int [d\varphi] \delta(\vec{\varphi} \cdot \vec{\varphi} - \beta) e^{-S_0[\varphi]}}$$

onde a medida da integral funcional é

$$[d\varphi \delta(\vec{\varphi} \cdot \vec{\varphi} - \beta)] = \prod_{i,x} d\varphi_i(x) \delta[\vec{\varphi}(x) \cdot \vec{\varphi}(x) - \beta],$$

de forma que os valores assumidos pelos campos $\vec{\varphi}$ nos sítios estarão submetidos ao vínculo $\vec{\varphi} \cdot \vec{\varphi} = \beta$. Obtemos desta maneira, a partir do modelo $\lambda\varphi^4$, o modelo *Sigma Não-Linear*.

Agora que já definimos os modelos com os quais iremos trabalhar, faremos uma exposição dos resultados da aproximação de campo médio que motivaram os nossos trabalhos.

Capítulo 2

O Método de Campo Médio

2.1 O Método de Campo Médio

A aproximação de campo médio é um método tradicionalmente utilizado em Mecânica Estatística e útil para obtenção de resultados não perturbativos em TQC. Ele pode ser utilizado tanto nos modelos polinomiais $\lambda\varphi^4$ quanto nos modelos Sigma.

A qualidade de seus resultados melhora com o número de dimensões espaço-temporais do modelo. O método funciona bem para $d = 3$, melhora para $d = 4$, enquanto para dimensões 1 ou 2 ele falha completamente.

Este método consiste da substituição das interações entre um particular sítio $\vec{n}_1 \in \Lambda$ com o restante da rede pela interação deste sítio com um campo médio de fundo não sujeito a flutuações estatísticas. O cálculo deste campo médio deve envolver uma soma sobre todos os sítios \vec{n}_2 que interagem com \vec{n}_1 , levando em conta que estes podem estar a diferentes distâncias de \vec{n}_1 e que deve-se tomar uma média temporal desta soma para eliminar flutuações estatísticas.

2.2 Resultados no Modelo Sigma

Apresentaremos nesta seção o cálculo da largura local dos campos no modelo Sigma.

Re-escrevendo a ação do modelo Sigma com o fator β escrito explicitamente,

$$S[\vec{\varphi}] = -\frac{1}{2} \sum_{\ell} (\Delta_{\ell} \vec{\varphi}) \cdot (\Delta_{\ell} \vec{\varphi}) = -\frac{\beta}{2} \sum_{\ell} (\Delta_{\ell} \vec{\phi}) \cdot (\Delta_{\ell} \vec{\phi}),$$

o vínculo nos valores de $\vec{\phi}$ será

$$\vec{\phi} \cdot \vec{\phi} = 1,$$

ou seja, teremos o campo $\vec{\phi}$ variando na esfera unitária S_{N-1} .

Aplicamos a aproximação de campo médio congelando o valor de $\vec{\phi}$, em todos os sítios menos um, no valor $(0, 0, 0, 0 \dots, M)$. Onde M é, a princípio, um valor arbitrário mas podemos impor a ele que adquira o valor da média de $\vec{\phi}$ na rede; isto pode ser obtido facilmente ajustando auto-consistentemente o valor de M pelo valor da magnetização M .

É importante notar que ao impormos um valor sobre os campos de determinados sítios perdemos por completo o vínculo sobre os campos destes.

Como escolhemos, arbitrariamente, a N -ésima direção como a direção de orientação de $\vec{M} = (0, 0, 0, 0 \dots, M)$, calculamos o valor esperado da magnetização $M = \langle \phi_N \rangle$ através de

$$M = \frac{\int_{-\infty}^{+\infty} d\phi_1 \dots d\phi_N \phi_N \delta(\vec{\phi} \cdot \vec{\phi} - 1) e^{2d\beta M \phi_N}}{\int_{-\infty}^{+\infty} d\phi_1 \dots d\phi_N \delta(\vec{\phi} \cdot \vec{\phi} - 1) e^{2d\beta M \phi_N}}. \quad (2.1)$$

Ajustando auto-consistentemente o valor de M para ser igual a M , podemos substituir M por M em eq.2.1

$$M = \frac{\int_{-\infty}^{+\infty} d\phi_1 \dots d\phi_N \phi_N \delta(\vec{\phi} \cdot \vec{\phi} - 1) e^{2d\beta M \phi_N}}{\int_{-\infty}^{+\infty} d\phi_1 \dots d\phi_N \delta(\vec{\phi} \cdot \vec{\phi} - 1) e^{2d\beta M \phi_N}}. \quad (2.2)$$

Substituindo $A = 2d\beta M$ e considerando como componente radial $\phi = \sqrt{\vec{\phi} \cdot \vec{\phi}}$, onde θ_{N-1} é o ângulo polar medido a partir do eixo de ϕ_N temos,

$$M = \frac{\oint d\Omega^{N-1} \int_0^\infty d\phi \phi^N \cos(\theta_{N-1}) \delta(\phi^2 - 1) e^{A\phi \cos(\theta_{N-1})}}{\oint d\Omega^{N-1} \int_0^\infty d\phi \phi^{N-1} \delta(\phi^2 - 1) e^{A\phi \cos(\theta_{N-1})}},$$

onde Ω^{N-1} é o elemento de ângulo sólido de N dimensões.

Realizando a integral radial obtemos

$$M = \frac{\oint d\Omega^{N-1} \cos(\theta_{N-1}) e^{A \cos(\theta_{N-1})}}{\oint d\Omega^{N-1} e^{A \cos(\theta_{N-1})}}.$$

Como sabemos que a transição de fase do modelo é de segunda ordem temos que $\beta \approx \beta_c \rightarrow M \approx 0$, assim expandimos M em torno de 0, mantendo apenas termos de primeira ordem (as potências ímpares de $\cos(\theta_{N-1})$ não contribuem), obtendo

$$\begin{aligned} M &= \frac{\oint d\Omega^{N-1} \cos(\theta_{N-1}) [1 + A \cos(\theta_{N-1})]}{\oint d\Omega^{N-1} [1 + A \cos(\theta_{N-1})]} \Rightarrow \\ M &= A \frac{\oint d\Omega^{N-1} \cos^2(\theta_{N-1})}{\oint d\Omega^{N-1}}. \end{aligned} \quad (2.3)$$

A integral angular em $d\Omega_{N-1}$ pode ser encontrada em [9],

$$\frac{\oint d\Omega^{N-1} \cos^2(\theta_{N-1})}{\oint d\Omega^{N-1}} = \frac{1}{N}. \quad (2.4)$$

Como $A = 2d\beta M$ o fator de M da equação 2.3 se cancelará, isto feito podemos impor a condição $M = 0$ que nos fornece a aproximação de campo médio para β_c do modelo Sigma

$$\beta_c(N, d) = \frac{N}{2d}. \quad (2.5)$$

Devemos observar que β é o valor da largura da distribuição local dos campos $\vec{\varphi}$. Assim, este resultado pode também ser aplicado à largura dos campos do modelo Sigma, pois o vínculo 1.7 implica em

$$\langle \varphi^2 \rangle = \beta \langle \phi^2 \rangle = \beta.$$

Uma vez que no modelo Sigma temos o vínculo $\vec{\varphi} \cdot \vec{\varphi} = \beta$, que a simetria dos campos é recuperada no ponto crítico e que, portanto, neste ponto os campos estão homogeneamente distribuídos no espaço interno, cada uma das N componentes φ_i^2 tem como valor esperado $1/2d$.

2.3 Resultados no Modelo $\lambda\varphi^4$

Estamos interessados em obter o valor esperado de $\langle \varphi_i^2(\vec{n}) \rangle$ na aproximação de campo médio e para isso usaremos uma relação que será obtida no cálculo do valor esperado de M nesta mesma aproximação.

Realizaremos o cálculo da magnetização da componente φ_N dos campos, $M = \langle \varphi_N \rangle$, que pode ser escrita na aproximação de campo médio como

$$M = \frac{\int_{-\infty}^{+\infty} d\varphi_1 \dots d\varphi_N \varphi_N e^{2dM\varphi_N} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}{\int_{-\infty}^{+\infty} d\varphi_1 \dots d\varphi_N e^{2dM\varphi_N} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}.$$

Fazendo uso de coordenadas esféricas no espaço interno, onde $\varphi_N = \varphi \cos(\theta_{N-1})$, temos

$$M = \frac{\oint d\Omega^{N-1} \int_0^\infty d\varphi \varphi^{N+1} \cos(\theta_{N-1}) e^{2dM\varphi_N} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}{\oint d\Omega^{N-1} \int_0^\infty d\varphi \varphi^{N-1} e^{2dM\varphi_N} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}. \quad (2.6)$$

O valor esperado de M pode ser calculado exatamente através de funções de Bessel I_ν , mas, como estamos interessados apenas no comportamento na vizinhança da curva crítica, podemos expandir $e^{2dM\varphi_N}$ em primeira ordem para $M \ll 1$, obtendo

$$M = 2dM \frac{\oint d\Omega^{N-1} \int_0^\infty d\varphi \varphi^{N+1} \cos^2(\theta_{N-1}) e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}{\oint d\Omega^{N-1} \int_0^\infty d\varphi \varphi^{N-1} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}.$$

O fator de M se cancela e, realizando a integral angular com [9], obtemos a relação

$$\frac{N}{2d} = \frac{\int_0^\infty d\varphi \varphi^{N+1} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}{\int_0^\infty d\varphi \varphi^{N-1} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}$$

que já nos fornece de forma implícita uma equação para a curva crítica, pois foi obtida sob a hipótese de que $M \ll 1$, tendo como parâmetros livres apenas α e λ . Ela pode ser escrita na forma

$$\int_0^\infty d\varphi \varphi^{N+1} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]} = \frac{N}{2d} \int_0^\infty d\varphi \varphi^{N-1} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}. \quad (2.7)$$

