



**Fundação Universidade Federal de Mato Grosso do Sul  
Instituto de Física  
Programa de Pós-Graduação em Ciência dos Materiais**



Christian Anton Philipps

**Código computacional do Modelo de Anderson de  
Dois Canais**

**Dissertação de Mestrado**

Campo Grande - MS

2018

Christian Anton Philipps

## **Código computacional do Modelo de Anderson de Dois Canais**

Dissertação apresentada ao Instituto de Física da Universidade Federal de Mato Grosso do Sul, sendo parte dos requisitos para a obtenção do título de Mestre em Ciência dos Materiais.

Universidade Federal de Mato Grosso do Sul - UFMS

Instituto de Física - INFI

Programa de Pós-Graduação

Orientador: Prof. Dr. João Vítor Batista Ferreira

Campo Grande - MS

Novembro/2018



**Ata de Defesa de Dissertação**  
**Programa de Pós-Graduação em Ciência dos Materiais**  
**Mestrado**

Aos vinte e seis dias do mês de outubro do ano de dois mil e dezoito, às nove horas, na Casa da Ciência/INFI, da Fundação Universidade Federal de Mato Grosso do Sul, reuniu-se a Banca Examinadora composta pelos membros: Joao Vitor Batista Ferreira (UFMS), Leandro Moreira de Campos Pinto (UFMS) e Marcos Serrou do Amaral (UFMS), sob a presidência do primeiro, para julgar o trabalho do aluno: **CHRISTIAN PHILIPPS**, CPF 71006859152, do Programa de Pós-Graduação em Ciência dos Materiais, Curso de Mestrado, da Fundação Universidade Federal de Mato Grosso do Sul, apresentado sob o título "**Código Computacional do Modelo de Anderson de Dois Canais.**" e orientação de Joao Vitor Batista Ferreira. O presidente da Banca Examinadora declarou abertos os trabalhos e agradeceu a presença de todos os Membros. A seguir, concedeu a palavra ao aluno que expôs sua Dissertação. Terminada a exposição, os senhores membros da Banca Examinadora iniciaram as arguições. Terminadas as arguições, o presidente da Banca Examinadora fez suas considerações. A seguir, a Banca Examinadora reuniu-se para avaliação, e após, emitiu Parecer expresso conforme segue:

**EXAMINADOR**

Dr. Joao Vitor Batista Ferreira (Interno)

Dr. Fabio Mallmann Zimmer (Interno) (Suplente)

Dr. Leandro Moreira de Campos Pinto (Externo)

Dr. Marcos Serrou do Amaral (Interno)

Dr. Valter Aragao do Nascimento (Interno) (Suplente)

ASSINATURA	AVALIAÇÃO
	Aprovado
//	//
	Aprovado
	APROVADO
//	//

**RESULTADO FINAL:**

Aprovação

Aprovação com revisão

Reprovação

**OBSERVAÇÕES:**

Em documento anexo a banca conferiu a nota média de 9,7 (nove vírgula sete) e conceito A.

Nada mais havendo a ser tratado, o Presidente declarou a sessão encerrada e agradeceu a todos pela presença.

**Assinaturas:**

Presidente da Banca Examinadora

Aluno

*Este trabalho é dedicado aos meus pais, por sempre terem me oferecido apoio.*

# Agradecimentos

Primeiro, eu gostaria de agradecer ao meu orientador, professor João Vítor Batista Ferreira, e ao nosso coordenador anterior do Mestrado em Ciência dos Materiais, professor Heberton Wender Luiz dos Santos, por sua paciência e ajuda com os documentos para receber duas vezes o visto de estudante para o Brasil. Também, nossa secretária do programa de pós-graduação, Flávia Baú. Sem estes esforços, eu não teria tido a possibilidade de participar nas disciplinas do mestrado e contribuir na pesquisa do professor João Vítor.

Segundo, um grande agradecimento pelo professor Marcos Serrou do Amaral por participar na minha banca de avaliação na minha qualificação e apresentação final. As sugestões me ajudaram a escrever a minha dissertação de forma mais compreensível e clara. Eu agradeço também a professora Isabela Porto Cavalcante pela participação na qualificação e por conversas muito esclarecedoras.

Mais uma vez, eu gostaria de mencionar o professor Marcos Serrou e o técnico Rodrigo Ferreira Gonçalves Fioravante por disponibilizar um computador da UFMS para obter os resultados computacionais.

Além disso, eu agradeço o professor Fábio Mallmann Zimmer e o mestre Paulo Guilherme Batista Shiota pelas discussões.

Acima de tudo, eu queria expressar a minha gratidão pela ajuda de escrever a dissertação em português ao meu professor João Vítor e à professora de português pelo programa Idiomas sem Fronteiras, Laura de Souza Martins Brasil Ovelar.

Finalmente, eu agradeço o meu amigo Reginaldo pela hospitalidade generosa.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

*It does not help that some politicians and journalists assume the public is interested only in those aspects of science that promise immediate practical benefits to technology or medicine. Some work on the most interesting problems of biological or physical science does have obvious practical value, but some does not, especially research that addresses problems lying at the boundaries of scientific knowledge. To earn society's support, we have to make true what we often claim: that today's basic scientific research is part of the culture of our times.*

*(Steven Weinberg)*

# Resumo

Atualmente, impurezas magnéticas isoladas em metal são estudadas devido ao Efeito Kondo supercompensado e suas aplicações em nanotecnologia. Elas são representadas por modelos de Anderson multicanal. Muitos destes modelos ainda não têm solução analítica ou são de difícil manipulação para determinação de propriedades físicas como calor específico, suscetibilidade magnética e densidade espectral. O Grupo de Renormalização Numérica (GRN) é uma técnica que pode determinar os autovalores dos hamiltonianos destes modelos usando processos iterativos. Mas é uma técnica de difícil implementação computacional devido ao grande grau da liberdade do sistema. Nesta dissertação, um programa existente em linguagem não documentado, o qual usa o pacote *givens* para a diagonalização, é analisado e comparado em termos de tempo de execução com um novo programa *C++* orientado a objetos que usa a biblioteca *Eigen*. O programa novo foi escrito de maneira modular e documentado em detalhes para fazer adaptações futuras simples. Cálculos analíticos foram feitos para os primeiros dois passos do processo iterativo do GRN como validação. Além disso, pretendemos obter uma licença de produto tecnológico do Instituto de Física da Universidade Federal de Mato Grosso do Sul.

**Palavras-chave:** Modelo de Anderson Multicanal, Diagonalização Numérica Computacional, Programação Orientada a Objeto.

# Abstract

Isolated magnetic impurities in metals are recently studied due to their overcompensated effects and their applications in nanotechnology. They are represented by multichannel Anderson models. Many of these models still don't have an analytic solution or are difficult to manipulate to determine physical properties as the specific heat, magnetic susceptibility and spectral density. The numerical renormalization group (NRG) is a method which can obtain the eigenvalues of hamiltonians of these models. However, the technical implementation is a complex task. In this thesis, an existing non-documented program which uses the *givens* package for diagonalization is analyzed and compared in terms of run time to a new object-orientated C++ program which uses the *Eigen* library. The new program was modular written and documented in detail to make future adaptation simple. Analytic calculations were done for the first two steps of the iterative process of the NRG as validation. Furthermore, it is aimed to obtain a license as technical product of the Physical Institute of the Federal University of Mato Grosso do Sul.

**Keywords:** Multichannel Anderson Model, Numerical Diagonalization, Object-Orientated Programming.



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>2</b>	<b>MOTIVAÇÃO - MODELO DE ANDERSON DE DOIS CANAIS</b>	<b>13</b>
2.1	Propriedades físicas	13
2.1.1	Densidade espectral	15
2.1.2	Suscetibilidade magnética e calor específico	16
2.2	Diagonalização do hamiltoniano	17
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>19</b>
3.1	Efeito Kondo	19
3.2	Modelo de Anderson de Dois Canais	21
3.3	Grupo de Renormalização Numérica	26
3.3.1	Discretização logarítmica da banda de condução.	27
3.3.2	Diagonalização iterativa	31
3.3.3	Os vetores da base	32
3.3.4	Os elementos de matriz de $H_N$	34
3.3.5	Invariantes	35
3.3.6	Hamiltoniano da iteração $N = -1$	37
<b>4</b>	<b>OBJETIVOS</b>	<b>38</b>
<b>5</b>	<b>METODOLOGIA</b>	<b>39</b>
5.1	Documentação do código computacional com o programa Doxygen	39
5.2	Implementação computacional	42
<b>6</b>	<b>RESULTADOS</b>	<b>43</b>
6.1	Análise do programa do Modelo de Anderson de Dois Canais	43
6.2	<i>Two Channel Single Impurity Anderson Model</i>	47
6.2.1	Discretização logarítmica da banda de condução	47
6.2.2	Cálculo analítico e computacional da iteração $N = 0$	52
6.2.3	Cálculo analítico e computacional da iteração $N = 1$	61
6.2.4	Programa computacional em C++	66

---

6.3	Início do processo de licenciamento . . . . .	80
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>81</b>
<b>A</b>	<b>PROCEDIMENTO PARA GERAR NOVOS ESTADOS . . . . .</b>	<b>83</b>
<b>B</b>	<b>ELEMENTOS DE MATRIZ E INVARIANTES . . . . .</b>	<b>84</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>89</b>

# 1 Introdução

Com o avanço da nanotecnologia tornou-se exequível a fabricação de dispositivos utilizando pontos quânticos - 'Quantum Dots'. Trata-se da construção de objetos esféricos na escala de  $10^{-9}$ m em metais ou semicondutores. O tamanho diminuto faz surgir propriedades físicas que não existem nestes materiais em escala macroscópica. Além da óbvia utilidade na eletrônica usual como transistores e leds, esses dispositivos são estudados em fotônica e spintrônica. Além de outras técnicas experimentais, a fabricação e caracterização de pontos quânticos podem ser realizadas usando a Microscopia de Tunelamento por Varredura - 'Scanning Tunneling Microscopy (STM)'. Esta microscopia tornou-se tão eficiente que nos dias atuais é possível a observação de um átomo de Cobalto na superfície de Cobre. Por ser de elemento químico diferente do substrato, o átomo de Cobalto é denominado de impureza. No caso de impurezas magnéticas em metais, a perfeita compreensão dos dados obtidos pelo STM baseia-se no conhecimento da densidade espectral da impureza sob Efeito Kondo. Este fenômeno é a blindagem magnética que um átomo sofre pelos elétrons de condução da rede cristalina do material do substrato ou da ponta de prova do STM. O Efeito Kondo é observado em 'Quantum Dots' [1, 2, 3] e moléculas [4]. Dentre as técnicas experimentais modernas em que o Efeito Kondo desempenha papel importante, a Microscopia de Tunelamento por Varredura destaca-se pela visualização da superfície de materiais com resolução no espaço e na energia em escalas atômicas [5, 6]. Estudos têm sido realizados para observar a dependência da ressonância Kondo na ponta de prova do STM em função da localização espacial entre ela e a impureza. STM é uma ferramenta experimental importante na área de Ciência dos Materiais e é necessário existir ferramenta computacional apropriada para análise dos dados quando se trata da observação do fenômeno Kondo.

O Efeito Kondo pode ser estudado pelo Modelo de Anderson de Dois Canais [7]. O hamiltoniano desse modelo é solucionado numericamente utilizando a técnica

conhecida como Grupo de Renormalização Numérica, que deve ser implementada através de programa computacional apropriado. Apesar de ter sido criada nos anos de 1970, somente recentemente diferentes pesquisadores têm se especializado na utilização dessa técnica, devido a dificuldade de sua compreensão e implementação. O Dr. João Vítor Batista Ferreira construiu um código eficaz em linguagem C++ para essa tarefa quando realizou sua pós-graduação no grupo de Física Teórica do IFSC-USP na década de 1990. Um dos poucos grupos a estudar este tema no Brasil. Eficaz mas não eficiente, esse programa foi concebido sem os modernos recursos das linguagens de programação e sem a devida documentação. Um código computacional construído com recursos modernos e devidamente documentado permite a sua compreensão de maneira mais simples, seu aprimoramento de maneira mais fácil e sua adaptação para determinação de propriedades físicas diferentes daquelas para que ele fora projetado inicialmente.

Esta dissertação descreve a implementação de um código computacional do Modelo de Anderson de Dois Canais. De maneira sucinta são abordados temas como impurezas magnéticas em metais, Efeito Kondo, Modelo de Anderson de Dois Canais e as propriedades físicas que podem ser obtidas deste modelo. Com um pouco mais de detalhe é descrita a técnica do Grupo de Renormalização Numérica (GRN) para diagonalização do hamiltoniano desse modelo. O foco deste trabalho é a análise do programa atual utilizado para implementar essa técnica e a escrita de um novo código computacional.

## 2 Motivação - Modelo de Anderson de Dois Canais

Na seção 2.1 é apresentado, em caráter apenas informativo, a importância do estudo de impurezas magnéticas em metais e suas propriedades físicas. Na seção 2.2 é explicado a diagonalização do Modelo de Anderson de Dois Canais.

### 2.1 Propriedades físicas

Os avanços tecnológicos em Ciência dos Materiais nas últimas décadas resultaram em diferentes técnicas de sondagem de varredura [8]. No entanto, a maioria delas se baseia nos mesmos princípios e na mesma técnica: Microscopia de Tunelamento por Varredura - 'Scanning Tunneling Microscopy (STM)'. Um microscópio de tunelamento de varredura pode ser usado para medir as propriedades eletrostáticas de uma amostra. Uma ponta condutora, que idealmente termina em um único átomo, é colocada em alguns angstroms acima da superfície da amostra. A distância é medida monitorando a corrente de tunelamento através da ponta de prova. Ao aplicar uma tensão de polarização entre a amostra e a ponta, é possível inferir a densidade local de estados (LDOS) dos estados ocupados e desocupados da amostra. A resolução da equação de Schroedinger para elétrons em estados de superfície resulta em funções de onda que se deterioram exponencialmente no vácuo. Se a ponta STM é colocada próxima o suficiente da superfície, há uma probabilidade finita para os elétrons nos estados de superfície se tunelarem para a ponta de prova. Existem vários parâmetros que determinam a probabilidade de tunelamento. Mais importante, os elétrons só podem ocupar os estados quânticos desocupados. Se os estados quânticos da ponta e da amostra estiverem ocupados até um nível de Fermi igual a  $E_F$ , nenhuma corrente pode ser observada. Aplicada uma tensão de polarização, os níveis de Fermi são deslocados energeticamente, deixando estados

vazios serem ocupados na amostra ou na ponta. Portanto, o sinal da corrente elétrica detectada na ponta de prova é determinado pelo sinal do campo potencial aplicado. Em segundo lugar, a função de onda de vácuo dos elétrons decai exponencialmente com o aumento da distância até a amostra. A probabilidade de tunelamento reduz cerca de uma ordem de magnitude para cada angstrom de distância. Portanto, a variação de altura no regime de sub-angstrom é detectável. Além disso, a corrente depende da densidade local de estados na amostra. Interpretando de acordo com a mecânica quântica, o STM pode ser descrito por um hamiltoniano constituído por três partes: o hamiltoniano não perturbado da amostra  $H_{\text{am}}$ , o hamiltoniano não perturbado da ponta  $H_{\text{po}}$  e um termo de transferência  $H_t$ , que representa o acoplamento e atua como uma pequena perturbação. Somando, o hamiltoniano completo é  $H = H_{\text{am}} + H_{\text{po}} + H_t$ .

Usando a teoria de perturbação de primeira ordem e a Regra de Ouro de Fermi, pode-se obter a corrente de tunelamento, que é proporcional a LDOS no nível de Fermi e função da altura da ponta de prova. Pequenas variações na corrente elétrica na ponta de prova são detectadas e registradas. O modo de experimento pode ser mantendo a altura da ponta de prova constante, para amostras pequenas e com topografia suficientemente plana. O outro modo de operação é de corrente tunelante constante, onde a altura é ajustada via retro-alimentação da corrente. Nos dois casos pode ocorrer a influência do Efeito Kondo na corrente de tunelamento. Essa influência pode ser analisada pela densidade espectral.

Além do STM, outras técnicas experimentais permitem acesso aos estados quânticos de elétrons de condução que sofrem influência do Efeito Kondo. Isso demonstra a utilidade da diagonalização de hamiltonianos de modelos de impureza magnética em metais. Os autovalores de energia permitem a descrição de diferentes propriedades físicas como calor específico e suscetibilidade magnética. A seguir, relatam-se brevemente algumas propriedades que podem ser determinadas após a obtenção das energias dos estados quânticos.

### 2.1.1 Densidade espectral

A densidade espectral é uma propriedade física que pode ser obtida experimentalmente por técnicas como Espectroscopia de Fotoemissão de Raios-X (XPS) [9]. Raios-X caracterizados por fótons com energia  $\hbar\omega$  incidem na impureza arrancando elétrons que são detectados e transformados em corrente elétrica. Em linguagem de segunda quantização, esse processo pode ser descrito pelo Hamiltoniano  $H_t = \sum_{\vec{k}} \langle \vec{k} | W | d \rangle c_k^\dagger c_d$  sendo  $c_d$  o operador que destrói um elétron no estado quântico da impureza,  $c_k^\dagger$  o operador que cria um elétron no estado de condução eletrônica e  $W$  o elemento de matriz de radiação eletromagnética de dipolo, conhecido como potencial vetor no gauge de Coulomb. Para calcular a probabilidade de transição entre o estado inicial  $i$  e possíveis estados finais  $f$  em função da energia utiliza-se a Regra de Ouro de Fermi resultando na densidade espectral [10, 11]. Em anos recentes, o avanço do STM permitiu a análise experimental de impurezas magnéticas mesmo que ela esteja em algumas camadas cristalinas de profundidade. Por exemplo, Pruser e colaboradores mediram o Efeito Kondo em torno de uma impureza no 'bulk' tanto em função da profundidade quanto em função da distância lateral à posição da impureza [6]. Essas informações permitem inferir a densidade de estados eletrônicos na região em volta da impureza. A determinação teórica da densidade espectral é obtida com os autovalores de energia do hamiltoniano do modelo.

### 2.1.2 Suscetibilidade magnética e calor específico

A obtenção de dados experimentais de impurezas magnéticas em metais em baixas temperaturas está em evidência na literatura [12]. A determinação teórica destas grandezas utilizando modelos tipo Anderson também é bem estabelecida. A suscetibilidade magnética  $\chi$  em função da temperatura  $T$  pode ser calculada da seguinte forma

$$\frac{k_B T \chi(T)}{(g\mu_B)^2} = [\langle S_z^2 \rangle - \langle S_z \rangle^2] \quad (2.1)$$

e o calor específico  $C$ , também em função da temperatura  $T$ , assim,

$$\frac{C(T)}{k_B} = \beta^2 [\langle E^2 \rangle - \langle E \rangle^2], \quad (2.2)$$

$k_B$  a constante de Boltzmann,  $g$  o fator giromagnético, sendo  $\mu_B$  o magneton de Bohr,  $S_z$  a componente  $z$  do spin,  $\beta$  a temperatura termodinâmica inversa,  $E$  autovalor de energia e  $\langle \dots \rangle$  valor esperado obtido pelo traço do hamiltoniano diagonalizado.

Todas essas propriedades dependem da determinação dos autovalores do hamiltoniano do modelo utilizado.



## 2.2 Diagonalização do hamiltoniano

A diagonalização do hamiltoniano pelo Grupo de Renormalização Numérica se baseia na discretização logarítmica da banda de condução [13, 14]. Mesmo sendo mais eficiente que a discretização linear, ainda assim o procedimento numérico introduz oscilações espúrias em  $\chi(T)$  e  $C(T)$  que se evidenciam à medida que a representação da banda de condução se afasta do contínuo. Esse problema é amenizado pelo método que é conhecido como Intercalamto [15, 16]. A discretização logarítmica e a determinação das propriedades físicas é realizada várias vezes, cada vez usando diferentes parâmetros para a discretização. No final realiza-se média dos resultados anteriores. As oscilações são sensivelmente reduzidas com esse processo, veja a figura 1.

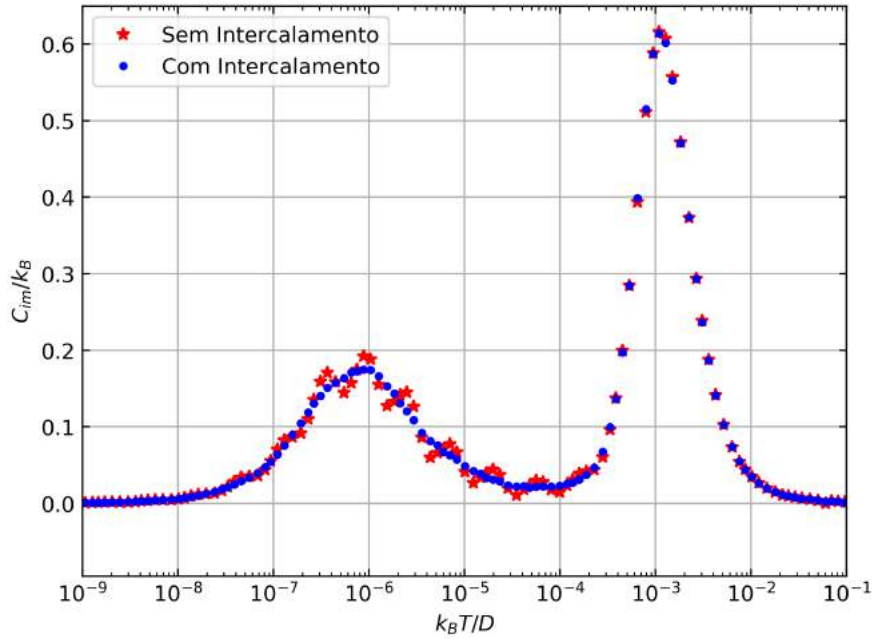


Figura 1 – Duas curvas do calor específico da impureza  $C_{im}$  dividido pela constante de Boltzmann  $k_B$  em função da temperatura  $T$  multiplicado pela constante de Boltzmann e dividido pela metade da largura da banda de condução. Na curva azul o procedimento de Intercalamto com 8 curvas individuais é feito. A vermelha mostra uma curva individual. Os mesmos parâmetros físicos  $V_1 = 0.005$ ,  $V_2 = 0.01$  e  $\Delta = 0.002$  são usados 3.2.

O Intercalamento economiza memória computacional mas exige que o programa seja executado várias vezes, aumentando enormemente o tempo. Espera-se que a reescrita do código permita melhor uso da memória computacional e a diminuição no tempo de processamento total. O próximo capítulo descreve resumidamente a teoria do Hamiltoniano do Modelo de Anderson de Dois Canais (MADC), usado pelo pesquisador J. V. B. Ferreira, que foi desenvolvido em 1997 pelos pesquisadores D. L. Cox e V. L. Líbero. Esse modelo foi criado para possibilitar o aparecimento do comportamento não líquido de Fermi, algo que não era possível usando os hamiltonianos tradicionais de Kondo e Anderson [17, 18].

## 3 Fundamentação teórica

### 3.1 Efeito Kondo

O Efeito Kondo é um fenômeno típico em ligas metálicas. É o acoplamento antiferromagnético entre um orbital atômico  $d$  ou  $f$  localizado com os elétrons de uma banda de condução. Em temperatura suficientemente baixa este acoplamento torna-se tão intenso que pode anular (blindar) completamente o caráter magnético deste orbital atômico. A intensidade desta blindagem depende basicamente de dois fatores: da agitação térmica dos elétrons de condução e de campos externos, como por exemplo, campos da rede cristalina. Mesmo em temperatura muito baixa, este efeito pode nunca acontecer se o campo externo fizer com que os elétrons de condução comecem a competir entre si durante esta blindagem magnética.

Inúmeros artigos científicos têm sido publicados sobre esse fenômeno desde 1930, quando houve a observação experimental indireta desse efeito na medição de resistência elétrica de diferentes metais [19]. Foi o desenvolvimento das técnicas de resfriamento que possibilitou essa descoberta, pois o Efeito Kondo, em geral, é sobrepujado por outros fenômenos em temperaturas mais altas, como por exemplo a vibração da rede cristalina [20] - podemos considerar que o Efeito Kondo é perceptível para temperaturas abaixo de 100K. Apesar de sua observação, o Efeito Kondo só foi compreendido teoricamente na década de 1960 com os trabalhos de J. Kondo e P. Anderson [21, 22].

Ambos os artigos propunham como modelo estados quânticos atômicos localizados interagindo com elétrons de condução, mas através de mecanismos de interação aparentemente diferentes. Esses modelos ficaram conhecidos como Modelo de Kondo e Modelo de Anderson. Demorou pouco tempo para se verificar que o modelo proposto por Kondo era apenas um caso particular daquele publicado por Anderson [23]. Esses trabalhos esclareceram que um átomo de Fe em um bloco

de Au provoca este acoplamento magnético entre ele e os elétrons de condução do material. Neste caso, o átomo de Fe é denominado impureza magnética, pois é um elemento químico estranho na rede cristalina do material hospedeiro. Os dois modelos enfatizam que o Efeito Kondo é um fenômeno de muitos corpos, e que a diagonalização dos respectivos é uma tarefa árdua. Somente na década de 1980 a solução analítica destes modelos foi alcançada [24], mas a determinação de propriedades físicas através das soluções encontradas muitas vezes não é fácil. Além disso, verificou-se que estes modelos não eram capazes de descrever corretamente o Efeito Kondo em novas ligas metálicas que estavam sendo criadas em laboratório.

