

Yuri Rossi Tonin

**Processamento de imagens de Ressonância
Magnética para avaliação de cirrose e função
hepática.**

São Carlos, Brasil

Setembro 2018

Agradecimentos

Em primeiro lugar, agradeço e dedico este trabalho a meus pais, os quais não pouparam esforços para me proporcionar o melhor conforto e educação possíveis. Sem vocês, eu não teria chegado até aqui. Essa conquista também é de vocês. Obrigado!

Não posso deixar também de agradecer em especial alguns professores que foram fundamentais, não apenas para meu aprendizado, mas também para que eu mantivesse meu entusiasmo com o curso e continuasse sonhando com o futuro.

Agradeço imensamente ao professor Emanuel Henn, por ter me acolhido como aluno de Iniciação Científica ainda no começo da graduação e por ter me ensinado tanto sobre o meio acadêmico e sobre a Física em si, sempre com muita atenção e empatia, nesses vários anos que trabalhamos juntos. Muito obrigado!

Agradeço também ao professor Paulo Liboni, que com suas excepcionais aulas e dedicação, ajudou-me a acreditar novamente em meu potencial e a perseverar no bacharelado. Fico feliz em tê-lo como grande amigo depois de todos esses anos. Obrigado!

Agradeço ao professor Fernando Paiva, que me orientou na execução deste trabalho de maneira extremamente amigável e me ajudou sempre que necessário, fazendo deste trabalho uma das experiências mais empolgantes que tive ao longo da graduação. Muito obrigado!

Ao professor Fernando Araújo Moreira, obrigado pelo constante apoio nesta fase final da graduação.

Por último, agradeço a minha irmã Yane e a Maria Luisa Consalter, por terem revisado este trabalho e por terem sido parte dos momentos mais felizes que tive durante os anos de graduação.

*“The reasonable man adapts himself to the world:
the unreasonable one persists in trying to adapt the world to himself.
Therefore all progress depends on the unreasonable man.”*

George Bernard Shaw

Sumário

1	RESUMO	7
2	OBJETIVOS	9
3	INTRODUÇÃO	11
3.1	A Ressonância Magnética	11
3.2	Fluxo magnético para detecção de sinais	12
3.3	Os dipolos elétrico e magnético	12
3.4	Interação do momento de dipolo com um campo magnético	14
3.5	Modelo semi-clássico: introduzindo o spin no modelo clássico	15
3.6	A equação de movimento para o spin	16
3.7	Magnetização	18
3.7.1	Magnetização de equilíbrio de uma amostra	19
3.8	Sistema de coordenadas girante	19
3.9	Pulso de RF e Ressonância	20
3.9.1	Pulso de excitação e o ângulo de flip θ	23
3.10	Relaxação	24
3.10.1	A interação spin-rede	24
3.10.2	A interação spin-spin	25
3.10.3	Equação de Bloch	27
3.11	Spin Eco	28
3.12	Contraste	32
3.13	Seleção de Cortes	33
3.14	Codificação de Frequência	33
3.14.1	Equação de imagem unidimensional e a Transformada de Fourier	35
3.14.2	O espaço-k	36
3.14.3	Exemplos analítico e com simulação	37
3.15	Codificação de fase	40
3.15.1	Equação de imagem multidimensional	40
3.15.2	O espaço-k bidimensional	41
3.16	Gradiente Eco	43
3.16.1	Tempo de Repetição T_R	45
3.16.2	Comparação de sequências Spin Eco e Gradiente Eco	46
3.17	Filtros no espaço-k: compreendendo frequências espaciais	47
4	MATERIAIS E MÉTODOS	51

4.1	Aquisição com tempo de repetição T_R curto	51
4.2	Método 1: Linearização	53
4.3	Método 2: Razão entre os sinais	54
4.3.1	Correção para aquisições distintas	55
4.4	Limitações dos Métodos	55
4.4.1	Ângulos ideais	56
4.4.2	Incertezas no ângulo θ	56
4.5	Aquisição das imagens de Ressonância Magnética	57
4.6	Interface Gráfica	59
5	RESULTADOS	61
5.1	Análise com a Interface Gráfica	61
5.2	Simulações	61
5.3	Fonte do erro	64
5.4	Obtenção do fator “F” de Normalização	64
6	DISCUSSÃO	69
7	CONCLUSÃO	71
8	AVALIAÇÃO DO ORIENTADOR	73
	REFERÊNCIAS	75
	APÊNDICES	79
	APÊNDICE A – RESOLUÇÃO ANALÍTICA DA TF DO SINAL . .	81
	APÊNDICE B – CÓDIGO DA SIMULAÇÃO EM PYTHON . . .	83
	APÊNDICE C – CÓDIGO DA INTERFACE GRÁFICA EM PYTHON .	89

1 Resumo

O diagnóstico de doenças hepáticas tem na biópsia a sua melhor opção. No entanto, problemas como o custo, erros de amostragem e variações entre diferentes observadores fazem deste método de diagnóstico o melhor até o momento, mas não o ideal ([BEDOSSA; CARRAT, 2009](#)).

Neste trabalho, buscamos implementar um método de processamento de imagem de Ressonância Magnética (RM) para diagnóstico de doenças no fígado. Para tal, extraímos o tempo de relaxação longitudinal T_1 do fígado por meio de imagens de RM. Trabalhos recentes mostraram que, quando um agente de contraste denominado *ácido gadoxético* é injetado no paciente ([BESA et al., 2015](#)), T_1 é alterado de forma diferente para tecidos doentes e saudáveis. Portanto, ao diferenciar T_1 , também diferenciamos o estado de saúde do tecido e podemos realizar um diagnóstico.

O trabalho foi realizado de Fevereiro de 2017 a Setembro de 2018, sendo orientado pelo professor Fernando Fernandes Paiva do Instituto de Física de São Carlos. Durante os primeiros 6 meses do projeto, foram estudados os fundamentos da Ressonância Magnética e realizadas simulações sobre fenômeno e sobre o processo de formação de imagem, apresentados aqui no capítulo [3](#).

Durante os meses seguintes, foram realizados o desenvolvimento e implementação do processamento propriamente dito. O principal objetivo, o qual também demandou maior esforço durante o projeto, foi o desenvolvimento de uma Interface Gráfica que poderia ser utilizada pelo usuário para realizar o processamento de forma rápida e eficaz, fornecendo um diagnóstico imediatamente após o exame de RM. O objetivo seria fornecer a Interface Gráfica ao Instituto D'Or de Pesquisa e Ensino (IDOR), colaborador do projeto, para ser utilizada em futuros diagnósticos.

Os dados utilizados para o processamento e análises foram adquiridos pelo IDOR. Alguns problemas foram encontrados durante a implementação do método. Realizando simulações e análises, pudemos identificar a fonte desses problemas como inerente à aquisição das imagens. Felizmente, tais problemas puderam ser contornados com um processo de normalização, ao custo de deteriorar a precisão quantitativa do método. No entanto, os resultados concordam com outros trabalhos na literatura ([KATSUBE et al., 2011; HAIMERL et al., 2013; BESA et al., 2015](#)) e reforçam que esta é alternativa promissora para diagnóstico de função hepática.

2 Objetivos

Buscamos desenvolver e implementar um método de processamento de imagens de Ressonância Magnética por meio da extração do tempo de relaxação longitudinal T_1 de tecidos hepáticos, a fim de diagnosticar doenças como cirrose, esteatose, fibrose e hepatite.

Apesar de alguns trabalhos já reportarem a utilização de T_1 junto de um agente de contraste como meio de diagnóstico, muitas vezes o tratamento matemático do método é parcialmente negligenciado. Portanto, queríamos também esclarecer a modelagem utilizada e reforçar a eficácia do método.

Finalmente, buscamos implementar o processamento dentro de uma Interface Gráfica de fácil utilização, a qual poderia ser utilizada pelo usuário final (no caso, nossos colaboradores do Instituto D'Or) para diagnosticar o fígado de um paciente imediatamente após a realização do exame.

3 Introdução

Introduzimos neste capítulo os principais conceitos físicos que explicam o fenômeno da Ressonância Magnética. Também apresentamos o comportamento do sinal de RM e como ocorre seu processamento. A principal referência foi o livro “Magnetic Resonance Imaging: Physical Principles and Sequence Design” ([BROWN et al., 2014](#)). Demais referências são citadas explicitamente no texto.

3.1 A Ressonância Magnética

Segundo a Mecânica Clássica, uma das propriedades conservativas da matéria é o momento angular orbital \vec{L} . Com o advento da Mecânica Quântica, descobriu-se que a matéria possui um segundo tipo de momento angular, o *momento angular intrínseco* \vec{J} , comumente chamado de *spin*.

A Ressonância Magnética trata da interação do spin com campos magnéticos. Como o spin é uma propriedade quântica da matéria, essa é uma interação fundamentalmente quântica. Felizmente, os resultados obtidos pela Mecânica Quântica para esse sistema são análogos a fenômenos clássicos bem conhecidos, como a precessão. Assim, podemos realizar uma modelagem clássica mais simples dos fenômenos, a qual fornece uma noção intuitiva sobre a dinâmica dos spins na RM.

Devido ao escopo deste trabalho, não apresentaremos os fundamentos quânticos da RM, os quais podem ser encontrados na referência principal ([BROWN et al., 2014](#)). Nas seções seguintes, introduzimos os modelos clássicos e fornecemos os valores numéricos de algumas quantidades de maior relevância. Fundamentos teóricos mais detalhados podem ser encontrados na literatura ([BROWN et al., VLAARDINGERBROEK; BOER](#)).

Primeiro, descrevemos o fenômeno em linhas gerais. Vários elementos químicos presentes no corpo humano possuem spin não-nulo, os quais funcionam como fonte de sinal magnético quando perturbados. Em aplicações biomédicas, a RM utiliza principalmente o spin dos núcleos de Hidrogênio (próton). Na presença de um campo magnético externo, o conjunto dos spins que compõem o tecido biológico funciona como um pequeno imã que gera um campo magnético ao seu redor. Aplica-se então um pulso magnético que retira o conjunto de spins de sua posição de equilíbrio. O sinal de ressonância é adquirido por meio da variação do fluxo magnético gerado pelos spins enquanto eles retornam ao equilíbrio, fenômeno chamado de *relaxação*. Diferentes tecidos biológicos relaxam em tempos distintos e, portanto, geram sinais distintos, fornecendo assim um mecanismo de contraste. Perante a aplicação de uma sequência específica de campos magnéticos sobre a amostra, o sinal

proveniente dos spins também possui informação sobre a posição de cada um deles. Com isso, o processamento do sinal permite reconstruir uma imagem.