O cálculo de $\langle \varphi_N^2 \rangle$ é realizado de maneira muito parecida,

$$\begin{aligned} \langle \varphi_N^2 \rangle &= \frac{\int_{-\infty}^{+\infty} d\varphi_1 \dots d\varphi_N \varphi_N^2 e^{2dM\varphi_N} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}{\int_{-\infty}^{+\infty} d\varphi_1 \dots d\varphi_N e^{2dM\varphi_N} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}} \Rightarrow \\ \langle \varphi_N^2 \rangle &= \frac{\oint d\Omega^{N-1} \int_0^\infty d\varphi \varphi^{N+1} \cos^2(\theta_{N-1}) e^{2dM\varphi} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}{\oint d\Omega^{N-1} \int_0^\infty d\varphi \varphi^{N-1} e^{2dM\varphi} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}. \end{aligned}$$

Expandindo novamente M para $M \approx 0$ temos

$$\langle \varphi_N^2 \rangle = N \frac{\int_0^\infty d\varphi \varphi^{N+1} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}{\int_0^\infty d\varphi \varphi^{N-1} e^{-[(d+\alpha/2)\varphi^2 + (\lambda/4)\varphi^4]}}. \quad (2.8)$$

Substituindo a equação 2.7 em 2.8 obtemos

$$\langle \varphi_N^2 \rangle = \frac{1}{2d}, \quad (2.9)$$

um valor constante para a largura dos campos em toda a curva crítica. Deve-se notar que este resultado é válido para qualquer valor de λ , o que o torna válido tanto para o modelo Gaussiano quanto para o modelo $\lambda\varphi^4$.

Estes dois resultados, 2.9 e 2.5, nos indicam que dentro da aproximação de campo médio o valor esperado da largura dos campos é idêntico nos modelos Sigma, Gaussiano e $\lambda\varphi^4$.

Se esta constância fosse comprovada no limite $N \rightarrow \infty$, isto nos permitiria ligar a solução do modelo Gaussiano, 1.2, à do modelo Sigma. Um resultado como esse nos proveria de um formidável meio para determinação da posição do ponto crítico do modelo Sigma.

Capítulo 3

O Método de Monte Carlo

Antes de passarmos à descrição dos resultados obtidos, faremos uma exposição do método de Monte Carlo, que foi utilizado para realizar as integrais funcionais nos cálculos dos observáveis, bem como dos algoritmos com os quais o implementamos.

Do ponto de vista da mecânica estatística, os sistemas estudados serão sempre sistemas em equilíbrio. Devemos observar que a TQC euclidiana é equivalente a mecânica estatística em $d = 4$.

Assim, seja E_C a energia de uma particular configuração da rede e supondo que o sistema esteja em equilíbrio térmico; a probabilidade do sistema se encontrar na configuração C é proporcional a

$$\exp(-\beta E_C),$$

conhecido como *fator de Boltzmann*. Para que tenhamos uma distribuição de probabilidades normalizada, a probabilidade p_C da configuração C é definida como

$$p_C = \frac{1}{Z} e^{-\beta E_C}, \quad (3.1)$$

onde o fator de normalização Z é definido como a soma do fator de Boltzmann sobre todas as possíveis configurações do sistema,

$$Z = \sum_{\forall C \in \Omega} e^{-\beta E_C}. \quad (3.2)$$

Esta quantidade é normalmente chamada de *função de partição do sistema*.

A distribuição de probabilidades definida por 3.1 é denominada de *distribuição de probabilidades de Gibbs* e com ela poderemos calcular o valor médio de um observável O desde que:

- conheça-se Ω , o conjunto de todas as possíveis configurações do sistema;

- possamos calcular o valor da energia E_C e do observável O_C para todas configurações $C \in \Omega$;
- possamos realizar a soma

$$\langle O \rangle = \sum_{\forall C \in \Omega} p_C O_C = \frac{1}{Z} \sum_{\forall C \in \Omega} O_C e^{-\beta E_C}. \quad (3.3)$$

3.1 Sorteio de Configurações

O problema é que, embora o esquema apresentado acima para o cálculo de observáveis na rede seja completamente consistente do ponto de vista conceitual, só temos condições de realizar a somatória 3.3 para um número extremamente pequeno de casos.

Quando não possuímos meios para calcular exatamente 3.3 teremos de nos valer de métodos numéricos para tentar obter seu valor.

Este é o caso quando tentamos calcular o valor de algum observável na ação de alguns dos modelos já descritos. Mesmo usando campos em $O(1)$ (único caso em que espaço de configurações será finito) em redes relativamente pequenas, o tamanho do espaço de configurações é grande demais para que qualquer outra alternativa seja viável.

Por exemplo, no modelo de Ising (ou modelo Sigma Não-Linear com simetria $O(1)$) temos, associado a cada sítio \vec{n} , um campo $\varphi_{\vec{n}} \in \{-1, +1\}$, de maneira que, se formos estudar o modelo numa rede bi-dimensional 4×4 , teremos $2^{16} = 65536$ possíveis configurações. Neste caso, com os atuais recursos computacionais, poderíamos calcular todas as possíveis configurações do sistema sem grandes problemas. Todavia, se estivermos interessados em estudar redes não muito maiores, como 10×10 por exemplo, teríamos $2^{100} \approx 10^{30}$ configurações, um número que está muito, muito além de nossas capacidades atuais.

Quando não existem meios analíticos para integração de uma função há duas alternativas: integração numérica e integração estocástica. A integração estocástica, ou método de Monte Carlo, é competitiva com a integração numérica para integrais de dimensionalidade alta. Para algumas aplicações o método de Monte Carlo é a única escolha possível.

O método de Monte Carlo aplicado às integrações nos modelos interagentes consiste da obtenção de uma sequência de configurações descorrelacionadas na rede Λ de N^d sítios. Assim, estimaremos as propriedades do sistema a partir de uma pequena amostragem do espaço de configurações.

Os valores dos observáveis de interesse são então medidos em cada configuração; ao final do processo tomamos as médias dos valores destes observáveis em todas as configurações da sequência. Assim, se o número de configurações utilizadas para o cálculo da média for m , este processo nos dará para uma quantidade O a estimativa

$$\langle O \rangle_m = \frac{1}{Z_m} \sum_i^m O_{\varphi,i} e^{-S_i[\varphi]} \quad \text{com} \quad Z_m = \sum_i^m e^{-S_i[\varphi]}. \quad (3.4)$$

3.1.1 Seleção Aleatória

Discutamos agora como realizar o passeio no espaço de configurações, ou seja, como realizar a seleção das configurações a serem utilizadas.

A abordagem mais simples e direta seria simplesmente sortear configurações aleatoriamente de forma homogênea no espaço de configurações e estimar o valor dos observáveis de interesse a partir da média desta amostragem.

No entanto, a variância de uma soma de variáveis independentes é a soma das variâncias individuais. Assim as variâncias do numerador e denominador em 3.4, usando sorteio puramente aleatório, serão

$$\frac{1}{m} \text{var} (O e^{-S_i[\varphi]}) \quad \text{e} \quad \frac{1}{m} \text{var} (e^{-S_i[\varphi]}), \quad (3.5)$$

para o numerador e denominador respectivamente. Nestas expressões temos o fator $1/m$, que resulta numa redução do desvio padrão por $1/\sqrt{m}$, o que está longe de ser razoável caso tenhamos uma variância grande no fator $\exp(-S_i[\varphi])$, que é justamente o que acontece.

O problema é que estamos incluindo, com frequência igual, configurações com pesos absolutamente discrepantes, muitas com contribuições absolutamente desprezíveis para o valor final de 3.4. As contribuições de diferentes configurações dentro de uma mesma rede chegam a diferir em até $\exp(2\beta N^2)$ para redes bidimensionais $N \times N$ no modelo de Ising, fazendo com que a variância divirja muito fortemente com o tamanho da rede.

O que precisamos é selecionar configurações que darão contribuições significativas a 3.4 com a maior frequência possível, sem no entanto abandonar a distribuição correta na escolha das configurações.

A melhor estratégia para realizar isso é selecionar configurações de acordo com a distribuição de probabilidades de Gibbs, ou seja independentemente do valor de O_C , pois assim o *ensemble* de configurações obtido pode ser utilizado no cálculo de qualquer observável e não apenas de algum observável em particular.

3.2 Monte Carlo

O que queremos obter é uma sequência de configurações $C_i \in \Omega$ na qual a frequência de ocorrência de cada C_i seja proporcional à probabilidade de Gibbs $p(C_i)$. Para tanto, precisaremos realizar um *processo de Markov*, um método para geração aleatória de novos estados, no qual a escolha do próximo estado da sequência depende apenas

do conhecimento do estado imediatamente anterior e de nada mais. Uma sequência gerada por esse método será chamada de *sequência de Markov*.

As regras que definem um processo de Markov podem ser expressas como um conjunto de probabilidades de transição, para cada par $C, C' \in \Omega$, existe uma probabilidade de transição $P(C \rightarrow C')$ entre esses estados. Estas probabilidades devem ser normalizadas,

$$\sum_{C' \in \Omega} P(C \rightarrow C') = 1.$$

Como queremos ainda que nossa sequência de Markov esteja vinculada a uma distribuição de probabilidades de Gibbs será preciso impor duas outras condições sobre a probabilidade de transição $P(C \rightarrow C')$:

- *ergodicidade*: qualquer configuração $C' \in \Omega$ deve ser acessível, em um número finito de passos, a partir de uma configuração inicial $C \in \Omega$;
- *balanço*: a distribuição de probabilidades de Boltzmann $P_b(C) = \exp(-S[C])$ deve ser mantida invariante no processo.