A interação entre o orbital localizado e os estados de condução de um metal resulta em fenômenos intrigantes bastante estudados, como valência flutuante, compostos de férmions pesados, supercondutividade e outros mais. Atualmente, novos modelos de impureza magnética em metal têm sido propostos para a descrição mais realística de ligas metálicas [17].

## 3.2 Modelo de Anderson de Dois Canais

O modelo proposto por Kondo e o modelo proposto por Anderson, entre outros, passaram a ser conhecidos como modelos de impureza magnética isolada de um canal, após a publicação do artigo de Ph. Nozières e A. Blandin [25] em 1980. Estes pesquisadores argumentaram que para tornar esses modelos mais realistas é necessário considerar a degenerescência orbital dos estados atômicos localizados, interação spin-órbita e campo da rede cristalina. Usando teoria de grupo e outras considerações, Nozières e Blandin mostraram que o resultado final de todas essas interações era a possibilidade de formação de mais de um acoplamento (ressonância) entre a impureza e os elétrons da banda de condução do metal. Essas ressonâncias receberam o nome de canais: os elétrons de condução ganharam mais um número quântico que designa com qual canal ele vai interagir com a impureza. A consequência disso é o fenômeno de frustração quando os elétrons de dois canais diferentes passam a se prejudicar na tarefa de blindar magneticamente a impureza. Nos modelos tradicionais, um canal, isto nunca pode acontecer [26, 27].

Ao contrário dos modelos citados anteriormente, o Modelo de Anderson de Dois Canais usado nesta dissertação não é bastante conhecido. É um modelo mais recente e foi criado pelo grupo de pesquisa teórica no Instituto de Física de São Carlos da Universidade de São Paulo (IFSC-USP).

O Modelo de Anderson de Dois Canais ('Two Channel Single Impurity Anderson Model - TCSIAM') descreve uma impureza magnética em um metal não magnético, e a impureza é um átomo com camada  $d$  ou  $f$  semi-preenchida. Ele foi proposto pelos pesquisadores D. L. Cox e V. L. Líbero com a proposta de representar ligas metálicas com comportamento não usual [28, 29]. O TCSIAM está ilustrado na figura 2. O metal é representado pela sua banda de condução contínua. Essa é isotrópica, semi-preenchida e tem uma largura total de  $2D$  ( $2D \sim 10\text{eV}$ ). O parâmetro  $D$  é usado como unidade de energia e o nível Fermi da banda como sendo o nível zero de energia. A impureza é representada apenas pelo seu orbital de valência: estado fundamental e dois estados excitados. O estado fundamental  $|m, \sigma\rangle$  é um dubleto que tem  $m$  elétrons, spin total  $\sigma = \pm 1/2$  e energia  $E_0$ . Os estados excitados  $|m + 1, \alpha\rangle$  e  $|m - 1, \alpha\rangle$  são dois dubletos que têm  $m \pm 1$  elétrons, spin

total nulo, energia  $E_x$  e canal  $\alpha = 1$  ou  $2$ . O termo de acoplamento (hibridização) entre a banda de condução e a impureza é  $V_\alpha$ , que não depende da temperatura. A banda de condução funciona como um reservatório térmico com o qual a impureza está em contato. Então, apesar de  $V_\alpha$  não depender da temperatura, a blindagem magnética que o conjunto de elétrons exerce na impureza acaba dependendo da temperatura.

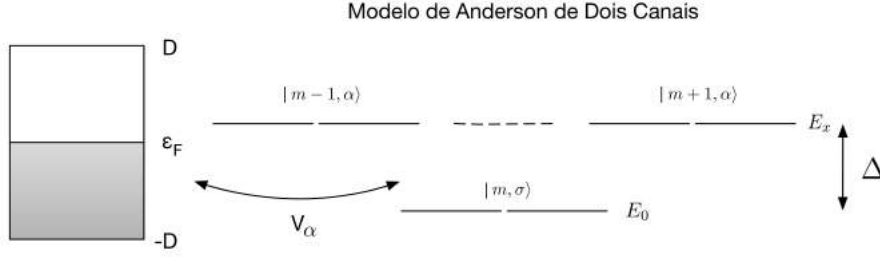


Figura 2 – Diagrama do Modelo de Anderson de Dois Canais. Interação entre a banda de condução semi-preenchida, largura  $2D$ , e nível de Fermi no meio  $\epsilon_F$  via  $V_\alpha$  com a impureza que consiste nos três estados: estado fundamental  $|m, \sigma\rangle$  com energia  $E_0$ , estados excitados  $|m - 1, \alpha\rangle$  e  $|m + 1, \alpha\rangle$  com energia  $E_x$ . A variável  $\Delta$  descreve a diferença entre as duas energias iniciais do TCSIAM: energia excitada  $E_x$  e fundamental  $E_0$ .

Em linguagem de segunda quantização, o TCSIAM é descrito por um hamiltoniano de três componentes:

$$H = H_b + H_{im} + H_{hi} \quad (3.1)$$

$$H_b = \sum_{\vec{k}, \sigma, \alpha} \epsilon_{\vec{k}} c_{\vec{k}, \sigma, \alpha}^\dagger c_{\vec{k}, \sigma, \alpha}, \quad (3.2)$$

$$H_{im} = E_x \sum_{\alpha} |m - 1, \alpha\rangle \langle m - 1, \alpha| + |m + 1, \alpha\rangle \langle m + 1, \alpha| + E_0 \sum_{\sigma} |m, \sigma\rangle \langle m, \sigma|, \quad (3.3)$$

$$H_{hi} = \sum_{\vec{k}, \sigma, \alpha} V_\alpha (f_{-1, \sigma, \alpha}^\dagger c_{\vec{k}, \sigma, \alpha} + f_{-1, \sigma, \alpha} c_{\vec{k}, \sigma, \alpha}^\dagger). \quad (3.4)$$

$H_b$ ,  $H_{im}$  e  $H_{hi}$  representam a banda de condução, impureza e o mecanismo de hibridização respectivamente. Das Referências [28, 29] seguem as definições:

- $c_{\vec{k},\sigma,\alpha}^\dagger$  cria um elétron livre na banda de condução do metal com momentum  $\vec{k}$ , spin  $\sigma = \pm 1/2$ , canal  $\alpha = 1$  ou  $2$  e energia  $\epsilon_k$ , como a banda é isotrópica  $\epsilon_{\vec{k}} = \epsilon_k$ ;
- $|m, \sigma\rangle$  é o estado fundamental da impureza denominado de *estado de spin*, tem  $m$  elétrons, spin total  $\sigma = \pm 1/2$  e energia  $E_0$ ;
- $|m \pm 1, \alpha\rangle$  é o estado excitado da impureza que tem  $m \pm 1$  elétrons, spin total zero e energia  $E_x$ . Está acoplado com o canal  $\alpha = 1$  ou  $2$ . A combinação  $|m + 1, \alpha\rangle$  e  $|m - 1, -\alpha\rangle$  forma aquilo que denominamos *estado de domínio do canal  $\alpha$*  ( $\alpha = 1 \rightarrow -\alpha = 2$  e  $\alpha = 2 \rightarrow -\alpha = 1$ );
- A diferença de energia entres estes estados é denominada de delta e definido por  $\Delta = E_x - E_0$ ;
- $f_{-1,\sigma,\alpha}^\dagger$  e  $f_{-1,\sigma,\alpha}$  são os *operadores de hibridização*, sendo responsáveis pela mudança de estado da impureza. Os dois operadores são definidos pelas seguintes expressões

$$f_{-1,\sigma,\alpha}^\dagger = |m, \sigma\rangle \langle m - 1, -\alpha| + (2\sigma) |m + 1, \alpha\rangle \langle m, -\sigma|; \quad (3.5)$$

e

$$f_{-1,\sigma,\alpha} = |m - 1, -\alpha\rangle \langle m, \sigma| + (2\sigma) |m, -\sigma\rangle \langle m + 1, \alpha|; \quad (3.6)$$

Como os operadores de hibridização representam a interação entre a banda de condução e a impureza é mostrado na figura 3.

- $V_\alpha$  é o elemento da matriz de acoplamento entre o estado de spin e o estado de canal  $\alpha$  da impureza.  $V_1$  e  $V_2$  podem ser iguais ou diferentes;
- Quando  $V_1 \neq V_2$  o sistema está no *regime líquido de Fermi*;
- Quando  $V_1 = V_2$  o sistema está no *não líquido de Fermi*.

A interação entre a banda e a impureza representada pelas energias de hibridização  $V_1$  e  $V_2$  indica a propensão dos elétrons se acoplarem através do canal 1 e 2, respectivamente, e podem assumir ou não os mesmos valores. Quando  $V_1 \neq V_2$  o spin da impureza é completamente blindado pelo spin dos elétrons de

condução. Quando  $V_1 = V_2$  ocorre o Efeito Kondo supercompensado, a blindagem acaba resultando em spin líquido diferente de zero pela ação dos elétrons dos dois canais. As grandezas  $V_1$ ,  $V_2$  e  $\Delta$  são os parâmetros físicos do modelo. Ajustando seus valores pode representar o sistema físico em outros regimes e obter diferentes comportamentos das propriedades físicas.

Para ilustrar o significado dos operadores de hibridização,  $f_{-1,\sigma,\alpha}^\dagger c_{\vec{k},\sigma,\alpha}^-$  e  $f_{-1,\sigma,\alpha} c_{\vec{k},\sigma,\alpha}^\dagger$ , veja na equação (3.4) e figura 3. Eles entram no Hamiltoniano do Modelo de Anderson de Dois Canais em combinação com os operadores da banda de condução. O primeiro termo da hibridização  $H_{hi}$  permite dois tipos de interação e são eles:

$$f_{-1,\sigma,\alpha}^\dagger c_{\vec{k},\sigma,\alpha}^- = \left( |m, \sigma\rangle \langle m-1, -\alpha| + (2\sigma) |m+1, \alpha\rangle \langle m, -\sigma| \right) c_{\vec{k},\sigma,\alpha}^- . \quad (3.7)$$

O primeiro termo desses dois operadores descreve o processo de: destruir um elétron na banda de condução com momentum  $\vec{k}$ , spin  $\sigma$  e canal  $\alpha$ , mudando o estado excitado da impureza com  $m-1$  elétrons e canal  $-\alpha$  para o estado fundamental da impureza com  $m$  elétrons e spin  $\sigma$ .

O segundo termo é a interação de: destruir um elétron na banda de condução com momentum  $\vec{k}$ , spin  $\sigma$  e canal  $\alpha$ , mudando o estado fundamental da impureza com  $m$  elétrons e spin  $-\sigma$  para o estado excitado da impureza com  $m+1$  elétrons e canal  $\alpha$ .



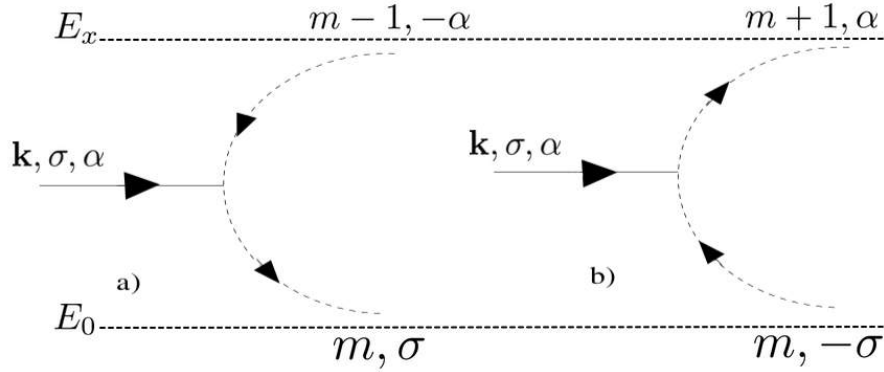


Figura 3 – Duas de quatro interações do Modelo de Anderson de Dois Canais entre a banda de condução e a impureza. Processo a) destruir um elétron na banda de condução com momentum  $\vec{k}$ , spin  $\sigma$  e canal  $\alpha$ , mudando o estado excitado da impureza com  $m-1$  elétrons e canal  $-\alpha$  para o estado fundamental da impureza com  $m$  elétrons e spin  $\sigma$ , e processo b) destruir um elétron na banda de condução com momentum  $\vec{k}$ , spin  $\sigma$  e canal  $\alpha$ , mudando o estado fundamental da impureza com  $m$  elétrons e spin  $-\sigma$  para o estado excitado da impureza com  $m+1$  elétrons e canal  $\alpha$ .

Adaptação da fonte: FERREIRA, J. V. B. Modelo de Anderson de dois canais. Tese (Doutorado em Física Básica) — Universidade de São Paulo, São Carlos, Instituto de Física de São Carlos, São Paulo, 2000.

Uma interação direta entre o estado fundamental e excitado da impureza não é incluído no Modelo de Anderson de Dois Canais. O segundo termo da hibridização,  $f_{-1,\sigma,\alpha}^\dagger c_{\vec{k},\sigma,\alpha}^\dagger$ , permite os dois processos inversos. Por isso, o modelo tem quatro processos de interação.

O Grupo de Renormalização Numérica (GRN) diagonaliza este hamiltoniano numericamente e será apresentado a seguir.

### 3.3 Grupo de Renormalização Numérica

O Grupo de Renormalização Numérica (GRN) foi criado na década de 1970 por K. G. Wilson [30, 31]. Usando idéias de renormalização da teoria de campo, Wilson desenvolveu esta técnica para solucionar o hamiltoniano do Modelo de Kondo, utilizando cálculos numéricos na diagonalização de matrizes [13]. Anos depois ele ganharia o Prêmio Nobel de Física por causa disso. Usando o mesmo procedimento, ele próprio, Krishna-Murthy e J. W. Wilkins investigaram o Hamiltoniano de Anderson tradicional [14]. A aplicação desta técnica em modelos multicanais foi tema de pesquisa do grupo de Física Teórica em USP-São Carlos. [32, 33]. Um importante avanço foi realizado no acréscimo de mais um parâmetro de discretização da banda de condução [34]. Este novo parâmetro mostrou-se útil também para o cálculo de propriedades termodinâmicas como suscetibilidade magnética e calor específico [15, 16]. A Ref. [35] é um artigo de revisão sobre o GRN, publicado em 2008.

Diagonalizar o Hamiltoniano de Anderson diretamente é um processo complexo. Para facilitar a tarefa de diagonalização, é aplicada ao Hamiltoniano original uma transformação, denominada transformação do GRN,  $\mathfrak{S}$ , de tal modo que ele se torne uma sequência de Hamiltonianos interligados  $H = H_0 + H_1 + H_2 + \dots$ , e que seja criada uma relação de recorrência, que permite determinar os elementos de matriz  $H_N$  conhecendo os autovalores e autovetores da iteração anterior,  $H_N = \mathfrak{S}[H_{N-1}]$ . A primeira etapa dessa transformação é a discretização da banda de condução.

### 3.3.1 Discretização logarítmica da banda de condução.

Nos parágrafos seguintes será apresentada resumidamente a transformação do hamiltoniano do TCSIAM para a forma que é implementada nos códigos computacionais. Este procedimento matemático é muito laborioso e envolve muitos detalhes. São usados conceitos como Transformada Discreta de Fourier e Método de Ortogonalização de Gram-Schmidt. Como o objetivo desta dissertação é a análise e construção do código computacional que diagonaliza o hamiltoniano já transformado do TCSIAM, estes procedimentos analíticos são apresentados de maneira apenas descritiva.

A banda de condução é um contínuo e por isso tem infinitos valores de energia para representar os estados quânticos de condução. Para haver solução numérica é necessário encontrar um número máximo de valores de energia que sejam representativos da Física do fenômeno. Essa discretização deve ser invariante mediante uma mudança de escala nas energias próximas ao nível de Fermi, assim como ocorre na banda de condução contínua. Para isso, a discretização feita é logarítmica. No GRN tradicional utilizado por Wilson [13] é usado o parâmetro de discretização  $\lambda$  (um número inteiro positivo maior que a unidade), que também é utilizado no TCSIAM, juntamente com o parâmetro  $z$  (um número racional não negativo), que faz parte do GRN generalizado utilizado neste trabalho. Sendo assim, o GRN de Wilson é um caso particular quando  $z = 1$ .

A discretização é dividida em duas etapas: a primeira é responsável por ordenar o acoplamento dos elétrons de condução do metal com a impureza, da maior escala de energia para a menor; a segunda etapa é responsável por introduzir uma base de operadores de Lanczos  $\{f_n\}$  [36], que possuem a característica de se acoplarem apenas com os operadores vizinhos, ou seja,  $f_n$  se acopla com  $f_{n-1}$  e com  $f_{n+1}$ . Como  $H_{-1}$  é o único termo referente a impureza, somente  $f_0$  se acopla com a mesma.

Na primeira etapa da discretização a energia  $\epsilon_k$  é expandida em torno da energia de Fermi ( $-D < \epsilon_k < D$ ), em função do módulo do momentum, pois considera-se a banda de condução do metal isotrópica e que só depende do módulo do momentum. Os operadores  $c_{\vec{k},\sigma,\alpha}$  são substituídos pelos operadores  $c_{\epsilon,\sigma,\alpha}$  (que

obedecem à relação de anti-comutação  $\{c_{\epsilon,\sigma,\alpha}^\dagger, c_{\epsilon',\sigma,\alpha}\} = \delta(\epsilon - \epsilon')/\sqrt{\rho(\epsilon)}$ , onde  $\sqrt{\rho(\epsilon)}$  é a densidade de estados [37, 38] e é usada a variável adimensional  $\epsilon \equiv \epsilon_k/D$ . Wilson demonstrou que uma boa aproximação para a densidade de estados é a expressão  $\rho_0(\epsilon) = 1/(2D)$ .

Na segunda etapa da discretização a banda de condução é representada por uma matriz na forma tridiagonal, para isso é construída uma nova base  $\{f_{n,\sigma,\alpha}^\dagger\}$  [39] e o resultado é o seguinte:

$$\frac{H_b}{D} = \sum_{k,\sigma,\alpha} \frac{\epsilon_k}{D} c_{k,\sigma,\alpha}^\dagger c_{k,\sigma,\alpha} = \sum_{\alpha,\sigma,n=0}^{\infty} \epsilon_n^z \left( f_{n,\sigma,\alpha}^\dagger f_{n+1,\sigma,\alpha} + f_{n+1,\sigma,\alpha}^\dagger f_{n,\sigma,\alpha} \right), \quad (3.8)$$

onde os valores  $\epsilon_n^z$  são encontrados numericamente através de um procedimento recursivo [34] e discutidos no capítulo 6.2.1. Para valores grandes de  $n$ , a expressão  $\epsilon_n^z$  se aproxima de um comportamento assintótico:

$$\epsilon_n^z = \frac{1 + \lambda^{-1}}{2} \lambda^{1-z-n/2}, \quad n \geq 8. \quad (3.9)$$

Agora o Hamiltoniano é truncado, definindo-se um valor máximo para  $n$ , denominado  $N$ . O valor de  $N$  é escolhido de acordo com a escala de energia de interesse, para propriedades termodinâmicas é  $\epsilon_T = k_B T$  ( $k_B$  é a constante de Boltzmann e  $T$  a temperatura). O erro introduzido pelo truncamento é perfeitamente controlado, já que o único operador acoplado com a impureza é  $f_{0,\sigma,\alpha}^\dagger$ . O valor máximo  $N$  deve ser tal que a menor escala de energia  $\epsilon_N$  seja:

$$\epsilon_N \ll \epsilon_T \Rightarrow \frac{1 + \lambda^{-1}}{2} \lambda^{1-z-N/2} \ll k_B T \quad (3.10)$$

e o Hamiltoniano  $H_b$  pode ser restringido até  $N - 1$ , resultando em

$$\frac{H_b}{D} = \sum_{\alpha,\sigma,n=0}^{N-1} \epsilon_n \left( f_{n,\sigma,\alpha}^\dagger f_{n+1,\sigma,\alpha} + f_{n+1,\sigma,\alpha}^\dagger f_{n,\sigma,\alpha} \right). \quad (3.11)$$

O menor valor de energia é então  $\epsilon_{N-1} = (1 + \lambda^{-1}) \lambda^{1-z-(N-1)/2}/2$ . Por facilidades no cálculo numérico é interessante que o menor valor de energia seja da ordem de 1 e isso é conseguido dividindo  $H_b/D$  por  $(1 + \lambda^{-1}) \lambda^{-(N-1)/2}/2$ .

Todas estas transformações também são realizadas nos outros termos do hamiltoniano. Assim, pode-se reescrever a equação (3.1) usando a seguinte economia

de notação:  $\eta = 2/[D(1 + \lambda^{-1})]$ ,  $\eta E_0 \rightarrow E_0$ ,  $\eta E_x \rightarrow E_x$  e  $\eta V_\alpha \sqrt{2\rho_0 D} \rightarrow V_\alpha$ ,  $H/D \rightarrow H$  é escrito como:

$$\begin{aligned} H_N &= \lambda^{\frac{N-1}{2}} \left\{ \sum_{\alpha, \sigma, n=0}^{N-1} \epsilon_n (f_{n\sigma\alpha}^\dagger f_{n+1\sigma\alpha} + h.c.) + \right. \\ &+ \sum_{\sigma, \alpha} V_\alpha (f_{-1, \sigma, \alpha}^\dagger f_{0, \sigma, \alpha} + f_{-1, \sigma, \alpha} f_{0, \sigma, \alpha}^\dagger) + E_0 \sum_{\sigma} |m, \sigma\rangle \langle m, \sigma| \\ &+ \left. E_x \sum_{\alpha} [(|m-1, \alpha\rangle \langle m-1, \alpha| + |m+1, \alpha\rangle \langle m+1, \alpha|)] \right\}. \end{aligned} \quad (3.12)$$

Para recuperar o Hamiltoniano original é suficiente para fazer o limite:

$$H = \lim_{N \rightarrow \infty} \frac{D(1 + \lambda^{-1})}{2} \lambda^{-(N-1)/2} H_N. \quad (3.13)$$

A escala de energia original em uma dada iteração  $N$  pode ser recuperado com:

$$\exp(\beta H) = \exp\left(\beta \frac{1 + \lambda^{-1}}{2} D \lambda^{-(N-1)/2} H_N\right) \equiv \exp(\bar{\beta} H) \quad (3.14)$$

onde  $\bar{\beta}$  é definido,

$$\bar{\beta} = \beta \frac{1 + \lambda^{-1}}{2} D \lambda^{-(N-1)/2}, \quad (3.15)$$

e pode identificar cada iteração com uma respectiva escala de temperatura (à medida que  $N$  aumenta a temperatura diminui). A temperatura entra na variável  $\beta = 1/k_B T$ .

O TCSIAM apresenta uma simetria conhecida como carga axial em cada canal ( $J_\alpha$ ) [40]. Esta propriedade está relacionada com o fato que, no modelo, estados quânticos da impureza com  $m-1$  elétrons têm a mesma energia que estados quânticos da impureza com  $m+1$  elétrons. Assim, o operador que representa esta simetria e o hamiltoniano do TCSIAM comutam, logo os estados podem ser descritos pelos números quânticos desta simetria. O Hamiltoniano (3.12) conserva a carga axial  $J_\alpha$  em cada canal. Os autovalores do operador  $J_\alpha$  são usados como números quânticos ( $j_\alpha$ ) para descrever os estados do sistema. Ao contrário da carga axial, o spin não é conservado em cada canal separadamente. Mas o spin total  $S$  é, e comuta com  $J_\alpha$ . Isso permite também usar seus autovalores  $s$  como número quântico para descrever os estados do sistema.

Os operadores  $f_{n,\sigma,\alpha}^\dagger$  e  $f_{n,\sigma,\alpha}$  são componentes de um tensor de ordem 1/2. Então, com o Teorema de Wigner-Eckart [41], os estados do sistema podem ser desmembrados: com os números quânticos  $j_1$ ,  $j_2$  e  $s$ , resulta em

$$\langle j'_1, j'_{1z}, j'_2, j'_{2z}, s', s'_z | f_{n,\sigma,\alpha}^\dagger | j_1, j_{1z}, j_2, j_{2z}, s, s_z \rangle = \left\langle j_\alpha, j_{\alpha z}; \frac{1}{2}, \frac{1}{2} | j'_\alpha, j'_{\alpha z} \right\rangle \left\langle s, s_z; \frac{1}{2}, \sigma | s, s \right\rangle \langle j'_1, j'_2, s | \alpha | j_1, j_2, s \rangle, \quad (3.16)$$

$$\langle j'_1, j'_{1z}, j'_2, j'_{2z}, s, s | (-1)^n (-2\sigma) f_{n,\sigma,\alpha} | j_1, j_{1z}, j_2, j_{2z}, s, s_z \rangle = \left\langle j_\alpha, j_{\alpha z}; \frac{1}{2}, -\frac{1}{2} | j'_\alpha, j'_{\alpha z} \right\rangle \left\langle s, s_z; \frac{1}{2}, -\sigma | s, s \right\rangle \langle j'_1, j'_2, s | \alpha | j_1, j_2, s \rangle. \quad (3.17)$$

Dessa forma, os números quânticos da carga axial do canal 1,  $j_1$ , do canal 2,  $j_2$ , e do spin total,  $s$ , representam os estados da base em que  $H_N$  será diagonalizado:  $\{|j_1, j_2, s\rangle_N\}$ .

