



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA
GRADUAÇÃO EM FÍSICA

FERNANDO JOSÉ DE ALMEIDA

MÉTODOS PARA SIMULAÇÃO DE SLE CORDAL

FORTALEZA

2016

FERNANDO JOSÉ DE ALMEIDA

MÉTODOS PARA SIMULAÇÃO DE SLE CORDAL

Monografia de Bacharelado apresentada à
Coordenação da Graduação do Curso de Física,
da Universidade Federal do Ceará, como
requisito parcial para a obtenção do Título
de Bacharel em Física.

Orientador: Prof. Dr. André Auto Moreira

FORTALEZA
2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca do Curso de Física

-
- A446m Almeida, Fernando José de
Métodos para simulação de SLE cordal / Fernando José de Almeida. – 2016.
32 f. : il.
- Monografia (Graduação em Física) – Universidade Federal do Ceará, Centro de Ciências,
Departamento de Física, Curso de Bacharelado em Física, Fortaleza, 2016.
Orientação: Prof. Dr. André Auto Moreira.
Inclui bibliografia.
1. Loewner, equações de. 2. Modelos de rede. 3. Mapas conformais. I. Moreira, André Auto.
II. Título.

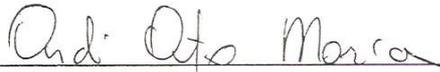
FERNANDO JOSÉ DE ALMEIDA

MÉTODOS PARA SIMULAÇÃO DE SLE CORDAL

Monografia de Bacharelado apresentada à
Coordenação da Graduação do Curso de Física,
da Universidade Federal do Ceará, como
requisito parcial para a obtenção do Título
de Bacharel em Física.

Aprovada em .

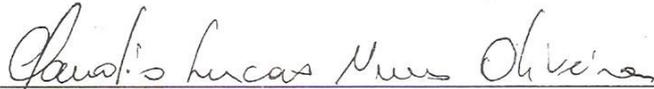
BANCA EXAMINADORA



Professor Dr. André Auto Moreira (Orientador)
Universidade Federal do Ceará (UFC)



Mestre Heitor Fernandes Credidio
Universidade Federal do Ceará (UFC)



Professor Dr. Claudio Lucas Nunes de Oliveira
Universidade Federal do Ceará (UFC)

À minha família.

AGRADECIMENTOS

Aos Meus Pais, pelo apoio e compreensão irrestritos.

Ao professor José Soares de Andrade Junior, pelas orientações, e por me dar a oportunidade de iniciar na pesquisa científica.

Ao professor André Auto Moreira, pela orientação, paciência e incentivo durante os dois anos como seu aluno, que tornaram possível a realização deste trabalho.

A todo o corpo docente do departamento de física da UFC, pelo conhecimento fornecido ao longo de toda a graduação; e aos professores do departamento de matemática da UFC que também tiveram participação fundamental no meu aprendizado.

A esta universidade, por todos os recursos empregados e que acompanham toda a formação de seus alunos, em sua rotina no campus.

Aos colegas de curso, em especial, Adaias Souza, Emanuel Fontelles, Filipe Martins, João Pedro, José Bentivi, Ken Aikawa, Michel Rodrigues, Nathanaell Sousa, Pablo Ramon, Pedro Henrique, Raul Peixoto, Sofia Magalhães e Wendel Oliveira, pela amizade e apoio que contribuíram para minha formação.

Por fim, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio à pesquisa científica proporcionado aos alunos durante a graduação.

RESUMO

Modelos bidimensionais que simulam o fenômeno das transições de fases são comuns em física estatística. Acredita-se que, em alguns casos, a evolução do sistema apresenta invariância conforme na região crítica. Caso essa suposição seja verdadeira, no limite contínuo, quando o espaçamento da rede tende a zero, as curvas aleatórias que formam a fronteira dos aglomerados na transição de fase poderiam ser descritas por um processo dinâmico denominado *SLE*. Neste trabalho definimos *SLE* cordal, e falamos sobre sua conexão com os modelos de rede. Mostramos também um método para resolução numérica das equações de Loewner, conhecido como SLE discreto. A partir dele é possível verificar a convergência dos modelos de rede em SLE_{κ} , o que leva a uma maior compreensão do comportamento dos sistemas na fase crítica, como por exemplo na obtenção de seus expoentes críticos. Finalmente, uma alternativa ao método discreto é mostrada, que produz um algoritmo mais eficiente na maioria das situações.

Palavras-chave: Equações de Loewner. SLE. Modelos de rede. mapas conformais.

ABSTRACT

Two dimensional models that simulate the phenomenon of phase transitions are common in statistical physics. It's believed that some models are conformally invariant in the system's critical region. If this assumption is true, in the continuum limit when the spacing of the mesh is zero, random traces that form the boundary of clusters at the phase transition could be described by a dynamic process called *SLE*. In this work we define chordal SLE, and talk about its connection to lattice models. We also show a method for numerical evaluation of the Loewner equation, using a method know as discrete SLE; from which is possible to verify the convergence of lattice models which leads to a greater understanding of the behaviour of such systems, like getting their critical-point exponents. Finally, we shown an alternative form of the discrete SLE, producing a more efficient algorithm in most situations.

Keywords: Loewner equations. SLE. Lattice models. Conformal mapping.

LISTA DE FIGURAS

Figura 1 – Diagrama pressão-temperatura da água	10
Figura 2 – Aplicação de g_t em um movimento browniano simples para alguns instantes de tempo	16
Figura 3 – Ilustração do processo de exploração em uma parte da rede.	23
Figura 4 – Processo de exploração cordal para $n = 10^4$ pontos de γ_t	24
Figura 5 – Função U_t obtida usando o método discreto.	25
Figura 6 – Gráfico da evolução da variância de U_t com o tempo.	25
Figura 7 – Comparação dos algoritmos 2 e 3 conhecendo-se U_t	29

SUMÁRIO

1	INTRODUÇÃO	10
2	EQUAÇÕES DE LOEWNER E DEFINIÇÃO DE SLE	13
2.1	Noções em variáveis complexas	13
2.1.1	Regiões do plano complexo	13
2.1.2	Domínios simplesmente conectados	13
2.1.3	Mapas conformes	14
2.1.3.1	Preservação dos ângulos	14
2.1.3.2	Invariância de escala para pontos próximos	14
2.2	Equações de Loewner na versão cordal	15
2.3	SLE e algumas propriedades	17
3	SLE DIRETO E APLICAÇÃO EM ALGUNS MODELOS DE REDE	18
3.1	Métodos de simulação para o SLE discreto: Algoritmo Zipper	18
3.1.1	Obtenção de $U(t)$ conhecendo-se γ_t	18
3.1.2	Obtenção de γ_t conhecendo-se U_t	21
3.2	Estimativa de k para alguns modelos de rede	23
3.2.1	Percolação crítica	23
4	PERSPECTIVAS PARA A SOLUÇÃO DA EQUAÇÃO DE LOEWNER	26
4.1	Solução para f_t	26
5	CONCLUSÃO	30
	REFERÊNCIAS	31

1 INTRODUÇÃO

Fenômenos de grande interesse em física são as transições de fase [2], pois ocorrem amplamente na natureza. Quando algum parâmetro (temperatura, por exemplo) é levado a um valor crítico, o sistema sofre drásticas alterações de suas propriedades físicas, devido a uma mudança global nas características microscópicas do sistema. Uma análise quantitativa do que realmente ocorre na vizinhança de uma transição de fase é em muitos casos inviável, pois a quantidade de variáveis e suas relações necessárias para a descrição do processo podem ser tão grandes quanto o número de constituintes básicos do sistema.

Um exemplo simples de transições de fase são as mudanças de estado da matéria, onde alterações significativas na organização microscópica dos componentes de um material são observadas quando uma temperatura característica ao mesmo é alcançada (mantendo a pressão ou o volume constante). As alterações provocam uma mudança qualitativa no material, como por exemplo nas mudanças de estado da água à volume constante (FIGURA 1). Na Figura 1 o ponto (T_c, P_c) é o ponto onde as fases líquida e gasosa tornam-se indistinguíveis: a partir de uma configuração $T > T_c$ e $P > P_c$, não é possível realizar a transição líquido-gasoso (ou vice-versa) variando-se somente pressão ou temperatura do material.