3.2 Fluxo magnético para detecção de sinais

Nesta seção, recordamos brevemente como o fluxo magnético é utilizado para gerar um sinal elétrico.

Considere um circuito S na presença de um campo magnético \vec{B} , análogo ao ilustrado na figura 1. O fluxo magnético Φ que atravessa S é definido por

$$\Phi = \int_S \vec{B} \cdot d\vec{S} \quad (3.1)$$

Pela lei de Lenz, temos que a variação de Φ no tempo resulta numa força eletromotriz (emf) sobre o circuito

$$\text{emf} = -\frac{d\Phi}{dt} \quad (3.2)$$

Portanto, as equações 3.1 e 3.2 fornecem

$$\text{emf} = -\int_S \frac{d\vec{B}}{dt} \cdot d\vec{S} \quad (3.3)$$

Ou seja, quanto maior a variação temporal do campo magnético no interior do circuito, maior será a força eletromotriz gerada. A corrente elétrica gerada pela emf pode ser utilizada como um sinal elétrico, a partir do qual inferimos o comportamento do campo magnético gerador do fluxo.

3.3 Os dipolos elétrico e magnético

Também é importante recordar o conceito de um dipolo. O dipolo elétrico é um objeto bem conhecido do Eletromagnetismo: trata-se de um par de cargas elétricas de sinais opostos $+q$ e $-q$ separadas por uma distância d (figura 2).

O campo elétrico \vec{E} gerado por esse par de cargas a uma distância $r \gg d$ é dado por (GRIFFITHS, 2014)

$$\vec{E}(\vec{r}) = \frac{1}{4\pi\epsilon_0} \frac{1}{r^3} [3(\vec{p} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}} - \vec{p}] \quad (3.4)$$

onde

$$\vec{p} = qd\hat{n} \quad (3.5)$$

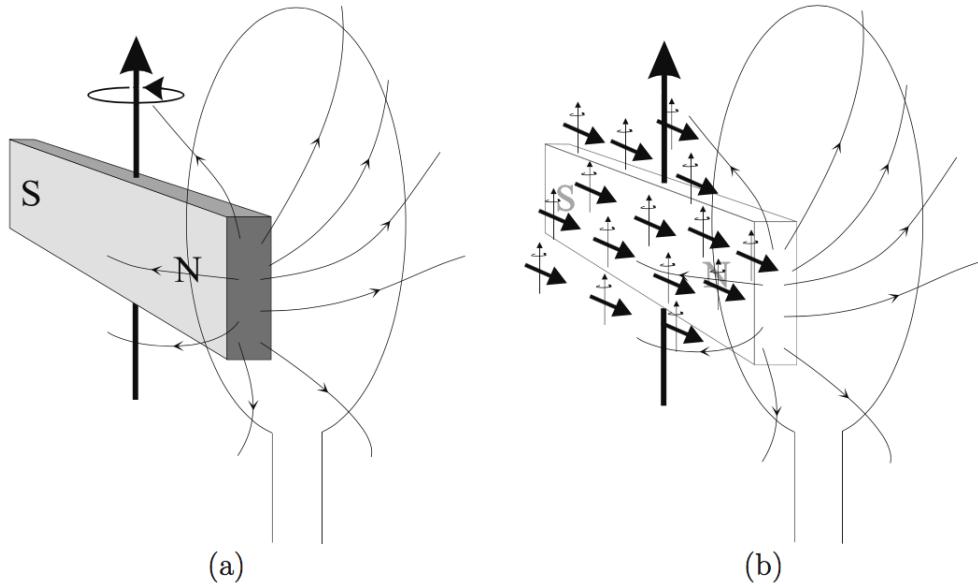


Figura 1 – Na figura (a) temos um imã gerador de um campo magnético, o qual atravessa um circuito. Se o imã rotaciona em torno do eixo da figura, o campo magnético que atravessa o circuito varia e, consequentemente, surge uma corrente elétrica. A figura (b) ilustra um conjunto de spins funcionando como o imã da figura ao lado. Figura retirada de ([BROWN et al., 2014](#)).

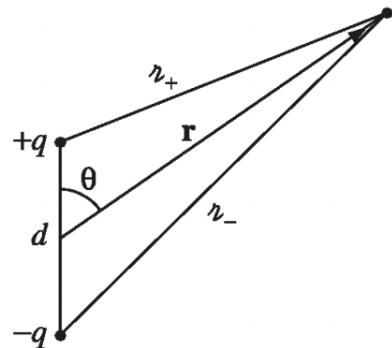


Figura 2 – Esquema de um dipolo. Figura retirada de (GRIFFITHS, 2014).

é denominado *momento de dipolo elétrico*, sendo \hat{n} o versor que aponta da carga negativa para a positiva.

Considere agora um circuito de pequenas dimensões pelo qual passa uma corrente elétrica I . Nesse caso, calcula-se que o campo magnético \vec{B} gerado pelo circuito a uma distância r (grande em relação às dimensões do circuito) é ([GRIFFITHS, 2014](#))

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \frac{1}{r^3} [3(\vec{\mu} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}} - \vec{\mu}] \quad (3.6)$$

onde $\vec{\mu} = I\vec{a}$, sendo \vec{a} o vetor de área do circuito cuja direção é perpendicular ao plano do circuito, i.e. é dada pela usual “regra da mão-direita”.

Percebe-se imediatamente a semelhança entre as equações 3.4 e 3.6, onde as quantidades \vec{p} e $\vec{\mu}$ são análogas. Por este motivo, dá-se a quantidade $\vec{\mu}$ o nome de *momento*

de dipolo magnético. Repare que $\vec{\mu}$ possui a mesma direção de \vec{a} , ou seja, é perpendicular ao plano do circuito.

3.4 Interação do momento de dipolo com um campo magnético

Apresentamos a seguir modelos clássicos que descrevem a interação do momento de dipolo com um campo magnético. Veremos posteriormente que o spin pode ser interpretado como um dipolo magnético. Portanto, entender essa interação dipolo-campo fornece uma noção intuitiva da dinâmica dos spins no fenômeno da RM.

O torque na presença de um campo uniforme

Considere um circuito fechado pelo qual passa uma corrente I (figura 3). Quando inserido num campo magnético uniforme \vec{B} , surge sobre cada elemento $d\vec{l}$ do circuito uma força, chamada força de Lorentz:

$$d\vec{F} = Id\vec{l} \times \vec{B} \quad (3.7)$$

Ao integrar a equação acima sobre o circuito, temos que a força resultante é nula: as forças em pontos opostos do circuito se cancelam mutuamente, como ilustra a figura 3. No entanto, o mesmo não é válido para o torque \vec{N} . Ao somar a contribuição de todos os elementos de torque $d\vec{N} = \vec{r} \times d\vec{F}$, o torque resultante \vec{N} só será nulo para o caso em que o campo \vec{B} e o momento de dipolo $\vec{\mu}$ estão alinhados (figura 3a). Para demais situações em que essa condição não é satisfeita (figura 3b), surge um torque resultante não-nulo dado por

$$\vec{N} = \vec{\mu} \times \vec{B} \quad (3.8)$$

o qual moverá o sistema à configuração de equilíbrio onde \vec{B} e $\vec{\mu}$ estarão alinhados.

Dipolo em um campo não-uniforme

Se expandirmos a análise para o caso de um campo \vec{B} não-uniforme, teremos uma força resultante dada por

$$\vec{F} = -\nabla U = \nabla(\vec{\mu} \cdot \vec{B}) \quad (3.9)$$

onde

$$U = -\vec{\mu} \cdot \vec{B} = -\mu B \cos(\theta) \quad (3.10)$$

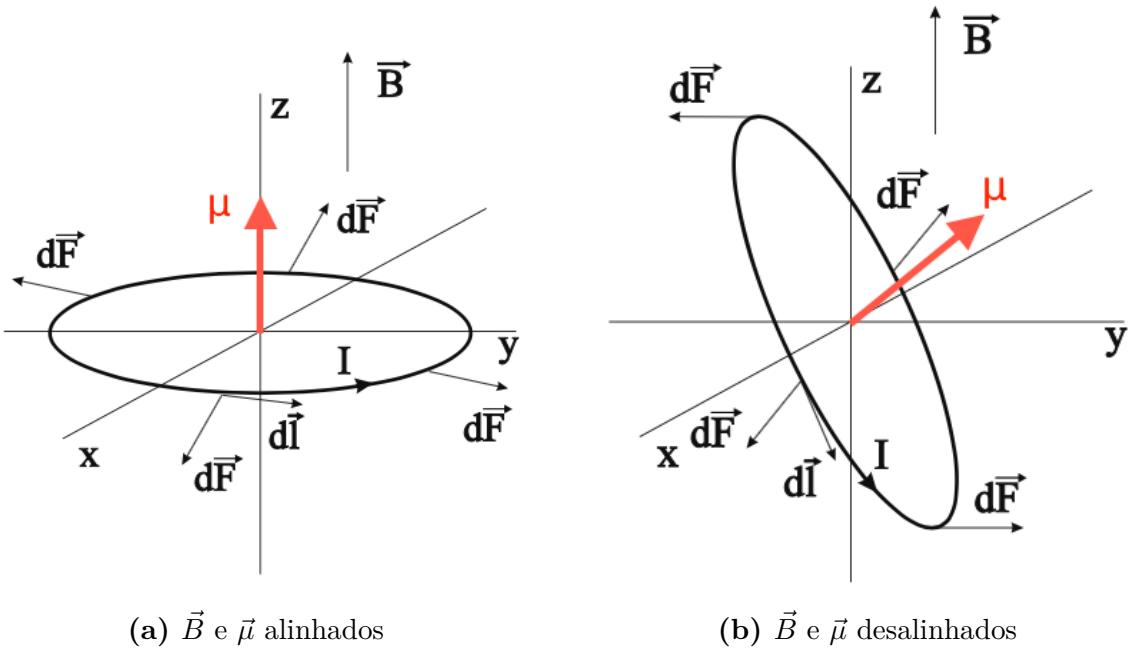


Figura 3 – Esquemas de circuitos fechados na presença de um campo magnético uniforme.
Figura adaptada de (BROWN et al., 2014)

é a energia potencial do dipolo e θ o ângulo entre os vetores $\vec{\mu}$ e \vec{B} . Dessa forma, temos que as orientações de maior e menor energia para o dipolo são paralelas ao campo aplicado ($\cos(\theta) = \pm 1$).