### 3.3.2 Diagonalização iterativa

Da equação 3.12, pode-se escrever o Hamiltoniano da seguinte forma:

$$H_N = \sqrt{\lambda}H_{N-1} + \xi_{N-1}(f_{N-1,\sigma,\alpha}^\dagger f_{N,\sigma,\alpha} + f_{N,\sigma,\alpha}^\dagger f_{N-1,\sigma,\alpha}), \quad (3.18)$$

$\xi_{N-1} = \sqrt{2}\lambda^{(N-1)/2}\epsilon_{N-1}$ , com  $\xi_{-1,\alpha} = \sqrt{\frac{2}{\lambda}}V_\alpha$  e soma sobre  $\alpha$  e  $\sigma$  subentendidas.

A equação 3.18 é o que permite uma diagonalização iterativa: isto é, a transformação do TCSIAM, é a transformação  $\mathfrak{S}$  do GRN.

Os auto-estados do hamiltoniano  $H_{N-1}$  são usados para construir a base de vetores da iteração seguinte. Para construir os vetores da base na iteração  $N$ , o produto direito dos auto-estados anteriores, com os elétrons de condução em cada canal, é feito. As simetrias dos operadores,  $\hat{j}_1, \hat{j}_2, \hat{s}$ , são usadas em cada passo para calcular os vetores da base. O processo continua até atingir uma iteração máxima de  $N_{max}$  que corresponda à escala de energia baixa suficiente para ver os fenômenos quânticos. Na figura 4 o processo iterativo do GRN é apresentado.

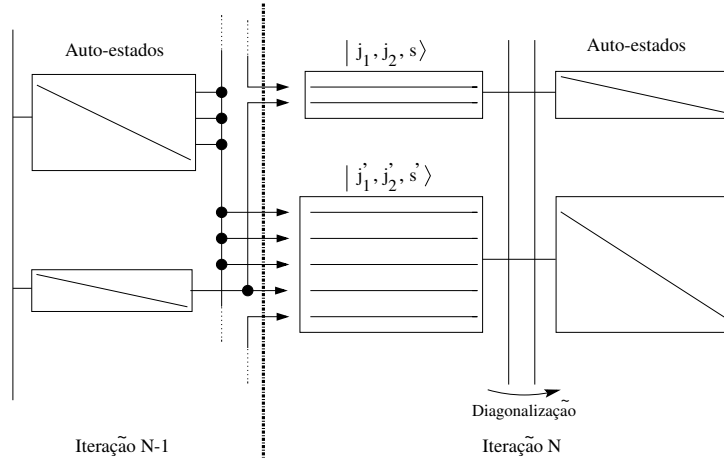


Figura 4 – Algoritmo do processo iterativo do GRN. No final da iteração  $N - 1$ , o hamiltoniano  $H_{N-1}$  já está completamente diagonalizado. Por isso, os auto-estados e autovalores são conhecidos. Cada autovetor da interação anterior gera vários estados de base para a iteração atual. Os estados com os mesmos números quânticos são agrupados em subespaços. Os elementos de matrizes são calculados através os invariantes coletivos. Essas matrizes simétricas são diagonalizadas depois para encontrar os novos auto-estados e autovalores da iteração atual  $N$ .

### 3.3.3 Os vetores da base

Os vetores de base da iteração  $N$  são obtidos multiplicando-se cada auto-estado da iteração anterior,  $N - 1$ , por um dos 16 estados de canal da tabela 1. Cada canal pode estar em um dos quatro seguintes estados  $|\uparrow\downarrow\rangle$ ,  $|0\rangle$ ,  $|\uparrow\rangle$  ou  $|\downarrow\rangle$ . Os estados juntos resultam em 16 auto-estados dos operadores  $\hat{j}_1^2$ ,  $\hat{j}_{1z}$ ,  $\hat{j}_2^2$ ,  $\hat{j}_{2z}$ ,  $\hat{s}^2$  e  $\hat{s}_z$ . O hamiltoniano do Modelo de Anderson de Dois Canais não tem operadores com o componente  $z$  (3.2-3.4). Por isso, somente os valores  $j_1$ ,  $j_2$  e  $s$  entram nos seguintes cálculos. Os elementos dentro de um mesmo grupo, subespaço, apresentam os mesmos números quânticos. Para diferenciá-los, o índice de contagem  $r$  é introduzido.

Tabela 1 – As combinações dos estados de cada canal formam os vetores para criar a base da iteração atual. Estados com mesmos números quânticos estão no mesmo subespaço e são diferenciados pelo índice de contagem  $r$ .

	canal 1, canal 2⟩	$j_1, j_2, s$
1	$ \uparrow\downarrow, \uparrow\downarrow\rangle$	1/2, 1/2, 0
2	$(-1)^N  \uparrow\downarrow, 0\rangle$	1/2, 1/2, 0
3	$(-1)^N  0, \uparrow\downarrow\rangle$	1/2, 1/2, 0
4	$ 0, 0\rangle$	1/2, 1/2, 0
1	$ \uparrow, 0\rangle$	0, 1/2, 1/2
2	$(-1)^N  \uparrow, \uparrow\downarrow\rangle$	0, 1/2, 1/2
3	$ \downarrow, 0\rangle$	0, 1/2, 1/2
4	$(-1)^N  \downarrow, \uparrow\downarrow\rangle$	0, 1/2, 1/2
1	$(-1)^N  \uparrow\downarrow, \uparrow\rangle$	1/2, 0, 1/2
2	$ 0, \uparrow\rangle$	1/2, 0, 1/2
3	$(-1)^N  \uparrow\downarrow, \downarrow\rangle$	1/2, 0, 1/2
4	$ 0, \downarrow\rangle$	1/2, 0, 1/2
1	$ \uparrow, \uparrow\rangle$	0, 0, 1
2	$\frac{1}{\sqrt{2}}( \uparrow, \downarrow\rangle +  \downarrow, \uparrow\rangle)$	0, 0, 1
3	$ \downarrow, \downarrow\rangle$	0, 0, 1
4	$\frac{1}{\sqrt{2}}( \uparrow, \downarrow\rangle -  \downarrow, \uparrow\rangle)$	0, 0, 1

Cada auto-estado,  $(|j'_1, j'_2, s'\rangle_{N-1})$ , da iteração anterior,  $N - 1$ , é combi-



nado com esses estados de canal para estabelecer a base da iteração atual,  $N$ , na maneira que os vetores desta base também sejam auto-estados dos operadores  $\hat{j}_1^2, \hat{j}_{1z}, \hat{j}_2^2, \hat{j}_{2z}, \hat{s}^2$  e  $\hat{s}_z$ . Começando com um auto-estado da iteração anterior,  $|j'_1, j'_2, s'\rangle_{N-1}$ , os procedimentos para gerar novos estados (filhos)  $|j_1, j_2, s\rangle_N$  da iteração atual são mostrados na tabela 15. No total são 16 procedimentos possíveis para criar filhos e cada um deles recebe mais um número quântico que é chamado de tipo  $t$ . O tipo guarda a informação como o estado da iteração atual foi criado. É importante observar que autovetores  $|j'_1, j'_2, s'\rangle_{N-1}$  diferentes podem resultar em estados com os mesmos números quânticos na iteração seguinte os quais são agrupados. Esses estados com mesmos números quânticos são juntados para formar os subespaços da iteração  $N$ , usando o Teorema de Wigner-Eckart e os coeficientes de Clebsh-Gordan. Como o hamiltoniano, conseqüentemente, os vetores da base não dependem de  $j_{1z}, j_{2z}$  ou  $s_z$ , pois não há campo magnético ou campo elétrico externo. Por causa da ausência dos campos externos, a redução das variáveis é feita,  $j_{1z} = j_1, j_{2z} = j_2$  e  $s_z = s$ .

Com essa base de vetores, as matrizes são explicadas na seção seguinte.

### 3.3.4 Os elementos de matriz de $H_N$

Os elementos de matriz  $\langle j'_1, j'_2, s', t' | H | j_1, j_2, s, t \rangle_N$  são calculados utilizando transformação do GRN, equação (3.18). Por economia de notação, tanto  $j_1, j_2$  e  $s$  como  $j'_1, j'_2$  e  $s'$  são números quânticos de estados quaisquer da iteração atual. Os elementos diagonais de  $H_N$  são os autovalores da iteração anterior multiplicados por  $\sqrt{\lambda}$ . Os operadores  $f_{N-1, \sigma, \alpha}^\dagger f_{N, \sigma, \alpha}$  ou  $f_{N, \sigma, \alpha}^\dagger f_{N-1, \sigma, \alpha}$  determinam os elementos não-diagonais. O Teorema de Wigner-Eckart e a álgebra vetorial são usados para calculá-los. Os operadores  $\{f_{n, \sigma, \alpha}^\dagger\}$  são responsáveis pela criação ou destruição dos elétrons. Por isso, no Modelo de Anderson de Dois Canais, os elementos não-diagonais são diferentes de zero somente em um dos dois casos:

$$i) \quad j'_1 = j_1, \quad j'_2 = j_2 \pm 1/2, \quad (3.19)$$

$$ii) \quad j'_1 = j_1 \pm 1/2, \quad j'_2 = j_2. \quad (3.20)$$

Os estados da iteração  $N$  são bem determinados pelos procedimentos para criar filhos, veja no apêndice A. Os 16 tipos  $t$  definem as regras de criação. As duas condições acima explicam a combinação do tipo  $t$  do estado *ket* com o tipo  $t'$  do estado *bra*. A tabela 2 mostra todas as transições que são em confirmação com as duas condições (3.19, 3.20). As outras combinações de tipos que não são mostradas na tabela 2 são nulos. No apêndice B encontram-se todos os 40 elementos de matriz não-diagonais que podem resultar em valor não nulo. Como exemplo, o elemento de matriz entre o estado *ket* do tipo 5 e estado *bra* do tipo 1 é mostrado.

$$\langle 1 | H | 5 \rangle_N = (-1)^N \sqrt{\frac{2s}{2s+1}} \langle j_1, j_2 - \frac{1}{2}, s - \frac{1}{2} || 1 || j_1 - \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1}. \quad (3.21)$$

Tabela 2 – Todas as transições possíveis do *ket* para o *bra* dentro de uma iteração.

2, 3, 5, 6 → 1	4, 7, 8, 9 → 2	4, 8, 9, 10 → 3
11, 12 → 4	7, 8, 9, 13 → 5	8, 9, 10, 13 → 6
11, 14, → 7	11, 12, 14, 15 → 8	11, 12, 14, 15 → 9
12, 15, → 10	16 → 11	16 → 12
14, 15 → 13	16 → 14	16 → 15

### 3.3.5 Invariantes

O elemento de matriz da equação (3.21) é basicamente formado por um coeficiente multiplicado pelo termo  $\langle j_1, j_2 - \frac{1}{2}, s - \frac{1}{2} || 1 || j_1 - \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1}$ . Este termo, pelo fato de não depender das componentes  $j_{1z}$ ,  $j_{2z}$  e  $s_z$  é denominado de invariante coletivo. Ele pode ser de canal 1 ou de canal 2. Na verdade, esses invariantes coletivos são combinações de invariantes individuais dos estados *pai* e possuem a seguinte forma genérica  $\langle j_1, j_2, s, r || || j'_1, j'_2, s', r' \rangle_{N-1}$ , onde  $r$  é o índice de contagem. Cada *bra* ou *ket* representa um autovetor do subespaço da interação anterior. Para se referir a um estado específico desse subespaço é usada a notação  $|| j'_1, j'_2, s', r \rangle$ ,  $r$ -ésimo auto-estado do subespaço estabelecido pelo  $j'_1, j'_2$  e  $s'$ .

Os invariantes são calculados utilizando o Teorema de Wigner-Eckart e os coeficientes de Clebsh-Gordan. Nesse cálculo, a combinação de  $j_1, j_2, s$  com  $j'_1, j'_2$  e  $s'$  só é diferente de zero em uma entre as quatro maneiras possíveis da tabela 3. Em cada canal existem 20 invariantes individuais possíveis. No apêndice B encontra-se todos os 40 invariantes individuais do Modelo de Anderson de Dois Canais.

Tabela 3 – Quatro combinações possíveis para os invariantes.

<i>a:</i>	$j'_1 = j_1$	$j'_2 = j_2 - 1/2$	$s' = s - 1/2$
<i>b:</i>	$j'_1 = j_1$	$j'_2 = j_2 - 1/2$	$s' = s + 1/2$
<i>c:</i>	$j'_1 = j_1 - 1/2$	$j'_2 = j_2$	$s' = s - 1/2$
<i>d:</i>	$j'_1 = j_1 - 1/2$	$j'_2 = j_2$	$s' = s + 1/2$

O calculo do invariante coletivo é um produto onde as dimensões dos dois autovetores podem ser diferentes. Como um exemplo, o invariante da iteração  $N = 0$  do segundo auto-estado *ket* bidimensional,  $|\frac{1}{2}, 0, 1\rangle$ , e do primeiro auto-estado *bra* tridimensional,  $|\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle$ , no canal 2, é mostrado na equação (3.24) para ilustrar a relação entre invariante coletivo e individual. As duas representações dos dois auto-estados são:

$$|\frac{1}{2}, 0, 1, 2\rangle = h_1|2\rangle + h_2|7\rangle, \quad (3.22)$$

$$\langle \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1| = w_1\langle 1| + w_2\langle 2| + w_3\langle 5|. \quad (3.23)$$

O segundo *ket* vem do subespaço com base de tipos 2 e 7, o primeiro *bra* tem os tipos 1, 2 e 5 como base na iteração  $N = 0$ .

$$\begin{aligned}
\left\langle \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1 \middle| 2 \middle| \frac{1}{2}, 0, 1, 2 \right\rangle_0 &= \sum_{i=1}^3 \sum_{j=1}^2 w_i h_j \left\langle t_i \middle| 2 \middle| t_j \right\rangle_0 \\
&= w_1 h_1 \left\langle 1 \middle| 2 \middle| 2 \right\rangle_0 + w_1 h_2 \underbrace{\left\langle 1 \middle| 2 \middle| 7 \right\rangle_0}_{=0} + \\
&w_2 h_1 \underbrace{\left\langle 2 \middle| 2 \middle| 2 \right\rangle_0}_{=0} + w_2 h_2 \underbrace{\left\langle 2 \middle| 2 \middle| 7 \right\rangle_0}_{=0} + \\
&w_3 h_1 \underbrace{\left\langle 5 \middle| 2 \middle| 2 \right\rangle_0}_{=0} + w_3 h_2 \left\langle 5 \middle| 2 \middle| 7 \right\rangle_0
\end{aligned} \tag{3.24}$$

No cálculo do invariante coletivo, quatro invariantes individuais são zeros porque os tipos não combinam ou não permitem uma transição nesses canais. Veja no apêndice B todas combinações de tipos que resultam em um invariante individual não nulo nos dois canais. O cálculo de montar um subespaço na iteração  $N = 1$ , incluindo os cálculos dos invariantes, é mostrado na seção 6.2.3.

### 3.3.6 Hamiltoniano da iteração $N = -1$

Para iniciar o processo iterativo é necessário fornecer os auto-estados, os autovalores e os invariantes da iteração inicial  $N = -1$  e diagonalizar a matriz do Hamiltoniano  $H_{-1}$ . Essa etapa é feita analiticamente. Através da equação (3.12) o Hamiltoniano da iteração  $N = -1$  é obtido:

$$H_{-1} = E_0 \sum_{\sigma} |m, \sigma\rangle \langle m, \sigma| + E_x \sum_{\alpha} |m-1, \alpha\rangle \langle m-1, \alpha| + |m+1, \alpha\rangle \langle m+1, \alpha|. \quad (3.25)$$

## 4 Objetivos

Este projeto de mestrado teve como objetivo geral: Desenvolver um novo código computacional usando os conceitos do Grupo de Renormalização Numérica, que diagonalize o Modelo de Anderson de Dois Canais. Este novo código deve ser de fácil compreensão, devidamente documentado para futuros aperfeiçoamentos e que permita seja licenciado como produto tecnológico do INFI-UFMS.

Para isso, os objetivos específicos foram:

- Aprender sobre impurezas magnéticas em metais, seus modelos e sobre a técnica do Grupo de Renormalização Numérica.
- Analisar o código computacional do Dr. J. V. B. Ferreira: tempo de execução e estilo de programação.
- Escrever um novo código e comparar com o anterior.
- Fazer documentação utilizando pacote apropriado e de livre acesso.
- Iniciar processo de licenciamento como produto tecnológico do Instituto de Física da Universidade Federal do Mato Grosso do Sul (INFI-UFMS).

## 5 Metodologia

### 5.1 Documentação do código computacional com o programa Doxygen

O desenvolvimento de grandes programas de simulação e análise ou mesmo pacotes de software é um processo trabalhoso - então, por que gastar ainda mais tempo com a documentação do código-fonte? E o que é uma boa documentação? Em primeiro lugar, um programa bem documentado é mais fácil de se entender por outros desenvolvedores e evita o processo muito demorado de ler apenas o código fonte original. Uma boa documentação descreve muito claramente porque uma parte de um programa (script, função ou classe) foi criada e o que faz. Assim, o próximo usuário obtém uma visão geral básica da funcionalidade atual, o que facilita a adição e a adaptação de novos módulos. Em segundo lugar, o processo de documentação pode ser visto como um processo de controle do código. Reescrever o programa novamente ajuda a melhorar o estilo. A convenção de indentação de cada linguagem deve ser mantida e as variáveis devem conter nomes auto-documentados. O processo de documentação favorece essas orientações gerais. Além disso, é verificado novamente se a parte de um programa é realmente necessária ou se apenas é uma duplicata de uma funcionalidade já existente. Dessa forma, a codificação dupla é apontada e, conseqüentemente, pode ser evitada. Outro ponto importante é que uma boa documentação ilustra todas as conexões e dependências entre diferentes scripts, funções e classes. O novo desenvolvedor consegue ter uma ideia de como a execução de uma tarefa funciona sem sequer usar o programa. O ponto mais importante a favor da documentação é o argumento de comunicação e clareza. Um programa bem documentado torna o uso dele efetivo e eficiente para outro usuário sem que seja necessário o contato direto com o programador original. Além disso, a perda de conhecimento de como o software escrito deve ser aplicado

pode ser prevenida por uma documentação bem mantida se o desenvolvedor deixar a instituição. Uma boa documentação simplifica também o desenvolvimento em grupo de um programa de simulação ou análise devido à clareza de cada parte. Pode-se dizer que o tempo necessário para escrever e manter a documentação atualizada é bem investido, como demonstram as vantagens acima mencionadas. Além disso é parte essencial no processo de patentear o código computacional.

Um gerador de documentação gratuito muito difundido para C++ é a plataforma Doxygen [42]. Tem sido desenvolvido por Dimitri van Heesch desde 1997 e também pode ser usado para muitas outras linguagens de programação (Python, Java, C, Fortran, Matlab, etc.). Este gerador de documentação é executado através de um código-fonte, que deve ser documentado em um formato específico Doxygen, e cria um arquivo 'html' de documentação que inclui todas as vantagens mencionadas anteriormente. O desenvolvedor Dimitri van Heesch afirma que "o Doxygen é a ferramenta padrão de fato para gerar documentação de fontes C++ anotadas". A documentação de um código-fonte geralmente é estabelecida em dois níveis, um para cada um dos dois grupos-alvo comuns. O nível superior é para o usuário que aplica apenas a funcionalidade já existente do programa. A documentação foca aqui na entrada e saída do programa e como o parâmetro deve ser configurado ou modificado. O nível mais baixo é para o desenvolvedor real que precisa conhecer todos os pequenos detalhes de cada parte (scripts, funções, classes). Os valores e tipos de entrada e saída de cada pequena parte do código-fonte são descritos. As dependências das chamadas de função e a forma como as classes interagem entre si também são protocoladas.

O programa de simulação e análise atual é denominado de 'Two Channel Single Impurity Anderson Model'. Ele está escrito em C++ em estilo de programação funcional e orientada a objetos. A segmentação do programa em funções únicas é denominada programação funcional [43]. Uma função é um subprograma que cumpre uma determinada tarefa do programa. Uma função pode ter variáveis de entrada e saída. Pode ser chamada sempre que necessário. A duplicação do código-fonte pode ser evitada pela separação de todo o código-fonte em funções. É principalmente suficiente saber qual tarefa as funções únicas cumprem e em que ordem elas são chamadas para se entender a execução de um programa escrito em um estilo



funcional. Não é essencial compreender cada detalhe dos algoritmos de cada função para uma compreensão geral do programa. Na programação orientada a objetos a ênfase é a construção de objetos: conjuntos constituídos por dados relacionados e suas operações. O resultado é que o código computacional é reutilizável com adaptações em novos problemas. Outras características são a facilidade da depuração do código e a alocação e liberação dinâmica, que permite aproveitamento eficiente de memória. A programação orientada a objetos é determinada pelo conceito de que as variáveis e as funções são encapsuladas em classes [44]. Uma classe é uma estrutura abstrata que define de que maneira um objeto dessa classe é construído e quais características, atributos, e quais funções, métodos, ela carrega. As características de um objeto são chamadas de propriedades ou atributos que possuem valores que lhe são atribuídos. Objetos diferentes da mesma classe são distinguidos e comparados por esses valores. As propriedades são, portanto, as variáveis. As funções implementadas de um objeto são chamadas de métodos. Os métodos só podem ser chamados pelo próprio objeto e podem ter variáveis de entrada e saída como funções comuns. O método mais importante de cada classe é o construtor que cria o objeto ou instância de uma classe. O construtor normalmente espera variáveis de entrada que serão atribuídas às propriedades do objeto. O objeto de uma classe na programação orientada a objetos é a combinação de variáveis e funções que atuam nas variáveis na programação funcional [45].

## 5.2 Implementação computacional

No TCSIAM o número de estados  $e$ , conseqüentemente, o tamanho da matriz Hamiltoniana, cresce na proporção de  $16^N$ . Após diagonalizada uma iteração, por exemplo, iteração  $H_{N-1}$ , seus auto-estados e autovalores são usados para construir os vetores da base da iteração seguinte  $N$ , e assim por diante. A partir dos estados da iteração  $N$  se constrói a base dos subespaços que formam o Hamiltoniano. Cada subespaço é identificado pelos números quânticos  $j_1$ ,  $j_2$  e  $s$ . Um elemento particular de um subespaço é identificado pelos três números quânticos mais um índice de contagem. Com a base calculam-se os elementos de matriz, utilizando-se os elementos invariantes já calculados na iteração anterior. Os subespaços são então diagonalizados. Com os autovalores dos subespaços calculam-se as propriedades termodinâmicas desejadas. O número de iterações  $N$  é arbitrário: ele está relacionado com a temperatura que desejamos alcançar e quanto maior seu valor, mais baixa a temperatura que se alcança.

O programa tem vários parâmetros de entrada sendo os principais os seguintes: quatro parâmetros que se referem ao modelo ( $E_0$ ,  $E_x$ ,  $V_1$ ,  $V_2$ ) e que por isso são denominados de parâmetros físicos, três que se referem à discretização logarítmica ( $N$ ,  $\lambda$ ,  $z$ ), dois que se referem ao controle de memória do programa ( $Corte$ ,  $T$ ) e dois que controlam o número de pontos obtidos ( $nBB$ ,  $N$ ). Mesmo usando leis de conservação e alocação dinâmica de memória, é inviável calcular todos os estados de  $H_N$  para  $N$  grande. Assim calculam-se os estados até um valor máximo de energia arbitrado, que é denominado de  $Corte$  ou até que o respectivo subespaço atinja o tamanho máximo definido por  $T$ . Procura-se estabelecer estes valores tão menores quanto possível, isto é, sem comprometer a precisão das propriedades físicas calculadas.

Para retirar as oscilações espúrias originárias da discretização logarítmica é necessário realizar o método de Intercalamento utilizando várias curvas de suscetibilidade ou de calor específico com diferentes valores de  $z$ . A situação mais crítica ocorre na curva de calor específico do caso isotrópico ( $V_1 = V_2$ ) quando a degenerescência é muito grande. Nessa situação é necessário fazer a média com oito ou dezesseis valores diferentes de  $z$  [16].

## 6 Resultados

### 6.1 Análise do programa do Modelo de Anderson de Dois Canais

O programa Modelo de Anderson de Dois Canais (MADC) do Dr. J. V. B. Ferreira foi escrito na linguagem C++ baseado na filosofia de programação orientada a objetos na década de 1990 [37, 46, 47]. De fato, o estilo de programação utilizado é uma combinação dos estilos script, funcional e orientado a objeto. C++ era um linguagem mais nova nesta época e trouxe a vantagem de controlar e regular a memória usada. Isso permite a manipulação de grande quantidade dados que são os estados quânticos, os invariantes e conseqüentemente as matrizes. O controle do crescimento do Hamiltoniano,  $16^N$ , se manifestou umas das tarefas fundamentais na implementação do GRN.