Figura 1 – Diagrama pressão-temperatura da água



Fonte: Elaborado pelo autor. A curva de pressão de vapor acaba em (T_c, P_c) , e a partir deste ponto as fases tornam-se indistinguíveis.

Uma característica presente na transição de fase é uma variável denominada parâmetro de ordem: uma quantidade observável que decresce com $|T_c - T|^\beta$ em uma vizinhança abaixo do ponto crítico e é zero acima do mesmo. β é uma constante positiva.

No exemplo dado acima, na transição líquido-gás o parâmetro de ordem é a diferença $\rho_l - \rho_g$ entre as densidades da água no estado líquido e gasoso, que vai a zero acima da região crítica. Outras quantidades (como capacidade térmica por exemplo) podem obedecer também a lei de potências da forma $|T_c - T|^{-\nu}$, onde ν é uma constante positiva; ou seja, parâmetros que vão pra infinito acima da região crítica. Os expoentes β , ν e outros que o sistema pode ter, relacionados com algum parâmetro que varia na região crítica exclusivamente pela variação da temperatura; são denominados *expoentes de ponto crítico*.

Ao longo dos experimentos muitos sistemas de diferentes materiais apresentaram os mesmos expoentes de ponto crítico. Esse padrão revelava a possibilidade da universalidade [3] de comportamento dos materiais na fase crítica do sistema. Posteriormente outros expoentes críticos foram descobertos e os sistemas passaram a ser agrupados em classes de universalidade. Dois sistemas com os mesmos valores de expoentes críticos pertencem à mesma classe de universalidade.

Pode-se inferir a partir da ideia da universalidade que somente um reduzido conjunto de parâmetros microscópicos são necessários para descrever um sistema quando próximo a uma transição de fase. Particularidades microscópicas podem ser completamente ignoradas. Modelos simples podem ser criados para analisar o comportamento crítico de tais sistemas. Muitos deles podem ser descritos utilizando-se os modelos de rede [9] (*lattice models*). Modelo de rede é uma versão discreta do plano em que os vértices (arestas) representam pontos desse plano; e as arestas (vértices) são as conexões entre pontos vizinhos do sistema. As conexões são feitas em função de alguma propriedade de interesse na análise, e que os elementos do sistema têm em comum caso estejam conectados. Modelos de percolação; modelo de Ising; e modelo de Potts de q estados (*q-state Potts model*) são alguns exemplos de modelos de rede.

Acreita-se que alguns sistemas físicos (os descritos pelos modelos citados acima estão nessa lista) apresentam invariância conforme na região crítica, no limite em que o espaçamento da rede tende a zero [11]. No sentido de elucidar essa questão, Schramm [12] combinou um processo dinâmico chamado evolução de Loewner [13] com o cálculo estocástico. Assim, uma família de curvas aleatórias denominadas SLE (*Schramm-Loewner evolution* ou *Stochastic Loewner evolution*) com parâmetro κ , ou SLE_κ foram definidas. Para o caso dos modelos de rede, se tomarmos uma curva aleatória que contorna o aglomerado (região de vértices conectados) que tenha as dimensões do sistema na transição de fase; ela deverá convergir para uma SLE no limite contínuo. O modelo do caminho aleatório (*simple random walk*), com algumas restrições, converge para SLE nesse limite. Essas restrições geram variações desse modelo. Quando retiramos todos os ciclos (*loop-erased*

random walk), foi mostrado que a curva resultante é SLE_2 [15]; quando simplesmente proibimos que o percurso tenha ciclos (*self-avoiding walk*), a curva gerada parece ser $SLE_{8/3}$ [16]. Alguns outros modelos que comprovadamente geram curvas SLE no limite contínuo são: percolação crítica [17] e árvores de extensão uniforme (*uniform spanning trees*)[15].

O corpo deste trabalho é dividido em 4 capítulos. No primeiro capítulo definiremos SLE partindo da evolução de Loewner. Algumas propriedades da SLE serão discutidas. Não focaremos nos detalhes de como são obtidas as equações de Loewner, onde um estudo mais aprofundado em análise complexa e cálculo estocástico são necessários. No segundo capítulo abordaremos alguns modelos de rede e suas conexões com SLE. No capítulo 3 resolveremos numericamente a equação de Loewner por um método conhecido como *SLE discreto*, onde usaremos os resultados para construir um algoritmo de que obtém a função U_t . Usaremos esse método para estimar o valor de κ para o modelo de percolação crítica e verificar sua convergência para o SLE_6 . No terceiro e último capítulo mostraremos algumas perspectivas de um algoritmo mais eficiente que resolve a equação de Loewner diretamente por uma expansão em série de Laurent.

2 EQUAÇÕES DE LOEWNER E DEFINIÇÃO DE SLE

2.1 Noções em variáveis complexas

Alguns conceitos e definições serão importantes para as seções seguintes, então resumimos as questões mais pertinentes. Usamos [1] como referência para essa seção.

2.1.1 Regiões do plano complexo

Dado um ponto z_0 no plano complexo, define-se uma *vizinhança* ϵ de z_0 como todos os pontos z internos ao círculo centrado em z_0 e de raio ϵ . Ou seja, todos os pontos z tal que

$$|z - z_0| < \epsilon. \quad (2.1)$$

Quando ϵ é irrelevante para a discussão, chamamos 2.1 apenas de vizinhança.

Um ponto z_0 é dito im *ponto interior* a um conjunto $S \subset \mathbb{C}$ se existe alguma vizinhança de z_0 somente com pontos $z \in S$. Por outro lado, z_0 é um *ponto exterior* a S se existe alguma vizinhança de z_0 somente com pontos z não pertencentes a S . Se z_0 não é interior nem exterior a S , então z_0 é um *ponto de fronteira* de S . Todos os pontos de fronteira de S juntos formam a *fronteira* de S .

Um conjunto é dito *aberto* se não contém seus pontos de fronteira. Por outro lado, o conjunto é dito *fechado* se não contém todos os seus pontos de fronteira. Um conjunto aberto S é dito *conectado* se podemos ligar quaisquer dois pontos $z_1, z_2 \in S$ por um número finito de segmentos de linha que ficam inteiramente em S . Um conjunto aberto e conectado é chamado *domínio* (qualquer vizinhança é, portanto, um domínio). Um domínio com algum de seus pontos de fronteira é denominado *região*.

2.1.2 Domínios simplesmente conectados

Uma função

$$z(t) = x(t) + iy(t) \quad (a \leq t \leq b), \quad (2.2)$$

onde x, y são funções contínuas do parâmetro real t ; forma um conjunto de pontos chamado *curva*. A curva é dita *simples* se não toca em si mesma em nenhum ponto ($z(t_1) \neq z(t_2)$ quando $t_1 \neq t_2$). Um *arco de Jordan* ou *curva fechada simples* é uma curva simples exceto para $z(a) = z(b)$. Se $z(t)$ tem primeira derivada contínua, a curva é dita *diferenciável*. Se uma curva é diferenciável e $z'(t) \neq 0$ para $a < t < b$, o vetor tangente (o número complexo

pode ser visto como um vetor na forma $z = a + ib = (a, b)$

$$\mathbf{T} = \frac{1}{|z|}(Re(z'), Im(z')) \quad (2.3)$$

é bem definido ao longo da curva, com inclinação $argz'(t)$; nesse caso a curva é dita *suave*.

Um número finito de curvas suaves $\gamma_1[a_1, b_1], \gamma_2[a_2, b_2], \dots, \gamma_n[a_n, b_n]$ justapostas de forma que $\gamma_1(b_1) = \gamma_2(a_2), \gamma_2(b_2) = \gamma_3(a_3), \dots, \gamma_{n-1}(b_{n-1}) = \gamma_n(a_n)$ é denominado *caminho*. Se $\gamma(a_1) = \gamma(b_n)$, então o caminho é *fechado simples*.