Na Mecânica Quântica, partículas de spin 1/2 (como o próton) possuem apenas dois estados possíveis de energia: spin *up* e spin *down*. Dizemos que há, respectivamente, um alinhamento paralelo e anti-paralelo dos spins, em analogia aos dois valores extremos de U provenientes da equação 3.10.

3.5 Modelo semi-clássico: introduzindo o spin no modelo clássico

Até o momento, afirmamos que o spin pode ser considerado um momento magnético $\vec{\mu}$, mas lidamos apenas com equações e construções clássicas. É natural questionarmos de onde surge, efetivamente, a relação entre o spin e o momento magnético $\vec{\mu}$.

Experimentalmente, obtém-se uma relação direta entre o momento angular intrínseco \vec{J} e o momento de dipolo magnético $\vec{\mu}$ da amostra:

$$\vec{\mu} = \gamma \vec{J} \quad (3.11)$$

A constante γ é chamada razão giromagnética e depende do elemento químico da amostra.

Tabela 1 – Alguns elementos também comuns no corpo humano, como o carbono, foram omitidos, pois seu spin (e , portanto, seu momento magnético) é nulo. O sinal negativo para o momento magnético indica um alinhamento anti-paralelo. Tabela retirada de (BROWN et al., 2014).

Núcleo	Spin	Momento Magnético	γ	Abundância no corpo humano
Hidrogênio	1/2	2.7928	42.58	88 M
Sódio	3/2	2.2175	11.27	80 mM
Fósforo	1/2	1.1316	17.25	75 mM
Oxigênio	5/2	-1.8938	-5.77	17 mM
Flúor	1/2	2.6289	40.08	4 μ M

Para o próton livre,

$$\gamma = 2.675 \times 10^8 \text{ rad s}^{-1} \text{ T}^{-1} \quad (3.12)$$

Com tal precisão numérica, este valor de γ é o mesmo para os prótons ligados ao oxigênio da água. São estes prótons os de maior interesse em imagens de Ressonância Magnética. Ao invés de γ , é comum utilizar em equações

$$\gamma = \frac{\gamma}{2\pi} = 42.58 \text{ MHz T}^{-1} \quad (3.13)$$

A tabela 1 compara os valores de γ para elementos comuns do corpo humano. Repare que o Hidrogênio possui o maior valor, o que implica um maior momento magnético. Como veremos na seção 3.7, quanto maior for $\vec{\mu}$, maior será a magnetização \vec{M} da amostra. Isso, aliado à abundância do Hidrogênio no corpo humano, resulta num sinal maior. Daí a preferência em utilizar o spin do Hidrogênio na RM ao invés do spin de outros elementos.

3.6 A equação de movimento para o spin

De acordo com a Mecânica Clássica, a variação do momento angular no tempo é igual ao torque exercido sobre aquele sistema. Portanto, na ausência de momento angular orbital \vec{L} , temos:

$$\vec{N} = \frac{d\vec{J}}{dt} \quad (3.14)$$

Igualando as equações 3.8 e 3.14 e substituindo 3.11, obtemos a equação diferencial

$$\frac{d\vec{\mu}}{dt} = \gamma \vec{\mu} \times \vec{B} \quad (3.15)$$

Está é uma versão simplificada da chamada equação de Bloch, a qual descreve a dinâmica do spin na presença de um campo externo.

Pela equação 3.15, a taxa de variação no tempo do vetor $\vec{\mu}$ é proporcional a um produto vetorial que envolve o próprio $\vec{\mu}$. Qualitativamente, isso indica que a magnitude desse vetor mantém-se constante no tempo, enquanto que sua direção varia. Com o auxílio da figura 4, analisamos agora como se dá essa variação.

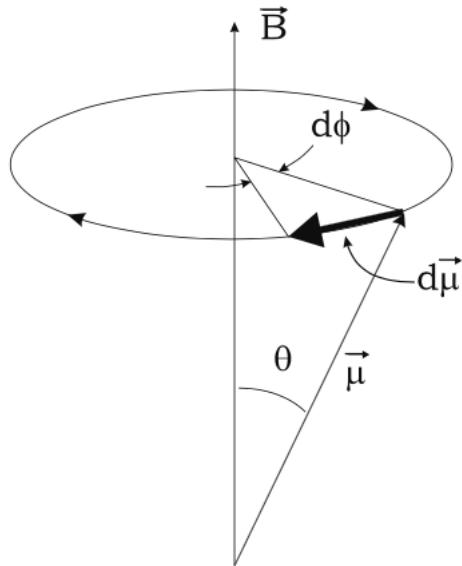


Figura 4 – Esquema da precessão do spin em volta do campo magnético

Considere a variação diferencial $d\vec{\mu}$ do momento magnético num tempo dt .

$$d\vec{\mu} = \gamma\vec{\mu} \times \vec{B} dt \quad (3.16)$$

Vemos que $d\vec{\mu}$ é perpendicular ao plano definido por $\vec{\mu}$ e \vec{B} , ou seja, $\vec{\mu}$ rotaciona ao redor de \vec{B} numa mesma trajetória (desde que \vec{B} mantenha-se constante no tempo).

Pela figura 4, vemos que a variação diferencial da magnitude do momento de dipolo $|d\vec{\mu}|$ é dada por

$$|d\vec{\mu}| = \vec{\mu} \sin \theta |d\phi| \quad (3.17)$$

Por outro lado, a equação 3.16 também implica

$$|d\vec{\mu}| = \gamma\mu B \sin \theta dt \quad (3.18)$$

Igualando as equações 3.17 e 3.18, obtemos a chamada *frequência de precessão de Larmor*.

$$\omega_0 = \left| \frac{d\phi}{dt} \right| = \gamma B \quad (3.19)$$

Ou seja, na condição de equilíbrio, o modelo semi-clássico mostra que o vetor momento de dipolo (spin) precessiona ao redor do campo estático com velocidade angular ω_0 devido

à presença do torque \vec{N} , diferindo do modelo clássico onde há um alinhamento perfeito entre $\vec{\mu}$ e \vec{B} .

3.7 Magnetização

Ao analisar amostras macroscópicas, é conveniente definir o momento magnético por unidade de volume, ou magnetização \vec{M} . Dado um volume V contendo um número grande de prótons, mas ainda pequeno o suficiente para que o campo magnético externo \vec{B}_{ext} seja aproximadamente constante em V , define-se a magnetização a partir da soma dos i momentos magnéticos individuais:

$$\vec{M} = \frac{1}{V} \sum_i \vec{\mu}_i \quad (3.20)$$

Se somarmos as i equações 3.15 e dividirmos por V , obtemos

$$\frac{1}{V} \sum_i \frac{d\vec{\mu}_i}{dt} = \frac{\gamma}{V} \sum_i \vec{\mu}_i \times \vec{B}_{ext} \quad (3.21)$$

ou

$$\frac{d\vec{M}}{dt} = \gamma \vec{M} \times \vec{B}_{ext} \quad (3.22)$$

Ou seja, temos a equação de Bloch também para a magnetização. Podemos analisar a 3.22 separadamente para sua componente longitudinal M_z e para a componente transversal $\vec{M}_t = M_x \hat{x} + M_y \hat{y}$. Por simplicidade, escolhemos o sistema de coordenadas alinhado com o campo magnético externo: $\vec{B}_{ext} = B_0 \hat{z}$. Assim, as componentes ficam

$$\frac{dM_z}{dt} = 0 \quad (3.23)$$

e

$$\frac{d\vec{M}_t}{dt} = \gamma \vec{M}_t \times \vec{B}_{ext} \quad (3.24)$$

Essas equações são válidas para prótons livres, i.e. não consideramos interações dos prótons com sua vizinhança ou entre si. Veremos na seção 3.10 como a introdução de tais interações altera as equações 3.23 e 3.24, resultando na equação de Bloch em sua forma geral.

3.7.1 Magnetização de equilíbrio de uma amostra

Mencionamos na seção 3.4 que partículas de spin 1/2, como o íon Hidrogênio, possuem apenas dois níveis de energia possíveis. Esses níveis são (GRIFFITHS, 2005)

$$E_+ = +\frac{\hbar\omega_0}{2} \quad \text{e} \quad E_- = -\frac{\hbar\omega_0}{2} \quad (3.25)$$

onde \hbar é a constante de Planck reduzida e ω_0 a frequência de Larmor. A diferença de energia entre os dois estados é $\Delta E = \hbar\omega_0$; E_+ é chamado spin *up* (alinhamento paralelo), enquanto E_- é chamado spin *down* (alinhamento anti-paralelo).

Em temperatura ambiente T , essa diferença $\Delta E = \hbar\omega_0$ é milhões de vezes menor que a energia térmica média kT do sistema, onde k é a constante de Boltzmann. Como consequência, o número de spins paralelos excede por pouco o número de spins anti-paralelos, pois o excesso depende da razão $\hbar\omega_0/kT$:

$$\text{excesso de spins} \approx N \frac{\hbar\omega_0}{2kT} \quad (3.26)$$

sendo $\hbar\omega_0/kT \ll 1$ para temperatura ambiente e N o número total de spins.

Como o sinal de RM provém justamente da detecção desse excesso de spins paralelos, a equação anterior indica *a priori* que não haveria um sinal significativo em condições ambientes. No entanto, é preciso lembrar que há um grande número de átomos de Hidrogênio em alguns gramas de tecido do corpo humano, uma vez que a quantidade de água é alta. Portanto, o valor de N é grande e ajuda a compensar os efeitos térmicos.

A magnetização total de equilíbrio devido aos spins da amostra é dada por

$$M_0 = \rho_0 \frac{\hbar^2 \gamma^2}{4kT} B_0 \quad (3.27)$$

onde ρ_0 é o número de prótons por unidade de volume (densidade de prótons). Portanto, temos que a magnetização pode gerar um sinal grande o suficiente desde que os valores de ρ_0 , B_0 e γ também sejam altos. Como dito, utiliza-se o Hidrogênio em RM, elemento que, além de abundante, apresenta o maior valor de γ dentre os principais elementos do corpo humano (vide tabela 1). Já em relação a intensidade do campo \vec{B}_0 , os valores são bem altos: a maioria dos campos estáticos utilizados possuem pelo menos 1 T. Para efeito de comparação, o campo magnético de uma imã de geladeira possui aproximadamente 5 mT (Information on MRI Technique, 2005).

3.8 Sistema de coordenadas girante

A fim de simplificar algumas equações e assim visualizar o comportamento do vetor $\vec{\mu}$ de forma mais intuitiva, é conveniente descrevê-lo em um sistema de coordenadas

girante.