Nos parágrafos seguintes, os detalhes técnicos, como a programa pode ser iniciado e qual são os arquivos mais importantes, são explicados.

Todos os arquivos (*.cpp*, *.h*, *.o* e o *makefile*) são guardados na mesma pasta *and2UFCodigos2017.1*. Para iniciar a execução do programa, o usuário acessa esta pasta e executa o seguinte comando

```
echo \-0.005 \0.001 \0.02 \0.002 \6 \0.25 \40 \18.0 \6 \18.0 \80 \0.05 > entrada.dat
```

para criar um arquivo *entrada.dat* com todos os parâmetros ( $E_o$ ,  $E_x$ ,  $V_1$ ,  $V_2$ ,  $\lambda$ ,  $z$ ,  $N$ ,  $C$ ,  $N_{in}$ ,  $C_{in}$ ,  $T$  e  $eBar$ ). O compilador e o linker são invocados pelo comando *make* que vai ler as informações do *makefile*.

O arquivo *multichannel.cpp* mostrado na figura 5 é a função principal que lê

os parâmetros que se referem ao modelo  $(E_0, E_x, V_1, V_2)$ , à discretização logarítmica  $(N, \lambda, z)$  e ao controle de memória  $(Corte, N2, CorteN2, nval)$  e inicia as iterações em seguida. Os três auto-estados são iniciados e a primeira iteração  $N = -1$  é chamada explicitamente (*Iterh\_1*). A iteração  $N = -1$  é a inicial e tratada de maneira diferente. As demais são tratadas pelos mesmos procedimentos (*Iterh\_n*).

```
//      PROGRAM MULTICHANNEL
//      Diagonalization Two-channel Anderson Model
//      by numerical renormalization group

int main(){

    Indices ***Element;
    Invariants *****Eminv;
    Entrada();
    Iterh_1(&Element, &Eminv);
    Iterh_N(&Element, &Eminv);

    return 0;
}
```

Figura 5 – Programa Model de Anderson de Dois Canais: Função principal, executável, para iniciar os elementos de matriz, calcular os invariantes, ler os argumentos de entrada, executar a iteração inicial e continuar até iteração  $N$ .

Valgrind é um programa de código aberto para fazer análises dinâmicas [48]. Ele pode ser usado para detectar vazamentos de memória, 'threading bugs' ou analisar o tempo de execução em detalhes. Os desenvolvedores são voluntários e o financiamento é baseado em doações. Callgrind que vem com Valgrind é um programa para visualizar os resultados da análise do tempo de execução [49].

A figura 6 mostra a análise do programa Modelo de Anderson de Dois Canais. Quantas vezes uma rotina (função, classe) é chamada e quanto tempo em porcentagem o programa gasta numa parte é representado. Além disso, as dependências entre as rotinas são mostradas. Baseado nesta pre-análise do programa MADC, foi possível ver que o programa gasta 52% em rotinas escritas por Dr. J. V. B. Ferreira e 48% na biblioteca *givens*, que é usada para diagonalizar os subespaços numericamente. Isso foi a motivação fundamental de reescrever e

otimizar o programa. Melhorar a biblioteca que faz a diagonalização é normalmente uma tarefa bem difícil e reque um conhecimento profundo no campo da otimização. Trocar a biblioteca para *Eigen* ou *Lapack* seria uma opção. Mais comum é a ideia de reescrever a parte do desenvolvedor e tentar otimizar isso. A seção 6.2.4 descreve esse novo código.

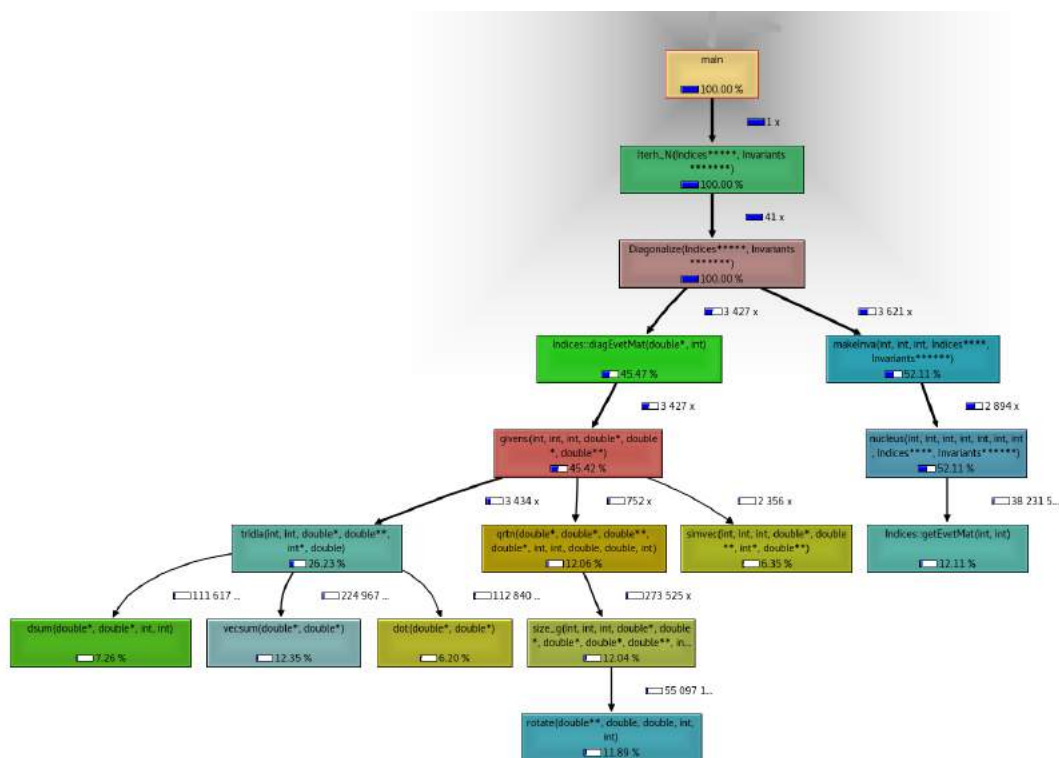


Figura 6 – Análise do tempo de execução com os programas Valgrind e Callgrind do programa Modelo de Anderson de Dois Canais. Dependências das funções individuais e o tempo relativo que o programa gasta em cada função.

Uma segunda análise de tempo de execução foi feita para analisar o tempo por iteração, veja figura 7. O programa Modelo de Anderson de Dois Canais tem duas funcionalidades de controlar o crescimento dos subespaços, o corte da energia dos autovetores e a limitação do tamanho dos subespaços.

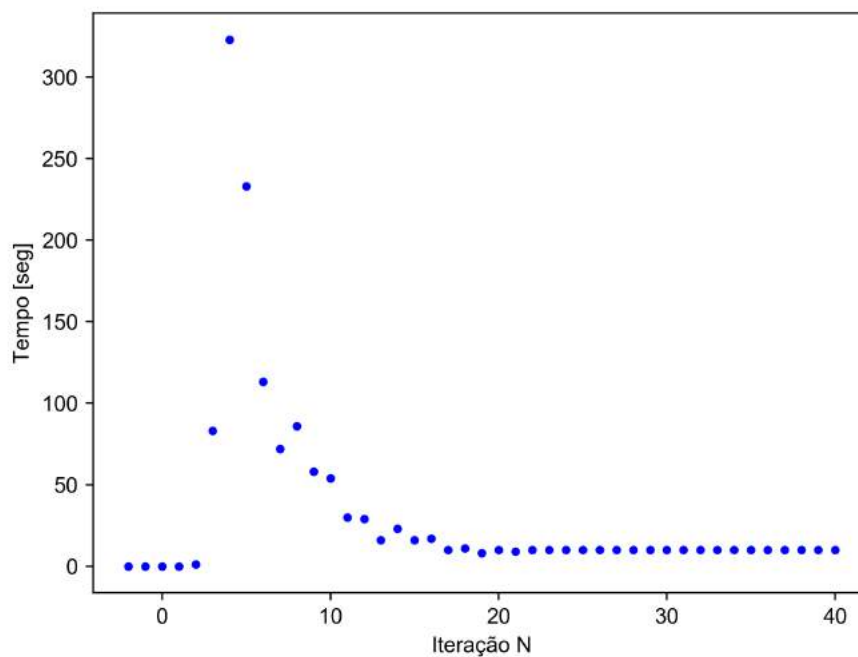


Figura 7 – Análise do tempo de execução por iteração do programa Modelo de Anderson de Dois Canais.

Uma análise dos arquivos, que o MADC gera, mostra que o primeiro controle de crescimento é o limite do tamanho das matrizes até a iteração  $N = 8$  e depois é o corte das energias que limita o crescimento exponencial.

## 6.2 Two Channel Single Impurity Anderson Model

Nos seguintes parágrafos a construção do programa, *Two Channel Single Impurity Anderson Model* - TCSIAM, usando os conceitos do Grupo de Renormalização Numérica, é apresentada.

### 6.2.1 Discretização logarítmica da banda de condução

No modelo físico a banda de condução eletrônica é um conjunto contínuo e infinito de valores de energia. Para ser implementado em código computacional é necessário transformar a banda de condução semi-preenchida em um conjunto discreto de valores de energia. Além disso, a alta densidade de estados com energia próximo do nível de Fermi é mantida usando a discretização logarítmica ao invés de linear.

A transformação do GRN, veja equação (3.18), é reescrita aqui

$$H_N = \sqrt{\lambda}H_D + \xi_{N-1}^z H_I \quad (6.1)$$

$$H_D = H_{N-1} \quad (6.2)$$

$$H_I = f_{N-1,\sigma,\alpha}^\dagger f_{N,\sigma,\alpha} + f_{N,\sigma,\alpha}^\dagger f_{N-1,\sigma,\alpha} \quad (6.3)$$

$$\xi_N^z = \sqrt{2}\lambda^{N/2}\epsilon_N^z. \quad (6.4)$$

Os fatores  $\epsilon_N^z$  são a despeito de um fator de  $\sqrt{2}\lambda^{N/2}$  as energias logarítmicas da banda de condução. Eles são os mesmos para cada iteração  $N$  e são determinados pela seguinte equação recursiva.

$$\prod_{n=0}^N (\epsilon_n^z)^2 = F_N(z, \lambda) - [(M_{N+1})^{2N+2}]_{11} \quad (6.5)$$

$$F_N(z, \lambda) = (1 - \lambda^{-z}) \left[ \frac{1 + \lambda^{-z}}{2} \right]^{2N+2} + (1 - \lambda^{-1}) \left[ \frac{1 + \lambda^{-1}}{2} \right]^{2N+2} \frac{\lambda^{-(2N+3)z}}{1 - \lambda^{-(2N+3)}} \quad (6.6)$$

$$[M_{N+1}]_{ij} = \epsilon_{i-1}^z \delta_{i,j-1} + \epsilon_{j-1}^z \delta_{j,i-1} \quad (i, j = 1, 2, \dots, N+1) \quad (6.7)$$

$$\epsilon_0^z = (1 - \lambda^{-2})(1 + \lambda^{-1})/4 \sum_{m=1}^{m_{max}} \lambda^{3(1-z-m)} + (1 - \lambda^{-2z})(1 + \lambda^{-z})/4 \quad (6.8)$$

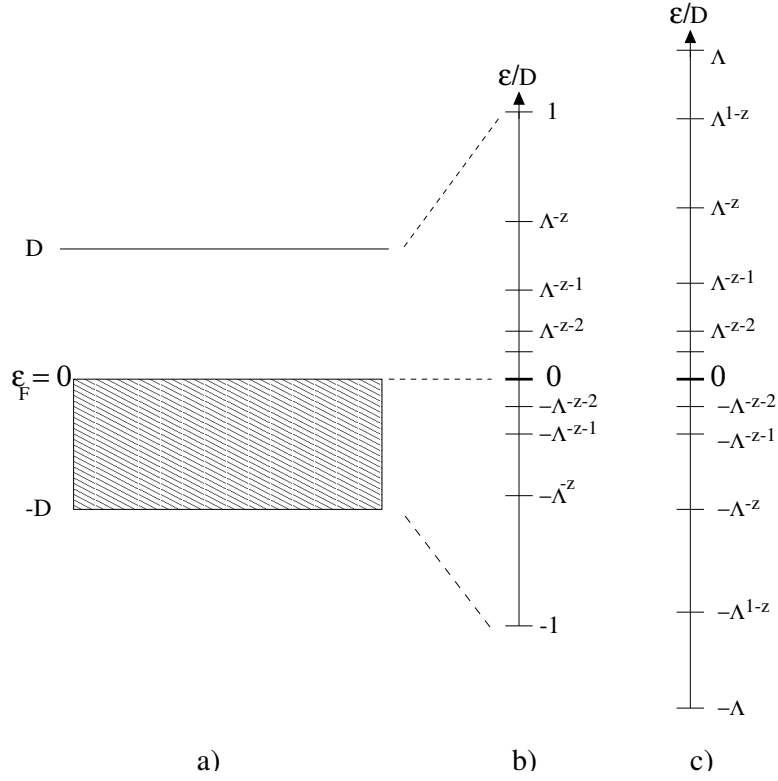


Figura 8 – a) Banda de condução semi-preenchida, com largura  $2D$ , contínua e isotrópica b) Discretização logarítmica da banda de condução pelos parâmetros  $\lambda > 1$  e  $z > 0$  c) Multiplicação dos nível de energia por  $\lambda^n$ , com  $n$  inteiro e positivo. Notação: O  $\Lambda$  na figura é igual como o  $\lambda$ .

Fonte: FERREIRA, J. V. B. Modelo de Anderson de dois canais. Tese (Doutorado em Física Básica) — Universidade de São Paulo, São Carlos, Instituto de Física de São Carlos, São Paulo, 2000.

O parâmetro  $m_{max}$  é definido com o valor 100 no programa TCSIAM porque um valor mais alto não muda mais o resultado. Um comportamento assintótico é obtido no limite  $N \rightarrow \infty$  pelos fatores  $\epsilon_N^z$ .

$$\epsilon_N^z = \frac{1 + \lambda^{-1}}{2} \lambda^{1-z-N/2} \quad (6.9)$$

No entanto, uma dificuldade prática surge da convergência rápida dos dois valores  $F_N(z, \lambda)$  e  $[(M_{N+1})^{2N+2}]_{11}$ . Por isso, a precisão computacional quádrupla é usada no TCSIAM, e a mudança para calcular os fatores com a expressão assintótica 6.9 já é usada na iteração  $N = 8$ . As tabelas 4 e 5 mostram os fatores  $\epsilon_N^z$  calculados



com precisão quádrupla para  $\lambda = 3$  e dois valores de  $z$  em comparação com os cálculos numéricos do artigo de Yoshida et al [34].

Tabela 4 – Fatores para  $\lambda = 3$  e  $z = 1$  para o modelo TCSIAM, que usa a equação (6.5) para determinar os valores numericamente até iteração  $N = 7$  e depois a expressão assintótica (6.9), em comparação com os valores do artigo de Yoshida [34], que usa a equação (6.5) para todas iterações.

N	$\epsilon_N^z$ TCSIAM	$\epsilon_N^z$ Yoshida
0	$5.54700196 \times 10^{-01}$	$5.54700196 \times 10^{-01}$
1	$3.49370517 \times 10^{-01}$	$3.49370517 \times 10^{-01}$
2	$2.14482487 \times 10^{-01}$	$2.14482487 \times 10^{-01}$
3	$1.26748308 \times 10^{-01}$	$1.26748308 \times 10^{-01}$
4	$7.37713247 \times 10^{-02}$	$7.37713247 \times 10^{-02}$
5	$4.27081557 \times 10^{-02}$	$4.27081557 \times 10^{-02}$
6	$2.46800766 \times 10^{-02}$	$2.46800766 \times 10^{-02}$
7	$1.42533969 \times 10^{-02}$	$1.42533900 \times 10^{-02}$
8	$8.23045267 \times 10^{-03}$	$8.23003456 \times 10^{-03}$
9	$4.75185407 \times 10^{-03}$	$4.75177360 \times 10^{-03}$
10	$2.74348422 \times 10^{-03}$	$2.74346878 \times 10^{-03}$
11	$1.58395136 \times 10^{-03}$	$1.58794010 \times 10^{-03}$

Tabela 5 – Fatores para  $\lambda = 3$  e  $z = 0.5$  para o modelo TCSIAM, que usa a equação (6.5) para determinar os valores numericamente até iteração  $N = 7$  e depois a expressão assintótica (6.9), em comparação com os valores do artigo de Yoshida [34], que usa a equação (6.5) para todas iterações.

N	$\epsilon_N^z$ TCSIAM	$\epsilon_N^z$ Yoshida
0	$5.67544821 \times 10^{-01}$	$5.67544821 \times 10^{-01}$
1	$4.60312552 \times 10^{-01}$	$4.60312552 \times 10^{-01}$
2	$4.03544580 \times 10^{-01}$	$4.03544580 \times 10^{-01}$
3	$2.55183371 \times 10^{-01}$	$2.55183371 \times 10^{-01}$
4	$1.35339032 \times 10^{-01}$	$1.35339032 \times 10^{-01}$
5	$7.50818175 \times 10^{-02}$	$7.50818175 \times 10^{-02}$
6	$4.29351236 \times 10^{-02}$	$4.29351236 \times 10^{-02}$
7	$2.47225482 \times 10^{-02}$	$2.47225488 \times 10^{-02}$
8	$1.42555622 \times 10^{-02}$	$1.42614966 \times 10^{-02}$
9	$8.23045267 \times 10^{-03}$	$8.23159055 \times 10^{-03}$
10	$4.75185407 \times 10^{-03}$	$4.75207278 \times 10^{-03}$
11	$2.74348422 \times 10^{-03}$	$2.74352676 \times 10^{-03}$

Os valores do grupo de Yoshida e os do programa TCSIAM são os mesmos até a iteração  $N = 6$  para os dois parâmetros de  $z$ . Um pequeno desvio durante as iterações 7 e 11 é obtida, a qual é no máximo no terceiro lugar depois do ponto.

Yoshida também calculou as energias logarítmicas da banda de condução com uma expressão exata e mostrou que os valores numéricos deles, mostrados nas duas tabelas acima, desviam bem pouco. Ele sugeriu, calcular os valores com a expressão recursiva, equação (6.5) até iteração  $N = 11$  e usar a fórmula assintótica, equação (6.9) para as seguintes iterações. Ele conseguiu garantir que os valores desviam no máximo 0,1% dos valores exatos com este procedimento.

Como os valores numéricos do TCSIAM são quase os mesmos do que os valores numéricos do Yoshida, eles foram considerados adequados para os cálculos

computacionais.

A energia da banda de condução da iteração inicial,  $N = -1$ , é diferente.

$$\xi_{-1,\alpha} = \sqrt{\frac{2}{\lambda}} V_\alpha \quad (6.10)$$

Este é a única vez que os parâmetros,  $V_1$  e  $V_2$ , entram diretamente no Modelo de Anderson de Dois Canais.

### 6.2.2 Cálculo analítico e computacional da iteração $N = 0$

Com a teoria de Grupo de Renormalização Numérica é possível diagonalizar analiticamente o hamiltoniano do Modelo de Anderson de Dois Canais com uma impureza para a iteração inicial [46]. Dessa maneira, os autovalores e os autovetores da iteração inicial, nomeado aqui  $N = -1$ , são obtidos analiticamente. A notação  $|j_1, j_2, s, r\rangle$ , onde o  $j_1$  é a carga axial do canal 1,  $j_2$  é a carga axial do canal 2,  $s$  é o spin total e  $r$  é o índice de contagem de um estado dentro do subespaço é usada. O índice de contagem guarda a informação de quantos estados com os mesmos números quânticos,  $j_1$ ,  $j_2$  e  $s$ , existem na iteração atual. Os 3 autovetores da iteração inicial não são degenerados, resultando em um índice de contagem igual a 1.

Tabela 6 – Autovalores com o parâmetro da discretização logarítmica  $\lambda$  e os auto-estados na iteração  $N = -1$ .

auto-estados	autovalores
$ 0, 0, \frac{1}{2}, 1\rangle$	$\tilde{E}_0 = E_0/\lambda$
$ 0, \frac{1}{2}, 0, 1\rangle$	$\tilde{E}_x = E_x/\lambda$
$ \frac{1}{2}, 0, 0, 1\rangle$	$\tilde{E}_x = E_x/\lambda$

Os invariantes da iteração inicial podem ser calculados analiticamente também e são [46]

$$\langle 1/2, 0, 0 || 0, 0, 1/2 \rangle = \sqrt{2}, \quad (6.11)$$

$$\langle 0, 1/2, 0 || 0, 0, 1/2 \rangle = \sqrt{2}. \quad (6.12)$$

Para a primeira iteração  $N = 0$  realizam-se os passos do processo de grupo de renormalização: criar filhos, determinar o índice de contagem, montar e diagonalizar os subespaços. O tamanho das matrizes ainda é razoável para fazer o cálculo analítico.

#### **Criar novos estados quânticos e determinar o índice de contagem**

Veja no anexo A como os tipos são usados para criar novos estados quânticos, filhos, pelos autovetores da iteração anterior, pais. Depois da criação, os estados com os mesmos números quânticos são contados e os índices de contagem são determinados.

Tabela 7 – Cálculo analítico: Os autovetores, pais, da iteração  $N = -1$  e os correspondentes estados, filhos, da iteração  $N = 0$ , criados pelas regras da tabela 15 no anexo A. A notação  $|j_1, j_2, s, r\rangle_t$  é usada. A letra  $t$ , tipo, indica qual é o procedimento que resultou neste estado.

$N = -1$	Filhos - $N = 0$	$N = -1$	Filhos $N = 0$	$N = -1$	Filhos $N = 0$
$ 0, 0, \frac{1}{2}, 1\rangle$	$ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1\rangle_1$	$ 0, \frac{1}{2}, 0, 1\rangle$	$ \frac{1}{2}, 1, 0, 1\rangle_1$	$ \frac{1}{2}, 0, 0, 1\rangle$	$ 1, \frac{1}{2}, 0, 1\rangle_1$
	$ \frac{1}{2}, 0, 1, 1\rangle_2$		$ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 2\rangle_2$		$ 1, 0, \frac{1}{2}, 1\rangle_2$
	$ \frac{1}{2}, 0, 0, 1\rangle_3$		$ \frac{1}{2}, 0, 0, 2\rangle_4$		$ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 3\rangle_5$
	$ 0, \frac{1}{2}, 1, 1\rangle_5$		$ 0, 1, \frac{1}{2}, 1\rangle_5$		$ \frac{1}{2}, 0, 1, 2\rangle_7$
	$ 0, \frac{1}{2}, 0, 1\rangle_6$		$ 0, \frac{1}{2}, 1, 2\rangle_7$		$ \frac{1}{2}, 0, 0, 3\rangle_8$
	$ 0, 0, \frac{3}{2}, 1\rangle_7$		$ 0, \frac{1}{2}, 0, 2\rangle_8$		$ 0, \frac{1}{2}, 0, 3\rangle_{13}$
	$ 0, 0, \frac{1}{2}, 1\rangle_8$		$ 0, 0, \frac{1}{2}, 3\rangle_{11}$		$ 0, 0, \frac{1}{2}, 4\rangle_{14}$
	$ 0, 0, \frac{1}{2}, 2\rangle_9$				

O programa TCSIAM cria os filhos e determina os índices de contagem, veja a figura 9, em confirmação com o cálculo analítico, mostrado nas tabelas 7 e 8.

```

Iteration: 0
Number of subspaces: 11
Dimension of H_N: 22
mat      j_1      j_2      s      dim      types
0      0.500000    0.500000    0.500000    3      1,2,5
1      0.500000    0.000000    1.000000    2      2,7
2      0.500000    0.000000    0.000000    3      3,4,8
3      0.000000    0.500000    1.000000    2      5,7
4      0.000000    0.500000    0.000000    3      6,8,13
5      0.000000    0.000000    1.500000    1      7
6      0.000000    0.000000    0.500000    4      8,9,11,14
7      0.500000    1.000000    0.000000    1      1
8      0.000000    1.000000    0.500000    1      5
9      1.000000    0.500000    0.000000    1      1
10     1.000000    0.000000    0.500000    1      2
    
```

Figura 9 – Programa TCSIAM: Número de subespaços, dimensão total do Hamiltoniano  $H_0$  e os tipos dos 11 subespaços da iteração  $N = 0$ .

### Montar os subespaços

Todos os estados com os mesmos números quânticos,  $j_1$ ,  $j_2$  e  $s$ , fundam um subespaço. Os tipos definem a base. O hamiltoniano do TCSIAM é hermitiano e por isso as matrizes reais dos subespaços são simétricas. Os subespaços da iteração  $N = 0$  têm os seguintes tamanhos.

Tabela 8 – Cálculo analítico: A iteração  $N = 0$  tem um subespaço com tamanho  $4 \times 4$ , três com tamanho  $3 \times 3$ , dois com tamanho  $2 \times 2$  e cinco subespaços com  $1 \times 1$ . O tamanho total do Hamiltoniano  $H_0$ , estabelecido pelos 11 subespaços, é 22.