Um domínio D é dito *simplesmente conectado* se qualquer caminho fechado simples dentro dele só contenha pontos de D. Informalmente pode-se dizer que é um domínio sem "buracos".

2.1.3 Mapas conformes

2.1.3.1 Preservação dos ângulos

Um mapa conforme é uma função f que mapeia ponto a ponto um domínio simplesmente conectado $D \subset \mathbb{C}$ em outro domínio simplesmente conectado $D' \subset \mathbb{C}$; preservando os ângulos relativos entre quaisquer curvas que se interceptam em algum ponto do domínio. Essa é uma outra forma de dizer que a função f deve ser analítica em qualquer ponto de D e $f'(z) \neq 0, \forall z \in D$. Vejamos como isso é verdade aplicando f em duas curvas suaves $C_1 (z_1 = z_1(t), a_1 \leq t \leq b_1)$ e $C_2 (z_2 = z_2(t), a_2 \leq t \leq b_2)$ que se interceptam em um ponto $z_0 = z_1(t_0) = z_2(t_0)$. Temos que o ângulo do vetor tangente à curva imagem de C_1 no ponto $\omega_0 = f(z_0)$ é $\phi_1 = arg\omega_1 = argf'[z_1(t_0)] + argz'_1(t_0) = argf'[z_0] + \theta_1$; e para a curva imagem de C_2 é $\phi_2 = arg\omega_2 = argf'[z_2(t_0)] + argz'_2(t_0) = argf'[z_0] + \theta_2$. Nos dois casos a linha tangente às curvas C_1 e C_2 no ponto z_0 sofrem uma rotação por um ângulo $argf'(z_0)$. Juntando os dois resultados,

$$\phi_2 - \phi_1 = \theta_2 - \theta_1, \quad (2.4)$$

e as diferenças de ângulos são preservadas. Como f é analítica, esse tipo de propriedade é verificada em todos os pontos em uma vizinhança de z_0 , onde $f'(z) \approx f'(z_0)$. A função f portanto preserva os ângulos relativos sempre que for analítica e $f'(z) \neq 0, \forall z \in D$, como queríamos mostrar.

2.1.3.2 Invariância de escala para pontos próximos

Dado um ponto z_0 no domínio D, sua distância a qualquer ponto z também do domínio D é $|z - z_0|$; após a aplicação do mapa conforme f suas distâncias serão

$|f(z) - f(z_0)|$. Portanto o fator de escala é a fração

$$\frac{|f(z) - f(z_0)|}{|z - z_0|}. \quad (2.5)$$

Devido analiticidade da função f , existe uma vizinhança de z_0 onde $f'(z_0)$ é uma boa aproximação para 2.5. Esse resultado garante que mapas conformais apresentam invariância de escala localmente.

O resultado acima e o da seção (2.1.3.1) nos mostra a ligação entre invariância conforme e preservação da simetria nas vizinhanças de um ponto qualquer do domínio. Mesmo que a função f transforme o domínio D drasticamente em larga escala, regiões pequenas do domínio têm suas formas preservadas quando observadas em torno de um ponto da mesma.

Se um sistema físico que apresenta invariância conforme na região de transição de fase,

2.2 Equações de Loewner na versão cordal

Todas as afirmações adiante não serão provadas neste trabalho; foram apenas extraídas de materiais sobre o assunto ([13] e [8]), de modo que as equações de Loewner pudessem ser obtidas, e algumas propriedades das funções f_t e g_t que satisfazem essas equações pudessem ser exploradas mais adiante, onde resolvemos numericamente o SLE_κ .

Seja $\gamma(t)$ ($t \geq 0$) uma curva no semiplano superior complexo (\mathbb{H}), partindo da origem e que não toca a si mesma (esse tipo de curva é o que caracteriza o termo cordal usado durante o trabalho). Sob essas condições, $\mathbb{H} \setminus \gamma[0, t]$ é um domínio simplesmente conectado. Além disso, se $\gamma(t) \rightarrow \infty$ quando $t \rightarrow \infty$; existe um mapa conforme g_t que transforma $\mathbb{H} \setminus \gamma[0, t]$ em \mathbb{H} tal que $g_t(z) \rightarrow z$ quando $z \rightarrow \infty$. Sendo assim, podemos expandir g_t em uma série de Laurent, com coeficientes nulos para os termos de ordem maior que z^2 ,

$$g_t(z) = z + \frac{C(t)}{z} + O\left(\frac{1}{|z|^2}\right), \quad z \rightarrow \infty. \quad (2.6)$$

$C(t)$ é chamada de capacidade de $\gamma[0, t]$ no semiplano (*half-plane capacity*), e é estritamente crescente. É possível mostrar que podemos reparametrizar a curva $\gamma(t)$ para que $C(t)=2t$. Em consequência disso, $g(t)$ satisfaz a equação diferencial de Loewner:

$$\frac{\partial g_t}{\partial t} = \frac{2}{g_t(z) - U_t}; \quad g_0(z) = z. \quad (2.7)$$

Onde $U_t \in \mathbb{R}$ é a chamada função diretora (*driving function*). Temos também que $U_{t_k} = g_{t_k}(\gamma(t_k))$ (embora apenas por um processo limite, pois $\gamma(t_k)$ não faz parte do domínio de

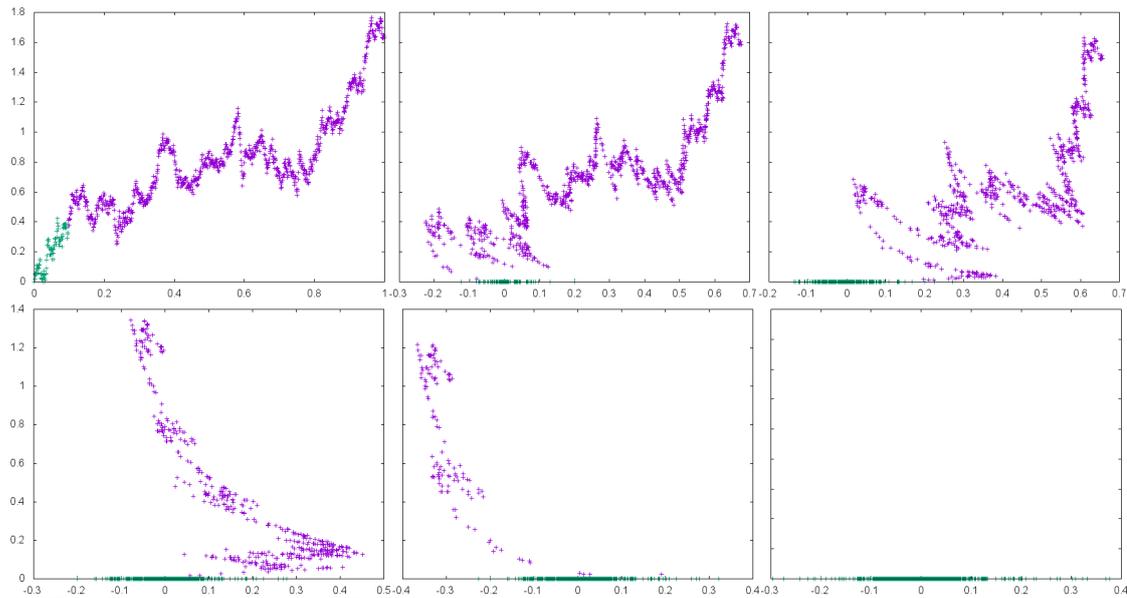
g, e sim da fronteira do sistema). Inversamente, a partir de uma função diretora conhecida, podemos obter os pontos que formam $\gamma(t)$ em um mapa que transforma o domínio \mathbb{H} no contradomínio $\mathbb{H} \setminus \gamma[0, t]$:

$$\frac{\partial f}{\partial t} = \frac{2}{U_t - z} \frac{\partial f}{\partial z}; \quad f_0(z) = z. \quad (2.8)$$

Essas são as duas versões da equação diferencial de Loewner.