3.2.1 Freiamento Crítico

Existem vários algoritmos diferentes para implementar iterações de Monte Carlo satisfazendo as condições descritas. Alguns deles alteram as configurações localmente, isto é, mudam apenas as variáveis em um sítio de cada vez, enquanto outros tentam mudar conglomerados inteiros simultaneamente. Esta distinção é muito importante, pois nos estudos de fenômenos críticos, os algoritmos para simulações estocásticas muitas vezes apresentam fenômeno de freiamento crítico (*critical slowing down*). Conforme nos aproximamos de pontos críticos, o tempo de decorrelação, τ , definido como o número de passos de Monte Carlo necessários para que a auto-correlação entre configurações desapareça, se torna cada vez maior. Empiricamente, sabemos que τ está relacionado com o *comprimento de auto-correlação* (definido no item 3.4) através de

$$\tau \approx (\xi^{ac})^z,$$

onde z é o *expoente crítico dinâmico* e ξ^{ac} o comprimento de auto-correlação. Em sistemas finitos, como os utilizados em simulações computacionais, τ aumenta com

$$\tau \approx N^z.$$

Como configurações não são estatisticamente independentes dentro de um intervalo de tempo da ordem τ , isto nos leva a um aumento nos erros estatísticos ou em um aumento no tempo de computação necessário para gerar novas configurações. Este

fenômeno pode ser entendido como uma perda na *velocidade* em que *passamos* dentro do espaço de configurações da rede, ou seja, quanto mais próximos estivermos da linha crítica, mais *tempo* terá de ser gasto para obter configurações estatisticamente independentes.

Para o tradicional algoritmo de Metropolis, um exemplo de algoritmo local, temos $z \approx 2$, levando τ a crescer muito rapidamente com o tamanho da rede. Isto implica num aumento muito grande nos erros estatísticos dos observáveis.

A principal razão para isso é que em algoritmos locais as mudanças se propagam por difusão. Para tratar este problema foram propostos algoritmos não locais [4] nos quais conglomerados (*clusters*) inteiros de sítios têm seu valor alterado de uma só vez. Estes algoritmos, conhecidos como *algoritmos de clusters*, conseguiram evitar de maneira muito eficiente o problema do freiamento crítico. O algoritmo de cluster que se mostrou mais eficaz e geral foi o algoritmo de Wolff, que apresenta, por exemplo, $z \simeq 0.14$ para o modelo de Ising tridimensional e $z = 0$ para $SO(N)$ com $N = 2, 3$.

3.2.2 Metropolis

O mais importante e mais utilizado algoritmo para processos markovianos é o algoritmo de Metropolis [3] desenvolvido em 1953. Em nosso trabalho o algoritmo de Metropolis foi utilizado para variar a componente radial dos campos.

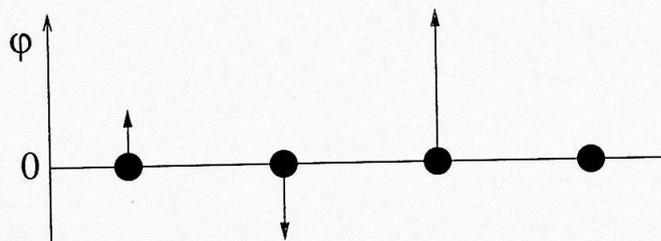


Figura 3.1: A componente radial de campos φ com simetria $SO(2)$.

Começando em uma configuração inicial qualquer C_1 , o algoritmo de Metropolis consiste de:

1. escolher algum novo estado C_2 de acordo com uma probabilidade de transição $P_T(C_2|C_1)$;
2. aceitar ou não a nova configuração C_2 de acordo com a probabilidade

$$P_A(C_2|C_1) = \min \left(1, \frac{P_T(C_1|C_2) \exp[S(C_2)]}{P_T(C_2|C_1) \exp[S(C_1)]} \right) \Rightarrow$$

$$P_A(C_2|C_1) = \min(1, \exp[S(C_2) - S(C_1)]) \quad \text{para } P_T(C_1|C_2) = P_T(C_2|C_1).$$
(3.6)

A probabilidade resultante $P_1 = P_T P_A$ satisfaz à condição de balanço detalhado. Em particular, para P_T simétrica, a informação necessária para decidir quanto à mudança para C_2 é obtida apenas da variação na ação $\Delta S_{C_1 \rightarrow C_2}$;

3. repetir os passos anteriores.

Definimos que uma iteração completa de Monte Carlo corresponde a repetir o procedimento de Metropolis em torno de N^d vezes.

Neste trabalho o algoritmo de Metropolis foi utilizado para variar φ , a componente radial dos campos $\vec{\varphi}$.

Para cada sítio \vec{n} da rede Λ tentamos variar o raio $\varphi(\vec{n})$ de acordo com

$$\begin{aligned}\delta\varphi(\vec{n}) &= \text{wrang} \times (\text{random}() - 0.5), \\ \varphi_{\text{trial}}(s) &= \varphi(\vec{n}) + \delta\varphi(\vec{n}),\end{aligned}$$

onde wrang é a amplitude máxima da mudança que tentamos implementar em $\varphi(\vec{n})$ e $\varphi_{\text{trial}}(\vec{n})$ é o novo valor do raio que será aceito com probabilidade

$$P_A(\varphi_{\text{trial}}, \varphi) = \varphi_{\text{trial}}^{N-1} \exp(-\Delta S[\vec{\varphi}]) - \varphi^{N-1} \text{random}().$$

A ordem com a qual os sítios são submetidos ao teste também é importante, neste trabalho utilizamos o método de *checkerboarding* para determinar a ordem da aplicação. Neste método os sítios são separados diagonalmente em dois grupos de forma que um sítio de um grupo só tenha vizinhos pertencentes ao outro. No caso bidimensional os dois grupos são divididos similarmente à disposição de cores de um tabuleiro de xadrez.

3.2.3 Wolff

Neste trabalho utilizamos o algoritmo de Wolff para variar a componente angular dos campos $\vec{\varphi}$.

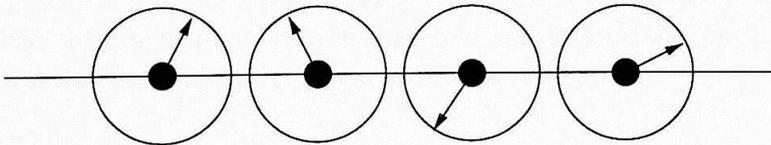


Figura 3.2: A componente angular de campos $\vec{\varphi}$ com simetria $SO(2)$.

Nesta exposição utilizaremos, por simplicidade, a ação do modelo Sigma Não-Linear dada em (1.3).

Este algoritmo implementa uma generalização para $N \geq 2$ da inversão da direção de campos de simetria $O(1)$, $\vec{\varphi} \rightarrow -\vec{\varphi}$. Esta generalização é feita através da escolha de uma direção arbitrária, $\hat{r} \in S_{N-1}$, e da reflexão dos campos com respeito ao hiperplano ortogonal a \hat{r} . Esta operação é realizada por $R(\hat{r})$,

$$R(\hat{r})\vec{\varphi}(\vec{n}) = \vec{\varphi}(\vec{n}) - 2(\vec{\varphi}(\vec{n}) \cdot \hat{r})\hat{r}.$$

Dadas estas definições podemos expor o algoritmo de Wolff. Este consiste da construção e reflexão de um cluster de sítios, C , através da:

1. escolha, aleatória, de uma direção $\hat{r} \in \mathbb{R}^N$;
2. escolha, aleatória, de um sítio $\vec{n} \in \Lambda$ e da inclusão de \vec{n} no cluster $C \in \Lambda$ (este é o primeiro sítio do cluster C que tentaremos construir);
3. ativação de cada *link*, ℓ , conectando um sítio $\vec{n}_1 \in C$ com um sítio $\vec{n}_2 \notin C$, com probabilidade

$$P_\ell(\vec{n}_1, \vec{n}_2) = \min(0, 1 - \exp[2\beta(\hat{r} \cdot \vec{\varphi}(\vec{n}_1))(\hat{r} \cdot \vec{\varphi}(\vec{n}_2))]);$$

se a probabilidade não for satisfeita marcamos este link como desativado de maneira que não seja visitado novamente;

4. repetição do passo 3 até que não hajam mais links para serem testados;
5. uma vez terminado o processo de iteração, invertemos todos sítios $\vec{n} \in C$ em relação ao hiper-plano ortogonal a \hat{r}

$$R(\hat{r})\vec{\varphi}(\vec{n}) = \vec{\varphi}(\vec{n}) - 2(\vec{\varphi}(\vec{n}) \cdot \hat{r})\hat{r}.$$

A ergodicidade do processo é garantida pelo fato de que há uma probabilidade não nula de que C seja composto por apenas um sítio. Para uma demonstração de que o algoritmo satisfaz o balanço detalhado ver [4] ou o capítulo de técnicas computacionais de [1].

Definimos uma iteração de Monte Carlo, com o algoritmo de Wolff, como a construção e inversão de um número de clusters que correspondam, em número total de sítios, a aproximadamente metade do tamanho da rede.

3.2.4 Calibrando os Algoritmos

Tanto no caso de Metropolis quanto no de Wolff será preciso calibrar o funcionamento dos algoritmos para que possamos obter o máximo de performance.

No caso de Metropolis, calibramos o parâmetro *wrang* e no de Wolff modificamos o número de clusters que são criados, *loopc*, uma vez que o tamanho médio dos clusters depende de β , um parâmetro físico. Em ambos os casos utilizamos o mesmo método, visando obter 50% de aceitação nos testes de Metropolis e 50% dos sítios invertidos numa iteração de Wolff.