Tamanho					
$4 \times 4$	$ 0, 0, \frac{1}{2}\rangle$				
$3 \times 3$	$ 0, \frac{1}{2}, 0\rangle$	$ \frac{1}{2}, 0, 0\rangle$	$ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle$		
$2 \times 2$	$ \frac{1}{2}, 0, 1\rangle$	$ 0, \frac{1}{2}, 1\rangle$			
$1 \times 1$	$ 1, 0, \frac{1}{2}\rangle$	$ 1, \frac{1}{2}, 0\rangle$	$ 0, 1, \frac{1}{2}\rangle$	$ \frac{1}{2}, 1, 0\rangle$	$ 0, 0, \frac{3}{2}\rangle$

O subespaço para o estado  $|0, 0, \frac{1}{2}\rangle$  é estabelecido pelos tipos 8, 9, 11 e 14.

$$H_0^{|0,0,\frac{1}{2}\rangle} = \frac{1}{\sqrt{\lambda}} \begin{pmatrix} E_0 & 0 & \sqrt{2}V_2 & \sqrt{2}V_1 \\ 0 & E_0 & \sqrt{6}V_2 & -\sqrt{6}V_1 \\ \sqrt{2}V_2 & \sqrt{6}V_2 & E_x & 0 \\ \sqrt{2}V_1 & -\sqrt{6}V_1 & 0 & E_x \end{pmatrix} \quad (6.13)$$

O subespaço para o estado  $|0, \frac{1}{2}, 0\rangle$  é estabelecido pelos tipos 6, 8 e 13.

$$H_0^{|0,\frac{1}{2},0\rangle} = \frac{1}{\sqrt{\lambda}} \begin{pmatrix} E_0 & -\sqrt{2}V_2 & 2\sqrt{2}V_1 \\ -\sqrt{2}V_2 & E_x & 0 \\ 2\sqrt{2}V_1 & 0 & E_x \end{pmatrix} \quad (6.14)$$

O subespaço para o estado  $|\frac{1}{2}, 0, 0\rangle$  é estabelecido pelos tipos 3, 4 e 8.

$$H_0^{|\frac{1}{2},0,0\rangle} = \frac{1}{\sqrt{\lambda}} \begin{pmatrix} E_0 & 2\sqrt{2}V_2 & -2\sqrt{2}V_1 \\ 2\sqrt{2}V_2 & E_x & 0 \\ -\sqrt{2}V_1 & 0 & E_x \end{pmatrix} \quad (6.15)$$

O subespaço para o estado  $|\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle$  é estabelecido pelos tipos 1, 2 e 5.

$$H_0^{|\frac{1}{2},\frac{1}{2},\frac{1}{2}\rangle} = \frac{1}{\sqrt{\lambda}} \begin{pmatrix} E_0 & \sqrt{2}V_2 & \sqrt{2}V_1 \\ \sqrt{2}V_2 & E_x & 0 \\ \sqrt{2}V_1 & 0 & E_x \end{pmatrix} \quad (6.16)$$

O subespaço para o estado  $|\frac{1}{2}, 0, 1\rangle$  é estabelecido pelos tipos 2 e 7.

$$H_0^{|\frac{1}{2},0,1\rangle} = \frac{1}{\sqrt{\lambda}} \begin{pmatrix} E_0 & -\sqrt{2}V_1 \\ -\sqrt{2}V_1 & E_x \end{pmatrix} \quad (6.17)$$

O subespaço para o estado  $|0, \frac{1}{2}, 1\rangle$  é estabelecido pelos tipos 5 e 7.

$$H_0^{|0, \frac{1}{2}, 1\rangle} = \frac{1}{\sqrt{\lambda}} \begin{pmatrix} E_0 & \sqrt{2}V_2 \\ \sqrt{2}V_2 & E_x \end{pmatrix} \quad (6.18)$$

Os seguintes subespaços são unidimensionais e por isso não precisam ser diagonalizados no próximo passo.

O subespaço para o estado  $|1, 0, \frac{1}{2}\rangle$  é estabelecido pelo tipo 2.

$$H_0^{|1, 0, \frac{1}{2}\rangle} = \frac{1}{\sqrt{\lambda}} E_x \quad (6.19)$$

O subespaço para o estado  $|1, \frac{1}{2}, 0\rangle$  é estabelecido pelo tipo 1.

$$H_0^{|1, \frac{1}{2}, 0\rangle} = \frac{1}{\sqrt{\lambda}} E_x \quad (6.20)$$

O subespaço para o estado  $|0, 1, \frac{1}{2}\rangle$  é estabelecido pelo tipo 5.

$$H_0^{|0, 1, \frac{1}{2}\rangle} = \frac{1}{\sqrt{\lambda}} E_x \quad (6.21)$$

O subespaço para o estado  $|\frac{1}{2}, 1, 0\rangle$  é estabelecido pelo tipo 1.

$$H_0^{|\frac{1}{2}, 1, 0\rangle} = \frac{1}{\sqrt{\lambda}} E_x \quad (6.22)$$

O subespaço para o estado  $|0, 0, \frac{3}{2}\rangle$  é estabelecido pelo tipo 7.

$$H_0^{|0, 0, \frac{3}{2}\rangle} = \frac{1}{\sqrt{\lambda}} E_0 \quad (6.23)$$

### Diagonalização dos subespaços

O programa *Mathematica*, versão 11.1.1.0 com uma licença de estudante é usado para diagonalizar os subespaços (6.13-6.18) analiticamente. Estes resultados analíticos servem como um controle dos resultados numéricos do novo programa TCSIAM. Os seguintes parâmetros de padrão são usados.

Tabela 9 – Parâmetros padrões da discretização logarítmica,  $\lambda$  e  $z$ , e do Modelo de Anderson de Dois Canais,  $E_0, E_x, V_1$  e  $V_2$ .

Parâmetros padrões					
$\lambda$	$z$	$E_0$	$E_x$	$V_1$	$V_2$
3	1	-0.001	0.002	0.001	0.003

As funções *Eigenvectors* e *Eigenvalues* da *Mathematica* fazem a diagonalização. Os autovetores  $\mathbf{v}_i$  com os correspondentes autovalores  $\nu_i$  são mostrados em seguinte para todas matrizes da iteração  $N = 0$ . O problema de autovalor ainda tem um grau de liberdade. Cada autovetor pode ter uma fase. As fases da biblioteca *Mathematica* e *Eigen* são diferentes. Os resultados analíticos são adaptados para que eles correspondem aos numéricos. A mudança somente se manifesta no sinal. Os subespaços são montados sem os prefatores da seção 6.2.1, porque eles não são necessários para a comparação com o programa TCSIAM. Os elementos das matrizes são mostrados com 8 dígitos, enquanto os autovetores são com 6, que é a saída padrão da biblioteca *Eigen* para matrizes e vetores. Os cálculos mesmo, no programa *Mathematica* e no TCSIAM, sempre são feitos com precisão quádrupla.

O subespaço  $|0, 0, \frac{1}{2}\rangle$  na base dos tipos 8, 9, 11 e 14:

$$H_0^{|0,0,\frac{1}{2}\rangle} = \begin{pmatrix} -0.00057735 & 0 & 0.00244949 & 0.000816497 \\ 0 & -0.00057735 & 0.00424264 & -0.00141421 \\ 0.00244949 & 0.00424264 & 0.0011547 & 0 \\ 0.000816497 & -0.00141421 & 0 & 0.0011547 \end{pmatrix} \quad (6.24)$$

Os autovetores e autovalores do subespaço  $|0, 0, \frac{1}{2}\rangle$ :

$$\begin{aligned} \mathbf{v}_1 = \begin{pmatrix} 0.293097 \\ 0.572978 \\ 0.753088 \\ -0.136561 \end{pmatrix} \quad \mathbf{v}_2 = \begin{pmatrix} 0.348556 \\ 0.681395 \\ -0.633264 \\ 0.114833 \end{pmatrix} \quad \mathbf{v}_3 = \begin{pmatrix} -0.432546 \\ 0.221261 \\ -0.155951 \\ -0.860016 \end{pmatrix} \\ \mathbf{v}_4 = \begin{pmatrix} 0.778143 \\ -0.398046 \\ -0.0866881 \\ -0.478056 \end{pmatrix} \quad \begin{aligned} \nu_1 &= 0.00533599 \\ \nu_2 &= -0.00475863 \\ \nu_3 &= 0.0019292 \\ \nu_4 &= -0.00135185 \end{aligned} \end{aligned} \quad (6.25)$$



O subespaço  $|0, \frac{1}{2}, 0\rangle$  na base dos tipos 6, 8 e 13:

$$H_0^{|0, \frac{1}{2}, 0\rangle} = \begin{pmatrix} -0.00057735 & -0.00244949 & 0.00163299 \\ -0.00244949 & 0.0011547 & 0 \\ 0.00163299 & 0 & 0.0011547 \end{pmatrix} \quad (6.26)$$

Os autovetores e autovalores do subespaço  $|0, \frac{1}{2}, 0\rangle$ :

$$\begin{aligned} \mathbf{v}_1 &= \begin{pmatrix} -0.599076 \\ 0.666216 \\ -0.444144 \end{pmatrix} & \mathbf{v}_2 &= \begin{pmatrix} -0.800692 \\ -0.498461 \\ 0.332307 \end{pmatrix} & \mathbf{v}_3 &= \begin{pmatrix} 0 \\ 0.5547 \\ 0.83205 \end{pmatrix} \\ \nu_1 &= 0.00335733 & \nu_2 &= -0.00277998 & \nu_3 &= 0.0011547 \end{aligned} \quad (6.27)$$

O subespaço  $|\frac{1}{2}, 0, 0\rangle$  na base dos tipos 3, 4 e 8:

$$H_0^{|\frac{1}{2}, 0, 0\rangle} = \begin{pmatrix} -0.00057735 & 0.00489898 & -0.000816497 \\ 0.00489898 & 0.0011547 & 0 \\ -0.000816497 & 0 & 0.0011547 \end{pmatrix} \quad (6.28)$$

Os autovetores e autovalores do subespaço  $|\frac{1}{2}, 0, 0\rangle$ :

$$\begin{aligned} \mathbf{v}_1 &= \begin{pmatrix} 0.643514 \\ 0.75502 \\ -0.125837 \end{pmatrix} & \mathbf{v}_2 &= \begin{pmatrix} -0.765434 \\ 0.634758 \\ -0.105793 \end{pmatrix} & \mathbf{v}_3 &= \begin{pmatrix} 0 \\ 0.164399 \\ 0.986394 \end{pmatrix} \\ \nu_1 &= 0.00533017 & \nu_2 &= -0.00475282 & \nu_3 &= 0.0011547 \end{aligned} \quad (6.29)$$

O subespaço  $|\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle$  na base dos tipos 1, 2 e 5:

$$H_0^{|\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle} = \begin{pmatrix} -0.00057735 & 0.00244949 & 0.000816497 \\ 0.00244949 & 0.0011547 & 0 \\ 0.000816497 & 0 & 0.0011547 \end{pmatrix} \quad (6.30)$$

Os autovetores e autovalores do subespaço  $|\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle$ :

$$\begin{aligned} \mathbf{v}_1 &= \begin{pmatrix} 0.583952 \\ 0.77013 \\ 0.25671 \end{pmatrix} & \mathbf{v}_2 &= \begin{pmatrix} -0.811788 \\ 0.553986 \\ 0.184662 \end{pmatrix} & \mathbf{v}_3 &= \begin{pmatrix} 0 \\ -0.316228 \\ 0.948683 \end{pmatrix} \\ \nu_1 &= 0.00301203 & \nu_2 &= -0.00243468 & \nu_3 &= 0.0011547 \end{aligned} \quad (6.31)$$

O subespaço  $|\frac{1}{2}, 0, 1\rangle$  na base dos tipos 2 e 7:

$$H_0^{|\frac{1}{2}, 0, 1\rangle} = \begin{pmatrix} -0.00057735 & -0.000816497 \\ -0.000816497 & 0.0011547 \end{pmatrix} \quad (6.32)$$

Os autovetores e autovalores do subespaço  $|\frac{1}{2}, 0, 1\rangle$ :

$$\begin{aligned} \mathbf{v}_1 &= \begin{pmatrix} 0.369048 \\ -0.92941 \end{pmatrix} & \mathbf{v}_2 &= \begin{pmatrix} -0.92941 \\ -0.369048 \end{pmatrix} \\ \nu_1 &= 0.00147891 & \nu_2 &= -0.000901563 \end{aligned} \quad (6.33)$$

O subespaço  $|0, \frac{1}{2}, 1\rangle$  na base dos tipos 5 e 7:

$$H_0^{|0, \frac{1}{2}, 1\rangle} = \begin{pmatrix} -0.00057735 & 0.00244949 \\ 0.00244949 & 0.0011547 \end{pmatrix} \quad (6.34)$$

Os autovetores e autovalores do subespaço  $|0, \frac{1}{2}, 1\rangle$ :

$$\begin{aligned} \mathbf{v}_1 &= \begin{pmatrix} -0.57735 \\ -0.816497 \end{pmatrix} & \mathbf{v}_2 &= \begin{pmatrix} -0.816497 \\ 0.57735 \end{pmatrix} \\ \nu_1 &= 0.00288675 & \nu_2 &= -0.0023094 \end{aligned} \quad (6.35)$$

Determinando as matrizes com os parâmetros padrões, tabela 9, e encontrando todos os autovalores da iteração  $N = 0$  analiticamente, eles podem ser usados para conferir se o processo de calcular os elementos de matriz e a diagonalização do programa TCSIAM funcionam corretamente. Os três seguintes figuras mostram que de fato os cálculos numéricos e analíticos resultam nos mesmos valores. Por isso, pode ser concluído que o TCSIAM calcula os elementos de matriz corretamente para a iteração inicial.

```

Subspace: |0.000000,0.000000,0.500000>
-0.00057735      0      0.00244949 0.000816497
      0 -0.00057735  0.00424264 -0.00141421
  0.00244949  0.00424264  0.0011547      0
0.000816497 -0.00141421      0      0.0011547
Types:
8      9      11      14
Eigenvector number j: 0
Representation of eigenvector number j
  0.348556
  0.681395
-0.633264
  0.114833
Eigenvalue of vec number j: -0.00475863
Eigenvector number j: 1
Representation of eigenvector number j
  0.778143
-0.398046
-0.0866881
-0.478056
Eigenvalue of vec number j: -0.00135185
Eigenvector number j: 2
Representation of eigenvector number j
-0.432546
  0.221261
-0.155951
-0.860016
Eigenvalue of vec number j: 0.0019292
Eigenvector number j: 3
Representation of eigenvector number j
  0.293097
  0.572978
  0.753088
-0.136561
Eigenvalue of vec number j: 0.00533599

```

Figura 10 – Programa TCSIAM: Subespaço  $|0, 0, \frac{1}{2}\rangle$  na iteração  $N = 0$  com a base dos tipos 8, 9, 11 e 14. Os 4 eigeenvetores e autovalores correspondentes são mostrados.

```

Subspace: |0.000000,0.500000,0.000000>
-0.00057735 -0.00244949 0.00163299
-0.00244949 0.0011547 0
0.00163299 0 0.0011547
Types:
6      8      13
Eigenvector number j: 0
Representation of eigenvector number j
-0.800692
-0.498461
0.332307
Eigenvalue of vec number j: -0.00277998
Eigenvector number j: 1
Representation of eigenvector number j
0
0.5547
0.83205
Eigenvalue of vec number j: 0.0011547
Eigenvector number j: 2
Representation of eigenvector number j
-0.599076
0.666216
-0.444144
Eigenvalue of vec number j: 0.00335733

Subspace: |0.500000,0.000000,0.000000>
-0.00057735 0.00489898 -0.000816497
0.00489898 0.0011547 0
-0.000816497 0 0.0011547
Types:
3      4      8
Eigenvector number j: 0
Representation of eigenvector number j
-0.765434
0.634758
-0.105793
Eigenvalue of vec number j: -0.00475282
Eigenvector number j: 1
Representation of eigenvector number j
0
0.164399
0.986394
Eigenvalue of vec number j: 0.0011547
Eigenvector number j: 2
Representation of eigenvector number j
0.643514
0.75502
-0.125837
Eigenvalue of vec number j: 0.00533017

Subspace: |0.500000,0.500000,0.500000>
-0.00057735 0.00244949 0.000816497
0.00244949 0.0011547 0
0.000816497 0 0.0011547
Types:
1      2      5
Eigenvector number j: 0
Representation of eigenvector number j
-0.811788
0.553986
0.184662
Eigenvalue of vec number j: -0.00243468
Eigenvector number j: 1
Representation of eigenvector number j
0
-0.316228
0.948683
Eigenvalue of vec number j: 0.0011547
Eigenvector number j: 2
Representation of eigenvector number j
0.583952
0.77013
0.25671
Eigenvalue of vec number j: 0.00301203
    
```

- (a) Subespaço  $|0, \frac{1}{2}, 0\rangle$  com a base dos tipos 6, 8 e 13. Os 3 eivenvetores e autovalores correspondentes são mostrados.
- (b) Subespaço  $|\frac{1}{2}, 0, 0\rangle$  com a base dos tipos 3, 4 e 8. Os 3 eivenvetores e autovalores correspondentes são mostrados.
- (c) Subespaço  $|\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle$  com a base dos tipos 1, 2 e 5. Os 3 eivenvetores e autovalores correspondentes são mostrados.

Figura 11 – Programa TCSIAM: Os três subespaços tridimensional na iteração  $N = 0$ .

```

Subspace: |0.500000,0.000000,1.000000>
-0.00057735 -0.000816497
-0.000816497 0.0011547
Types:
2      7
Eigenvector number j: 0
Representation of eigenvector number j
-0.92941
-0.369048
Eigenvalue of vec number j: -0.000901563
Eigenvector number j: 1
Representation of eigenvector number j
0.369048
-0.92941
Eigenvalue of vec number j: 0.00147891

Subspace: |0.000000,0.500000,1.000000>
-0.00057735 0.00244949
0.00244949 0.0011547
Types:
5      7
Eigenvector number j: 0
Representation of eigenvector number j
-0.816497
0.57735
Eigenvalue of vec number j: -0.0023094
Eigenvector number j: 1
Representation of eigenvector number j
-0.57735
-0.816497
Eigenvalue of vec number j: 0.00288675
    
```

- (a) Subespaço  $|\frac{1}{2}, 0, 1\rangle$  com a base dos tipos 2 e 7. Os 2 eivenvetores e autovalores correspondentes são mostrados.
- (b) Subespaço  $|0, \frac{1}{2}, 1\rangle$  com a base dos tipos 5 e 7. Os 2 eivenvetores e autovalores correspondentes são mostrados.

Figura 12 – Programa TCSIAM: Os dois subespaços bidimensional na iteração  $N = 0$ .

### 6.2.3 Cálculo analítico e computacional da iteração $N = 1$

Tabela 10 – Cálculo analítico: Os 28 subespaços da iteração  $N = 1$ . O tamanho total do Hamiltoniano  $H_1$  é 218.

Tamanho					
$24 \times 24$	$ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\rangle$				
$22 \times 22$	$ 0, 0, \frac{1}{2}\rangle$				
$18 \times 18$	$ \frac{1}{2}, 0, 1\rangle$	$ 0, \frac{1}{2}, 1\rangle$			
$17 \times 17$	$ \frac{1}{2}, 0, 0\rangle$	$ 0, \frac{1}{2}, 0\rangle$			
$11 \times 11$	$ 1, 0, \frac{1}{2}\rangle$	$ 0, 1, \frac{1}{2}\rangle$			
$10 \times 10$	$ 0, 0, \frac{3}{2}\rangle$				
$9 \times 9$	$ 1, \frac{1}{2}, 0\rangle$	$ \frac{1}{2}, 1, 0\rangle$			
$8 \times 8$	$ \frac{1}{2}, \frac{1}{2}, \frac{3}{2}\rangle$				
$7 \times 7$	$ 1, \frac{1}{2}, 1\rangle$	$ \frac{1}{2}, 1, 1\rangle$			
$5 \times 5$	$ 1, 1, \frac{1}{2}\rangle$				
$3 \times 3$	$ 1, 0, \frac{3}{2}\rangle$	$ \frac{1}{2}, 0, 2\rangle$	$ 0, 1, \frac{3}{2}\rangle$	$ 0, \frac{1}{2}, 2\rangle$	
$2 \times 2$	$ \frac{1}{2}, \frac{3}{2}, \frac{1}{2}\rangle$	$ 0, \frac{3}{2}, 0\rangle$	$ \frac{3}{2}, \frac{1}{2}, \frac{1}{2}\rangle$	$ \frac{3}{2}, 0, 0\rangle$	
$1 \times 1$	$ 0, 0, \frac{5}{2}\rangle$	$ 1, \frac{3}{2}, 0\rangle$	$ 0, \frac{3}{2}, 1\rangle$	$ \frac{3}{2}, 1, 0\rangle$	$ \frac{3}{2}, 0, 1\rangle$

```

Iteration: 1
Number of subspaces: 28
Dimension of H_N: 218
mat      j_1      j_2      s          dim      types
0      1.000000    1.000000    0.500000     5      1,1,1,2,5
1      1.000000    0.500000    1.000000     7      1,1,2,2,2,5,7
2      1.000000    0.500000    0.000000     9      1,1,1,3,3,3,4,6,8
3      1.000000    0.000000    0.500000    11      2,2,2,3,3,4,4,4,8,9,11
4      0.500000    1.000000    1.000000     7      1,1,2,5,5,5,7
5      0.500000    1.000000    0.000000     9      1,1,1,3,6,6,6,8,13
6      0.500000    0.500000    1.500000     8      1,2,2,5,5,7,7,7
7      0.500000    0.500000    0.500000    24      1,1,1,1,2,2,2,3,3,4,5,5,6,6,8,8,8,9,9,9,11,13,14
8      0.500000    0.000000    1.000000    18      2,2,2,2,3,4,4,7,7,7,8,8,9,9,11,11,11,14
9      0.500000    0.000000    0.000000    17      3,3,3,3,4,4,4,8,8,8,10,10,12,12,12,15,16
10     0.000000    1.000000    0.500000    11      5,5,5,6,6,8,9,13,13,13,14
11     0.000000    1.000000    1.000000    18      5,5,5,5,6,7,7,7,8,8,9,9,11,13,13,14,14,14
12     0.000000    0.500000    0.000000    17      6,6,6,6,8,8,8,10,10,12,13,13,13,15,15,15,16
13     0.000000    0.000000    0.500000    22      8,8,8,8,9,9,9,9,10,11,11,11,12,12,14,14,14,15,15,16,16,16
14     1.000000    0.000000    1.500000     3      2,2,7
15     0.500000    0.000000    2.000000     3      2,7,7
16     0.000000    0.000000    1.500000    10      7,7,7,7,8,9,11,11,14,14
17     0.000000    1.000000    1.500000     3      5,5,7
18     0.000000    0.500000    2.000000     3      5,7,7
19     0.000000    0.000000    2.500000     1      7
20     1.000000    1.500000    0.000000     1      1
21     0.500000    1.500000    0.500000     2      1,5
22     0.000000    1.500000    0.000000     2      6,13
23     0.000000    1.500000    1.000000     1      5
24     1.500000    1.000000    0.000000     1      1
25     1.500000    0.500000    0.500000     2      1,2
26     1.500000    0.000000    0.000000     2      3,4
27     1.500000    0.000000    1.000000     1      2
    
```

Figura 13 – Programa TCSIAM: Número de subespaços, dimensão total do Hamiltoniano  $H_1$  e os tipos dos 28 subespaços da iteração  $N = 1$ .

Os tamanhos das matrizes e os números quânticos dos subespaços na iteração  $N = 1$  do cálculo analítico, veja a tabela 10, e daqueles do programa TCSIAM,

veja a figura 13, são os mesmos. Como resultado, o cálculo analítico confirma que o programa monta os subespaços corretamente.

Para conferir que a computação dos invariantes coletivos funciona propriamente no TCSIAM, os subespaços até a dimensão 7 na iteração  $N = 1$  foram calculados analiticamente e comparados com os resultados numéricos.