Na Figura 2 usamos o método discreto (descrito mais adiante) para ilustrar como a função g_t retira a curva γ_t da fronteira do domínio $\mathbb{H} \setminus \gamma[0, t]$. A curva usada é um movimento browniano gerado especialmente para a simulação. O método de geração usado é o de deslocamento aleatório do ponto médio [14], que garante que a curva não toca em si mesma. O movimento browniano foi gerado com 1025 pontos separados por uma distância $\Delta x = 2^{-10}$ entre si. A curva é mostrada para alguns instantes específicos, após o mapa g_t alcançar um certo número de pontos.

Figura 2 – Aplicação de g_t em um movimento browniano simples para alguns instantes de tempo



Fonte: Elaborado pelo autor. Cada imagem representa a curva γ_t após a aplicação de g_t para: 0; 100; e 300 pontos, acima; e 600; 900; e 1025 pontos, abaixo (da esquerda pra direita nos dois casos).

2.3 SLE e algumas propriedades

SLE é um processo que utiliza as equações de Loewner tomando $U_t = \sqrt{k}B_t$, onde k é um parâmetro positivo e B_t é um movimento browniano com $\langle B_t \rangle = 0$ e variância $\langle |B_{t_2} - B_{t_1}|^2 \rangle = |t_2 - t_1|$. Esse modelo é denominado SLE_κ cordal. A equação de Loewner para esse modelo então fica

$$\frac{\partial g_t}{\partial t} = \frac{2}{g_t(z) - \sqrt{k}B_t}; \quad g_0(z) = z. \quad (2.9)$$

A curva γ_t associa-se então com o parâmetro k pela relação

$$\sqrt{k}B_t = g_{t_k}(\gamma(t_k)). \quad (2.10)$$

O valor de k influi naturalmente no comportamento de γ_t . Observa-se três fases distintas de comportamento dessa curva. Para $k \in [0, 4]$, γ é uma curva simples na ampla maioria das situações. Também é improvável que a mesma toque o eixo dos reais em $t > 0$. Para $k \in (4, 8]$, γ não é mais simples e intercepta o eixo dos reais, mas não preenche o espaço ($d_f < 2$). Para $k > 8$, a curva além de não ser simples e tocar o eixo dos reais se comporta de forma a preencher o plano complexo ($d_f = 2$). Foi provada uma relação [10] entre a dimensão fractal da curva γ e o parâmetro k , que é compatível com o comportamento da curva SLE_κ destacado acima:

$$d_f = \min\left(2, 1 + \frac{k}{8}\right). \quad (2.11)$$

Podemos comparar as curvas geradas pelos modelos de rede na transição de fase, onde sabe-se o valor da dimensão fractal, com a curva SLE_κ extraída da relação 2.11. Há algumas formas de fazer esse estudo. Mostraremos no capítulo 2 o método direto, onde resolveremos diretamente a equação de Loewner usando um método de discretização da SLE, de modo a obter a driving function U_t ; e sabendo que

$$\langle |U_{t_2} - U_{t_1}|^2 \rangle = k|t_2 - t_1|, \quad (2.12)$$

podemos estimar o valor de k analisando a evolução temporal da variância de U em relação ao tempo, como veremos adiante. Usaremos o modelo de percolação crítica como exemplo.

3 SLE DIRETO E APLICAÇÃO EM ALGUNS MODELOS DE REDE

3.1 Métodos de simulação para o SLE discreto: Algoritmo Zipper

3.1.1 Obtenção de $U(t)$ conhecendo-se γ_t

Seja g_t o mapeamento já mencionado ($\mathbb{H} \setminus \gamma[0, t] \rightarrow \mathbb{H}$) com $C(t)=2t$:

$$g_t(z) = z + \frac{2t}{z} + O\left(\frac{1}{z^2}\right), \quad (3.1)$$

e a equação de Loewner 2.7. Dado um instante t_k , temos que $g_{t_k}(\gamma_{t_k})$ mapeia γ_{t_k} para um ponto do eixo real, visto que U_{t_k} é real e $U_{t_k} = g_{t_k}(\gamma(t_k))$. Com isso, a curva após $g_{t_k}(\gamma(t_k))$ não parte mais da origem, mas do ponto U_t , e o restante $\gamma[t_k, t]$ é mapeado de forma a obtermos uma nova curva contínua. Para que a curva seja mapeada para começar na origem, no lugar de g_t podemos usar o mapeamento:

$$h_t(z) = g_t(z) - U_t. \quad (3.2)$$

Seja z_1, z_2, \dots, z_k um conjunto de pontos pertencentes a $\gamma(t)$. Se dividirmos t em intervalos $0 = t_0 < t_1 < \dots < t_k = t$, onde $\gamma(t_k) = z_k$; podemos aplicar sucessivos mapeamentos h_1, h_2, \dots, h_k intermediários a cada instante t_k , que vão parcialmente retirando a curva do semiplano superior, até que toda ela seja retirada. Assim, no instante t_1 o mapa h_1 leva $\mathbb{H} \setminus \gamma[t_0 = 0, t_1]$ para \mathbb{H} , com a nova curva sendo deslocada para $\gamma'[0, t_k - t_1]$, e o ponto z_1 é levado para a origem; no instante t_2 o mapa h_2 leva $\mathbb{H} \setminus \gamma[0, t_k - t_1]$ para \mathbb{H} , com a nova curva sendo deslocada para $\gamma''[0, t_k - t_2]$ e o ponto z_2 é levado para a origem; e assim sucessivamente até que no instante t_k o mapa h_k leve $\mathbb{H} \setminus \gamma^{(k)}[0, t_k - t_{k-1}]$ para \mathbb{H} , com toda a curva $\gamma[0, t_k]$ sendo retirada. O resultado geral é que $h_k \circ h_{k-1} \circ \dots \circ h_1$ mapeia $\mathbb{H} \setminus \gamma[0, t_k] \rightarrow \mathbb{H}$, como desejado. Cada mapa h_k intermediário tem sua própria expansão no infinito:

$$h_k(z) = z - \Delta U_k + \frac{2\Delta t_k}{z} + O\left(\frac{1}{z^2}\right), \quad z \rightarrow \infty, \quad (3.3)$$

onde $\Delta U_k = U_k - U_{k-1}$ e $\Delta t_k = t_k - t_{k-1}$. Então,

$$h_1(z) = z - \Delta U_1 + \frac{\Delta t_1}{z} + O\left(\frac{1}{z^2}\right), \quad (3.4)$$

$$h_2 \circ h_1(z) = z - U_2 + \frac{2t_2}{z - \Delta U_k + 2\frac{\Delta t_k}{z} + O\left(\frac{1}{z^2}\right)} + O\left(\frac{1}{z^2}\right), \quad (3.5)$$

que pode ser simplificado usando $z \rightarrow \infty$:

$$h_2 \circ h_1(z) = z - U_2 + \frac{2t_2}{z} + O\left(\frac{1}{z^2}\right). \quad (3.6)$$

Generalizando,

$$h_k \circ \dots \circ h_2 \circ h_1(z) = z - U_t + \frac{2t_k}{z} + O\left(\frac{1}{z^2}\right), \quad (3.7)$$

onde $U_t = \sum_{i=1}^k \Delta U_i$ e $t = \sum_{i=1}^k \Delta t_i$.