O processo de calibre dos parâmetros consiste em estimar a taxa de rejeição dos testes para diferentes valores de *wrang* ou do número de sítios que têm seu valor mudado com a construção de *loopc* clusters. Sabemos que *wrang* = 0 implica que

a taxa de rejeição, $T(\text{wrang}) \in [0, 1]$, seja zero, além disso verificamos que a taxa de rejeição seja monotonicamente crescente com wrang .

Usamos então o método de diferenças finitas para tentar encontrar um valor de wrang tal que $T(\text{wrang}) \approx 0.5$.

Iniciamos os cálculos utilizando o único ponto conhecido a priori e um valor inicial qualquer para wrang como, por exemplo, 1. Realizamos vários testes com este valor de $\text{wrang}_{\text{ini}}$ obtendo $T(\text{wrang}_{\text{ini}})$. Em seguida, ajustamos uma reta usando estes dois pontos obtendo assim uma nova estimativa para wrang tal que $T(\text{wrang}) = 0.5$, este procedimento é repetido muitas vezes até que o valor da estimativa se estabilize. Caso a estimativa nos dê um valor negativo para wrang , tomamos como novo wrang metade do valor atual.

O cálculo do valor de loopc é realizado de forma análoga.

3.3 Back Rotation

Nesta seção descreveremos o método de *back rotation*, utilizado em parte dos cálculos computacionais deste trabalho.

É bem conhecido o fato de que uma simulação de um modelo $SO(N)$ em uma rede Λ finita com condições de contorno periódicas resultará em

$$\langle \vec{\varphi} \rangle = 0.$$

No entanto se considerarmos a *magnetização* de uma particular configuração da rede C_j da rede,

$$\vec{M}_{C_j} = \vec{\varphi}_{C_j}(0) = \frac{1}{N^d} \sum_{\vec{n} \in \Lambda} \varphi_{C_j, \vec{n}} \neq 0,$$

o que acontece é que a direção de \vec{M} não é constante e varia dentro de uma esfera S^{N-1} , o que nos leva a $\langle \vec{M} \rangle \rightarrow 0$.

O *back rotation* é um método numérico que utilizamos para eliminar as flutuações da direção de quebra de simetria das configurações da rede sem, no entanto, alterar o resultado final do limite do contínuo.

Este método consiste em fixar uma direção para a magnetização em uma das componentes dos campos da rede. Assim para campos $\vec{\varphi}$ com simetria $SO(N)$, escolhemos uma particular φ_N componente e impomos que a magnetização esteja na direção desta componente e seja positiva: $\sum_i \varphi_i \equiv M \propto \hat{\varphi}_N$. Isto é obtido através de uma rotação dos campos em todos os sítios.

Para isso, após o término da aplicação dos algoritmos de Monte Carlo (Wolff e Metropolis), quando obtemos uma nova configuração, medimos a sua magnetização. Em seguida determinamos a rotação que seria necessária para mapear \vec{M} na direção positiva de $\hat{\varphi}_N$ e aplicamos esta rotação aos campos de todos os sítios.

Uma vez que os modelos são invariantes por rotações de $SO(N)$, esta última operação não viola as hipóteses do algoritmo de Wolff e nos permite manter fixa a direção de quebra de simetria.

Este procedimento pode ser entendido como uma eliminação do *modo-zero* dos campos em todas as componentes exceto a N -ésima,

$$\langle M_{BR} \rangle = (0, \dots, 0, \langle M \rangle).$$

Com isto obtemos uma representação dos modelos em redes finitas em que $\langle \vec{\varphi} \rangle \neq 0$, o que sem dúvida é diferente da usual, sem entretanto, mudar o limite do contínuo.

3.4 Auto-Correlações entre Configurações

No cálculo dos observáveis assumimos que as configurações são totalmente decorrelacionadas, o que nunca é verdadeiro. Precisamos então estimar as auto-correlações entre as configurações para que possamos corrigir as barras de erro de nossas médias estatísticas.

Assim para uma sequência $\{(O_1, \dots, O_m) | m \gg 1\}$ de valores de um observável O , com média

$$\bar{O} = \frac{1}{m} \sum_{i=0}^m O_i,$$

a auto-correlação entre medidas de um observável é calculada da seguinte maneira,

$$g_O(j) = \frac{1}{m-j} \sum_{i=0}^{m-j} (O_i - \bar{O})(O_{i+j} - \bar{O}).$$

Para obter uma função de auto-correlação normalizada, para a qual $f(0) = 1$, definimos a função de auto-correlação como

$$f_O(i) = \frac{g_O(i)}{g_O(0)}. \quad (3.7)$$

O comprimento de auto-correlação de O , ξ_O^{ac} , é obtido através do ajuste de uma exponencial $\exp(-j/\xi_O^{ac})$ à 3.7. Usando esta quantidade é que corrigimos a barra de erro de O multiplicando-a por $1 + \xi_O^{ac}/m$.

Capítulo 4

A Proposta de Trabalho

Nosso objetivo era verificar se a relação indicada pelos cálculos de campo médio seria observada em simulações estocásticas dos modelos, pois neste caso teríamos como calcular de forma exata, no limite do contínuo, o valor de β_c dos modelos Sigma.

A principal questão que se apresenta é como aproximar a linha crítica no limite $N \rightarrow \infty$. Utilizamos dois métodos diferentes de convergência para (λ_c, α_c) e β_c no limite $N \rightarrow \infty$.

4.1 Mínimo de α_R

A primeira abordagem do problema de mapeamento da linha crítica do modelo $\lambda\varphi^4$ foi através da determinação do ponto de mínimo de α_R dentro do espaço de parâmetros (α, λ) . A quantidade α_R é definida através de

$$\langle |\tilde{\varphi}(\vec{k})|^2 \rangle = \frac{1}{N^d} \frac{1}{\rho^2(\vec{k}) + \alpha_R}, \quad \text{onde } \rho^2(\vec{k}) \equiv 4 \sum_{\mu=1}^N \text{sen}^2 \left(\frac{\pi k_\mu}{N} \right). \quad (4.1)$$

Esta quantidade é calculada nos programas através de uma inversão desta equação e de um ajuste ao valor numérico de $\sigma_0^2 = \langle \tilde{\varphi}^2 \rangle - \langle \tilde{\varphi} \rangle^2$. O mínimo de α_R foi calculado como função de $\theta = \arctan(-\lambda/\alpha)$, para um dado raio $R = \sqrt{\alpha^2 + \lambda^2}$.

Como α_R está relacionada com a massa renormalizada, m_R , através de

$$\alpha_R = (m_R a)^2,$$

onde a é o espaçamento entre os vértices, teremos no limite do contínuo, quando faremos com que o espaçamento entre os sítios vá a zero, $\alpha_R \rightarrow 0$.

Estimando os valores de α e λ que determinam a posição da linha crítica através de cálculos perturbativos pode-se obter a relação

$$\alpha_c + 3\lambda_c \sigma_0^2 = 0, \quad (4.2)$$

e com o cálculo da função de dois pontos obtemos

$$\alpha_R = \alpha + 3\lambda\sigma_0^2 \quad (4.3)$$

na fase simétrica e

$$\alpha_R = -2\alpha - 6\lambda\sigma_0^2, \quad (4.4)$$

na fase de simetria quebrada. Deve-se notar que esta não é uma fórmula fechada uma vez que σ_0^2 depende de α_R .

Assim, ao tomarmos o limite de $N \rightarrow \infty$, obtendo conseqüentemente $\alpha_R \rightarrow 0$, estaremos levando tanto (4.3) quanto (4.4) em (4.2) e portanto caminhando para a linha crítica.

4.2 Limite com α_R Constante

A abordagem através do mínimo de α_R não funcionou muito bem porque o critério depende muito fortemente de R quando R é pequeno, apresentando de fato uma divergência em $R = 0$ quando nos aproximamos do ponto gaussiano, por isso tentamos alternativamente mapear a linha crítica através de outra abordagem: tomando o limite com α_R constante ao longo da curva crítica.

Este método consiste em definir um critério para, dado N , escolher um valor para α_R de maneira que quando $N \rightarrow \infty$ tenhamos $\alpha_R \rightarrow 0$, mantendo entretanto α_R constante como função de α e λ . De posse deste valor tentamos mapear uma linha no plano (α, λ) a partir do ponto $(\alpha = \alpha_R, \lambda = 0)$ na qual α_R tenha sempre o mesmo valor.

A relação escolhida para variarmos α_R com N foi

$$\alpha_R = \frac{36}{N^2}.$$

Estas simulações foram todas realizadas dentro da fase simétrica do modelo, pois assim teríamos como calcular valores para σ_0^2 usando resultados do modelo Gaussiano, além de podermos testar o funcionamento do código comparando seus resultados com os resultados analíticos do caso $\lambda = 0$.

Comparando os resultados obtidos com os dois métodos no modelo $\lambda\varphi^4$, decidimos continuar apenas com o método de α_R constante nas simulações do modelo Sigma, pois neste caso podíamos contar com o cálculo do resultado analítico do modelo Gaussiano, sem termos de nos preocupar com a inclusão ou não nas somatórias no espaço de momentos do termo correspondente ao modo zero.

Capítulo 5

Resultados

Nossas simulações consistiram de:

1. simulações com poucas sub-médias com o objetivo de localizar a região do mínimo de $\alpha_R(\alpha, \lambda)$ ou de $\alpha_R = 36/N^2$, dados N e R (raio); como apresentado na figura 5.1(a);
2. uma série final de simulações de maior precisão na região do mínimo ou de $\alpha_R = 36/N^2$; como apresentado na figura 5.1(b);
3. uma última simulação no ponto indicado pelo ajuste polinomial dos resultados das simulações do item 2.