O cálculo analítico do subespaço  $|1, 1, \frac{1}{2}\rangle$  que é estabelecido pelos tipos 1, 2 e 5, é mostrado como exemplo. O estado de tipo 1 vem de um subespaço tridimensional e por isso gera 3 estados com os números quânticos  $j_1 = 1$ ,  $j_2 = 1$  e  $s = \frac{1}{2}$ . Os estados dos tipos 2 e 5 são de subespaços unidimensionais e contribuem somente uma vez.

$$H_1^{|1,1,\frac{1}{2}\rangle} = \begin{matrix} & 1 & 1 & 1 & 2 & 5 \\ \begin{matrix} 1 \\ 1 \\ 1 \\ 2 \\ 5 \end{matrix} & \begin{pmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{pmatrix} & \end{matrix} \quad (6.36)$$

Os elementos diagonais são os autovalores da iteração anterior, veja seção 6.2.2, multiplicado por raiz de  $\lambda$ . A notação usada no seguinte é  $\langle t'|H|t\rangle_N^{r',r}$ , onde  $t$  e  $t'$  indicam os tipos,  $N$  o número da iteração atual e  $r$  e  $r'$  os índices de contagem para diferenciar autovetores do mesmo subespaço da iteração anterior.

$$\langle 1|H|1\rangle_1^{1,1} = \sqrt{3} \cdot 0.00301203 = 5.2169 \cdot 10^{-3} \quad (6.37)$$

$$\langle 1|H|1\rangle_1^{2,2} = \sqrt{3} \cdot (-0.00243468) = -4.2169 \cdot 10^{-3} \quad (6.38)$$

$$\langle 1|H|1\rangle_1^{3,3} = \sqrt{3} \cdot 0.0011547 = 1.9999 \cdot 10^{-3} \quad (6.39)$$

$$\langle 2|H|2\rangle_1^{1,1} = \sqrt{\lambda} \cdot \frac{1}{\sqrt{\lambda}} \cdot E_x = 0.002 \quad (6.40)$$

$$\langle 5|H|5\rangle_1^{1,1} = \sqrt{\lambda} \cdot \frac{1}{\sqrt{\lambda}} \cdot E_x = 0.002 \quad (6.41)$$

Os elementos não diagonais que vem do mesmo procedimento, têm os mesmos tipos, são zeros, porque os autovetores da iteração anterior do mesmo

subespaço são ortogonais.

$$\langle 1|H|1\rangle_1^{1,2} = 0 \quad (6.42)$$

$$\langle 1|H|1\rangle_1^{1,3} = 0 \quad (6.43)$$

$$\langle 1|H|1\rangle_1^{2,3} = 0 \quad (6.44)$$

Os tipos que não tem interação, veja as tabelas 16 e 17 para todas interações possíveis, são zeros também. Esse é o caso para o tipo 2 e 5 aqui.

$$\langle 2|H|5\rangle_1^{1,1} = 0 \quad (6.45)$$

Os outros elementos não diagonais são definidos pelos prefatores, veja seção 6.2.1, multiplicado pelos fatores e invariantes coletivos, explicado no apêndice B. Os invariantes coletivos mesmos são constituídos pelos invariantes individuais. Os prefatores não são considerados aqui porque eles não são necessários para a comparação com os resultados do programa TCSIAM. Veja capítulo 3.3.5 para a notação dos estados no invariante coletivo e seção 6.2.2 para os valores. A representação do invariante coletivo é  $\langle j'_1, j'_2, s' | canal | j_1, j_2, s \rangle_{N-1}$  e dos estados somente pelos tipos  $|t\rangle$ . A forma mais completa dos estados seria  $|j_1, j_2, s, t\rangle$ .

$$\langle 1|H|2\rangle_1^{1,1} = -1\sqrt{\frac{1}{2}} \underbrace{\langle \frac{1}{2}, 1, 0 || 2 || \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \rangle_0}_{\text{invariante coletivo}} \quad (6.46)$$

$$= -1\sqrt{\frac{1}{2}} \left[ \underbrace{1 \langle 1 || 2 || 1 \rangle_0}_{\text{invariante individual} = 0} 0.583952 + \underbrace{1 \langle 1 || 2 || 2 \rangle_0}_{\text{invariante individual} \neq 0} 0.77013 + \right. \\ \left. \underbrace{1 \langle 1 || 2 || 5 \rangle_0}_{\text{invariante individual} = 0} 0.25671 \right] \quad (6.47)$$

O primeiro termo é zero porque os invariantes individuais nunca têm uma interação entre os mesmos tipos. O terceiro termo é zero porque os tipos 1 e 5 não combinam no canal 2. O invariante individual dos tipos 1 e 2 dá um valor não nulo no canal 2. Veja a tabela 19 para a formula do invariante individual desses tipos e nesse canal. Importante é que os números quânticos  $j'_1$ ,  $j'_2$  e  $s'$  do vetor *bra* entram nesse cálculo. O resultado é  $-\sqrt{2}$  aqui. O apêndice B mostra todas as fórmulas das

contribuições possíveis.

$$\begin{aligned} \langle 1|H|2\rangle_1^{1,1} &= -1\sqrt{\frac{1}{2}} \underbrace{\langle \frac{1}{2}, 1, 0 || 2 || \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \rangle_0}_{\text{invariante coletivo}} = -1\sqrt{\frac{1}{2}} \underbrace{1 \langle \frac{1}{2}, 1, 0, 1 || 2 || \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 2 \rangle_0}_{\text{invariante individual}} 0.77013 \\ &= 0.77013 \end{aligned}$$

Para os outros elementos de matriz somente os resultados são mostrados.

$$\langle 1|H|2\rangle_1^{1,2} = 0.553986 \quad (6.48)$$

$$\langle 1|H|2\rangle_1^{1,3} = -0.316228 \quad (6.49)$$

$$\langle 1|H|5\rangle_1^{1,1} = 0.25671 \quad (6.50)$$

$$\langle 1|H|5\rangle_1^{1,2} = 0.184662 \quad (6.51)$$

$$\langle 1|H|5\rangle_1^{1,3} = 0.948683 \quad (6.52)$$

O subespaço hermitiano dos números quânticos  $j_1 = 1$ ,  $j_2 = 1$  e  $s = \frac{1}{2}$  estabelecido pelos tipos 1, 2 e 5 na iteração  $N = 1$  é mostrado em seguida com os elementos de matriz até três dígitos.

$$H_1^{[1,1,\frac{1}{2}]} = \begin{matrix} & \begin{matrix} 1 & 1 & 1 & 2 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 1 \\ 1 \\ 2 \\ 5 \end{matrix} & \left( \begin{array}{ccccc} 5.216 \cdot 10^{-3} & 0 & 0 & 0.770 & 0.257 \\ 0 & -4.216 \cdot 10^{-3} & 0 & 0.554 & 0.185 \\ 0 & 0 & 1.999 \cdot 10^{-3} & -0.316 & 0.949 \\ 0.770 & 0.554 & -0.316 & 0.002 & 0 \\ 0.257 & 0.185 & 0.949 & 0 & 0.002 \end{array} \right) \end{matrix}$$

O programa TCSIAM constrói o mesmo subespaço na mesma iteração corretamente, veja a figura 14. Somente a ordem dos estados que vêm do mesmo subespaço anterior é diferente. Essa ordem variada dos vetores da base não muda os autovalores e autovetores resultando da diagonalização. De fato, apenas os autovalores determinam as grandezas físicas e por isso a ordem do cálculo analítico e da computação numérica não precisam ser igual.



```

Iteration N = 1
Compute Matrix i: 0 with size 5
Subspace: |1.000000,1.000000,0.500000>
-0.00421699      0      0      0.553986      0.184662
      0      0.002      0      -0.316228      0.948683
      0      0      0.00521699      0.77013      0.25671
      0.553986      -0.316228      0.77013      0.002      0
      0.184662      0.948683      0.25671      0      0.002
Types:
1      1      1      2      5

```

Figura 14 – Programa TCSIAM: Representação de matriz do subespaço dos números quânticos  $j_1 = 1$ ,  $j_2 = 1$  e  $s = \frac{1}{2}$  na iteração  $N = 1$ . Os tipos, 1, 1, 1, 2 e 5, que estabelecem o subespaço cinco dimensional.

Para todas as matrizes até dimensão 7, veja a tabela 10, os cálculos foram feitos para obter resultados analíticos para comparar com os resultados numéricos do programa TCSIAM. Apesar da ordem dos vetores da base, todos verificam os resultados.

Para ter certeza absoluta que o novo código determina os invariantes coletivos propriamente, seria necessário calcular as matrizes maiores de dimensão 7 da iteração  $N = 1$  também. Por falta de tempo nem todas matrizes foram calculadas analiticamente. Uma divergência das computações numéricas não é esperada pelas outras matrizes porque o procedimento de cálculo é o mesmo.

## 6.2.4 Programa computacional em C++

O novo programa, que tem o nome do modelo TCSIAM, 'Two Channel Single Impurity Anderson Model', foi escrito em C++ moderno a partir do zero. A intenção é de estabelecer um novo código computacional que é bem documentado, para facilitar o uso por outros pesquisadores, e modular, para deixar adaptações futuras simples. Além disso, o novo TCSIAM serve como confirmação dos resultados do programa antigo MADC, Modelo de Anderson de Dois Canais, do Dr. J. V. B. Ferreira. C++ moderno significa aqui o uso extensivo da biblioteca padrão STL, 'standard template library', e da biblioteca bem testada *Eigen*. Além disso, aplica a mistura de estilos de programação: processual, modular, orientado a objetos e funcional.

O novo código é escrito inteiramente em inglês e usa a nomenclatura que se orienta nas 'Google C++ Style Guide', Google guias de estilo de C++ [50]. Os nomes dos arquivos do sistema operacional seguem a notação do 'Camel Case', por exemplo 'createStates.cpp'. Ao contrário, as classes sempre começam com 'TCSIAM', por exemplo 'TCSIAMParameter'. As propriedades, ou atributos, e as funções de membros das classes, ou métodos, usam nomes em minúsculos e incluem sublinhados para separar expressões longas, por exemplo 'file\_path\_read\_rel' e 'set\_default\_parameter'.

As quatro classes do Modelo de Anderson de Dois Canais são apresentadas a seguir. Primeiro, as propriedades mais importantes são explicadas e depois a importância da classe dentro do programa é descrita. Todos os atributos das classes são privados para controlar o acesso e a mudança deles melhor. As classes seguem a ideologia de encapsulamento. Por isso, cada propriedade de uma classe tem um 'setter', para mudar o valor, e 'getter', para receber o valor. Além disso, todas as classes tem um construtor para criar um objeto desta classe e um destrutor para poder deletar essa instância. Esses métodos são públicos. Sem isso, um acesso ou a mudança dos valores seria impossível. A implementação é mostrado baseado no exemplo da classe 'TCSIAMState' na figura 15.

A classe 'TCSIAMState' define um estado no processo iterativo do Grupo de Renormalização Numérica. Ela tem sete atributos que são explicados na tabela

seguinte.

Tabela 11 – Propriedades e as explicações delas da classe 'TCSIAMState'.

propriedade	explicação
$j_1$	Número quântico: carga axial do canal 1
$j_2$	Número quântico: carga axial do canal 2
$s$	Número quântico: spin total
$t$	Número quântico: tipo de canal a partir
$r$	Índice de contagem, enumeração dos estados com os mesmos números quânticos $j_1$ , $j_2$ e $s$ para iteração atual
$N$	Iteração em qual estado é criado
$*p$	Ponteiro para criador, autovetor da iteração anterior

Os estados para a iteração atual são criados na função 'createStates' e armazenados no atributo *state* do objeto da classe 'TCSIAMResult'. Todos os estados com os mesmos números quânticos  $j_1$ ,  $j_2$  e  $s$  constroem um subespaço. A propriedade  $r$ , índice de contagem, é usada para distinguir os estados com os mesmos números quânticos. O índice de contagem é definido na função 'indexing'. Um ponteiro *bruto* para o autovetor da iteração anterior que criou o estado é guardado no atributo  $*p$ . Esta referência ao seu criador é necessária para o cálculo dos elementos de matriz.

```

7 class TCSIAMState {
8
9     private:
10         /// Quantum number: Axial charge channel 1
11         Real    j1;
12         /// Quantum number: Axial charge channel 2
13         Real    j2;
14         /// Quantum number: Total spin
15         Real    s;
16         /// Quantum number: Type of channel from which state was created
17         unsigned t;
18         /// Counting index for current iteration, enumeration of states with the same
19         unsigned r;
20         /// Iteration in which state was created
21         unsigned N;
22         /// Pointer to creator, eigenvector of previous iteration
23         TCSIAMEigVec *p;
24
25     public:
26         TCSIAMState();
27         TCSIAMState(    const Real j1, const Real j2, const Real s,
28                       const unsigned t, const unsigned r, const unsigned N,
29                       TCSIAMEigVec *const p);
30         ~TCSIAMState();
31
32         void set_j1(const Real &input);
33         void set_j2(const Real &input);
34         void set_s(const Real &input);
35         void set_t(const unsigned &input);
36         void set_r(const unsigned &input);
37         void set_N(const unsigned &input);
38         void set_p(TCSIAMEigVec *const input);
39
40         const Real&    get_j1() const;
41         const Real&    get_j2() const;
42         const Real&    get_s() const;
43         const unsigned& get_t() const;
44         const unsigned& get_r() const;
45         const unsigned& get_N() const;
46         TCSIAMEigVec*  get_p() const;
47 };

```

Figura 15 – O 'headerfile', 'TCSIAMState.hpp', da classe 'TCSIAMState' do programa TCSIAM. As propriedades privadas da classe,  $j_1$ ,  $j_2$ ,  $s$ ,  $t$ ,  $r$ ,  $N$  e  $*p$  são mostrados. O construtor de padrão sem argumento de entrada é na linha 26. Aquele com todos os valores para os atributos fica na próxima linha. Todos os 'setters' com as referências para os argumentos de entrada estão entre a linha 32 e 38. Os correspondentes 'getters' são entre 40 e 46.

Ler, escrever e armazenar todos os parâmetros relativos ao TCSIAM são as tarefas da classe 'TCSIAMParameter'. Como o Modelo de Anderson de Dois Canais e a implementação do Grupo de Renormalização Numérica são bem grandes e complicados, o 'TCSIAMParameter' tem muitos atributos. Os mais importantes são explicados aqui.

Tabela 12 – Propriedades e as explicações delas da classe 'TCSIAMParameter'.

propriedade	explicação
$v1\_j1$	Número quântico: carga axial do canal 1 do auto-estado fundamental da iteração inicial
$v1\_j2$	Número quântico: carga axial do canal 2 do auto-estado fundamental da iteração inicial
$v1\_s$	Número quântico: spin total do auto-estado fundamental da iteração inicial
$E_0$	Autovalor do auto-estado fundamental, valor típico 0.005 D
$E_x$	Autovalor dos auto-estados excitados, valor típico 0.0001 D
$V_1$	Intensidade de hibridização do canal 1
$V_2$	Intensidade de hibridização do canal 2
$D$	Largura da banda de condução, valor típico 5 eV
$lam$	Primeiro parâmetro de discretização da banda de condução, valor típico $\lambda = 6$
$z$	Segundo parâmetro de discretização da banda de condução, expoente de $\lambda$ , valores típicos $\lambda = 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0, 1.125$
$N$	Número da iteração atual
$C\_mode$	Booleano para determinar se o corte da energia é aplicado
$C$	Energia de corte, valor típico 18 D

O objeto da classe 'TCSIAMParameter' é criado no construtor da classe 'TCSIAMResult' e salvo no seu atributo, tal que todos os parâmetros são acessíveis onde o objeto 'result' é usado. Parâmetros são os valores do modelo ( $E_0$ ,  $E_x$ ,  $V_1$ ,  $V_2$ ) e os parâmetros numéricos ( $\lambda$ ,  $z$ ,  $C$ ,  $N_{max}$ ) e os valores do sistema, que são específico para o TCSIAM e não podem ser alterados. Valores padrões para todos os parâmetros são usados se a função de membro 'read\_parameter' não for chamado. Para usar parâmetros definidos pelo usuário para o TCSIAM, veja o executável 'calculation' na figura 16.

A terceira classe 'TCSIAMEigVec' é derivada da classe 'TCSIAMState' e herda todas as propriedades dela. Ela determina a estrutura de um autovetor no processo iterativo do Grupo de Renormalização Numérica. Como um auto-estado é um estado especial, a classe tem algumas propriedades adicionais.

Tabela 13 – Propriedades e as explicações delas da classe 'TCSIAMEigVec'.

propriedade	explicação
<i>values</i>	Componentes do autovetor
<i>types</i>	Vetores da base do autovetor que são os tipos $t$
<i>eig_val</i>	Autovalor
<i>dim</i>	Dimensão do auto-estado

Os autovetores são uma variável-chave no Grupo de Renormalização Numérica. Os autovetores da iteração anterior,  $N - 1$ , são usados para criar novos estados quânticos ('createStates'). Autovetores da iteração atual,  $N$ , são obtidos pela diagonalização dos subespaços na função 'diag'. Para cada autovetor e seu autovalor correspondente, um objeto 'TCSIAMEigVec' é inicializado. Todos autovetores são criados a partir de uma matriz e não de um único estado e, portanto, não possuem nenhum tipo  $t$ . Por isso, o atributo tipo é definido como zero. O mesmo vale para o índice de contagem. Cada autovetor é único devido à matriz simétrica e, portanto, o índice de contagem é definido como sendo número um. Os autovetores são representados por seus componentes multiplicados com os vetores da base que são os tipos. Todos os objetos do 'TCSIAMEigVec' são armazenados no atributo *eig\_vec\_new* do objeto da classe 'TCSIAMResult'. Eles são usados para calcular as propriedades físicas (calor específico, susceptibilidade magnética) para alguns valores de temperatura. Cada iteração corresponde a várias temperaturas. Quanto maior a iteração  $N$ , quanto menor as temperaturas  $T$ . Após o cálculo das propriedades físicas, a função 'reset', método da classe 'TCSIAMResult', é chamada para liberar a memória dos autovetores que não são mais necessários.

A última classe é aquela que guarda todos os resultados e leva-os para cada função que ela pode acessar eles.

Tabela 14 – Propriedades e as explicações delas da classe 'TCSIAMResult'.

propriedade	explicação
<i>parameter</i>	Torna os parâmetros acessíveis para os cálculos
<i>pref</i>	Energias da discretização logarítmica da banda de condução que são os prefatores para os elementos de matriz não diagonais
<i>states</i>	Estados da iteração atual
<i>p_states_sort</i>	Ponteiros para estados da iteração atual classificados em subespaços com tipos ascendentes
<i>subspaces</i>	O tamanho deste vetor fornece o número dos subespaços, os elementos fornecem o numero de estados com os mesmos $j_1$ , $j_2$ e $s$ para cada subespaço
<i>indices</i>	Soma cumulativa da propriedade 'subspaces' com 0 adicional no início, determinando o início e o fim para cada subespaço
<i>types</i>	Tipos dos quais os estados atuais são derivados
<i>eig_val</i>	Todos os autovalores que são menores que a energia de corte $C$ se o corte for aplicado, se não todas as energias
<i>eig_vec_old</i>	Auto-estados da iteração anterior
<i>eig_vec_new</i>	Auto-estados da iteração atual

Um objeto da classe 'TCSIAMResult' é criado no início da rotina principal 'calculation' para definir os parâmetros padrões ou definidos pelo usuário, e inicializar os autovetores para a primeira iteração  $N = -1$ . Todas as variáveis necessárias para uma única iteração  $N$  são armazenadas nos atributos do objeto. Isso inclui os autovetores da iteração anterior  $N - 1$ , os novos estados criados e os novos autovetores da iteração atual  $N$ . Além disso, o método 'reset' é usado para liberar memória entre iterações que não é mais necessária.

A seguir, o curso do programa, chamado cronológico das funções, é explicado.

O executável, 'calculation', é a rotina principal que enceta a diagonalização do Modelo de Anderson de Dois Canais através do Grupo de Renormalização Numérica.

Ele, veja a figura 16, inicializa um objeto da classe 'TCSIAMResult' com os valores padrões para os parâmetros que determinam o modelo ( $V_1$ ,  $V_2$ ,  $E_0$ ,  $E_x$ ) e a

discretização logarítmica da banda de condução ( $\lambda, z$ ). Se especificado na linha 22, o programa é executado com parâmetros definidos pelo usuário. O processo iterativo do Grupo de Renormalização Numérica é iniciado pela função 'initialization' e depois continua pela 'iteration'. O objeto 'result' salva as informações sobre os estados, autovetores e autovalores da iteração atual e anterior. Por razões de memória, apenas a quantidade mínima de variáveis é armazenada e liberada quando não é mais necessária.

```
13 #include "calculation.hpp"
14
15 using namespace Eigen;
16 using namespace std;
17
18 int main(){
19
20     TCSIAMResult result;
21
22     if (1){
23         const string file_path_rel = "../input/input.txt";
24         result.read_user_defined_parameter(file_path_rel);
25     }
26
27     initialization(result);
28
29     iteration(result);
30 };
```

Figura 16 – O executável, 'calculation', do programa TCSIAM. O 'headerfile' 'calculation.hpp' é incluído tal que o compilador conhece as funções e classes que são desenvolvidas pelo programador. Os dois 'namespaces' da biblioteca de padrão STL, 'Standard Template Library', e de *Eigen* são usados. A função 'main' inicializa um objeto da classe 'TCSIAMResult' com os parâmetros de padrão, nomeado 'result'. A linha 22 determina se o programa roda com parâmetros que o usuário especifica no arquivo 'input.txt'. Esses parâmetros são lidos na linha 24. As funções 'initialization' e 'iteration' são chamadas com o objeto 'result' como argumento de entrada.

A função 'initialization', veja figura 17, inicializa o TCSIAM e executa a primeira etapa da iteração de  $N = -1$  a  $N = 0$ .

Em primeiro lugar, os três auto-estados da iteração inicial são montados com os números quânticos e autovalores deles, veja a tabela 6. Depois a banda de condução é discretizada com o processo, descrito na seção 6.2.1, para obter os prefatores dos elementos de matriz para todas iterações. Os novos estados são



criados a partir dos autovetores da iteração anterior na função 'createStates'. Em seguida, o índice de contagem  $r$  dos novos estados é atribuído na rotina 'indexing' para distinguir estados que possuem os mesmos números quânticos  $j_1$ ,  $j_2$  e  $s$ . Depois, a função 'diagN0' cria os subespaços e diagonaliza-os para obter os autovetores e autovalores da iteração  $N = 0$ . Em seguida, os autovetores da iteração  $N = 0$  são reatribuídos para serem os antigos da próxima iteração  $N = 1$ , através do método 'reset' do objeto 'result'. Os cálculos das grandezas físicas ainda não estão implementados.

```
9 void initialization(TCSIAMResult &result){
10
11     setEigVec(result);
12
13     comPref(result);
14
15     createStates(result);
16
17     indexing(result);
18
19     diagN0(result);
20
21     //computeSpecHeat(result);
22     //computeMagSusz(result);
23
24     result.reset();
25 }
```

Figura 17 – A função 'initialization' do programa TCSIAM. Ela leva a referência do objeto 'result' como argumento de entrada e por consequência não precisa um argumento de saída, 'void'. As funções 'setEigVec', que inicializa os autovetores da iteração  $N = -1$ , 'comPref', que calcula os prefatores, 'createStates', que cria estados novos, 'indexing', que determina o índice de contagem, 'diagN0', que diagonaliza a iteração  $N = 0$ , e o método 'reset' do objeto 'result', para liberar memória, são chamadas. Os cálculos das grandezas físicas ainda não estão implementados, 'computeSpecHeat' e 'computeMagSusz'.

A função 'iteration', veja figura 18, do TCSIAM é bem parecida como 'initialization'. Ela tem todos os processos para realizar uma iteração inteira e executa-as começando com  $N = 0$  até a máxima  $N_{max}$ . A diferença é que as rotinas 'setEigVec' e 'comPref' não são chamadas, porque os eigenvetores iniciais somente

são usados na primeira iteração  $N = -1$ , e os prefatores são calculados para todas iterações no começo e guardados no objeto 'result'. Ao invés de executar a função 'diagN0', 'diag' é usada aqui. Isso é porque, elementos de matriz são calculados diferentes no processo inicial. Finalmente, a função 'iteration' chama ela mesma para iniciar a próxima iteração.

```
16 void iteration(TCSIAMResult &result){
17
18     auto para = result.get_parameter();
19     const auto N = para.get_N();
20     const auto N_max = para.get_N_max();
21
22     if (N <= N_max){
23
24         createState(result);
25
26         indexing(result);
27
28         diag(result);
29
30         //computeSpecHeat(result);
31         //computeMagSusz(result);
32
33         result.reset();
34
35         iteration(result);
36     }
37 }
```

Figura 18 – A função 'iteration' do programa TCSIAM. Ela leva a referência do objeto 'result' como argumento de entrada. A iteração atual e máxima são copiadas no início e é conferido se a iteração atual ainda é menor igual da máxima. As funções 'createStates', que cria estados novos, 'indexing', que determina o índice de contagem, 'diag', que diagonaliza a iteração atual, e o método 'reset' do objeto 'result', para liberar memória, são chamadas. Os cálculos das grandezas físicas ainda não estão implementados, 'computeSpecHeat' e 'computeMagSusz'.