O tipo de mapeamento que mencionaremos é o mais simples para resolver o problema (*vertical slit*), onde cada h_i mapeia \mathbb{H} menos a linha vertical $[Re(\omega_i), \omega_i]$ para \mathbb{H} , com $\omega_i = h_{i-1} \circ h_{i-2} \circ \dots \circ h_1(z_i)$; em particular, $h_k(\omega_k) = 0$. Começamos o programa então aplicando h_1 em todos os pontos de $\gamma(t)$, garantindo apenas que a linha vertical que começa em $Re(\omega_1)$ e termina em ω_1 seja mapeada para o eixo real; em particular, $h_1(\omega_1) = h_1(z_1) = 0$. Os pontos restantes de $h_1(z_k)$ formam uma nova curva que começa em $h_1(z_2) = \omega_2$. O passo seguinte é aplicar o mapa h_2 em todos os pontos da curva resultante do passo anterior. A linha vertical que começa em $Re(\omega_2)$ e termina em ω_2 é mapeada para o eixo real. O ponto $h_2(\omega_2) = h_2 \circ h_1(z_1)$ é mapeado para a origem. Os pontos restantes de $h_2 \circ h_1(z_k)$ formam uma nova curva que começa em $h_2 \circ h_1(z_2) = \omega_2$. O processo é repetido até que finalmente aplicamos o mapa h_k , e após isso a curva γ_t inicial foi toda mapeada para o eixo real. Note-se que a curva mapeada em cada etapa i começa em ω_i , e não na origem, condição necessária para uma SLE cordal; a linha mapeada está a uma distância $Re(\omega_i)$ do ponto zero. No entanto, quanto mais próximo o ponto $\omega_1 = z_1$ de zero, os consequentes ω_k na simulação também o serão (pois os mapas h_i aproximam a curva cada vez mais para o eixo real em cada etapa). O programa fornece uma boa aproximação se escolhermos z_1 o suficiente próximo de zero.

Podemos obter uma forma funcional para o mapa h_k que resolva nosso problema para as condições apresentadas. Para h_k , a reta vertical no intervalo $[Re(\omega_k), \omega_k]$ é formada por pontos tais que $z = Re(\omega_k) + iIm(z)$ e $Im(z) \leq Im(\omega_k)$. Temos as restrições que $h_k(z) \in \mathbb{H}$ e $h_k(\omega_k) = 0$. Então $h_{k_{teste}} = -(z - Re(\omega_k))^2$ já resolve a primeira restrição. A segunda restrição pode ser resolvida fazendo-se $h_{k_{teste}} + Im(\omega_k)^2$ (para manter a primeira restrição $h_k(z) \in \mathbb{H}$ válida). Assim, ficamos com $h'_{k_{teste}} = (z - Re(\omega_k))^2 + Im(\omega_k)^2$, que resolve o problema. Porém, comparando com 3.3, o mapa h_k não tem termos $O(z^2)$. Devemos então diminuir a ordem do resultado de $h'_{k_{teste}}$, sem alterar as restrições mencionadas anteriormente. A solução é mostrada abaixo, que é a versão final h_k :

$$h_k(z) = i\sqrt{-(z - Re(\omega_k))^2 - Im(\omega_k)^2}. \quad (3.8)$$

Agora resta comparar 3.3 (para $z \rightarrow \infty$) e 3.10:

$$h_k^2 \simeq z^2 - 2z\Delta U_k + 4\Delta t_k + \Delta U_k^2 = z^2 - 2z\operatorname{Re}(\omega_k)^2 + \operatorname{Re}(\omega_k)^2 + \operatorname{Im}(\omega_k)^2,$$

$$\Delta U_k = \operatorname{Re}(\omega_k), \quad \Delta t_k = \frac{\operatorname{Im}(\omega_k)^2}{4}. \quad (3.9)$$

Podemos então escrever 3.10 do seguinte modo:

$$h_k(z) = i\sqrt{-(z - \Delta U_k)^2 - 4\Delta t_k}. \quad (3.10)$$

O resultado acima é a base para o algoritmo Zipper utilizado neste trabalho. Em Listing 1, mostramos uma implementação em C para o algoritmo descrito acima. O tempo gasto com o processamento é da ordem de $O(n^2)$, sendo n o número de pontos de γ fornecidos. Isso porquê é preciso computar, durante todo o programa, i mapas para cada ponto z_i ($h_i \circ h_{i-1} \circ \dots \circ h_1$), cada um gastando tempo linear (apenas a aplicação da fórmula funcional de h_i 3.10). Logo, a computação dos primeiros i pontos gasta tempo $O(i^2)$. Como o último ponto é o n -ésimo, o algoritmo tem complexidade $O(n^2)$, o que não é muito eficiente quando se é preciso processar uma quantidade considerável de pontos.

```

1 void getDrivFrmGamma(int Size, complex double *gamma, double* U, double*
   t)
2 {
3     int i, j;
4     U[0] = t[0] = 0;
5     for (i = 1; i < Size; i++){
6         U[i] = U[i - 1] + creal(gamma[i]);
7         t[i] = t[i - 1] + pow(cimag(gamma[i]), 2)/4;
8         for (j = i + 1; j < Size; j++)
9             gamma[j] = I*csqrt(-cpow(gamma[j] - creal(gamma[i]), 2) - pow
   (cimag(gamma[i]), 2));
10    }
11 }

```

Algoritmo 1: Código em C99 para obter pontos da função U_t a partir da curva γ_t em \mathbb{H}

3.1.2 Obtenção de γ_t conhecendo-se U_t

Nesse caso o mapa conforme utilizado faz o inverso do descrito na seção anterior. Vimos anteriormente que cada mapa h_i parte de $\mathbb{H} \setminus [Re(\omega_i), \omega_i]$ para \mathbb{H} , com $\omega_i = h_{i-1} \circ h_{i-2} \circ \dots \circ h_1(z_i)$. Em particular $h_k(\omega_k) = 0$, ou seja:

$$h_k \circ h_{k-1} \circ \dots \circ h_1(z_k) = 0. \quad (3.11)$$

Seja a função $l_k = h_k^{-1}$. Podemos então escrever 3.11 da seguinte forma:

$$l_k^{-1} \circ l_{k-1}^{-1} \circ \dots \circ l_1^{-1}(z_k) = 0.$$

Se aplicarmos $(l_1 \circ l_2 \circ \dots \circ l_k)$ à esquerda de cada lado da equação, obtemos

$$z_k = l_1 \circ l_2 \circ \dots \circ l_k(0). \quad (3.12)$$

Para encontrarmos a forma funcional de l , obtemos a inversa do mapa h_k obtida na seção anterior. Substituindo os valores em 3.9 em 3.10:

$$h_k(z) = i\sqrt{-(z - Re(\omega_k))^2 - Im(\omega_k)^2}. \quad (3.13)$$

Calculando a inversa $h_k^{-1} = f_k$ utilizando 3.10:

$$l_k(z) = \sqrt{z^2 - Im(\omega_k)^2} + Re(\omega_k). \quad (3.14)$$

Nesse caso, declaramos k pontos que começam com o valor zero. Chamemos tais pontos de v_1, v_2, \dots, v_k . No primeiro passo aplica-se l_i a cada ponto v_i para $i \in [1, k]; i \in \mathbb{Z}$. Como inicialmente todos são zero, obtemos

$$l_i(0) = \sqrt{-Im(\omega_i)^2} + Re(\omega_i) = \omega_i, \quad i \in [1, \dots, k]. \quad (3.15)$$