Considere um sistema de coordenada cuja base são os versores \hat{x}' , \hat{y}' e \hat{z}' , girando com velocidade angular $\vec{\Omega}$ em torno do sistema do laboratório \hat{x} , \hat{y} e \hat{z} . Um vetor $\vec{\mu}$ é descrito no sistema do laboratório por

$$\vec{\mu}(t) = \mu_x(t)\hat{x} + \mu_y(t)\hat{y} + \mu_z(t)\hat{z} \quad (3.28)$$

No sistema girante, o mesmo vetor torna-se

$$\vec{\mu}'(t) = \vec{\mu}(t) = \mu_{x'}(t)\hat{x}'(t) + \mu_{y'}(t)\hat{y}'(t) + \mu_{z'}(t)\hat{z}'(t) \quad (3.29)$$

As variações temporais de $\vec{\mu}$ nos dois sistemas se relacionam de acordo com a equação

$$\frac{d\vec{\mu}}{dt} = \left(\frac{d\vec{\mu}}{dt} \right)' + \vec{\Omega} \times \vec{\mu} \quad (3.30)$$

onde

$$\left(\frac{d\vec{\mu}}{dt} \right)' = \frac{d\vec{\mu}_{x'}}{dt}\hat{x}' + \frac{d\vec{\mu}_{y'}}{dt}\hat{y}' + \frac{d\vec{\mu}_{z'}}{dt}\hat{z}' \quad (3.31)$$

Comparando 3.15 e 3.30, obtemos que a variação do vetor $\vec{\mu}$ no sistema girante é análoga à variação no sistema do laboratório

$$\left(\frac{d\vec{\mu}}{dt} \right)' = \gamma \vec{\mu} \times \vec{B}_{eff} \quad (3.32)$$

mas com o campo externo \vec{B} sendo substituído por um campo efetivo

$$\vec{B}_{eff} = \vec{B} + \frac{\vec{\Omega}}{\gamma} \quad (3.33)$$

Note que a direção do vetor $\vec{\Omega}$ não foi especificada. Assim, a direção do campo efetivo não será necessariamente a mesma do campo externo \vec{B} , o que influenciará em como a precessão será vista do sistema girante.

Na figura 5a temos um sinal oscilante de frequência ω_0 e amortecido. Já na figura 5b, temos este mesmo sinal visto de um sistema girante também com frequência ω_0 . Vemos que o mesmo se torna um simples decaimento exponencial.

3.9 Pulso de RF e Ressonância

Como mencionado, o sinal em Ressonância Magnética provém da variação do fluxo magnético produzido por $\vec{\mu}$ (ou macroscopicamente, pela magnetização \vec{M}) quando este

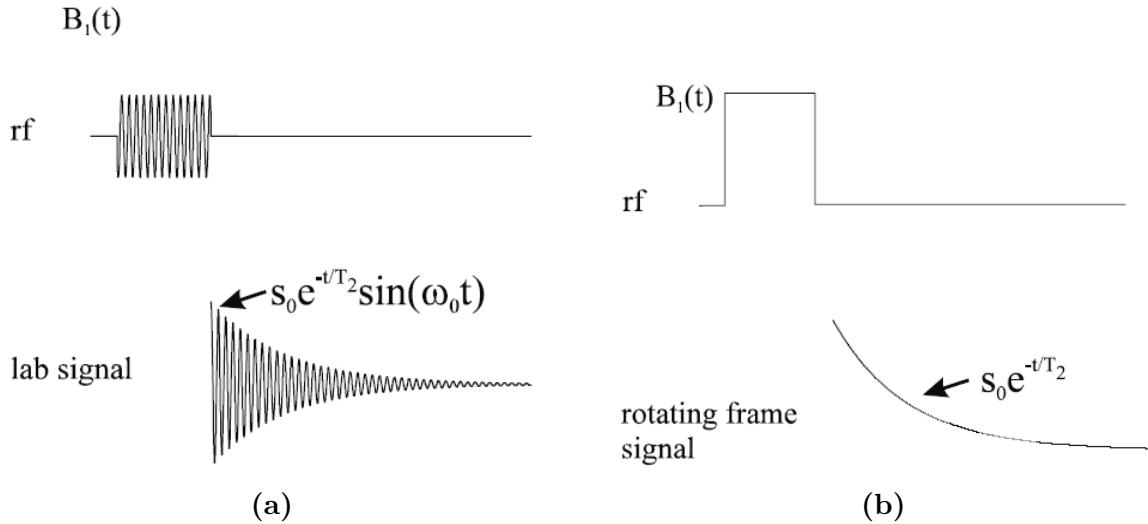


Figura 5 – Comparação de sinais nos sistemas do laboratório e girante. Figura retirada de (BROWN et al., 2014).

relaxa de um estado perturbado para o estado de equilíbrio. Normalmente, perturba-se o sistema com a aplicação de um campo magnético circularmente polarizado à esquerda

$$\vec{B}_1^{\text{circ}} = B_1(\hat{x} \cos(\omega t) - \hat{y} \sin(\omega t)) \quad (3.34)$$

que no sistema girante é simplesmente

$$\vec{B}_1^{\text{circ}} = B_1 \hat{x}' \quad (3.35)$$

Tal fato também é ilustrado na figura 5, onde um sinal oscilante no laboratório torna-se um pulso constante no sistema girante.

A adição do campo B_1 altera a equação 3.32 da seguinte forma

$$\left(\frac{d\vec{\mu}}{dt} \right)' = \gamma \vec{\mu} \times [\hat{z}'(\omega_0 - \omega) + \hat{x}'\omega_1] \quad (3.36)$$

onde ω_1 é a frequência de precessão devido à presença exclusiva do campo B_1 :

$$\omega_1 = \gamma B_1 \quad (3.37)$$

Nessa situação, $\vec{\mu}$ no sistema girante precessa ao redor de \vec{B}_{eff} , sendo este dado por

$$\vec{B}_{eff} = [\hat{z}'(\omega_0 - \omega) + \hat{x}'\omega_1]/\gamma \quad (3.38)$$

É de particular interesse o caso em que

$$\omega = \omega_0 \quad (3.39)$$

Essa é chamada de *condição de ressonância*, a qual dá origem ao nome “Ressonância Magnética”. Neste caso, \vec{B}_1 recebe o nome de *pulso de radiofrequência* (RF) e a equação 3.38 é simplificada para

$$\left(\frac{d\vec{\mu}}{dt} \right)' = \omega_1 \vec{\mu} \times \hat{x}' \quad (3.40)$$

Novamente, temos uma equação com a mesma forma da 3.15. Portanto, haverá precessão com frequência ω_1 ao redor do vedor \hat{x}' , i.e. a mesma direção de \vec{B}_1^{circ} . Assim, a escolha da direção de precessão depende exclusivamente do pulso de RF aplicado.

Consequentemente, na condição de ressonância até mesmo um pulso de baixa amplitude $B_1 = \omega_1/\gamma$ seria suficiente para rotacionar o spin para o plano transversal, fato esse desejado, pois a detecção do sinal provém justamente da magnetização transversal da amostra. Por outro lado, quanto mais distante da condição de ressonância estivermos, maior será a componente \hat{z}' na equação 3.38 e, portanto, o campo efetivo estará mais próximo do campo estático, de modo que a componente transversal será menor. A figura 6 ilustra o comportamento do vetor $\vec{\mu}$ no sistema girante e do laboratório para os casos dentro e fora de ressonância.

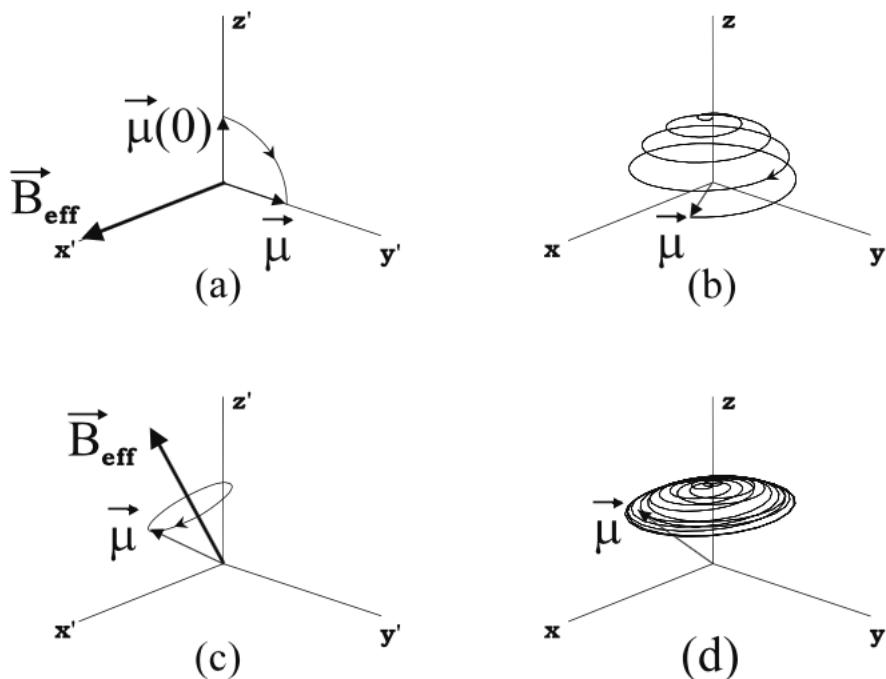


Figura 6 – Movimento do vetor magnetização nos sistemas girante e do laboratório. Ao aplicar o pulso de RF $\pi/2$ (figura a), o vetor é deslocado da posição de equilíbrio (alinhada ao eixo \hat{z}') movimentando-se perpendicularmente ao campo efetivo. Em 6b, temos o mesmo vetor visualizado agora no sistema do laboratório: ele parte do equilíbrio e desloca-se até o plano xy em caminho espiral devido à precessão. Fora da condição de ressonância, o campo não estará alinhado ao eixo \hat{x}' , como ilustra 6c. Nesse caso o vetor irá se deslocar pouco em relação a vertical, como ilustra a trajetória em 6d.