A função 'createStates', veja figura 19, usa os autovetores da iteração anterior para criar estados novos da iteração atual. Para cada autovetor da iteração anterior, que é definida pelos números quânticos  $j_1$ ,  $j_2$  e  $s$ , a função 'newTypes' é chamada para determinar quais dos 16 novos tipos podem ser usados para criar novos estados. Em seguida, os novos estados são armazenados não ordenados no atributo, *states*,

do objeto 'result'. Cada novo estado mantém um ponteiro bruto para seu autovetor original do qual ele foi criado. A próxima função no procedimento é a função 'indexing'.

```

14 void createStates(TCSIAMResult &result){
15
16     auto para = result.get_parameter();
17     const auto N = para.get_N();
18     auto& eig_vec_old = result.get_eig_vec_old(); // const cannot be used here to get proper pointer later
19     auto& states = result.get_states();
20
21     for (auto& eig_vec : eig_vec_old){
22         const auto j1 = eig_vec.get_j1();
23         const auto j2 = eig_vec.get_j2();
24         const auto s = eig_vec.get_s();
25         const auto types = newTypes(eig_vec); // create only possible types
26         for (const auto type : types){
27             Real j1_new;
28             Real j2_new;
29             Real s_new;
30             const unsigned r = 0; // will be set properly in indexing.cpp
31             switch (type){
32                 case 1: j1_new = j1 + 0.5; j2_new = j2 + 0.5; s_new = s + 0.0; break;
33                 case 2: j1_new = j1 + 0.5; j2_new = j2 + 0.0; s_new = s + 0.5; break;
34                 case 3: j1_new = j1 + 0.5; j2_new = j2 + 0.0; s_new = s - 0.5; break;
35                 case 4: j1_new = j1 + 0.5; j2_new = j2 - 0.5; s_new = s + 0.0; break;
36                 case 5: j1_new = j1 + 0.0; j2_new = j2 + 0.5; s_new = s + 0.5; break;
37                 case 6: j1_new = j1 + 0.0; j2_new = j2 + 0.5; s_new = s - 0.5; break;
38                 case 7: j1_new = j1 + 0.0; j2_new = j2 + 0.0; s_new = s + 1.0; break;
39                 case 8: j1_new = j1 + 0.0; j2_new = j2 + 0.0; s_new = s + 0.0; break;
40                 case 9: j1_new = j1 + 0.0; j2_new = j2 + 0.0; s_new = s + 0.0; break;
41                 case 10: j1_new = j1 + 0.0; j2_new = j2 + 0.0; s_new = s - 1.0; break;
42                 case 11: j1_new = j1 + 0.0; j2_new = j2 - 0.5; s_new = s + 0.5; break;
43                 case 12: j1_new = j1 + 0.0; j2_new = j2 - 0.5; s_new = s - 0.5; break;
44                 case 13: j1_new = j1 - 0.5; j2_new = j2 + 0.5; s_new = s + 0.0; break;
45                 case 14: j1_new = j1 - 0.5; j2_new = j2 + 0.0; s_new = s + 0.5; break;
46                 case 15: j1_new = j1 - 0.5; j2_new = j2 + 0.0; s_new = s - 0.5; break;
47                 case 16: j1_new = j1 - 0.5; j2_new = j2 - 0.5; s_new = s + 0.0; break;
48             }
49             TCSIAMEigVec *const p_eig_vec = &eig_vec;
50             TCSIAMState state_new(j1_new, j2_new, s_new, type, r, N, p_eig_vec);
51             states.push_back(state_new);
52         }
53     }
54 }

```

Figura 19 – A função 'createStates' do programa TCSIAM. Ela leva a referência do objeto 'result' como argumento de entrada. O número da iteração e os auto-estados da iteração anterior são lidos no início. Depois, a função 'newTypes', que determina qual novos estados podem ser criados baseado nos números quânticos do auto-estado, é chamada. Os estados com os tipos possíveis são criados resultando em novos números quânticos  $j1\_new$ ,  $j2\_new$  e  $s\_new$ . Um ponteiro *bruto* para o autovetor da iteração anterior é inicializado. Finalmente, os novos estados com os novos números quânticos e o ponteiro são criados e guardados no atributo *states* do objeto 'result'. Este procedimento é feito para cada autovetor da iteração anterior.

A função 'indexing', veja a figura 20, agrupa os estados com os mesmos números quânticos  $j_1$ ,  $j_2$  e  $s$  em subespaços e os indexa para distingui-los. Essa é uma das funções fundamentais do programa TCSIAM. Ela classifica os novos estados e torná-los facilmente acessíveis para cálculos seguintes. Os estados com os mesmos números quânticos são agora indexados de maneira ascendente com o índice de contagem  $r$  começando com 1. Com isso, os estados são agrupados

em subespaços. Para evitar a cópia de dados nesse processo de classificação, os ponteiros *brutos* para os estados não ordenados são criados e, em seguida, os ponteiros são classificados. Eles são armazenados na propriedade *p\_states\_sort* do objeto 'result'. Além disso, os três vetores - 'subspaces', 'indices' e 'types' - são computados. Esses três vetores são armazenados nos atributos do 'TCSIAMResult' objeto e são essenciais para a posterior diagonalização dos subespaços na função 'diag'.

```

20 void indexing(TCSIAMResult &result){
21
22     auto& states = result.get_states();
23     auto& p_states_sort = result.get_p_states_sort();
24     auto& types = result.get_types();
25     auto& subspaces = result.get_subspaces();
26     auto& indices = result.get_indices();
27
28     const unsigned num_of_states = states.size();
29     for (unsigned i = 0; i < num_of_states; i++){
30         TCSIAMState *const state_1 = &states[i];
31         const auto r_1 = state_1->get_r();
32         if (0 == r_1){ // condition that state is not indexed yet
33             state_1->set_r(1);
34             const auto t_1 = state_1->get_t();
35             p_states_sort.push_back(state_1);
36             types.push_back(t_1);
37             const auto j1_1 = state_1->get_j1();
38             const auto j2_1 = state_1->get_j2();
39             const auto s_1 = state_1->get_s();
40             unsigned d = 1;
41             for (unsigned j = i + 1; j < num_of_states; j++){
42                 TCSIAMState *const state_2 = &states[j];
43                 const auto r_2 = state_2->get_r();
44                 if (0 == r_2){ // condition that state to compare is not indexed yet
45                     const auto j1_2 = state_2->get_j1();
46                     const auto j2_2 = state_2->get_j2();
47                     const auto s_2 = state_2->get_s();
48                     // Condition to be in the same subspace
49                     if (j1_1 == j1_2 && j2_1 == j2_2 && s_1 == s_2){
50                         d = d + 1;
51                         state_2->set_r(d);
52                         const auto t_2 = state_2->get_t();
53                         p_states_sort.push_back(state_2);
54                         types.push_back(t_2);
55                     }
56                 }
57             }
58             subspaces.push_back(d);
59             const unsigned sum = accumulate(subspaces.begin(), subspaces.end(), 0);
60             indices.push_back(sum);
61         }
62     }
63     indices.insert(indices.begin(), 0);
64
65     sortStates(result);
66 }

```

Figura 20 – A função 'indexing' do programa TCSIAM. Ela leva a referência do objeto 'result' como argumento de entrada. Depois, confere se um estado ainda não tem o índice de contagem determinado, veja linha 32. Se ele ainda não tem, o valor 1 é dado. Todos os outros estados que ainda não tem índice de contagem, ver linha 44, e possuem os mesmos números quânticos, veja linha 49, recebem valores ascendentes. Além disso, os vetores *subspaces*, *indices* e *types*, que são explicados na tabela 14, são criados. No fim, a função 'sortStates' é chamada para ordenar os vetores da base dentro de um subespaço com tipos ascendentes.

A função 'diag', veja as figuras 21 e 22, calcula as matrizes simétricas dos subespaços através de invariantes e diagonaliza-as com a biblioteca *Eigen*. Os dois processos, computação das matrizes simétricas e diagonalização dos subespaços, são executados em uma função para evitar o armazenamento da matriz. A matriz

em si não é necessária para calcular as propriedades físicas como o calor específico ou a susceptibilidade magnética. Apenas os autovalores são usados. Veja a seção 3.3.4 para saber como a matriz é calculada. A função 'SelfAdjointEigenSolver' da biblioteca *Eigen* é usada para diagonalizá-la. Um corte de energia é aplicado aos autovalores maiores que  $C$ , um atributo da classe 'TCSIAMParameter'. Os autovetores correspondentes acima de  $C$  são descartados também e não podem criar novos estados na próxima iteração. Os autovetores com grandes autovalores são estados que estão muito distante energeticamente do nível de Fermi e, portanto, têm menor probabilidade de interagir com a impureza.

```

33     const unsigned subspace_num = subspaces.size();
34     for (unsigned i = 0; i < subspace_num; i++){
35         const unsigned mat_size = subspaces[i];
36         Mat subspace = Mat::Zero(mat_size, mat_size);
37         const unsigned j_start = indices[i];
38         const unsigned j_end = indices[i+1];
39
40         // Compute symmetric matrix
41         for (unsigned j = j_start; j < j_end; j++){
42             const TCSIAMState *const state_1 = p_states_sort[j]; // state of current iteration
43             const auto t_1 = state_1->get_t(); // type of current state
44             const TCSIAMEigVec *const p_eig_vec_1 = state_1->get_p(); // pointer to creator
45             const auto val_dia = p_eig_vec_1->get_eig_val();
46
47             const Real s1_j1 = state_1->get_j1();
48             const Real s1_j2 = state_1->get_j2();
49             const Real s1_s = state_1->get_s();
50             const unsigned j_mat = j - j_start;
51
52             subspace(j_mat, j_mat) = sqrt(lam) * val_dia; // multiply prefactor for diagonal element
53             for (unsigned k = j + 1; k < j_end; k++){
54                 const TCSIAMState *const state_2 = p_states_sort[k]; // state of current iteration
55                 const auto t_2 = state_2->get_t(); // type of second current state
56                 const Real s2_j1 = state_2->get_j1();
57                 const Real s2_j2 = state_2->get_j2();
58                 const Real s2_s = state_2->get_s();
59                 if (t_1 != t_2){
60                     const auto val_off_dia = calMatEl(result, state_1, state_2);
61                     const unsigned k_mat = k - j_start;
62                     subspace(j_mat, k_mat) = val_off_dia;
63                     subspace(k_mat, j_mat) = val_off_dia;
64                 }
65             }
66         }

```

Figura 21 – Primeira parte da função 'diag' do programa TCSIAM - computação das matrizes simétricas: Para cada subespaço, uma matriz com todos elementos de zero é inicializado. Os vetores da base são usados para calcular os elementos diagonais. Cada tipo de vetor da base é comparado com todos os outros tipos da base. Se os tipos não são iguais, os vetores vêm de subespaços diferentes, eles podem resultar em um elemento de matriz não zero. Para determinar o valor, a função 'calMatEl' é chamada. Como a matriz é simétrica somente os valores acima do diagonal são calculados é atribuídos para aqueles embaixo também.

```

69     const TCSIAMState *const state_1 = p_states_sort[j_start];
70     const auto j1 = state_1->get_j1();
71     const auto j2 = state_1->get_j2();
72     const auto s = state_1->get_s();
73
74     // Diagonalize
75     SelfAdjointEigenSolver<Mat> es(mat_size);
76     es.compute(subspace);
77     const Vec eig_values = es.eigenvalues();
78     const Mat eig_vectors = es.eigenvectors();
79     const vector<unsigned> types_cur(&types[j_start], &types[j_end]);
80     for (unsigned j = 0; j < mat_size; j++){
81         const Real eig_value = eig_values(j);
82         if (1 == C_mode){ // apply energy cut
83             if (-1.0*C > eig_value || C < eig_value)
84                 break;
85         }
86         const Vec eig_vector = eig_vectors.col(j);
87         TCSIAMEigVec eig_vec(j1, j2, s, 0, 1, N, eig_value, mat_size);
88         eig_val.push_back(eig_value);
89         eig_vec.set_values(eig_vector);
90         eig_vec.set_types(types_cur);
91         eig_vec_new.push_back(eig_vec);
92     }
93     }
94     result.rescale_energy();
95 }

```

Figura 22 – Segunda parte da função 'diag' do programa TCSIAM - diagonalização das matrizes: Cada subespaço é diagonalizado com o pacote 'SelfAdjointEigenSolver' da biblioteca *Eigen*. Se o corte da energia é ligado, os autovetores e os autovalores maior do  $C$  ou menos do  $-C$  são descartados. Para cada auto-estado um objeto da classe 'TCSIAMEigVec' com os números quânticos do subespaço é criado e guardado no atributo *eig\_vec\_new* do objeto 'result'. Adicionalmente, os autovalores são armazenados na propriedade *eig\_val* do mesmo objeto. No fim, o método 'rescale\_energy' do objeto 'result' é chamado para mudar os autovalores da energia que o menor é zero.

### 6.3 Início do processo de licenciamento

No Brasil, é possível o registro e licenciamento de software para uso direto ou para uso de uma empresa. No caso de uso por uma empresa, ela fica responsável de fazer o gerenciamento e gerar recursos para a universidade. A legislação atual que trata deste assunto pode ser obtido no link:

[http://www.planalto.gov.br/ccivil\\_03/\\_ato2004-2006/2004/lei/110.973.htm](http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/lei/110.973.htm)

Aqui ressaltamos os dois parágrafos que importam:

"Art. 6. É facultado à ICT pública celebrar contrato de transferência de tecnologia e de licenciamento para outorga de direito de uso ou de exploração de criação por ela desenvolvida isoladamente ou por meio de parceria."

"Art. 13. É assegurada ao criador participação mínima de 5% (cinco por cento) e máxima de 1/3 (um terço) nos ganhos econômicos, auferidos pela ICT, resultantes de contratos de transferência de tecnologia e de licenciamento para outorga de direito de uso ou de exploração de criação protegida da qual tenha sido o inventor, obtentor ou autor, aplicando-se, no que couber, o disposto no parágrafo único do art. 93 da Lei n' 9.279, de 1996."

Na UFMS, o órgão que trata deste tema atualmente é:

Núcleo de Inovação Tecnológica (DIPIT/CEI/AGINOVA)

(67) 3345-7188 - 3345-7793 <[www.aginova.ufms.br/nit](http://www.aginova.ufms.br/nit)>

Como exemplo de software registrados pela UFMS, a AGINOVA disponibiliza um portfólio no link: <<https://aginova.ufms.br/nit/portfolio-de-programa-de-computador/>>

O procedimento para solicitação de registro de software estipulado para a AGINOVA, começa com o preenchimento de um questionário disponibilizado pelo link: <[https://docs.google.com/forms/d/e/1FAIpQLScGhfyk2izmA\\_9c-gRNOBqG7f-xmExOnF-plhkvOetvtnCu5A/viewform?formkey=dFFHVUtWdVBVbExTdUVIVUQ1RHRoTIE6MQ](https://docs.google.com/forms/d/e/1FAIpQLScGhfyk2izmA_9c-gRNOBqG7f-xmExOnF-plhkvOetvtnCu5A/viewform?formkey=dFFHVUtWdVBVbExTdUVIVUQ1RHRoTIE6MQ)>

Assim que o TCSIAM estiver apto a calcular grandezas físicas desejamos registrá-lo como software científico para uso livre.



## 7 Conclusão

Este capítulo descreve a análise dos resultados obtidos, as dificuldades encontradas e expressar as tarefas necessárias para completar o objetivo proposto.

Foi desenvolvido um novo código computacional, TCSIAM, baseado nos conceitos do Grupo de Renormalização Numérica. Ele diagonaliza o Modelo de Anderson de Dois Canais. A parte principal deste código está completa: processo iterativo e determinação de autovalores dos subespaços.

Para realizar esta tarefa foram feitas as etapas:

- Estudo da teoria sobre impurezas magnéticas em metais: Efeito Kondo, Modelo de Kondo e de Anderson. Este assunto é, em geral, desenvolvido em grupos consolidados de pesquisa em física da matéria condensada. Aqui no INFI/UFMS somente o Dr. J. V. B. Ferreira trabalha nesta linha de pesquisa. A teoria do Efeito Kondo é toda desenvolvida em linguagem de segunda quantização. Hamiltonianos são propostos para descrever os modelos e, a partir de então, trata-se de diagonalizar estes hamiltonianos, encontrar seus autovalores e propriedades físicas. Devido ao grande número de graus de liberdade da banda de condução eletrônica, procura-se encontrar o máximo de simetrias físicas nos modelos afim de facilitar a diagonalização. Como foi descrito nesta dissertação, enquanto o Modelo de Kondo e o Modelo de Anderson tradicionais são bem conhecidos na literatura, o Modelo de Anderson de Dois Canais é relativamente pouco conhecido por ser recente. Mas é um modelo capaz de descrever os dois anteriores e, além disso, apresentar o comportamento não líquido de Fermi. Por exemplo, a liga metálica *CeLaNiGe* apresenta suscetibilidade magnética e calor específico em temperaturas baixas diferentes daqueles previstos pelas teorias usuais.

- Estudo do Grupo de Renormalização Numérica: Esta etapa também apresentou a mesma dificuldade do item anterior. A ideia básica do GRN é transformar o hamiltoniano original em uma expressão que pode ser resolvido de maneira iterativa.

A partir dos auto-estados e autovalores de um passo, constrói-se o hamiltoniano seguinte, determina-se os seus auto-estados e autovalores, e assim por diante. Essa construção dos hamiltonianos é seguida pelas regras obtidas pelas simetrias do modelo físico. Merece destaque a importância na clareza da definição dos elementos denominados invariantes (Teorema de Wigner–Eckart). Nesta dissertação houve o cuidado de explicitar a diferença entre invariantes coletivo e individuais.

- Análise do código computacional do Dr. J. V. B. Ferreira: A proposta inicial era simplesmente modernizar o código desenvolvido por este pesquisador. Apesar de ser funcional, neste código está embutido vários procedimentos que não se justificam em uma primeira análise. E como ele não está devidamente documentado, é difícil sua manipulação.

- Construção do novo código e respectiva documentação. Esta etapa implicou no estudo do modelo físico e na técnica de renormalização numérica. Apesar de ser muito mais difícil do que simplesmente modernizar o código do Dr. J. V. B. Ferreira, a construção do novo código permitiu entender a ideia de renormalização numérica e como implementá-la.

O TCSIAM está funcionando e calculando corretamente os autovalores. Para esta verificação foram realizados exaustivos procedimentos analíticos seguindo exatamente as regras determinadas pelo Grupo de Renormalização Numérica. A implementação do cálculo de grandezas físicas como suscetibilidade magnética e calor específico não foram implementadas por falta de tempo. Mas, graças à programação modular documentada, não implicará uma grande dificuldade.

A comparação entre o tempo de execução do TCSIAM em relação ao código do Dr. J. V. B. Ferreira indica a possibilidade de aprimoramento do primeiro. A troca do pacote *Eigen* por outro pacote de diagonalização pode fazer o TCSIAM reduzir pela metade o seu tempo atual de processamento.

Em relação ao processo de licenciamento, de acordo com as informações obtidas da UFMS, parece fácil de completar depois dos primeiros contatos com a administração. Desejo colocar o TCSIAM acessível gratuitamente para qualquer pesquisador utilizar.

## A Procedimento para gerar novos estados

O primeiro passo de cada iteração do TCSIAM é criar os novos estados quânticos, filhos, a partir dos auto-estados anterior. Cada autovetor da iteração anterior é o gerador, pai, por possivelmente 16 filhos na iteração seguinte [46]. Os números quânticos,  $j'_1$ ,  $j'_2$  e  $s'$ , são restritos aos números positivos incluindo o zero. Se um pai  $|0, 0, 1/2\rangle$  ia criar um filho de tipo 13 resultando em  $|-1/2, 1/2, 1/2\rangle$ , o filho é descartado. O tipo 9 tem uma restrição adicional. O spin  $s$  do pai não pode ser 0. Com os tipos e as duas limitações, os estados para a iteração seguinte podem ser criados pelos pais da iteração anterior.

Conseqüentemente, o pai  $|j_1, j_2, s\rangle$  pode criar no máximo 16 filhos  $|j'_1, j'_2, s'\rangle$  com os procedimentos da tabela seguinte.

Tabela 15 – Tipo  $t$  e números quânticos novos. Nesta tabela  $j_1$ ,  $j_2$  e  $s$  são números quânticos do auto-estado da iteração  $N - 1$  que gera os estados com os números quânticos  $j'_1$ ,  $j'_2$  e  $s'$  da iteração  $N$  através dos 16 procedimentos.

tipo	números quânticos novos	tipo	números quânticos novos
$t$	$j'_1, j'_2$ e $s'$	$t$	$j'_1, j'_2$ e $s'$
1	$j_1 + \frac{1}{2}, j_2 + \frac{1}{2}, s + 0$	9	$j_1 + 0, j_2 + 0, s + 0$
2	$j_1 + \frac{1}{2}, j_2 + 0, s + \frac{1}{2}$	10	$j_1 + 0, j_2 + 0, s - 1$
3	$j_1 + \frac{1}{2}, j_2 + 0, s - \frac{1}{2}$	11	$j_1 + 0, j_2 - \frac{1}{2}, s + \frac{1}{2}$
4	$j_1 + \frac{1}{2}, j_2 - \frac{1}{2}, s + 0$	12	$j_1 + 0, j_2 - \frac{1}{2}, s - \frac{1}{2}$
5	$j_1 + 0, j_2 + \frac{1}{2}, s + \frac{1}{2}$	13	$j_1 - \frac{1}{2}, j_2 + \frac{1}{2}, s + 0$
6	$j_1 + 0, j_2 + \frac{1}{2}, s - \frac{1}{2}$	14	$j_1 - \frac{1}{2}, j_2 + 0, s + \frac{1}{2}$
7	$j_1 + 0, j_2 + 0, s + 1$	15	$j_1 - \frac{1}{2}, j_2 + 0, s - \frac{1}{2}$
8	$j_1 + 0, j_2 + 0, s + 0$	16	$j_1 - \frac{1}{2}, j_2 - \frac{1}{2}, s + 0$

## B Elementos de matriz e invariantes

Os elementos de matriz e os invariantes são específicos para cada modelo. O Kenneth G. Wilson calculou-os para o Modelo de Kondo em 1975 [13] e o H.R. Krishna-Murthy para o Modelo de Anderson de um Canal em 1980 [14]. Os resultados para o TCSIAM são da tese do Dr. J. V. B. Ferreira [46].

A transformação do GRN que é reescrita aqui

$$H_N = \sqrt{\lambda}H_D + \epsilon_{N-1}^z H_I \quad (\text{B.1})$$

$$H_D = H_{N-1} \quad (\text{B.2})$$

$$H_I = f_{N-1,\sigma,\alpha}^\dagger f_{N,\sigma,\alpha} + f_{N,\sigma,\alpha}^\dagger f_{N-1,\sigma,\alpha} \quad (\text{B.3})$$

é usada para calcular os elementos de matriz. Os elementos diagonais de  $H_N$  são os autovalores da iteração anterior multiplicados por raiz de  $\lambda$ . Os elementos não diagonais, que têm tipos diferentes, são bem mais complicados. Eles têm um fator que é dependente dos números quânticos,  $j_1$ ,  $j_2$  e  $s$ , do subespaço da iteração atual  $N$  e um invariante coletivo da iteração anterior  $N - 1$ , e são multiplicados com as energias da banda de condução,  $\xi_N^z = \sqrt{2}\lambda^{N/2}\epsilon_N^z$ , que são dependentes dos parâmetros numéricos  $\lambda$  e  $z$ . É importante observar, que os elementos de matriz na iteração  $N$  dependem dos invariantes coletivos da iteração anterior.

Um elemento de matriz é construído pelos três componentes.

$$\textit{Elemento de matriz} = \textit{prefator} \times \textit{fator} \times \textit{invariante coletivo} \quad (\text{B.4})$$

### Prefator

Os prefatores que são as energias da banda de condução, são discutidos na seção 6.2.1. Aqui, é somente lembrado que a energia da iteração inicial,  $N = -1$ , é diferente, com o valor de  $\xi_{-1,\alpha} = \sqrt{\frac{2}{\lambda}}V_\alpha$ . Este é a única vez que os parâmetros,  $V_1$  e  $V_2$ , entram diretamente no Modelo de Anderson de Dois Canais.

### Fatores e invariantes coletivos

Os 40 resultados para elementos de matriz não nulos com os invariantes coletivos são mostrado nas tabelas 16 e 17. As tabelas 18 e 19 são os correspondentes invariantes individuais que determinam o invariante coletivo.