Em particular, temos $v_1 = \omega_1 = z_1$, encontrando o primeiro ponto da curva γ_t . No passo seguinte, aplica-se l_{i-1} a cada ponto $v_i = \omega_i$ para $i \in [2, k]; i \in \mathbb{Z}$. Nesse caso, obtemos $l_{i-1}(\omega_i) = l_{i-1} \circ l_{i-1}^{-1} \circ \dots \circ l_1^{-1}(z_k) = l_{i-2}^{-1} \circ \dots \circ l_1^{-1}(z_k)$. Particularmente, para $i = 2$ temos $l_1(\omega_2) = l_{i-1} \circ l_1^{-1}(z_2) = z_2$, encontrando o segundo ponto da curva γ_t . Fica claro que a cada passo i um ponto z_i da curva γ_t é encontrado. O processo é repetido até o k -ésimo passo, onde encontra-se o ponto z_k . Como resultado da simulação, são obtidos os pontos z_1, z_2, \dots, z_k da curva γ_t . É evidente, dada a análise acima, que o programa faz o mapeamento inverso do descrito na seção anterior, isto é, parte do domínio \mathbb{H} e chega em $\mathbb{H} \setminus \gamma_t$. Podemos substituir 3.9 em 3.14 para obter

$$l_k(z) = \sqrt{z^2 - 4\Delta t_k} + \Delta U_k. \quad (3.16)$$

Em Listing 2 colocamos uma implementação em C do algoritmo que descrevemos acima. A complexidade temporal é $O(n^2)$, visto que cada ponto z_i é obtido após aplicar $l_1 \circ l_2 \circ \dots \circ l_i(0)$, totalizando i mapas de complexidade constante (apenas a forma funcional de f_i 3.16). Se tivermos n pontos a serem obtidos, o programa gasta tempo $O(n^2)$ em todo o processo.

```

1 void getGammaFrmDriv(int Size, complex double *gamma, double* U, double*
   t)
2 {
3     int i,j;
4     double dU[Size], dt[Size];
5     dU[0]=dt[0]=0;
6     gamma[0]=0;
7     for (i = 1; i < Size; i++){
8         gamma[i] = 0;
9         dU[i] = U[i]-U[i-1];
10        dt[i] = t[i]-t[i-1];
11    }
12    for (i = 1; i < Size; i++)
13        for (j = i; j < Size; j++)
14            gamma[j] = RaizComplexa(cpow(gamma[j],2)-4*dt[j-i+1])+dU[j-i
+1];
15 }

```

Algoritmo 2: Código para obter os pontos de uma curva em \mathbb{H} a partir de pontos da função U_t .

3.2 Estimativa de k para alguns modelos de rede

Usaremos o método de resolução da equação de Loewner para estimar o valor de k para modelo de percolação crítica.

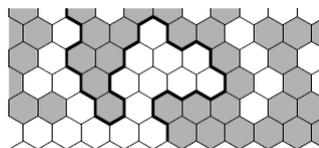
3.2.1 Percolação crítica

Difiniremos a percolação para sítios em uma rede triangular. Todos os vértices da rede são definidos independentemente como aberto, com probabilidade p ; ou fechado, com probabilidade $1-p$. p é a probabilidade da rede, uma configuração definida inicialmente. Sabe-se que para $p > 1/2$, é quase certo que exista um caminho de arestas conectadas (logo com os vértices abertos) que atravessa toda a rede, é e dito que ocorreu a percolação no sistema. O valor $p = 1/2$ é chamado de probabilidade crítica da rede triangular.

Um método equivalente e ainda mais simples de simular consiste em um método para colorir os hexágonos da rede dual. Basicamente, visita-se cada hexágono e sorteia-se um valor $p' \in [0 : 1]$. Se $p' \leq p$, onde p é um parâmetro de controle escolhido inicialmente; colore-se o hexágono com a cor azul. Caso $p' > p$, escolhe-se a cor a amarela. É sabido que para um valor de $p \leq 1/2$, a probabilidade de haver um aglomerado do tamanho da rede (e quando a ordem da rede tende ao infinito, o mesmo acontece com esse aglomerado) é quase nula. Para $p > 1/2$ exist, com probabilidade próxima de 1, um aglomerado azul do tamanho da rede. A fronteira que separa o aglomerado azul de todo o resto é a mesma curva gerada pelo método descrito no parágrafo anterior.

Como condição de contorno, os hexágonos que tocam uma de suas arestas no eixo real positivo são coloridos de azul, e os que tocam a linha real negativa são coloridos de amarelo. Na Figura 1 temos uma ilustração em pequena escala da rede na região crítica. Essas condições de contorno garantem que existe um único caminho de arestas dos hexágonos, começando da origem, que separa o aglomerado de hexágonos azuis ligados a semireta positiva dos reais; dos aglomerados de hexágonos amarelos ligados à semireta negativa dos reais. Tal caminho é chamado *processo de exploração cordal* de 0 a ∞ no semiplano.

Figura 3 – Ilustração do processo de exploração em uma parte da rede.



Fonte: [9].

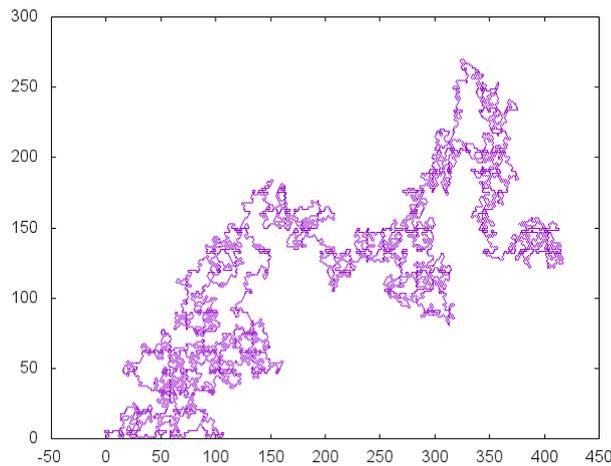
Utilizando um algoritmo [19] para gerar uma curva desse modelo, utilizamos

o algoritmo 01 para obter a função diretora U_t . Sabemos que essa curva é um movimento browniano com variância 2.12. O método que utilizamos é descrito em [20]. A curva é dividida em n partes de intervalos de Δt aproximadamente iguais e o somatório

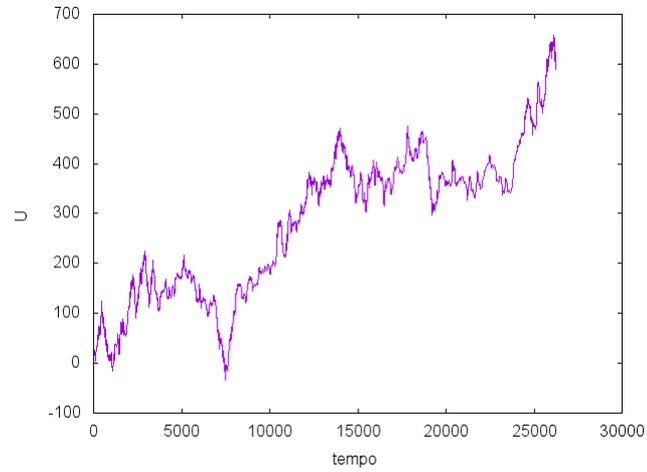
$$\langle |U_{t_i} - U_{t_i - \Delta t}|^2 \rangle \Delta t = \sum_{i=0}^N (U_{t_i} - U_{t_i - \Delta t})^2 / (N + 1), \quad (3.17)$$

é executado para valores arbitrários de n . N é o número de "pedaços" da curva que compõem o somatório, e depende do valor de n . Na Figura 05 temos a curva desejada. O coeficiente angular da reta aproximada é exatamente o valor de k , que no experimento resultou em $k \approx 6.19153 \pm 0.03521$. O valor que obtido é próximo ao $k=6$ obtido por Smirnov [17], embora o método usado para obter esse coeficiente ainda precise ser melhorado, dada as flutuações observadas no gráfico. Usando 2.11obtemos $d_f = 1,77$, próximo do valor 1,75 esperado.

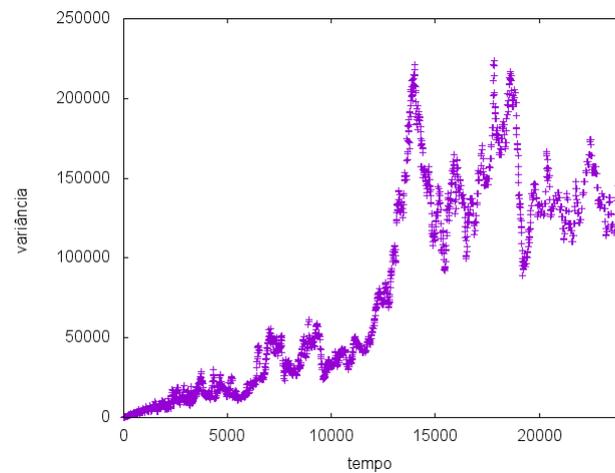
Figura 4 – Processo de exploração cordal para $n = 10^4$ pontos de γ_t .