3.9.1 Pulso de excitação e o ângulo de flip θ

Como mencionado, na condição de ressonância a direção dos campos \vec{B}_{eff} e \vec{B}_1 será a mesma e a frequência de precessão é dada por $\omega_1 = \gamma B_1$. Em geral, temos que a amplitude B_1 do pulso de RF é uma função no tempo. Ao integrar essa equação no tempo, podemos calcular o ângulo θ de deslocamento do vetor em relação ao eixo z . Esse ângulo é chamado *ângulo de flip*:

$$\theta(t) = \gamma \int_0^t B_1(t') dt' \quad (3.41)$$

Se B_1 for constante e o pulso de RF for ligado por um tempo τ , temos

$$\theta = \gamma B_1 \tau \quad (3.42)$$

Para prótons, um pulso de $\tau = 1$ ms e ângulo de flip $\theta = 90^\circ$, requer um campo $B_1 = 5.9 \mu\text{T}$, valor bem menor que o do campo estático. Pulsos que deslocam o vetor por um ângulo de 90° são chamados de pulsos $\pi/2$.

No entanto, usualmente $B_1 = B_1(t)$ e esse envelope do pulso de RF pode ser escolhido de forma a excitar uma amostra de diferentes maneiras. Preferencialmente, deseja-se que a excitação da amostra seja uniforme, isto é, que o ângulo de flip seja o mesmo ao longo da região de interesse. Na condição de ressonância e para ângulos de flip pequenos ($\theta < 30^\circ$), temos que o perfil da região excitada é aproximadamente igual a amplitude da Transformada de Fourier de $B_1(t)$ (BERNSTEIN; KING; ZHOU, 2004). Consequentemente, um dos envelopes $B_1(t)$ mais utilizados são os pulsos da forma $\text{sinc}(t)$

$$\text{sinc}(t) = \frac{\sin(t)}{t} \quad (3.43)$$

ilustrado na figura 7a. Isso ocorre pois a Transformada de Fourier de tal função é o pulso retangular $\text{rect}(f)$, ilustrada na figura 7b.

No entanto, tal par de Transformada de Fourier só ocorre realmente para um pulso $\text{sinc}(t)$ infinito, o que é inviável. Na prática, a função $\text{sinc}(t)$ é truncada no tempo, resultando em um pulso que se aproxima do pulso retangular conforme o tempo de aplicação do pulso se aproxima de $t = \infty$. Um exemplo do perfil de excitação nesse caso é mostrado na figura 8.

Neste caso, o ângulo de flip não será exatamente o mesmo em toda a região da amostra, o que pode implicar em incertezas no valor do sinal medido.

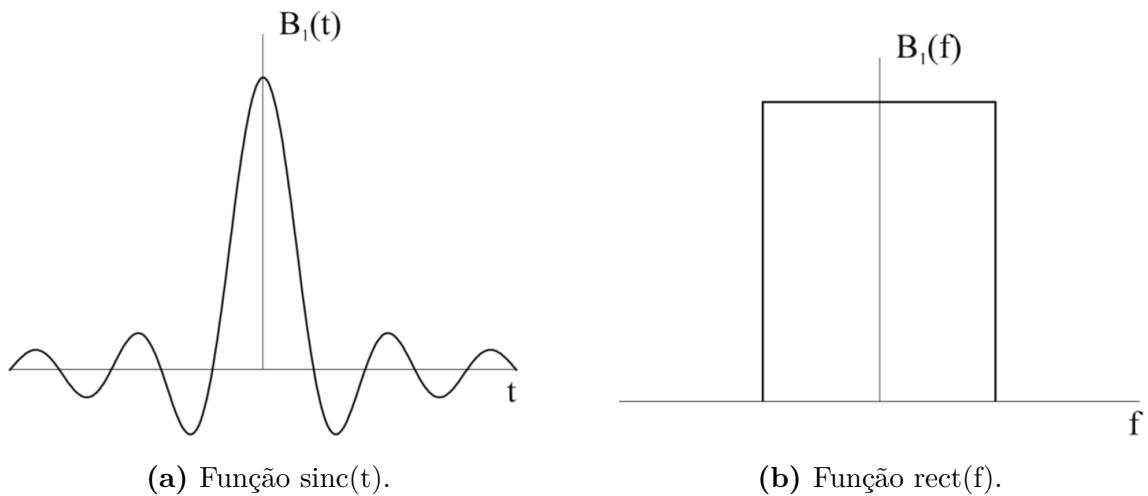


Figura 7 – Figuras retiradas de ([BROWN et al., 2014](#)).

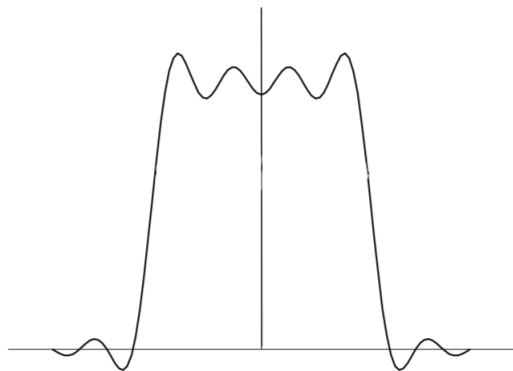


Figura 8 – A Transformada de Fourier da função $\text{sinc}(t)$, quando esta é truncada no tempo, não retorna uma função retangular perfeita, como na figura 7b. Quanto mais longo for o pulso no domínio do tempo, mais o pulso no domínio da frequência irá se aproximar da função retangular ideal. Figura adaptada de ([BROWN et al., 2014](#)).

3.10 Relaxação

Na dedução de [3.23](#) e [3.24](#), desconsideramos a interação dos spins com sua vizinhança. A seguir, veremos como a introdução de diferentes interações influenciam na relaxação da magnetização após a aplicação do pulso de RF. As equações [3.23](#) e [3.24](#) serão alteradas, resultando na forma completa da equação de Bloch.

3.10.1 A interação spin-rede

Quando inserimos a interação da rede com o spin, resultados da Mecânica Quântica mostram que a variação da componente M_z é proporcional à diferença $M_0 - M_z$ ([BROWN](#)

et al., 2014)

$$\frac{dM_z}{dt} = \frac{1}{T_1}(M_0 - M_z) \quad (3.44)$$

sendo M_0 a magnetização de equilíbrio e T_1 um fator empírico chamado *tempo de relaxação longitudinal*. Após a aplicação da RF, a evolução temporal de $M_z(t)$ é dada por

$$M_z(t) = M_z(0)e^{-t/T_1} + M_0(1 - e^{-t/T_1}) \quad (3.45)$$

Repare que para $t = 0$, resta apenas $M_z(0)$, isto é, a magnetização inicial após a aplicação do pulso de RF. Já para $t \rightarrow \infty$, recuperamos a magnetização de equilíbrio: $M_z(\infty) = M_0$.

A tabela 2 mostra valores típicos do tempo de relaxação T_1 para diferentes tecidos do corpo humano. Note que a variação de T_1 é relativamente grande de um tecido para outro. Portanto, se conseguirmos extrair o valor de T_1 dos diferentes tecidos, esse parâmetro pode funcionar como um bom mecanismo de contraste.

Tabela 2 – Tempo de relaxação longitudinal (T_1) e tranversal (T_2) para diferentes tecidos do corpo humano. Tabela retirada de (BROWN et al., 2014).

Tecido	T_1 (ms)	T_2 (ms)
Massa cinzenta	950	100
Massa branca	600	80
Músculo	900	50
Líquor	4500	2200
Gordura	250	60
Sangue	1200	100-200

3.10.2 A interação spin-spin

O campo magnético local que afeta um spin não provém apenas do campo externo aplicado, mas também dos campos criados pelos spins vizinhos. Pela equação 3.19, temos que pequenas variações do campo no espaço resultam em spins com diferentes frequências de precessão. Isso causa uma defasagem dos spin com o tempo e, consequentemente, uma redução da magnetização transversal \vec{M}_t , como mostra a figura 9.

Assim, caracteriza-se experimentalmente a taxa de redução da magnetização transversal por meio do *tempo de relaxação transversal* T_2 . Introduzindo essa variável, a equação 3.24 adquire a forma

$$\frac{d\vec{M}_t}{dt} = \gamma \vec{M}_t \times \vec{B}_{ext} - \frac{1}{T_2} \vec{M}_t \quad (3.46)$$

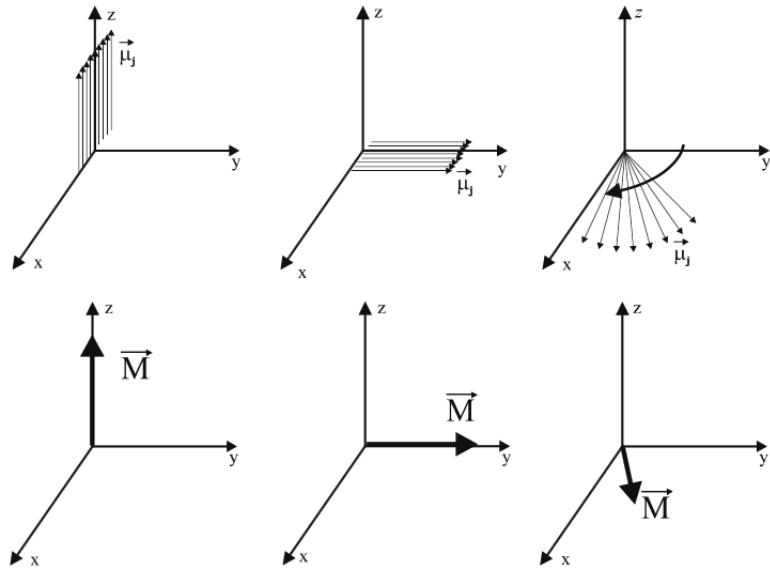


Figura 9 – Cada par de figuras na vertical mostra o alinhamento do spin e a magnetização resultante dos mesmos. Aplicada a RF, os spins são deslocados para o plano xy (figura central) e, então, são defasados com o tempo devido a diferenças locais do campo magnético. Figura retirada de (BROWN et al., 2014)

Podemos reescrevê-la no sistema de coordenadas girante a fim de visualizar facilmente o comportamento de \vec{M}_t no tempo:

$$\left(\frac{d\vec{M}_t}{dt} \right)' = -\frac{1}{T_2} \vec{M}_t \quad (3.47)$$

cuja solução é

$$\vec{M}_t(t) = \vec{M}_t(0) e^{-t/T_2} \quad (3.48)$$

Ou seja, a magnetização transversal decai exponencialmente com uma taxa $R_2 = 1/T_2$.

A relaxação transversal é, na verdade, uma sobreposição das interações spin-spin e spin-rede. Isso implica que a taxa R_2 de relaxação transversal será maior que a taxa R_1 de relaxação longitudinal, a qual provém apenas da interação spin-rede:

$$R_2 = 1/T_2 > R_1 = 1/T_1 \quad (3.49)$$

Ou seja,

$$T_2 < T_1 \quad (3.50)$$

Podemos corroborar a desigualdade acima pelos valores de T_2 na tabela 2.