Tabela 16 – Os primeiros 20 elementos de matriz não nulos.

$$\begin{aligned}
 1 - \langle 1|H|2\rangle_N &= (-1)^N \sqrt{\frac{2s}{2s+1}} \langle j_1 - \frac{1}{2}, j_2, s - \frac{1}{2} || 2 || j_1 - \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1} \\
 2 - \langle 1|H|3\rangle_N &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \langle j_1 - \frac{1}{2}, j_2, s + \frac{1}{2} || 2 || j_1 - \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1} \\
 3 - \langle 1|H|5\rangle_N &= (-1)^N \sqrt{\frac{2s}{2s+1}} \langle j_1, j_2 - \frac{1}{2}, s - \frac{1}{2} || 1 || j_1 - \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1} \\
 4 - \langle 1|H|6\rangle_N &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \langle j_1, j_2 - \frac{1}{2}, s + \frac{1}{2} || 1 || j_1 - \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1} \\
 5 - \langle 2|H|4\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \langle j_1 - \frac{1}{2}, j_2 + \frac{1}{2}, s || 2 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2} \rangle_{N-1} \\
 6 - \langle 2|H|7\rangle_N &= -(-1)^N \sqrt{\frac{2s-1}{2s}} \langle j_1, j_2, s - 1 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2} \rangle_{N-1} \\
 7 - \langle 2|H|8\rangle_N &= -(-1)^N \sqrt{\frac{1}{2}} \langle j_1, j_2, s || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2} \rangle_{N-1} \\
 8 - \langle 2|H|9\rangle_N &= (-1)^N \sqrt{\frac{s+1}{2s}} \langle j_1, j_2, s || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2} \rangle_{N-1} \\
 9 - \langle 3|H|4\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \langle j_1 - \frac{1}{2}, j_2 + \frac{1}{2}, s || 2 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2} \rangle_{N-1} \\
 10 - \langle 3|H|8\rangle_N &= -(-1)^N \sqrt{\frac{1}{2}} \langle j_1, j_2, s || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2} \rangle_{N-1} \\
 11 - \langle 3|H|9\rangle_N &= -(-1)^N \sqrt{\frac{s}{2s+2}} \langle j_1, j_2, s || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2} \rangle_{N-1} \\
 12 - \langle 3|H|10\rangle_N &= (-1)^N \sqrt{\frac{2s+3}{2s+2}} \langle j_1, j_2, s + 1 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2} \rangle_{N-1} \\
 13 - \langle 4|H|11\rangle_N &= (-1)^N \sqrt{\frac{2s}{2s+1}} \langle j_1, j_2 + \frac{1}{2}, s - \frac{1}{2} || 1 || j_1 - \frac{1}{2}, j_2 + \frac{1}{2}, s \rangle_{N-1} \\
 14 - \langle 4|H|12\rangle_N &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \langle j_1, j_2 + \frac{1}{2}, s + \frac{1}{2} || 1 || j_1 - \frac{1}{2}, j_2 + \frac{1}{2}, s \rangle_{N-1} \\
 15 - \langle 5|H|7\rangle_N &= (-1)^N \sqrt{\frac{2s-1}{2s}} \langle j_1, j_2, s - 1 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2} \rangle_{N-1} \\
 16 - \langle 5|H|8\rangle_N &= -(-1)^N \sqrt{\frac{1}{2}} \langle j_1, j_2, s || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2} \rangle_{N-1} \\
 17 - \langle 5|H|9\rangle_N &= -(-1)^N \sqrt{\frac{s+1}{2s}} \langle j_1, j_2, s || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2} \rangle_{N-1} \\
 18 - \langle 5|H|13\rangle_N &= \sqrt{\frac{2j_1+2}{2j_1+1}} \langle j_1 + \frac{1}{2}, j_2 - \frac{1}{2}, s || 1 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2} \rangle_{N-1} \\
 19 - \langle 6|H|8\rangle_N &= -(-1)^N \sqrt{\frac{1}{2}} \langle j_1, j_2, s || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2} \rangle_{N-1} \\
 20 - \langle 6|H|9\rangle_N &= (-1)^N \sqrt{\frac{s}{2s+2}} \langle j_1, j_2, s || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2} \rangle_{N-1}
 \end{aligned}$$

Tabela 17 – Os seguintes 20 elementos de matriz não nulos.

$$\begin{aligned}
 21 - \langle 6|H|10\rangle_N &= -(-1)^N \sqrt{\frac{2s+3}{2s+2}} \langle j_1, j_2, s+1 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2} \rangle_{N-1} \\
 22 - \langle 6|H|13\rangle_N &= \sqrt{\frac{2j_1+2}{2j_1+1}} \langle j_1 + \frac{1}{2}, j_2 - \frac{1}{2}, s || 1 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2} \rangle_{N-1} \\
 23 - \langle 7|H|11\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \langle j_1, j_2 + \frac{1}{2}, s - \frac{1}{2} || 2 || j_1, j_2, s - 1 \rangle_{N-1} \\
 24 - \langle 7|H|14\rangle_N &= -\sqrt{\frac{2j_1+2}{2j_1+1}} \langle j_1 + \frac{1}{2}, j_2, s - \frac{1}{2} || 1 || j_1, j_2, s - 1 \rangle_{N-1} \\
 25 - \langle 8|H|11\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \sqrt{\frac{s}{2s+1}} \langle j_1, j_2 + \frac{1}{2}, s - \frac{1}{2} || 2 || j_1, j_2, s \rangle_{N-1} \\
 26 - \langle 8|H|12\rangle_N &= -\sqrt{\frac{j_2+1}{2j_2+1}} \sqrt{\frac{2s+2}{2s+1}} \langle j_1, j_2 + \frac{1}{2}, s + \frac{1}{2} || 2 || j_1, j_2, s \rangle_{N-1} \\
 27 - \langle 8|H|14\rangle_N &= \sqrt{\frac{2j_1+2}{2j_1+1}} \sqrt{\frac{s}{2s+1}} \langle j_1 + \frac{1}{2}, j_2, s - \frac{1}{2} || 1 || j_1, j_2, s \rangle_{N-1} \\
 28 - \langle 8|H|15\rangle_N &= -\sqrt{\frac{j_1+1}{2j_1+1}} \sqrt{\frac{2s+2}{2s+1}} \langle j_1 + \frac{1}{2}, j_2, s + \frac{1}{2} || 1 || j_1, j_2, s \rangle_{N-1} \\
 29 - \langle 9|H|11\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \sqrt{\frac{s+1}{2s+1}} \langle j_1, j_2 + \frac{1}{2}, s - \frac{1}{2} || 2 || j_1, j_2, s \rangle_{N-1} \\
 30 - \langle 9|H|12\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \sqrt{\frac{s}{2s+1}} \langle j_1, j_2 + \frac{1}{2}, s + \frac{1}{2} || 2 || j_1, j_2, s \rangle_{N-1} \\
 31 - \langle 9|H|14\rangle_N &= -\sqrt{\frac{2j_1+2}{2j_1+1}} \sqrt{\frac{s+1}{2s+1}} \langle j_1 + \frac{1}{2}, j_2, s - \frac{1}{2} || 1 || j_1, j_2, s \rangle_{N-1} \\
 32 - \langle 9|H|15\rangle_N &= -\sqrt{\frac{2j_1+2}{2j_1+1}} \sqrt{\frac{s}{2s+1}} \langle j_1 + \frac{1}{2}, j_2, s + \frac{1}{2} || 1 || j_1, j_2, s \rangle_{N-1} \\
 33 - \langle 10|H|12\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \langle j_1, j_2 + \frac{1}{2}, s + \frac{1}{2} || 2 || j_1, j_2, s + 1 \rangle_{N-1} \\
 34 - \langle 10|H|15\rangle_N &= -\sqrt{\frac{2j_1+2}{2j_1+1}} \langle j_1 + \frac{1}{2}, j_2, s + \frac{1}{2} || 1 || j_1, j_2, s + 1 \rangle_{N-1} \\
 35 - \langle 11|H|16\rangle_N &= \sqrt{\frac{2j_1+2}{2j_1+1}} \langle j_1 + \frac{1}{2}, j_2 + \frac{1}{2}, s || 1 || j_1, j_2 + \frac{1}{2}, s - \frac{1}{2} \rangle_{N-1} \\
 36 - \langle 12|H|16\rangle_N &= \sqrt{\frac{2j_1+2}{2j_1+1}} \langle j_1 + \frac{1}{2}, j_2 + \frac{1}{2}, s || 1 || j_1, j_2 + \frac{1}{2}, s + \frac{1}{2} \rangle_{N-1} \\
 37 - \langle 13|H|14\rangle_N &= (-1)^N \sqrt{\frac{2s}{2s+1}} \langle j_1 + \frac{1}{2}, j_2, s - \frac{1}{2} || 2 || j_1 + \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1} \\
 38 - \langle 13|H|15\rangle_N &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \langle j_1 + \frac{1}{2}, j_2, s + \frac{1}{2} || 2 || j_1 + \frac{1}{2}, j_2 - \frac{1}{2}, s \rangle_{N-1} \\
 39 - \langle 14|H|16\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \langle j_1 + \frac{1}{2}, j_2 + \frac{1}{2}, s || 2 || j_1 + \frac{1}{2}, j_2, s - \frac{1}{2} \rangle_{N-1} \\
 40 - \langle 15|H|16\rangle_N &= \sqrt{\frac{2j_2+2}{2j_2+1}} \langle j_1 + \frac{1}{2}, j_2 + \frac{1}{2}, s || 2 || j_1 + \frac{1}{2}, j_2, s + \frac{1}{2} \rangle_{N-1}
 \end{aligned}$$

Tabela 18 – Os invariantes individuais do canal 1.  $\langle f_{N,1}^\dagger \rangle_N$ .

$$\begin{aligned}
 1 - \langle j_1, j_2, s, 1 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 5 \rangle &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \\
 2 - \langle j_1, j_2, s, 2 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 7 \rangle &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \\
 3 - \langle j_1, j_2, s, 3 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 8 \rangle &= -(-1)^N \sqrt{\frac{1}{2}} \\
 4 - \langle j_1, j_2, s, 3 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 9 \rangle &= -(-1)^N \sqrt{\frac{2s+3}{4s+2}} \\
 5 - \langle j_1, j_2, s, 4 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 11 \rangle &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \\
 6 - \langle j_1, j_2, s, 6 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 13 \rangle &= \sqrt{\frac{2j_1}{2j_1+1}} \\
 7 - \langle j_1, j_2, s, 8 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 14 \rangle &= -\frac{1}{2} \sqrt{\frac{2j_1}{2j_1+1}} \sqrt{\frac{4s+4}{2s+1}} \\
 8 - \langle j_1, j_2, s, 9 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 14 \rangle &= \frac{1}{2} \sqrt{\frac{2j_1}{2j_1+1}} \sqrt{\frac{4s}{2s+1}} \\
 9 - \langle j_1, j_2, s, 10 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 15 \rangle &= \sqrt{\frac{2j_1}{2j_1+1}} \\
 10 - \langle j_1, j_2, s, 12 || 1 || j_1 - \frac{1}{2}, j_2, s + \frac{1}{2}, 16 \rangle &= \sqrt{\frac{2j_1}{2j_1+1}} \\
 11 - \langle j_1, j_2, s, 1 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 6 \rangle &= (-1)^N \sqrt{\frac{2s}{2s+1}} \\
 12 - \langle j_1, j_2, s, 2 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 8 \rangle &= -(-1)^N \sqrt{\frac{1}{2}} \\
 13 - \langle j_1, j_2, s, 2 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 9 \rangle &= (-1)^N \sqrt{\frac{2s-1}{4s+2}} \\
 14 - \langle j_1, j_2, s, 3 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 10 \rangle &= (-1)^N \sqrt{\frac{2s}{2s+1}} \\
 15 - \langle j_1, j_2, s, 4 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 12 \rangle &= (-1)^N \sqrt{\frac{2s}{2s+1}} \\
 16 - \langle j_1, j_2, s, 5 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 13 \rangle &= \sqrt{\frac{2j_1}{2j_1+1}} \\
 17 - \langle j_1, j_2, s, 7 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 14 \rangle &= \sqrt{\frac{2j_1}{2j_1+1}} \\
 18 - \langle j_1, j_2, s, 8 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 15 \rangle &= \sqrt{\frac{1}{2}} \sqrt{\frac{2j_1}{2j_1+1}} \sqrt{\frac{2s}{2s+1}} \\
 19 - \langle j_1, j_2, s, 9 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 15 \rangle &= \sqrt{\frac{1}{2}} \sqrt{\frac{2j_1}{2j_1+1}} \sqrt{\frac{2s+2}{2s+1}} \\
 20 - \langle j_1, j_2, s, 11 || 1 || j_1 - \frac{1}{2}, j_2, s - \frac{1}{2}, 16 \rangle &= \sqrt{\frac{2j_1}{2j_1+1}}
 \end{aligned}$$

Tabela 19 – Os invariantes individuais do canal  $2 \langle f_{N,2}^\dagger \rangle_N$ .

$$\begin{aligned}
 1 - \langle j_1, j_2, s, 1 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 2 \rangle &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \\
 2 - \langle j_1, j_2, s, 3 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 4 \rangle &= \sqrt{\frac{2j_2}{2j_2+1}} \\
 3 - \langle j_1, j_2, s, 5 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 7 \rangle &= (-1)^N \sqrt{\frac{2s+2}{2s+1}} \\
 4 - \langle j_1, j_2, s, 6 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 8 \rangle &= -(-1)^N \sqrt{\frac{1}{2}} \\
 5 - \langle j_1, j_2, s, 6 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 9 \rangle &= (-1)^N \sqrt{\frac{2s+3}{4s+2}} \\
 6 - \langle j_1, j_2, s, 8 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 11 \rangle &= -\frac{1}{2} \sqrt{\frac{2j_2}{2j_2+1}} \sqrt{\frac{4s+4}{2s+1}} \\
 7 - \langle j_1, j_2, s, 9 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 11 \rangle &= -\frac{1}{2} \sqrt{\frac{2j_2}{2j_2+1}} \sqrt{\frac{4s}{2s+1}} \\
 8 - \langle j_1, j_2, s, 10 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 12 \rangle &= -\sqrt{\frac{2j_2}{2j_2+1}} \\
 9 - \langle j_1, j_2, s, 13 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 14 \rangle &= -(-1)^N \sqrt{\frac{2s+2}{2s+1}} \\
 10 - \langle j_1, j_2, s, 15 || 2 || j_1, j_2 - \frac{1}{2}, s + \frac{1}{2}, 16 \rangle &= \sqrt{\frac{2j_2}{2j_2+1}} \\
 11 - \langle j_1, j_2, s, 1 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 3 \rangle &= (-1)^N \sqrt{\frac{2s}{2s+1}} \\
 12 - \langle j_1, j_2, s, 2 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 4 \rangle &= \sqrt{\frac{2j_2}{2j_2+1}} \\
 13 - \langle j_1, j_2, s, 5 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 8 \rangle &= -(-1)^N \sqrt{\frac{1}{2}} \\
 14 - \langle j_1, j_2, s, 5 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 9 \rangle &= -(-1)^N \sqrt{\frac{2s-1}{4s+2}} \\
 15 - \langle j_1, j_2, s, 6 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 10 \rangle &= -(-1)^N \sqrt{\frac{2s}{2s+1}} \\
 16 - \langle j_1, j_2, s, 7 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 11 \rangle &= -\sqrt{\frac{2j_2}{2j_2+1}} \\
 17 - \langle j_1, j_2, s, 8 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 12 \rangle &= \sqrt{\frac{1}{2}} \sqrt{\frac{2j_2}{2j_2+1}} \sqrt{\frac{2s}{2s+1}} \\
 18 - \langle j_1, j_2, s, 9 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 12 \rangle &= -\sqrt{\frac{1}{2}} \sqrt{\frac{2j_2}{2j_2+1}} \sqrt{\frac{2s+2}{2s+1}} \\
 19 - \langle j_1, j_2, s, 13 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 15 \rangle &= (-1)^N \sqrt{\frac{2s}{2s+1}} \\
 20 - \langle j_1, j_2, s, 14 || 2 || j_1, j_2 - \frac{1}{2}, s - \frac{1}{2}, 16 \rangle &= \sqrt{\frac{2j_2}{2j_2+1}}
 \end{aligned}$$



## Referências

- 1 SUZUKI, T. J. Coherent transport of levitons through the Kondo resonance. *Physical Review B*, v. 95, n. 24, p. 241302, jun. 2017.
- 2 BRUN, B. et al. Electron Phase Shift at the Zero-Bias Anomaly of Quantum Point Contacts. *Physical Review Letters*, v. 116, n. 13, p. 136801, mar. 2016.
- 3 BOLECH, C. J.; SHAH, N. Prediction of the Capacitance Line Shape in Two-Channel Quantum Dots. *Physical Review Letters*, v. 95, n. 3, p. 036801, jul. 2005.
- 4 LIU, L. et al. Revealing the Atomic Site-Dependent  $g$  Factor within a Single Magnetic Molecule via the Extended Kondo Effect. *Physical Review Letters*, v. 114, n. 12, p. 126601, mar. 2015.
- 5 BORK, J. et al. A tunable two-impurity Kondo system in an atomic point contact. *Nature Physics*, v. 7, n. 11, p. 901–906, nov. 2011.
- 6 PRÜSER, H. et al. Long-range Kondo signature of a single magnetic impurity. *Nature Physics*, v. 7, n. 3, p. 203–206, mar. 2011.
- 7 FERREIRA, J. V. B. et al. Numerical Renormalization Group computation of temperature dependent specific heat for a two-channel Anderson model. *Journal of Magnetism and Magnetic Materials*, v. 324, n. 6, p. 1011–1016, mar. 2012. ISSN 0304-8853.
- 8 VOIGTLÄNDER, B. Scanning Probe Microscopy. In: *Scanning Probe Microscopy*. Berlin: Springer, Berlin, Heidelberg, 2015, (NanoScience and Technology). p. 1–11.
- 9 FROTA, H. O.; OLIVEIRA, L. N. Photoemission spectroscopy for the spin-degenerate Anderson model. *Physical Review B*, v. 33, n. 11, p. 7871–7874, jun. 1986.
- 10 PAULA, C. A.; SILVA, M. F.; OLIVEIRA, L. N. Low-energy spectral density for the Alexander-Anderson model. *Physical Review B*, v. 59, n. 1, p. 85–88, jan. 1999.
- 11 OLIVEIRA, F. F. de. *Densidade espectral da impureza para o modelo de Anderson*. Dissertação (Mestrado em Física Básica) — Universidade de São Paulo, São Carlos, fev. 2012.
- 12 YAMANE, Y. et al. Single-Site Non-Fermi Liquid Behaviors in a Diluted  $4f^2$  System  $Y_{1-x}Pr_xIr_2Zn_{20}$ . *Physical Review Letters*, v. 121, n. 7, ago. 2018.

- 13 WILSON, K. G. The renormalization group: Critical phenomena and the Kondo problem. *Reviews of Modern Physics*, v. 47, n. 4, p. 773–840, out. 1975.
- 14 KRISHNA-MURTHY, H. R.; WILKINS, J. W.; WILSON, K. G. Renormalization-group approach to the Anderson model of dilute magnetic alloys. I. Static properties for the symmetric case. *Physical Review B*, v. 21, n. 3, p. 1003–1043, fev. 1980.
- 15 OLIVEIRA, W. C.; OLIVEIRA, L. N. Generalized numerical renormalization-group method to calculate the thermodynamical properties of impurities in metals. *Physical Review B*, v. 49, n. 17, p. 11986–11994, maio 1994.
- 16 COSTA, S. C. et al. Numerical renormalization-group computation of specific heats. *Physical Review B*, v. 55, n. 1, p. 30–33, jan. 1997.
- 17 STEWART, G. R. Non-Fermi-liquid behavior in  $d$ - and  $f$ -electron metals. *Reviews of Modern Physics*, v. 73, n. 4, p. 797–855, out. 2001.
- 18 KILLER, U. et al. Unusual single-ion non-fermi-liquid behavior in  $\text{Ce}_{1-x}\text{La}_x\text{Ni}_9\text{Ge}_4$ . *Phys. Rev. Lett.*, v. 93, n. 21, p. 216404.
- 19 HAAS, W. J. de; BOER, J. de; BERG, G. J. van den. The electrical resistance of gold, copper and lead at low temperatures. *Physica*, v. 1, n. 7, p. 1115–1124, maio 1934.
- 20 HEWSON, A. C. *The Kondo Problem to Heavy Fermions*. Cambridge: Cambridge University Press, 1997.
- 21 KONDO, J. Resistance Minimum in Dilute Magnetic Alloys. *Progress of Theoretical Physics*, v. 32, n. 1, p. 37–49, jul. 1964.
- 22 ANDERSON, P. W. Localized Magnetic States in Metals. *Physical Review*, v. 124, n. 1, p. 41–53, out. 1961.
- 23 SCHRIEFFER, J. R.; WOLFF, P. A. Relation between the Anderson and Kondo Hamiltonians. *Physical Review*, v. 149, n. 2, p. 491–492, set. 1966.
- 24 ANDREI, N.; FURUYA, K.; LOWENSTEIN, J. H. Solution of the Kondo problem. *Reviews of Modern Physics*, v. 55, n. 2, p. 331–402, abr. 1983.
- 25 NOZIÈRES, P.; BLANDIN, A. Kondo effect in real metals. *Journal de Physique*, v. 41, n. 3, p. 193–211, mar. 1980.
- 26 SEAMAN, C. L. et al. Evidence for non-Fermi liquid behavior in the Kondo alloy  $\text{Y}_{1-x}\text{U}_x\text{Pd}_3$ . *Physical Review Letters*, v. 67, n. 20, p. 2882–2885, nov. 1991.

- 27 COX, D. L.; ZAWADOWSKI, A. Exotic Kondo effects in metals: Magnetic ions in a crystalline electric field and tunnelling centres. *Advances in Physics*, v. 47, n. 5, p. 599–942, set. 1998.
- 28 FERREIRA, J. a. V. B. et al. Magnetic susceptibility of a two-channel Anderson model. *Journal of Magnetism and Magnetic Materials*, v. 226-230, p. 132–133, maio 2001.
- 29 FERREIRA, J. a. V. B. et al. Non-Fermi liquid fixed points of a two-channel Anderson model. *Journal of Magnetism and Magnetic Materials*, v. 226-230, p. 196–198, maio 2001.
- 30 WILSON, K. G. Renormalization Group and Critical Phenomena. I. Renormalization Group and the Kadanoff Scaling Picture. *Physical Review B*, v. 4, n. 9, p. 3174–3183, nov. 1971.
- 31 WILSON, K. G. Renormalization Group and Critical Phenomena. II. Phase-Space Cell Analysis of Critical Behavior. *Physical Review B*, v. 4, n. 9, p. 3184–3205, nov. 1971.
- 32 OLIVEIRA, L. N.; WILKINS, J. W. Specific Heat of the Kondo Model. *Physical Review Letters*, v. 47, n. 21, p. 1553–1556, nov. 1981.
- 33 LÍBERO, V. L.; OLIVEIRA, L. N. Spectral density for fermion tunneling between two centers in a metallic environment. *Physical Review Letters*, v. 65, n. 16, p. 2042–2045, out. 1990.
- 34 YOSHIDA, M.; WHITAKER, M. A.; OLIVEIRA, L. N. Renormalization-group calculation of excitation properties for impurity models. *Physical Review B*, v. 41, n. 13, p. 9403–9414, maio 1990.
- 35 BULLA, R.; COSTI, T. A.; PRUSCHKE, T. Numerical renormalization group method for quantum impurity systems. *Reviews of Modern Physics*, v. 80, n. 2, p. 395–450, abr. 2008.
- 36 DAGOTTO, E. Correlated electrons in high-temperature superconductors. *Rev. Mod. Phys.*, v. 66, n. 3, p. 763–840.
- 37 FERREIRA, J. V. B. *Estudo do calor específico de um sistema de dois níveis acoplados a um banho fermiônico*. Dissertação (Mestrado em Física Básica) — Universidade de São Paulo, São Carlos, Instituto de Física de São Carlos, São Paulo, 1995.

- 38 COSTA, S. C. *Calor específico do modelo de Anderson de uma impureza por grupo de renormalização numérico*. Dissertação (Mestrado em Física Básica) — Universidade de São Paulo, São Carlos, Instituto de Física de São Carlos, São Paulo, 1995.
- 39 MANCINI, J. D.; MATTIS, D. C. Application of tridiagonalization to the many-body problem. *Physical Review B*, v. 28, n. 10, p. 6061–6063, nov. 1983.
- 40 JONES, B. A.; VARMA, C. M.; WILKINS, J. W. Low-Temperature Properties of the Two-Impurity Kondo Hamiltonian. *Physical Review Letters*, v. 61, n. 1, p. 125–128, jul. 1988.
- 41 FETTER, A. L.; WALECKA, J. D. *Quantum Theory of Many-Particle Systems*. Stanford: Courier Corporation, 2012.
- 42 KÖHLER, A. *Der C/C++-Projektbegleiter: C/C++ Projekte planen, dokumentieren, bauen und testen*. 1. Aufl. ed. Heidelberg: dpunkt-Verl, 2007.
- 43 FORBRIG, P.; KERNER, I. O.; CLEVE, J. (Ed.). *Programmierung: Paradigmen und Konzepte*. München: Fachbuchverl. Leipzig, 2006. (Lehr- und Übungsbuch Informatik).
- 44 WILLMS, A. *C-Programmierung lernen: anfangen, anwenden, verstehen*. München: Pearson Deutschland GmbH, 1998.
- 45 LANGTANGEN, H. P. *Object-Oriented Programming*. Berlin: Springer, Berlin, Heidelberg, 2016. (Texts in Computational Science and Engineering).
- 46 FERREIRA, J. V. B. *Modelo de Anderson de dois canais*. Tese (Doutorado em Física Básica) — Universidade de São Paulo, São Carlos, Instituto de Física de São Carlos, São Paulo, 2000.
- 47 DEITEL, P. J.; DEITEL, H. M. *C++ how to program*. Heidelberg: PearsonPrentice Hall, 2008.
- 48 VOLUNTÁRIOS, V. Valgrind home. Acessado no dia 14.10.2018: <http://valgrind.org/>.
- 49 WEIDENDORFER, J. KCachegrind. Acessado no dia 14.10.2018: <https://kcachegrind.github.io/html/Home.html>.
- 50 STROUSTRUP, B. Google C++ Style Guide. Acessado no dia 14.10.2018: <https://google.github.io/styleguide/cppguide.html>.