Fonte: Elaborado pelo autor.

Figura 5 – Função U_t obtida usando o método discreto.

Fonte: Elaborado pelo autor.

Figura 6 – Gráfico da evolução da variância de U_t com o tempo.Fonte: Elaborado pelo autor. O coeficiente linear do gráfico obtido foi $k \approx 6.19153 \pm 0.03521$.

4 PERSPECTIVAS PARA A SOLUÇÃO DA EQUAÇÃO DE LOEWNER

4.1 Solução para f_t

Começamos admitindo que $f(z,t)$ possui expansão em série de Laurent:

$$f(z) = \sum_{i=-\infty}^{\infty} a_i(t)z^i. \quad (4.1)$$

Temos que $f(z) \rightarrow z$ quando $z \rightarrow \infty$. Portanto, todos os termos de ordem $O(z^2)$ ou superior têm coeficientes nulos; além disso, $a_0 = 0$ e $a_1 = 1$. Portanto,

$$f(z) = z + \sum_{i=0}^{\infty} \frac{a_i(t)}{z^i}. \quad (4.2)$$

Vamos utilizar 4.2 para resolver cada termo da equação 2.8. Para $\frac{\partial f}{\partial t}$:

$$\frac{\partial f}{\partial t} = \sum_{i=0}^{\infty} \frac{1}{z^i} \frac{\partial a_i}{\partial t}. \quad (4.3)$$

Já $\frac{\partial f}{\partial z}$ pode ser escrito como:

$$\frac{\partial f}{\partial z} = 1 - \sum_{i=0}^{\infty} \frac{ia_i}{z^{i+1}}. \quad (4.4)$$

Para o termo restante na equação 2.8, podemos ajusta-lo transformando-o em uma série geométrica:

$$\frac{2}{U_t - z} = \frac{-2/z}{1 - U_t/z} = -\frac{2}{z} \sum_{i=0}^{\infty} \left(\frac{U_t}{z}\right)^i. \quad (4.5)$$

Substituindo os valores acima na equação 2.8:

$$\frac{\partial a_0}{\partial t} = 0; \quad \frac{\partial a_n}{\partial t} = -2U_t^{n-1} + 2 \sum_{i=0}^{n-2} ia_i U_t^{n-i-2}. \quad (4.6)$$

Começamos a simulação com um conjunto de pontos $(t_1; U_{t_1}), (t_2; U_{t_2}), \dots, (t_k; U_{t_k})$ conhecidos de U_t , além do ponto $t_0 = 0; U_{t_0} = 0$). A cada passo do programa, encontramos cada termo $\frac{\partial a_n}{\partial t}$ utilizando a fórmula recursiva 4.6. Após isso, podemos fazer a seguinte aproximação:

$$a_n(t_i) = \frac{\partial a_n}{\partial t} \Delta t_i + a_n(t_{i-1}), \quad (4.7)$$

onde $\Delta t_i = t_i - t_{i-1}$. Inicialmente, temos $a_n(t_0) = 0$. Portanto, temos todos os elementos pra resolver o primeiro passo e encontrar $a_n(t_1)$ (e consequentemente f_{t_1}). Seguindo este raciocínio podemos encontrar todos os coeficientes da série de Laurent em qualquer t , e

com isso determinar f_t . Note-se que para uma simulação computacional, temos que limitar o número de termos da expansão f_t 4.2, truncando a série para obter o resultado em uma precisão desejada. No algoritmo 3 apresentamos uma implementação deste método na linguagem C.

```

1 void f_Laurent(int Size, int imax, complex double *A, double* U, double*
    t)
2 {
3     // imax e o numero de termos da serie de Laurent (truncamento);
4     // k e a variavel de tempo, e A e o vetor de coeficientes de ft.
5     int t,i,j,k;
6     complex double zAnt;
7     for (i = 1; i < size; i++)
8         gamma[i] = 0;
9     for (i = 0; i < imax; i++)
10        A[i] = dA[i] = 0;
11    for(k = 1; k<size; k++){
12        for(i = 1; i < imax; i++){
13            dA[i] = -2*pow(U[k],i-1);
14            for (j = 1; j < i-1; j++)
15                dA[i] += 2*j*pow(U[k],i-j-2)*A[j];
16        }
17        A[i] += dA[i]*(t[k]-t[k-1]);
18    }
19 }

```

Algoritmo 3: Código em C99 para obter f_t por expansão em série de Laurent, conhecendo-se U_t .

A complexidade temporal do algoritmo acima é $O(nm^2)$, sendo n o número de pontos computados e m o número de termos da série de Laurent truncada obtida pela recursão 4.6; o que é extremamente eficiente visto que $m \ll n$ é uma constante determinada previamente. Seria ideal utilizar o algoritmo acima puramente, mas não é possível: há dois problemas na aproximação utilizada. Primeiramente, para pontos z próximos de zero, a série de Laurent diverge e os valores obtidos nessa região não são válidos. O contorno exato que define essa região de validade de f_t nessa aproximação depende da função U_t . O segundo defeito da abordagem é que ao aplicarmos $f_t(U_t) = \gamma(t)$ para obter os pontos da curva γ_t , não obteremos resultados corretos. Ocorre que $f_t(U_t) = \gamma(t)$ é apenas verificado em uma condição limite, já que nenhum ponto do domínio (\mathbb{H}) pode ser levado para um ponto da curva γ_t , pois a mesma não faz parte da imagem da função f_t ($\mathbb{H} \setminus \gamma_t$). Portanto a região onde a série de Laurent produz resultados inválidos inclui $z = U(t)$ no eixo real.

A seguir apresentaremos uma solução que desenvolvemos para os problemas apresentado no parágrafo anterior. Essencialmente, mesclamos os algoritmos 2 e 3. Pelo método discreto, para obter o ponto z_k , utilizamos a relação 3.12. Ao aplicarmos os n primeiros mapas l_i no ponto z_k ($n < k$),

$$z_k = l_1 \circ l_2 \circ \dots \circ l_{k-n}(\phi), \quad (4.8)$$

onde $\phi = l_{k-n+1} \circ l_{k-n+2} \circ \dots \circ l_k(0)$. Por outro lado, o mapa equivalente $l_{t_i} = l_1 \circ l_2 \circ \dots \circ l_i$ tem uma diferença apenas de translação em relação f_{t_i} (apenas para os pontos $\gamma(t) = U(t)$, sendo $U(0)=0$), de modo que vale a relação

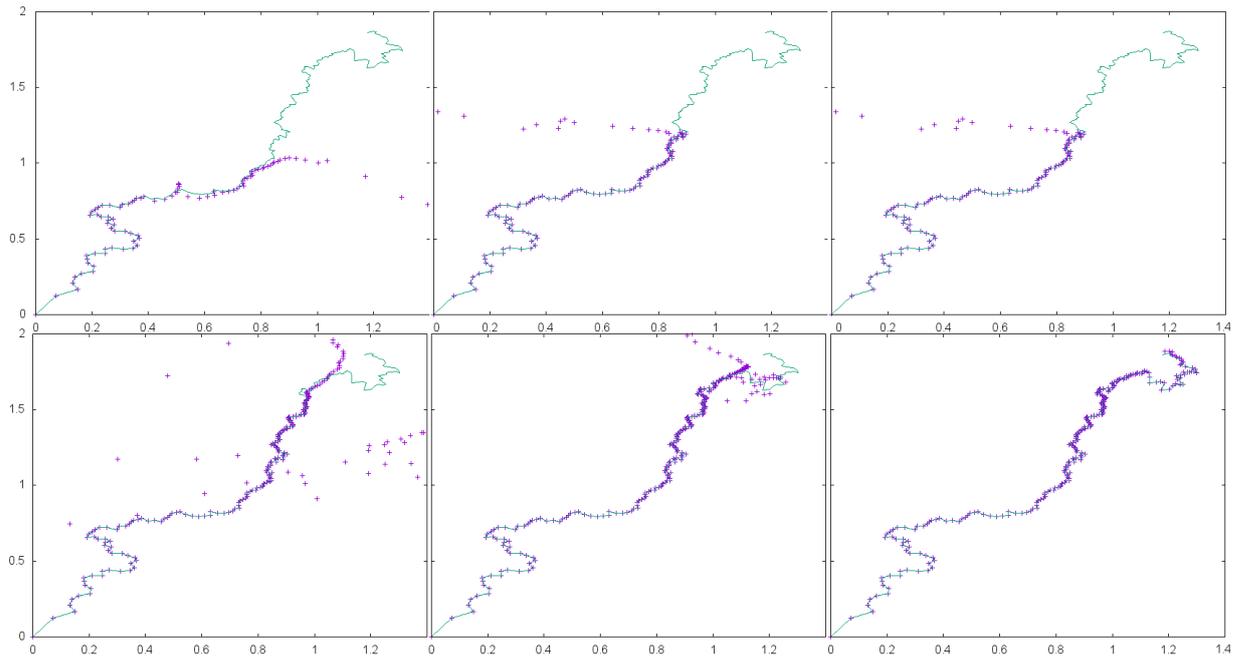
$$f_{t_i}(z_i + U_{t_i}) = l_{t_i}(z_i) = \gamma(t), \quad (4.9)$$