3.10.3 Equação de Bloch

Com todos os devidos efeitos considerados, temos a equação de Bloch em sua forma completa:

$$\frac{d\vec{M}}{dt} = \gamma \vec{M} \times \vec{B}_{ext} + \frac{1}{T_1} (M_0 - M_z) \hat{z} - \frac{1}{T_2} \vec{M} \quad (3.51)$$

As soluções para cada componente da equação 3.51 são

$$M_x(t) = e^{-t/T_2} (M_x(0) \cos(\omega_0 t) + M_y(0) \sin(\omega_0 t)) \quad (3.52)$$

$$M_y(t) = e^{-t/T_2} (M_y(0) \cos(\omega_0 t) - M_x(0) \sin(\omega_0 t)) \quad (3.53)$$

$$M_z(t) = M_z(0) e^{-t/T_1} + M_0 (1 - e^{-t/T_1}) \quad (3.54)$$

a partir das quais podemos tomar o limite $t \rightarrow \infty$ e confirmar que o sistema tende ao equilíbrio

$$M_x(\infty) = M_y(\infty) = 0 \quad \text{e} \quad M_z(\infty) = M_0 \quad (3.55)$$

Na figura 10 podemos visualizar a trajetória da ponta do vetor magnetização ao longo do tempo devido à sobreposição de ambas as relaxações.

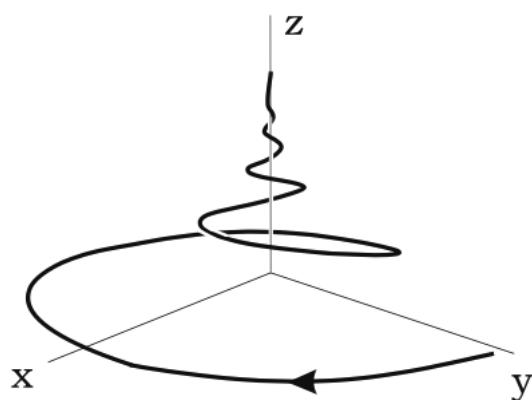


Figura 10 – Trajetória da ponta do vetor magnetização até o equilíbrio. O vetor precessa em torno do eixo z , gradualmente perdendo suas componentes transversais e, eventualmente, estabiliza-se com uma componente puramente longitudinal. Figura retirada de (BROWN et al., 2014)

3.11 Spin Eco

Além da interação spin-spin, inomogeneidades do campo externo \vec{B}_{ext} também podem causar a defasagem dos spins, o que acelera o decaimento com taxa R'_2 , de modo que $R_2^* = R_2 + R'_2$ e o tempo de decaimento torna-se T_2^* :

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T'_2} \quad (3.56)$$

Felizmente, tal efeito pode ser contornado utilizando o método *spin-eco*. Este consiste numa sequência de dois pulsos que provocam a refasagem dos spins. A figura 11 ilustra o comportamento da magnetização numa sequência spin eco. Primeiro, temos um pulso de RF $\pi/2$ na direção x' que move os spins alinhados à direção z (direção de \vec{B}_{ext}) para o eixo y' no plano transversal. Devido às inomogeneidades do campo \vec{B}_{ext} , ocorrerá uma defasagem, já que cada spin irá precessar com frequência distinta de acordo com a equação $\omega_0 = \gamma B_{ext}$.

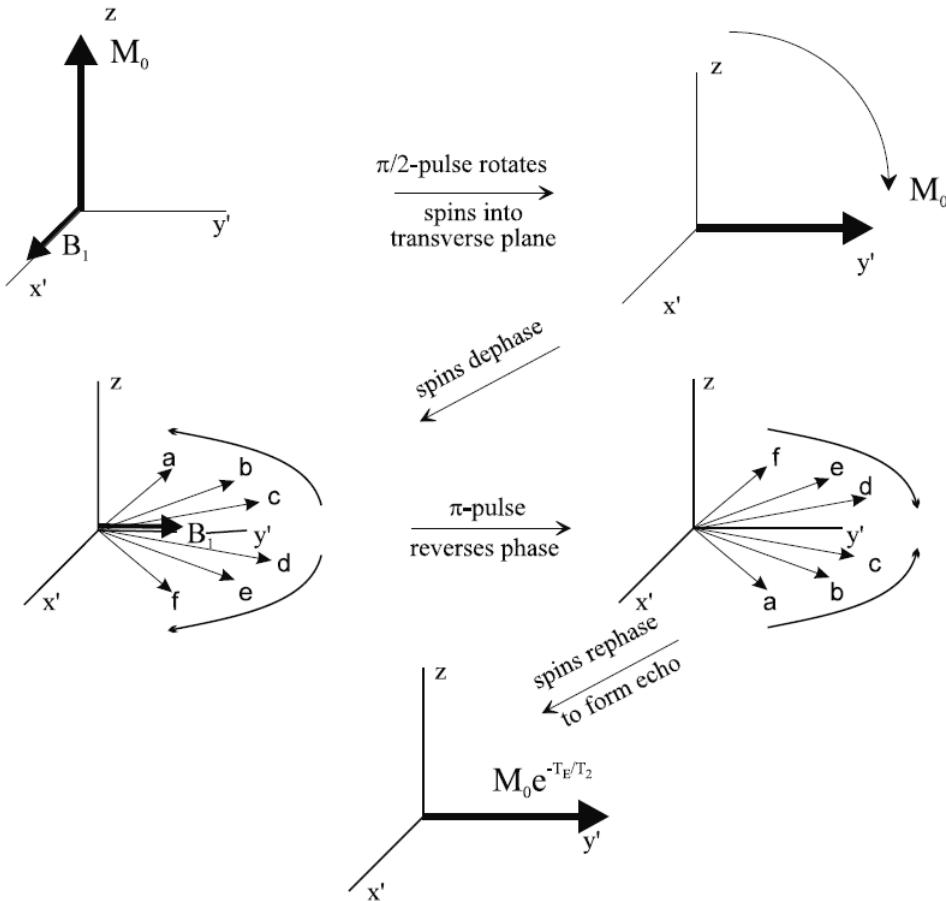


Figura 11 – Esquema do comportamento da magnetização em uma sequência spin-eco. Figura retirada de (BROWN et al., 2014)

O acúmulo de fase no sistema do laboratório para um spin na posição \vec{r} durante

um tempo t será

$$\phi(\vec{r}, t) = -\gamma(B_0 + \Delta B(\vec{r}))t \quad (3.57)$$

onde $\Delta B(\vec{r})$ é a variação local do campo magnético devido à inhomogeneidade.

No sistema girante, não teremos o termo de Larmor e o acúmulo de fase até um tempo $t = \tau$ será

$$\phi(\vec{r}, t) = -\gamma\Delta B(\vec{r})t \quad , \quad 0 < t < \tau \quad (3.58)$$

Em $t = \tau$, aplica-se um pulso π na direção y' , o qual rotaciona os spins por um ângulo de 180° ao redor do eixo y' . Assim, a fase imediatamente após o pulso ($t = \tau^+$) torna-se oposta a fase imediatamente antes do pulso ($t = \tau^-$):

$$\phi(\vec{r}, \tau^+) = -\phi(\vec{r}, \tau^-) = \gamma\Delta B(\vec{r})\tau \quad (3.59)$$

Portanto, para $t > \tau$ o acúmulo de fase total após ambos os pulsos será:

$$\begin{aligned} \phi(\vec{r}, t) &= \phi(\vec{r}, \tau^+) - \gamma\Delta B(\vec{r})(t - \tau) \\ &= \gamma\Delta B(\vec{r})(t - 2\tau) \\ &= \gamma\Delta B(\vec{r})(t - T_E) \quad , \quad t > \tau \end{aligned} \quad (3.60)$$

onde o *tempo de eco* T_E é definido como

$$T_E = 2\tau \quad (3.61)$$

Repare que para $t = T_E$, a fase volta a se anular para todo \vec{r} , indicando a ocorrência do eco, i.e. todos os spins possuem a mesma fase e, portanto, ocorre novamente um máximo do sinal.

Podemos entender intuitivamente o porquê deste comportamento com o auxílio da figura 12. Nela, introduzimos um diagrama de sequência, o qual ilustra a ocorrência dos pulsos e o sinal em função do tempo. Percebe-se que, logo após a aplicação do pulso $\pi/2$, temos o sinal em seu valor máximo. Este decai até que, em $t = \tau$, liga-se o pulso π . A fase de todos os spins será invertida (vide figura 11) e, como $\Delta B(\vec{r})$ permanece o mesmo em cada posição do espaço, a mesma variação de fase continuará a ocorrer, provocando agora uma refasagem dos spins. Assim, em $t = T_E$ todos voltam a fase $\phi = 0$, aumentando a magnetização.

Vemos pela figura 12 que, apesar da refasagem, a magnetização transversal em $t = T_E$ não será tão grande quanto a inicial, pois ainda assim ocorre a atenuação do sinal devido à T_2 (interação spin-spin).

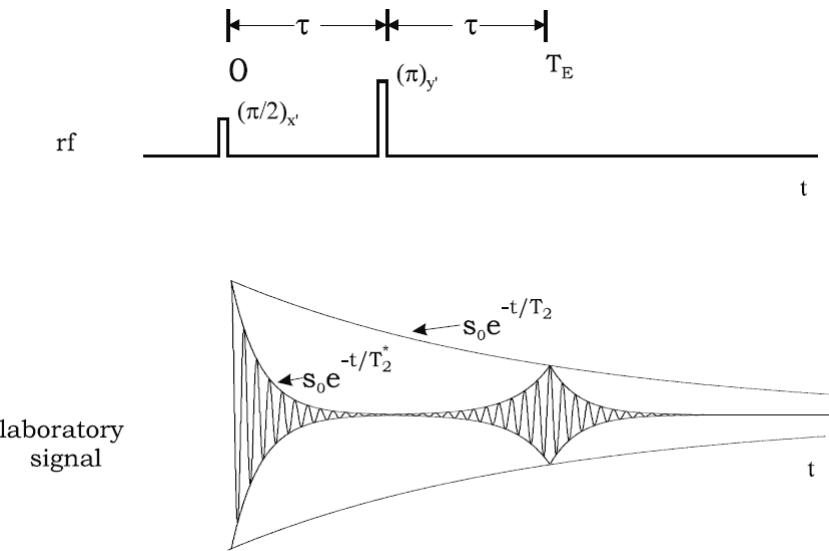


Figura 12 – Ilustração do sinal do laboratório no tempo em uma sequência spin eco. Figura retirada de ([BROWN et al., 2014](#))

Sequências spin eco são um dos tipos mais comuns de sequência utilizadas em RM. Um segundo tipo muito utilizado são as sequências *Gradiente Eco*, as quais apresentam algumas vantagens em relação ao método Spin Eco. Introduziremos tais sequências no capítulo seguinte após a introduzirmos outros conceitos de formação de imagem.