O número de sub-médias tomadas a cada conjunto de simulações variou entre 50 a 100 sub-médias de 2000 a 2500 iterações. Esta variação ocorreu conforme aumentávamos o tamanho das redes, para algumas redes o aumento no tempo de computação foi tal que nos forçou a diminuir o número de sub-médias e iterações.

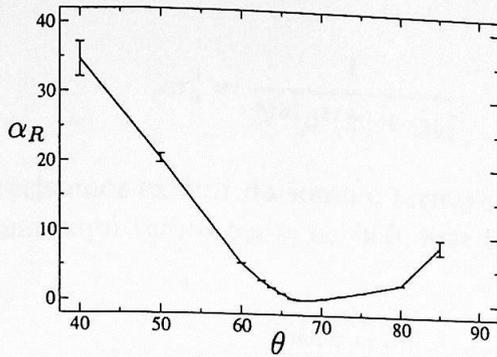
5.1 Mínimo de α_R

5.1.1 Simulações para Obtenção de α e λ

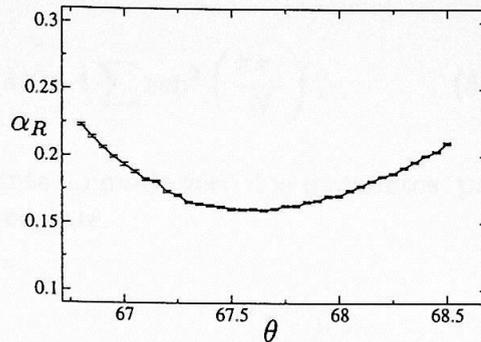
Para redes de 6 sítios foram realizadas simulações com os seguintes valores para o raio R : 10, 25, 50, 75, 100, 150, 200, 400, 600, 800 e 1000. Em redes de 8 e 10 sítios foram realizadas simulações apenas para raios com valores de 25, 50, 75, 100 e 150.

Dois motivos nos levaram a não realizar simulações para raios muito grandes em redes de 8 e 10 sítios: o aumento no tempo de computação das simulações decorrente do aumento do tamanho da rede e, principalmente, a observação da rápida convergência para a linha crítica como demonstrado na figura 5.2.

Uma vez estabelecido que podíamos inferir a localização dos pontos de mínimo a partir das simulações para redes menores, as primeiras sequências de simulações, que tinham a finalidade de mapear grosseiramente a posição do mínimo, foram eliminadas.



(a) Simulações para determinação grosseira da região do mínimo de α_R .



(b) Simulações para determinação fina do mínimo de α_R .

Figura 5.1: Cálculos realizados no modelo $\lambda\varphi^4$ em rede de $N = 6$ com raio $R = 25$, onde $\theta = \arctan(-\lambda/\alpha)$. Nestes cálculos $\vec{\varphi}$ tem simetria $O(1)$ e $d = 3$.

5.1.2 Resultados Obtidos

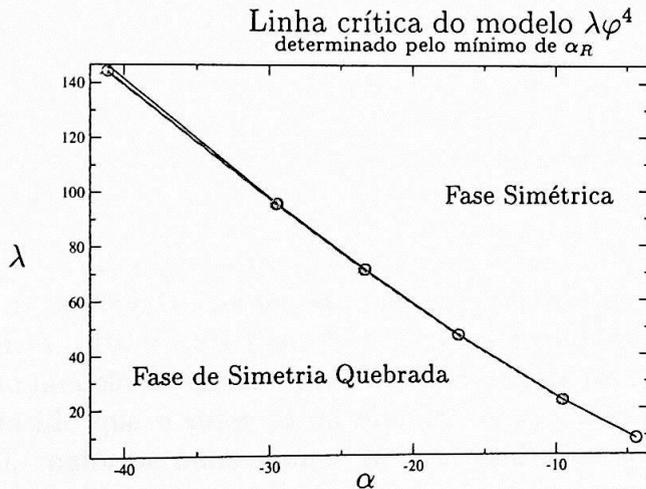


Figura 5.2: Linhas críticas determinadas minimizando-se α_R , para redes de 6, 8 e 10 vértices, em superposição. Nestes cálculos $\vec{\varphi}$ tem simetria $O(1)$ e $d = 3$.

O que observamos foi que, embora o valor da largura dos campos fosse essencialmente constante ao longo de quase toda a linha crítica, ele apresentava um aumento significativo para raios pequenos. Foi então levantada a possibilidade de que a convergência para raios menores fosse muito lenta, não podendo ser obtida (dentro de nossas capacidades computacionais) através desse limite.

No entanto, o salto entre os valores de σ_0^2 para raios pequenos e seu valor no ponto gaussiano ($\alpha = 0, \lambda = 0$), claramente visível na figura 5.3, precisa ser analisado. O que acontece é que, para usarmos a solução analítica do modelo Gaussiano no ponto gaussiano,

$$\sigma_0^2 = \frac{1}{N^d [\rho^2(\vec{k}) + \alpha_0]}, \quad \text{onde } \rho^2(\vec{k}) \equiv 4 \sum_{\mu} \text{sen}^2 \left(\frac{\pi k_{\mu}}{N} \right), \quad (5.1)$$

precisamos excluir da soma o termo correspondente ao modo zero dos momentos, pois como aqui temos $\alpha_R = \alpha_0 = 0$, este termo é divergente.

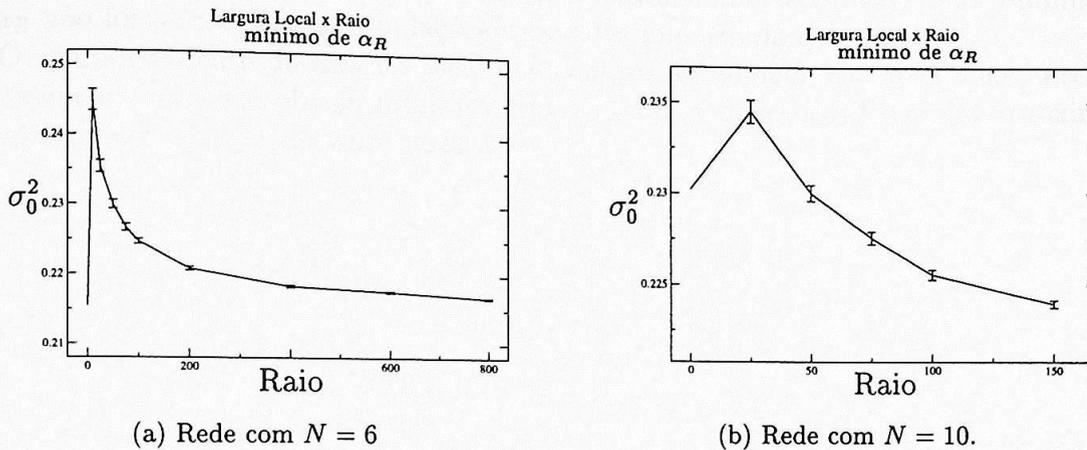


Figura 5.3: Valor de σ_0^2 na linha crítica do modelo $\lambda\varphi^4$. O método para o mapeamento da linha crítica utilizado no cálculo destes pontos foi a minimização de α_R . Nestes cálculos $\vec{\varphi}$ tem simetria $O(1)$ e $d = 3$.

Deve-se notar que a diferença entre as simulações de menor raio de cada rede N e o valor da solução do modelo Gaussiano são relativamente pequenas, diferem em 13% para $N = 6$ (menor raio 10) e 1.8% para $N = 10$ (menor raio 25). Usamos o valor da solução analítica do modelo Gaussiano para normalizar este resultado.

Entretanto, é nítido que o valor de σ_0 diminui, em relação ao valor da solução analítica de $\lambda = 0$, conforme aumentamos R . Os resultados de 5.3(a) indicam, no entanto, que o valor de σ_0 deve se estabilizar para R suficientemente grande. Ou seja que

$$\frac{d\sigma_0^2}{dR} \approx 0$$

para R suficientemente grande.

5.2 Simulações Realizadas a α_R Constante

Para realizarmos as simulações definimos um novo sistema de coordenadas com um novo eixo $\alpha' = \alpha + \frac{36}{N^2}$, de forma que tanto o raio R' quanto o ângulo θ' apresentados

nos gráficos têm sua origem no ponto $(\frac{36}{N^2}, 0)$ e não em $(0, 0)$. Para isto as sub-rotinas de controle dos programas de simulação tiveram de ser adaptadas.

Como pode ser observado na figura 5.4, o primeiro ponto, correspondente a $R' = 0$, não possui erros pois foi obtido através da solução analítica do modelo Gaussiano. Neste caso não há problema de divergência com o modo zero na somatória 5.1 pois $\alpha = 36/N^2$.

Foram realizadas simulações em redes de 6, 8 e 10 sítios com raios de 5, 10, 25, 50, 75, 100 e 150. Devido ao comportamento muito similar ao obtido com o mínimo de α_R não foram realizadas simulações com raios maiores do que 150.

O comportamento da largura local dos campos no modelo $\lambda\varphi^4$ a α_R constante é, em essência, o mesmo obtido minimizando α_R , com a diferença de que se aproxima de maneira consistente do valor para $\lambda = 0$.