de modo que em 4.8, a seguinte relação pode ser obtida:

$$z_k = f_{t_{k-n}}(\phi + U(t)). \quad (4.10)$$

Esse resultado nos permite aplicar um número n de mapas com o algoritmo 2, até que os pontos se distanciem o suficiente do eixo real e o algoritmo 3 possa ser usado. Na Figura 5 usamos a função $U_t = B_t$, onde B_t é um *movimento browniano* (MB). Nota-se o aumento da precisão do algoritmo a medida que aumentamos o número de passos n onde usamos o método de discretização. A linha cheia no gráfico é o resultado utilizando somente o algoritmo 2, enquanto os pontos são obtidos usando o algoritmo 3 para os valores de n mencionados. Para uma amostra dos pontos de $U(t)$ dividimos o intervalo $[0, 1]$ em 257 pontos, cada um com separação $\Delta t_i = \frac{1}{256}$. Nesse caso o algoritmo 4 teve uma eficiência próxima do algoritmo 2, visto que n corresponde a aproximadamente 58% dos pontos utilizados da função $U(t)$, logo são da mesma ordem em termos de complexidade temporal (ainda que na prática ele obtenha mais rapidamente os resultados).

O valor $n = n_v$ (valor mínimo de n para o qual a série de Laurent para f_t se torna uma boa aproximação) depende essencialmente do comportamento da função $U(t)$. Cabe um estudo posterior para determinar região de precisão da série de Laurent em termos da distância em relação ao eixo x dos pontos de γ_t .

Figura 7 – Comparação dos algoritmos 2 e 3 conhecendo-se U_t .

Fonte: Elaborado pelo autor. Aplicação do algoritmo 3 para os seguintes valores de n : 30; 50; 70; 100; 120 e 150 respectivamente. O número total de pontos da curva é 257.

5 CONCLUSÃO

Neste trabalho foi possível uma introdução ao estudo de curvas geradas por modelos de rede na transição de fase de um sistema, pelo processo conhecido como evolução de Schramm-Loewner (SLE). Uma definição foi apresentada e algumas propriedades do SLE foram discutidas, de modo a classificar os tipos de curvas que se adequam a esse processo.

Apresentamos um método de resolução das equações de Loewner, utilizando um método de discretização que permite simular a evolução da função U_t enquanto a curva γ_t é eliminada do plano. A partir desta função foi possível obter o expoente crítico de um modelo de rede bastante comum em física: percolação crítica. Estimamos o valor do parâmetro k e obtivemos $k \approx 6$ que já havia sido obtido em outros trabalhos; e com isso foi possível estimar a dimensão fractal da curva do processo de exploração, mostrando uma das utilidades do SLE; que é verificar a dimensão fractal de sistemas em transição de fase.

Por fim, foi apresentada uma forma alternativa para resolver as equações de Loewner, que fornece boas perspectivas na construção de uma simulação mais eficiente que o método discreto, que tem complexidade $O(n^2)$ (onde n é o número de pontos da curva analisada); e portanto ineficiente dependendo da escala em que se quer analisar as curvas. O algoritmo ainda pode ser melhorado em trabalhos futuros, no sentido de funcionar em harmonia com o método discreto e garantir uma complexidade temporal menor que a quadrática, mesmo no pior caso.

REFERÊNCIAS

- [1] BROWN, J. W; CHURCHILL, R. V. *Complex variables and applications*. 6. ed. New York: McGraw-Hill, 1996.
- [2] STANLEY, H. E. *Introduction to phase transitions and critical phenomena*. Oxford: Oxford University Press, 1987.
- [3] STANLEY, H. E. Scaling, universality, and renormalization: Three pillars of modern critical phenomena. *Reviews of modern physics*, v. 71, n. 2, p. S358, 1999.
- [4] CARDY, J. SLE for theoretical physicists. *Annals Phys*, v. 318, n. 1, p. 81-118, 2005.
- [5] KENNEDY, T. Numerical computations for the Schramm-Loewner Evolution. *J. Statist. Phys.* 137, 839-856 (2009).
- [6] KENNEDY, T. A fast algorithm for simulating the chordal Schramm-Loewner evolution. *J. Statist. Phys.* 128, 1125–1137 (2007).
- [7] KENNEDY, T. Computing the Loewner driving process of random curves in the half plane. *J. Statist. Phys.* 131, 803-819 (2008).
- [8] LAWLER, G. F. *Conformally Invariant Processes in the Plane*. Mathematical Surveys and Monographs, vol. 114, American Mathematical Society, 2005.
- [9] KAGER, w.; NIENHUIS, B. A guide to stochastic löwner evolution and its applications. *Journals of statistical physics*, Springer, v. 115, n 5-6, p. 1149-1229, 2004.
- [10] BEFFARA, V. The dimension of the sle curves. *The Annals of Probability*, JSTOR, p. 1421-1452, 2008.
- [11] CARDY, J. Conformal invariance. In: *Phase transitions and critical phenomena*, Academic Press, London, 1987.
- [12] SCHRAMM, O. Scaling limits of loop-erased random walks and uniform spanning trees. *Israel J. Math.*, v 118, n. 1, p. 221-228, 2000.
- [13] LÖWNER, K. Untersuchungen über schlichte konforme Abbildungen des Einheitskreises. *I. Mathematische Annalen*, Springer, v. 89, p. 103–121, 1923.
- [14] MANDELBROT, B. B. **The Fractal Geometry of Nature**. New York: W. H Freeman, 1982.
- [15] LAWLER, G. F; SCHRAMM, O.; WERNER, W. Conformal invariance of planar loop-erased random walks and uniform spanning trees. In: *Selected Works of Oded Schramm*. [S.l.]: Springer, p. 931-987, 2011.
- [16] LAWLER, G. F; SCHRAMM, O.; WERNER, W. On the scaling limit of planar self-avoiding walk. In: *Fractal geometry and applications: a jubilee of Benoît Mandelbrot*, Part 2, p. 339-364, 2004.

- [17] SMIRNOV, S. Critical percolation in the plane: Conformal invariance, Cardy's formula, scaling limits. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, Elsevier, v. 333, n. 3, p. 239-444, 2001.
- [18] SAXTON, M. J. Single-particle tracking: the distribution of diffusion coefficients. *Biophysical journal*, v. 72, n. 4, p. 1744, 1997.
- [19] ZIFF, R. M. Hull-generating walks. *Physica D: Nonlinear Phenomena*, v. 38, n. 1, p. 377-383, 1989.
- [20] QIAN, Hong; SHEETZ, Michael P.; ELSON, Elliot L. Single particle tracking: Analysis of diffusion and flow in two-dimensional systems. *Biophysical journal*, v. 60, n. 4, p. 910, 1991.