Formação de Imagem

Para construir uma imagem, inicialmente é preciso um mecanismo de contraste. Em Ressonância Magnética, esse mecanismo provém dos diferentes tempos de relaxação de cada tecido. Em segundo lugar, o sinal precisa conter informações sobre a localização espacial de cada um dos diferentes tecidos. Apresentamos nas seções seguintes como obter um sinal em função do tempo que contenha essas informações e mostramos também como extraí-las desse sinal. Em linhas gerais, utilizamos três *gradientes de campo magnético* para mapear o espaço tridimensional. Cada gradiente é responsável pelo mapeamento de uma coordenada espacial distinta (x , y ou z). Feito isso, o sinal medido no tempo (proveniente de todos tecidos) conterá informação sobre a localização e amplitude de cada tecido nesse ambiente mapeado. Para obter a imagem, precisamos apenas de um mecanismo que permita extrair essa informação do sinal como um todo.

Suponha que queiramos realizar imagens de um paciente posicionado como na figura 13, onde definimos três planos espaciais: sagital, transversal e coronal. O primeiro passo consiste na *seleção do corte*, i.e. diferenciar a posição dos tecidos no eixo z . Selecionado o corte, realiza-se o mapeamento da coordenada y no plano sagital, num processo denominado *codificação de fase*. Após esses dois passos, possuímos apenas uma codificação bidimensional do espaço, o que não é suficiente para obter imagens de um volume completo. A terceira e última etapa realiza a *codificação de frequência*, a qual completa o mapeamento tridimensional. Veremos cada um desses processos em detalhe nas seções seguintes.

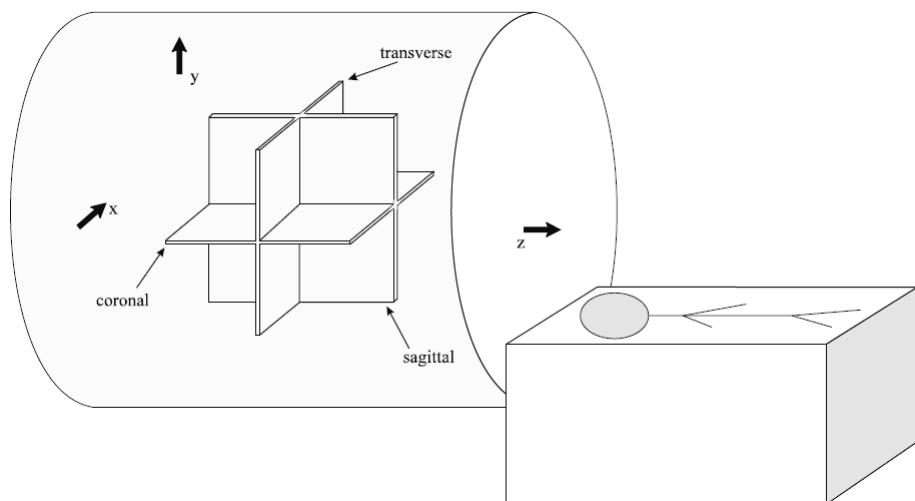


Figura 13 – Definição dos planos espaciais. Figura retirada de ([BROWN et al., 2014](#)).

3.12 Contraste

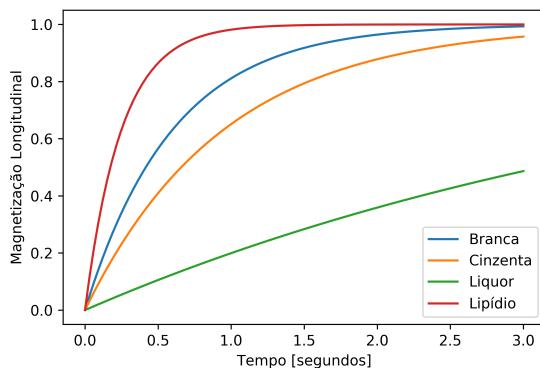
Um dos experimentos mais simples na Ressonância Magnética médica consiste em aplicar um pulso de RF $\pi/2$ a uma amostra sujeita a um campo magnético externo. Esse experimento recebe o nome de *decaimento de indução livre* (DIL), cujo sinal produzido é chamado *sinal de indução livre* (SIL). Tal sinal permite compreender o mecanismo de contraste na RM.

Dado uma amostra em um campo externo B_0 constante, imediatamente após aplicar um pulso $\pi/2$ ($t = 0^+$) a magnetização longitudinal torna-se nula, enquanto que a transversal se iguala a magnetização total M_0 . Dessa maneira, as equações 3.45 e 3.48 são reduzidas a:

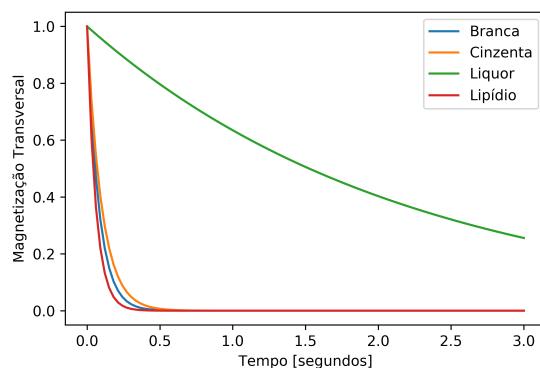
$$M_z(t) = M_0(1 - e^{-t/T_1}) \quad (3.62)$$

$$M_t(t) = M_0 e^{-t/T_2} \quad (3.63)$$

Na figura 14 temos essas funções ao longo do tempo utilizando os valores de T_1 e T_2 da tabela 2. Observe que a magnetização restante de cada tecido em um mesmo ponto no tempo é diferente. O sinal medido naquele instante será maior, por exemplo, para o lipídio do que para o líquor, caracterizando um contraste. Vale ressaltar que essa diferença na magnetização diminui conforme $t \rightarrow \infty$, pois a magnetização tende a voltar ao equilíbrio. Portanto, o tempo da aquisição torna-se um fato crítico na formação de imagem, caso contrário, o contraste poderá se tornar irrelevante.



(a) Magnetização Longitudinal



(b) Magnetização Transversal

Figura 14 – Magnetização de diferentes tecidos ao longo de 3 segundos. As curvas confirmam que relaxação transversal é maior que a longitudinal devido a $T_1 > T_2$. Repare também a ordem de grandeza do tempo de decaimento. Para tecidos sólidos, este é menor que 1 segundo. Já para o líquido (líquor), o tempo é da ordem de segundos. Por simplicidade, consideramos $M_0 = 1$.

3.13 Seleção de Cortes

Como visto pela equação $f = \gamma B_0$, a frequência de precessão é diretamente proporcional ao campo B_0 aplicado. Isso implica que, se introduzirmos um campo magnético $B(z, t)$ variante no espaço, os spins em diferentes posições irão precessar com frequências distintas. Portanto, ao aplicar um gradiente G_z constante na direção z , isto é,

$$\frac{\partial B_z}{\partial z} = G = \text{const.} \quad (3.64)$$

teremos que a frequência de precessão na posição z será

$$\begin{aligned} f(z) &= \gamma B_0 + \gamma G_z z \\ &= f_0 + \gamma G_z z \end{aligned} \quad (3.65)$$

Com isso, podemos excitar regiões distintas da amostra de forma seletiva simplesmente ajustando a frequência do pulso de RF. Por exemplo, se quisermos excitar os spins na posição z_0 , basta ajustarmos a frequência do pulso de RF para $f(z_0)$ pois, assim, apenas esses spins satisfarão a condição de ressonância.

Como o pulso possui uma largura de banda Δf , toda uma região Δz , chamada *espessura de corte*, ao redor de z_0 será excitada. Pela equação 3.65, temos que Δf é dado por

$$\begin{aligned} \Delta f &= (\gamma G_z z_0 + \gamma G_z \Delta z / 2) - (\gamma G_z z_0 - \gamma G_z \Delta z / 2) \\ &= \gamma G_z \Delta z \end{aligned} \quad (3.66)$$

A largura do Pulso pode ser definida, por exemplo, como a largura a meia altura do perfil obtido a partir da Transformada de Fourier de um pulso $B_1(t) \propto \text{sinc}(t)$, como ilustrado na figura 15.

Para adquirir cortes de outras regiões da amostra, basta alterarmos a frequência central $f(z_0)$ do pulso de RF, como mostra a figura 16. Ao variar o gradiente G_z , podemos obter diferentes tamanhos de corte Δz para uma mesma largura de banda Δf .

3.14 Codificação de Frequência

Feita a seleção do corte, ainda é necessário mapear o espaço nas dimensões x e y . Por simplicidade, iniciamos pela codificação de frequência, usualmente atribuída a dimensão x .

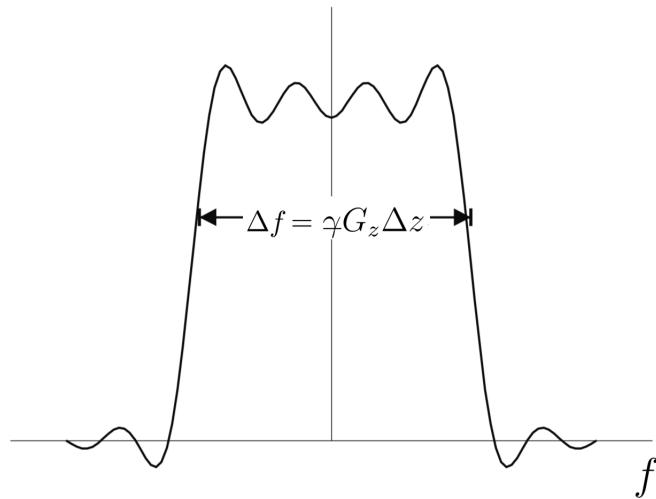


Figura 15 – Largura do pulso de excitação para um envelope temporal $B_1 \propto \text{sinc}(t)$ truncado no tempo. Figura adaptada de (BROWN et al., 2014).