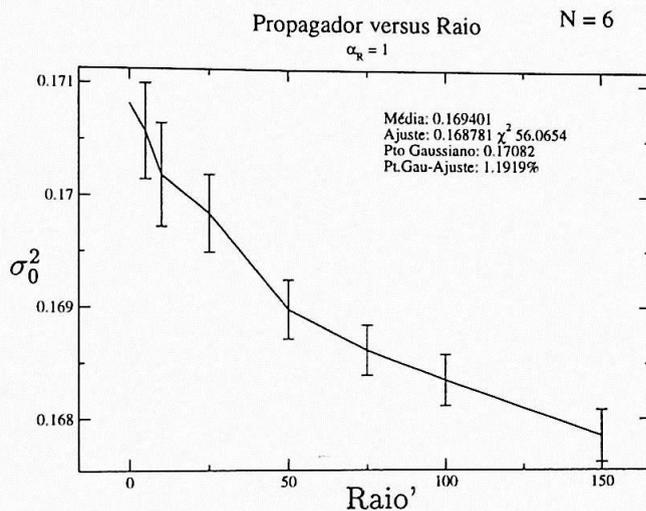


Figura 5.4: Largura local dos campos em redes de $N = 6$ sítios, $\alpha_R = 1$. Nestes cálculos $\vec{\varphi}$ tem simetria $O(1)$ e $d = 3$.

5.3 Modelo Sigma Não-Linear

Após nos convenceremos de que já tínhamos uma idéia suficientemente sólida sobre o comportamento de σ_0 ao longo da linha crítica do modelo $\lambda\varphi^4$, passamos a realizar simulações do modelo Sigma Não-Linear, cujo ponto crítico é o limite da linha crítica do modelo $\lambda\varphi^4$ quando $\lambda \rightarrow \infty$.

O critério utilizado para determinar o ponto crítico do modelo Sigma foi também a busca por $\alpha_R = 36/N^2$, pois consideramos o conjunto de dados obtidos no modelo $\lambda\varphi^4$ com $\alpha_R = 36/N^2$ mais consistente e cuidadosamente explorado do que os dados referentes ao mínimo de α_R . Assim, a sequência em N de pontos críticos do modelo Sigma poderia ser considerada como o limite da sequência de linhas críticas do $\lambda\varphi^4$ obtidas com α_R constante.

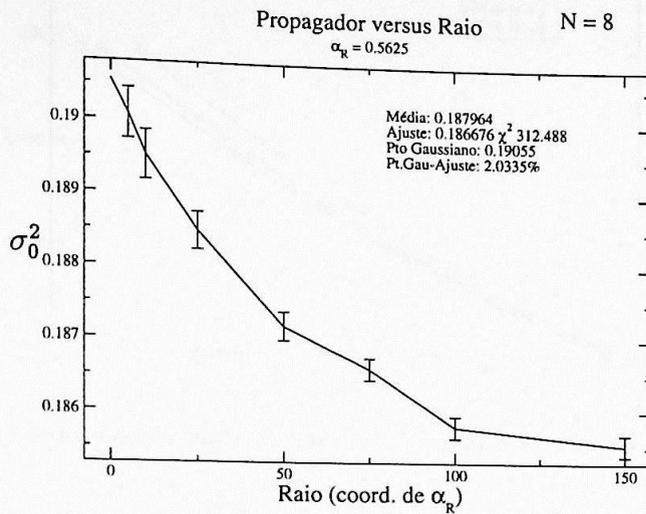


Figura 5.5: Largura local dos campos em redes de $N = 8$ sítios, $\alpha_R = 0.5625$. Nestes cálculos $\vec{\varphi}$ tem simetria $O(1)$ e $d = 3$.

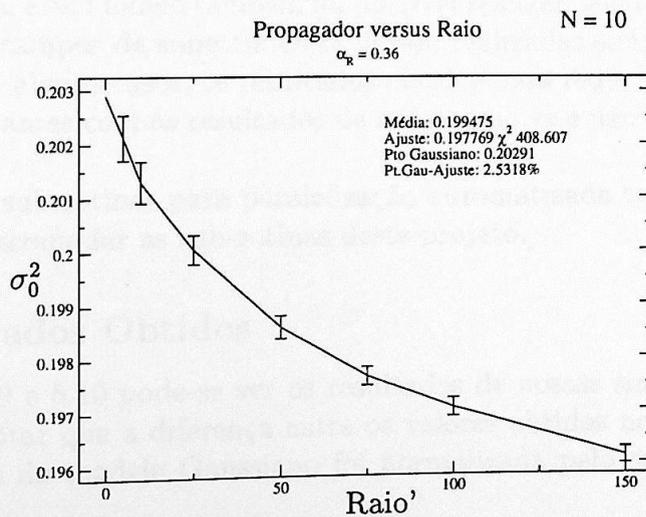


Figura 5.6: Largura local dos campos em redes de $N = 10$ sítios, $\alpha_R = 0.36$. Nestes cálculos $\vec{\varphi}$ tem simetria $O(1)$ e $d = 3$.

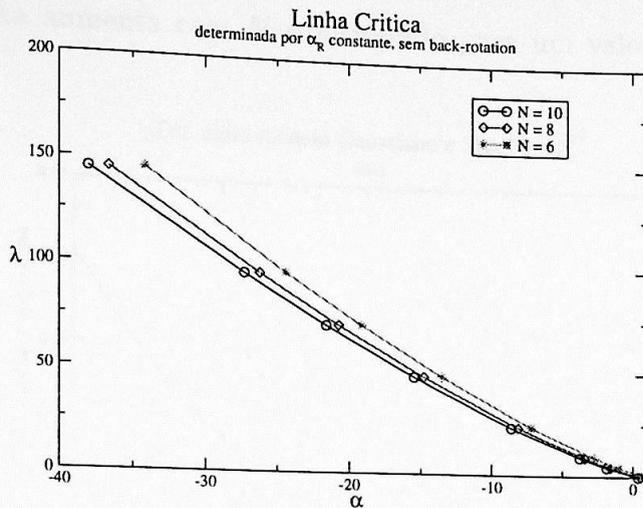


Figura 5.7: Linha crítica do modelo $\lambda\varphi^4$. Redes com N igual a 6, 8 e 10. Nestes cálculos φ tem simetria $O(1)$ e $d = 3$.

5.3.1 Simulações Realizadas

Como o modelo Sigma possui apenas um ponto crítico (em oposição à curva do $\lambda\varphi^4$) o trabalho computacional com este modelo é muito menos dispendioso tanto em termos de tempo de computação quanto em termos de volume de dados, o que nos permitiu levar as simulações a redes muito maiores do que as utilizadas no modelo $\lambda\varphi^4$.

Foram realizadas simulações com campos de simetria $O(1)$ em redes de 4 até 20 sítios como apresentado na figura 5.8. E para campos com simetria $SO(2)$ em redes de 5 até 16 sítios. Para este modelo também foi possível realizar cálculos em redes quadridimensionais com campos de simetria $O(1)$, foram realizadas simulações em redes de 4 até 11 sítios. Em alguns casos, os resultados obtidos para redes com $N = 4$ estavam totalmente discordantes com os resultados de redes maiores e não foram incluídos nos gráficos.

O conjunto de sub-rotinas para paralelização automatizada teve de ser estendido para que pudesse acomodar as sub-rotinas deste projeto.

5.3.2 Resultados Obtidos

Nas figuras 5.8, 5.9 e 5.10 pode-se ver os resultados de nossas simulações no modelo Sigma. Deve-se notar que a diferença entre os valores obtidos no modelo Sigma e o resultado analítico do modelo Gaussiano foi normalizada pelo seu valor no modelo Gaussiano

$$\Delta = \frac{\sigma_{\text{gaussiano}} - \sigma_{\text{sigma}}}{\sigma_{\text{gaussiano}}}$$

Os resultados obtidos são claros: existe uma diferença entre a largura local dos

dois modelos e esta aumenta com N convergindo para um valor finito no limite de $N \rightarrow \infty$.

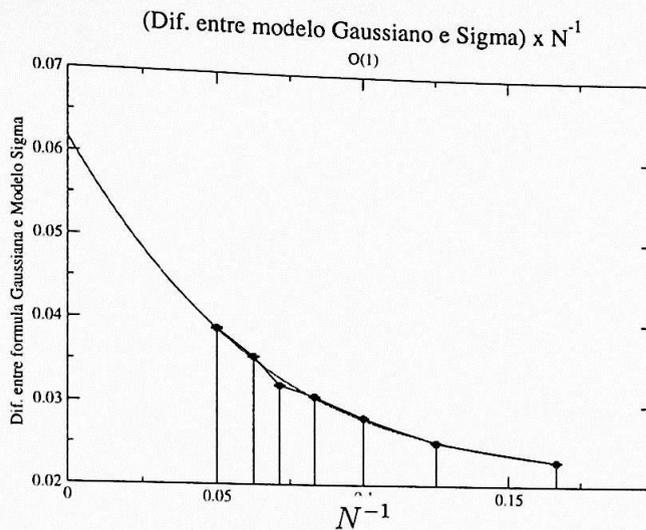


Figura 5.8: Diferença entre a fórmula analítica do modelo Gaussiano e resultados obtidos no modelo Sigma Não-Linear. Redes de $N = 6, 8, 10, 12, 14, 16$ e 20 . Campos $\vec{\varphi}$ com simetria $O(1)$.

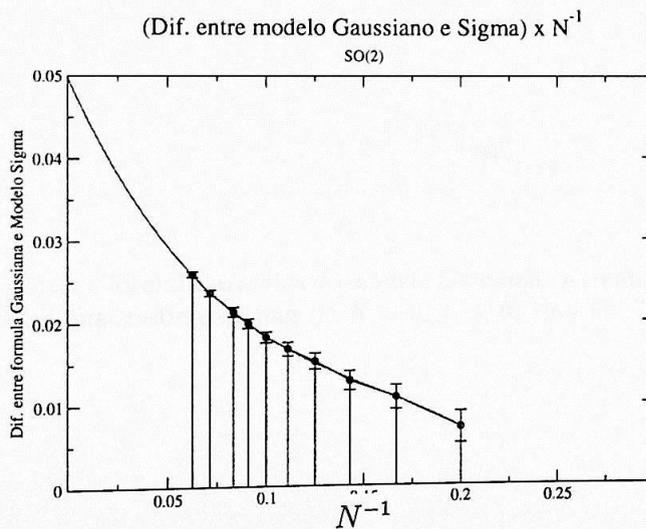


Figura 5.9: Diferença entre a fórmula analítica do modelo Gaussiano e resultados obtidos no modelo Sigma Não-Linear. Redes de $N = 5, 6, 8, 10, 12, 14$ e 16 . Campos $\vec{\varphi}$ com simetria $SO(2)$.

Diferença entre resultados dos modelos Sigma e Gaussiano $\times \frac{1}{N}$
 Redes quadri-dimensionais

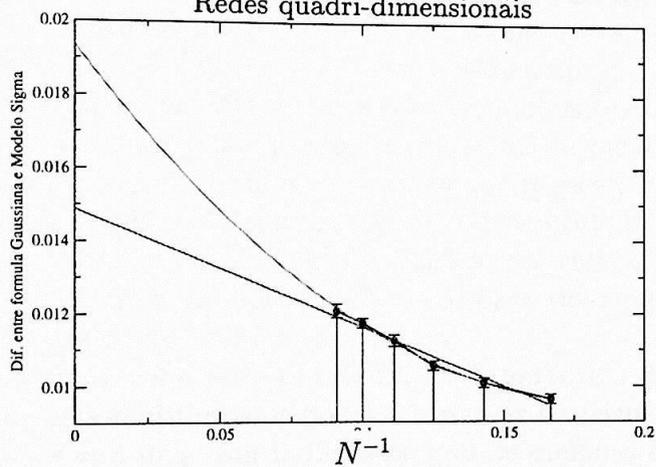


Figura 5.10: Diferença entre a fórmula analítica do modelo Gaussiano e resultados obtidos no modelo Sigma Não-Linear. Redes quadri-dimensionais de $N = 6, 7, 8, 9, 10$ e 11 . Campos $\vec{\varphi}$ com simetria $O(1)$.

Capítulo 6

Conclusões

Ao final dos trabalhos, foi possível constatar que os valores da largura média dos campos nos modelos Gaussiano, $\lambda\varphi^4$ e Sigma Não-Linear não são exatamente iguais em $d = 3$ ou $d = 4$. Apesar de muito pequena em alguns casos, a diferença existe e, aparentemente, aumenta com a ordem de simetria dos campos.

O valor de σ_0 no modelo $\lambda\varphi^4$ parece se afastar rapidamente do valor do resultado analítico do modelo Gaussiano, sendo possível notar sinais de que $d\sigma_0^2/dR$ diminui com o aumento de R . Curiosamente, o valor de estabilidade parece estar mais distante do resultado analítico do modelo Gaussiano do que o valor obtido no modelo Sigma.

Tanto para $O(1)$ e $SO(2)$ em $d = 3$, a diferença aumenta conforme tomamos o limite do contínuo convergindo no limite de $N \rightarrow \infty$ para valores finitos, da ordem de 6% para $O(1)$ e 5% para $SO(2)$ em $d = 3$.

Para o caso $O(1)$ em $d = 4$ o valor obtido foi 1.9%, embora a quantidade de dados para este caso tenha seja muito menor pode-se observar o mesmo tipo de comportamento observado em $d = 3$ mas com indícios de grande melhora em relação a $d = 3$. O que sugere como um possível prosseguimento para este trabalho:

1. um tratamento completo do caso $d = 4$, envolvendo tanto a obtenção de dados mais completos para o modelo Sigma quanto a análise do modelo $\lambda\varphi^4$;
2. a análise completa do caso $d = 5$.

Uma abordagem alternativa do problema seria realizar os mesmos cálculos em redes com condições de contorno fixas mas, como nos parece que o conjunto de resultados obtidos é conclusivo a respeito do problema, acreditamos que o resultado provavelmente seria o mesmo.

Assim, a medida das flutuações dos campos passa a se caracterizar como apenas um outro método numérico para determinação dos pontos críticos dos modelos Sigma.

Apêndice A

Tópicos em Computação Científica

O objetivo deste capítulo é apresentar conceitos e idéias que, acreditamos, são de grande valia na prática da computação científica nos dias de hoje.

A.1 Linguagens de Script e Computação Científica

Nesta seção caracterizaremos as linguagens de programação de *scripts* e serão expostos os motivos pelos quais estas devem vir a desempenhar um papel cada vez maior dentro da execução de tarefas computacionais e, em particular, dentro do âmbito da programação científica.

A.1.1 Linguagens de Script

Linguagens de *Script* possuem propósitos diferentes das linguagens de programação de sistemas¹. Linguagens de programação de sistema, como C e C++, são feitas para construir estruturas de dados e rotinas a partir do zero, lidando com os elementos de programação mais básicos. São linguagens compiladas e que exigem que o programador esteja mais ciente de como o programa será executado pelo *hardware* e cuide ele mesmo de grande quantidade de detalhes.

Linguagens de *script* como Python [23, 24], Perl [30], Bash [31] e Visual Basic são construídas com diferentes propósitos em mente. Uma das principais características destas linguagens é o sacrifício de performance em troca de flexibilidade e facilidade de programação, ou seja de tempo de desenvolvimento. No entanto, como a performance de um aplicativo de script é dominada pela performance de seus componentes que são, normalmente, implementados em linguagens de programação de sistema, este não chega a ser um problema crucial. Elas assumem a pré-existência de um conjunto de componentes e são feitas, primariamente, para conectar estes componentes.

Dizemos que estas são linguagens de programação de nível mais alto no sentido de que uma sentença executa dezenas de vezes mais instruções de máquina do que as sentenças de linguagens como C ou C++.

¹C, C++ e Pascal são exemplos de linguagens de programação de sistema.

Linguagens de *script* existem há várias décadas, mas recentemente temos observado um grande aumento em seu uso e importância. Isto se deve a vários fatores: o crescimento da Internet; o aumento da capacidade do *hardware* computacional; o aumento do uso de interfaces gráficas e até mesmo uma mudança no perfil dos programadores.

O crescimento da Internet contribuiu muito fortemente para o aumento da popularidade das linguagens de *script* pois permitiu, que uma vasta quantidade de aplicativos e programas existentes se tornasse acessível. A própria Internet é uma enorme ferramenta de união de componentes.

O aumento na qualidade destas linguagens é claramente visível, partindo das *shells* de Unix que não possuíam suporte para funções, chegando até Python ou Ruby que têm suporte à orientação a objeto.

O aumento na velocidade dos computadores faz com que vários nichos de aplicações passem a ser preenchidos por scripts, muitos aplicativos já são inteiramente desenvolvidos em linguagens de *script* e possuem excelente performance.

Outro fator que também contribuiu em todo esse processo foi a mudança de perfil dos programadores. No início da década de 80 a maior parte dos programadores tinha alto grau de qualificação e treinamento, trabalhando em grandes projetos. Dentro deste quadro era razoável que o programador gastasse vários meses de estudo para dominar as técnicas de programação de uma dada linguagem. Hoje, com o advento do computador pessoal, mais e mais programadores *casuais* estão tendo que desenvolver programas, pessoas cuja principal função não é programar e que usam o computador como uma ferramenta ocasional no exercício de seu trabalho. A absoluta maioria dos pesquisadores que hoje escrevem programas para realizar seu trabalho se encaixam neste perfil.

Estes programadores casuais não têm o tempo e a inclinação para gastar meses aprendendo uma linguagem de sistema antes de poderem passar a escrever programas que lhe sejam úteis. Linguagens de *script*, possuem em geral sintaxe mais simples, podendo ser aprendidas em muito menos tempo.

A.1.2 Aplicações na Área Científica

A computação científica se caracteriza por realizar simulações computacionais, buscar soluções para equações diferenciais e implementar análise e redução de dados. A maior parte dos códigos científicos está escrita em Fortran ou C e muitos acreditam que linguagens de *script* não tem nenhum papel a desempenhar nestas aplicações.

A maior parte destas tarefas é computacionalmente muito intensiva, de forma que em geral não poderemos escrever os algoritmos em Python ou Perl. No entanto, para uma vasta quantidade de aplicações científicas, já existem algoritmos desenvolvidos e exaustivamente testados, disponíveis sob forma de bibliotecas em repositórios de software científico na rede [32].

As linguagens de *script* se tornam então uma ferramenta muito útil e poderosa para o meio acadêmico, pois permitem ao pesquisador fazer uso destas bibliotecas

com menos esforço, tanto no aprendizado da linguagem quanto no desenvolvimento do código.

Devemos entender que, na absoluta maior parte das vezes, o pesquisador científico não é um programador profissional e quer gastar seu tempo se preocupando com a ciência por de trás de seus programas e não estudando detalhadamente uma linguagem de programação antes de poder escrever um programa que realmente faça alguma coisa.

Há também linguagens de script comerciais, desenvolvidas especificadamente para o desenvolvimento de aplicações científicas, como a do Maple, do Mathematica e do MatLab. Estas linguagens possuem a vantagem de possuir grandes bibliotecas para, dentre outras áreas, solução de equações diferenciais e manipulação algébrica. Contudo estes pacotes são por vezes demasiadamente caros à luz dos orçamentos de pesquisa, podendo também obstruir a colaboração científica entre diferentes grupos pois o compartilhamento ou desenvolvimento em conjunto de programas nestas linguagens só pode ocorrer caso os grupos envolvidos tenham acesso ao mesmo software proprietário.

A.2 Paralelização: PVM/MPI

Uma outra tendência muito forte que surgiu nos últimos anos é a paralelização dos programas de computação científica. Isto se deve à tendência de barateamento do custo de processadores e seu aumento de performance. Hoje a maneira mais barata de se obter uma máquina com altíssimo desempenho é através de um conglomerado de nós de processamento paralelo. Atualmente, é perfeitamente possível para um conjunto de poucos grupos de pesquisa obter uma máquina com dezenas ou centenas de nós de processamento paralelo.

No entanto, fazer uso real desta capacidade computacional requer a paralelização dos programas.

Nos anos 90, vimos o surgimento de dois tipos distintos de tecnologias para abordar esta (relativamente) nova necessidade de rodar em paralelo: suporte direto através do compilador e ambientes para troca de mensagens.

O modelo de suporte direto e automatizado através do compilador, no qual o programador se envolve muito menos com a tarefa de paralelizar o código, teve seus maiores sucessos através de FORTRAN 90 e *High Performance Fortran* (HPF). Neste modelo, o compilador tenta encontrar trechos do código que possam ser rodados em paralelo. No caso do HPF o usuário pode incluir diretivas que indiquem que trechos do código devem ou não ser distribuídos para execução em paralelo, mas os detalhes da paralelização são deixados por conta do compilador. No entanto, o alto custo destes compiladores e a falta da disponibilidade desta tecnologia para PCs fez com que esta opção não tivesse tanto sucesso junto à comunidade científica.

Os ambientes para troca de mensagens são conjuntos de bibliotecas que provêm meios para separação de um programa para execução em paralelo; de certa forma, esta é a linguagem *assembler* para o processamento paralelo. O programador é totalmente

responsável pela paralelização da aplicação. Os dois ambientes para troca de mensagens mais populares são o *Parallel Virtual Machine* (PVM) e o *Message Passing Interface* (MPI).

Embora seja por vezes argumentado [21] que MPI seja, sob alguns aspectos, mais robusto e simples do que PVM, PVM foi o ambiente escolhido para nosso trabalho. Essa escolha ocorreu essencialmente por termos acesso imediato a documentação de PVM [22] e pela disponibilidade de um módulo de PVM em Python [25] quando iniciamos este trabalho de paralelização².

A.2.1 O Modelo de Paralelização Implementado

A computação paralela com um sistema como o PVM pode ser realizada de várias maneiras que diferem em como é realizada a organização das tarefas. O modelo escolhido foi o chamado mestre-escravos. Nesta abordagem, um programa de controle separado *mestre* é responsável pela inicialização e espalhamento de processos, monitoramento e coleta de dados. Os programas *escravos* é que realizam a produção de dados propriamente dita.

Esta abordagem, foi escolhida por ser *suplementar* ao código Fortran já desenvolvido e não *complementar*, pois com essa suplementariedade, nós:

1. evitamos a introdução de bugs num código extremamente complexo e já relativamente estável;
2. mantivemos simples a compilação dos programas;
3. não vinculamos o uso dos programas às bibliotecas de PVM. Ou seja, mantivemos o maquinário de cálculo separado da interface;
4. obtivemos uma estrutura suplementar que pode facilmente ser re-utilizada em outros projetos do grupo de pesquisa com modificações mínimas (o que já está acontecendo);
5. podemos escolher a linguagem de programação a ser utilizada para o projeto.

A.3 Python

Uma vez identificada a necessidade de implementar uma estrutura externa, ao nosso código era preciso decidir como fazê-lo. Dentre as várias linguagens de script existentes hoje, Python se mostrou a mais adequada às nossas necessidades, pois:

1. dá suporte³ a programação orientada a objeto;

²Hoje já há suporte para o uso do protocolo de MPI em Python [26]

³Uma linguagem de programação *suporta* uma particular técnica ou estilo de programação se provê meios para que o uso desta técnica ou estilo seja alcançado com facilidade e conveniência. Uma linguagem não suporta uma técnica se, apesar de permitir seu uso, exige esforço ou habilidades excepcionais para a utilização desta.

2. possui sintaxe simples e um núcleo central relativamente pequeno; o que contribui fortemente para a legibilidade do código, o rápido aprendizado e evita dificuldades em voltar ao uso da linguagem depois de meses sem utilizá-la⁴;
3. é distribuída sob uma licença livre que, dentre várias outras coisas, nos garante que não teremos problemas para realizar atualizações ou colaborar com outros grupos de pesquisa;
4. há uma extensa documentação de Python livremente acessível, de maneira que um novo usuário não terá problemas em aprender a linguagem;
5. é uma linguagem com alta estabilidade e que possui uma grande comunidade para provisão de suporte através de *newsgroups*, grupos de interesse e da *Python Software Foundation* (PSA);
6. suporta inclusão de chamadas de funções em C e Fortran [33], permitindo o uso de seus *typecodes*.

Pelos motivos apresentados, acreditamos ser esta, atualmente, a linguagem de script mais adequada para programação científica, complementando códigos em C e Fortran.

INSTITUTO DE FÍSICA

Serviço de Biblioteca e Informação

Tombo: 3637
ex. 2

⁴Linux Journal *Hell is other people's Perl.*

Referências Bibliográficas

- [1] *Teoria Quântica de Campos na Rede, uma abordagem radical*, o Grupo Latt; o texto pode ser encontrado, na íntegra, em: <http://latt.if.usp.br/bazar/>.
- [2] C. N. Yang, T. D. Lee *Statistical Theory of Equations of State and Phase Transitions I. Theory of Condensation*, Phys. Rev. **87**, (1952) 404-409; T. D. Lee, C. N. Yang *Statistical Theory of Equations of State and Phase Transitions II. Lattice Gas and Ising Model*, Phys. Rev. **87**, (1952) 410-419.
- [3] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, J. Chem. Phys. **21** (1953), 1087; J. M. Hammersley and D. C. Handscomb, "Monte Carlo Methods", Methuen (1964).
- [4] J.-S. Wang and R. H. Swendsen, Physica A **167**, (1990), 565; U. Wolff, Phys. Rev. Lett. **62**, (1989), 361.
- [5] L. Kadanoff *Scaling Laws for Ising Models near T_c* ; Physics vol.2, N.6 pp 263-272 (1966).
- [6] C.B. Lang, *Computer Stochastic Simulations in Quantum Scalar Field Theory*, em *Stochastic Analysis and Applications in Physics* Proc. of the NATO ASI in Funchal, Madeira, Aug. 1993, ed. L. Streit.
- [7] P. Bevington, D. Robinson, *Data Reduction and Error Analysis for the Physical Sciences* Second Ed., McGraw-Hill (1992).
- [8] I. S. Gradshteyn e I. M. Ryzhik, *Table of Integral, Series and Products* fórmula 3.462, equação 1 e fórmula 3.461 eq. 2 e 3 página 337.
- [9] *ibid.* [8], fórmula 3.621, equações 3 e 4, página 369.
- [10] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes in C* Second Ed., Cambridge University Press (1997).
- [11] Reif, *Fundamentals of Statistical and Thermal Physics*, McGraw-Hill (1965).
- [12] Gardiner, C. W. *Handbook of Stochastic Methods*, Springer-Verlag (1983).
- [13] J. Binney, N. Dowrick, A. Fisher e M. Newman *The Theory of Critical Phenomena*, Oxford Science Publications (1995).

- [14] P. Brémaud *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues* Springer-Verlag (1999).
- [15] W. Feller *An Introduction to Probability Theory and Its Applications* John Wiley & Sons (1950).
- [16] G. Parisi *Statistical Field Theory*, Addison-Wesley Publishing (1988).
- [17] A. Papoulis *Probability, Random Variables and Stochastic Processes* McGraw-Hill (1965).
- [18] M. Le Bellac, *Quantum and Statistical Field Theory*, Oxford Science Publications (1991).
- [19] C. Rebbi (ed.), *Lattice Gauge Theories and Monte Carlo Simulations*, World Scientific (1983).
- [20] *Finite Lattice Systems with True Critical Behavior* J. L. deLyra, S. K. Foong and T. E. Gallivan, *The Physical Review* **D46**, (1992) 1643-1657.
- [21] K. Dowd, C. Severance, *High Performance Computing* Second Ed., O'Reilly (1998).
- [22] A. Geist, A. Beguelin, J. Dongarra *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing* (Scientific and Engineering Computation Series) MIT Press (1994).
- [23] M. Pilgrim, *Dive into Python*. Distribuído sob a licença LGPL (2001).
<http://diveintopython.org>
M. Lutz and D. Ascher *Learning Python*, O'Reilly (1999).
- [24] Python Language Home Page <http://www.python.org>.
- [25] Pypvm Home Page <http://pypvm.sourceforge.net/>.
- [26] Scientific Python Home Page:
<http://starship.python.net/~hinsen/scientific.html>
Pympi project page <http://sourceforge.net/projects/pympi/>.
- [27] J. Ousterhout *Scripting: Higher Level Programming for 21st Century*, IEEE Computer magazine, March 1998.
- [28] Beazley *Feeding a Large-scale Physics Application to Python*.
- [29] J. deLyra *Beowulf e o Saci: Como Construir e Operar o Seu Próprio Sistema PMC de Processamento Paralelo Baseado no Sistema Operacional Linux*, <http://latt.if.usp.br/pmc/Howto/>.
- [30] R. Schwartz, T. Christiansen *Learning Perl*, O'Reilly (1997).

- [31] M. Cooper *Advanced Bash-Scripting Guide* <http://linuxdoc.org/LDP/abs/html/>.
- [32] Netlib Repository <http://www.netlib.org>.
- [33] Swig Project - *Simplified Wrapper and Interface Generator*: <http://www.swig.org>
F2Py - *Fortran to Python Interface Generator*:
<http://cens.ioc.ee/projects/f2py2e/>.