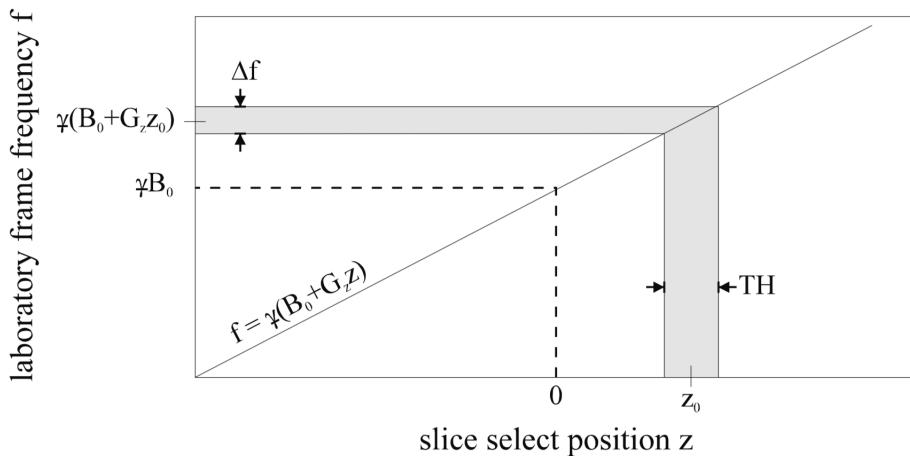


Figura 16 – Gráfico da frequência no sistema do laboratório em função da posição do corte. A região cinza indica que um corte de espessura TH centrado em z_0 será excitado para uma largura de banda Δf . Na posição $z = 0$, o gradiente é nulo e, portanto, a frequência de precessão é apenas γB_0 . Figura retirada de (BROWN et al., 2014).

Nesse procedimento, analogamente a seleção de corte, atribui-se uma frequência de precessão $\omega(x, t)$ distinta para os spins em diferentes posições no espaço por meio da aplicação de um gradiente constante G_x . O sinal total no tempo conterá informação sobre a localização dos diferentes pontos na direção x . Com o auxílio da Transformada de Fourier, podemos obter o sinal no espaço-k (ou espaço de frequências), onde cada frequência está diretamente relacionada a uma posição única no espaço.

Com a aplicação de um segundo campo B linear no espaço sobre o campo externo B_0 , temos

$$B_x(x, t) = B_0 + B(x, t) = B_0 + zG_x(t) \quad (3.67)$$

onde $G_x(t)$ é um gradiente de campo magnético constante na direção x :

$$\frac{\partial B_x}{\partial x} = G_x = \text{const.} \quad (3.68)$$

Com este novo campo total, a equação 3.19 é modificada pela nova componente de frequência $\omega_G(z, t) = \gamma x G(t)$:

$$\omega(x, t) = \omega_0 + \omega_G(x, t) \quad (3.69)$$

Os gradientes utilizados são da ordem de milli Tesla por metro e, portanto, pouco alteram o valor da frequência de precessão. Para um campo $B_0 = 1.5 \text{ T}$ e $G = 10 \text{ mT m}^{-1}$, obtemos $\omega_0 = 63.87 \text{ MHz}$ e $\omega_G = 0.4258 \text{ MHz}$, resultando numa variação de menos de 1% no valor da frequência após a aplicação do gradiente. No entanto, para que a codificação seja eficiente, as variações no campo na presença de G_x devem ser maiores que as variações devido às inhomogeneidades I do campo, caso contrário seria impossível distinguir o efeito do gradiente e das inhomogeneidades. Um gradiente $G_x = 10 \text{ mT m}^{-1}$ satisfaz essa condição para valores típicos de I. Como exemplo, para $I = 50 \text{ ppm}$ (partes por milhão) sobre uma região de 50 DSV (diâmetro de região esférica, do inglês, *diameter spherical volume*), a variação do campo ao longo de 1 metro devido à inhomogeneidades seria $\delta B = 0.03 \text{ mT}$, isto é, 3 ordens de grandeza a menos que a variação do campo devido a G_x nesta mesma região.

É importante destacar que a aquisição do sinal deve ocorrer enquanto o gradiente G_x encontra-se ligado, pois apenas assim o sinal total medido será composto por sinais de diferentes frequências. Caso contrário, a aplicação do gradiente por um tempo t anterior à aquisição corresponderá apenas a um acúmulo de fase distinto para os spins nas posições x :

$$\phi_G(x, t) = - \int_0^t \omega_G(t') dt' = -\gamma x \int_0^t G_x(t') dt' \quad (3.70)$$

Para um gradiente constante no tempo, a equação 3.70 é simplificada para

$$\phi_G(x) = -\gamma x G_x t \quad (3.71)$$

3.14.1 Equação de imagem unidimensional e a Transformada de Fourier

O mecanismo utilizado para extrair a informação espacial contida no sinal no tempo $s(t)$ é a Transformada de Fourier. Por simplicidade, consideremos primeiro o caso unidimensional.

O sinal na presença de um gradiente constante está diretamente relacionado com a densidade de spins no espaço $\rho(z)$ pela equação (BROWN et al., 2014)

$$s(t) = \int \rho(z) e^{i\phi_G(z,t)} dz \quad (3.72)$$

onde a exponencial imaginária diz respeito ao acúmulo de fase em cada região. Substituindo a equação 3.70, obtemos

$$s(k) = \int \rho(z) e^{-i2\pi kz} dz \quad (3.73)$$

onde $k = k(t)$ é chamada de *frequência espacial*:

$$k(t) = \gamma \int_0^t G(t') dt' \quad (3.74)$$

A equação 3.73 mostra que o sinal $s(k)$ nada mais é que a Transformada de Fourier da densidade de spins $\rho(z)$ da amostra, a qual pode ser obtida pela Transformada inversa

$$\rho(z) = \int s(k) e^{+i2\pi kz} dk \quad (3.75)$$

Vale ressaltar que, para cada ponto z do espaço, a integral é feita sobre todo o espaço- k . Ou seja, cada um dos pontos do espaço- k contém apenas uma parte da informação sobre um único ponto do espaço real e é necessário varrer todo o espaço- k a fim de obter a informação completa sobre este único ponto. Vemos pela equação 3.73 que o inverso também ocorre: um ponto do espaço- k só é obtido por meio de todos os pontos do espaço real.

3.14.2 O espaço- k

A Transformada de Fourier indica uma relação entre o sinal e a densidade de spins. Isso implica que podemos obter a densidade e construir uma imagem com ela, desde que tenhamos pontos suficientes no espaço- k .

No caso de um gradiente constante, a equação 3.74 é simplificada para

$$k = \gamma G t \quad (3.76)$$

Portanto, obter um dado com muitos pontos no espaço- k na presença de um gradiente G constante, é o mesmo que obter um sinal durante um tempo t grande. Idealmente, teríamos de realizar a aquisição por um tempo $t = \infty$ a fim de obter um espaço- k completo e reconstruir a imagem perfeitamente. No entanto, precisamos realizar aquisições durante um intervalo finito de tempo, não apenas por motivos de viabilidade econômica do experimento, mas também porque a própria precisão dos equipamentos torna-se um fator limitante,

já que o sinal decai exponencialmente no tempo devido à relaxação e torna-se pequeno demais para ser medido.

Pela equação acima, vemos também que uma segunda forma de obter valores distintos de k é alterar o gradiente G . Esse método consiste na sequência gradiente eco que, de fato, possui grande relevância na aquisição de imagens, como veremos na seção 3.16.

3.14.3 Exemplos analítico e com simulação

Introduzimos a seguir um exemplo analítico e outro computacional de como a informação espacial pode ser recuperada através do sinal no tempo.

Dois spins ideais

Suponha que tivéssemos dois spins na presença de um campo magnético externo e de um gradiente de campo G constante.

Para spins posicionados em $z = z_0$ e $z = -z_0$ e para $G > 0$, a equação 3.71 diz que cada spin irá, respectivamente, acumular fases $\phi(-z_0, t) = \gamma G z_0 t$ e $\phi(z_0, t) = -\gamma G z_0 t$ após um tempo t . A integral sobre a densidade de spins da equação 3.72 reduz-se a contribuição dos dois spins

$$\begin{aligned} s(t) &= s_0 e^{-i\gamma G z_0 t} + s_0 e^{i\gamma G z_0 t} \\ &= 2s_0 \cos(\gamma G t z_0) \end{aligned} \quad (3.77)$$

que pode ser rescrita para a variável k :

$$s(k) = 2s_0 \cos(2\pi k z_0) \quad 0 < k = \gamma G t \quad (3.78)$$

Pela Transformada de Fourier inversa, verifica-se que esse sinal está diretamente relacionado com a distribuição dos spins. De fato, por 3.75

$$\begin{aligned} \rho(z) &= \int_{-\infty}^{\infty} 2s_0 \cos(2\pi k z_0) e^{i2\pi k z} dk \\ &= s_0 \int_{-\infty}^{\infty} (e^{i2\pi k(z+z_0)} + e^{i2\pi k(z-z_0)}) dk \\ &= s_0 [\delta(z+z_0) + \delta(z-z_0)] \end{aligned} \quad (3.79)$$

Como esperado, o resultado é a soma de dois deltas de Dirac que indicam dois spins pontuais nas posições z_0 e $-z_0$.

Simulação de dois SIL e da TF

O exemplo acima considerou um caso ideal de dois spins pontuais, no qual a Transformada de Fourier foi feita de maneira analítica. Tratamos agora de um sinal simulado que decai no tempo sobre o qual é aplicado uma Transformada de Fourier discreta. A fim de compreender como o sinal simulado se assemelha ao sinal que é detectado na prática, devemos detalhar como se dá a aquisição deste sinal.

Para detectar um sinal devido à precessão da magnetização transversal M_t , posiciona-se uma bobina perpendicular ao plano transversal. Dessa forma, esperamos obter um sinal oscilante como na figura 17a. No entanto, essa configuração não permite dizer se a frequência angular do vetor magnetização é positiva ou negativa, isto é, não sabemos se rotação ocorre em sentido horário ou anti-horário.

Para resolver esse problema, realizamos a chamada *detecção em quadratura*, na qual se utiliza uma segunda bobina de detecção, cujo eixo é perpendicular ao eixo da primeira (por exemplo, uma com eixo alinhado à direção x e a outra alinhada à direção y). Toma-se então o sinal de cada bobina como os termos real e imaginário de uma exponencial imaginária:

$$\begin{aligned} s(t) &= s_0 e^{-t/T} \left(\underbrace{\cos(i2\pi ft)}_{\text{Termo da 1ª bobina}} \pm i \underbrace{\sin(i2\pi ft)}_{\text{Termo da 2ª bobina}} \right) \\ &= s_0 e^{-t/T} e^{i2\pi ft} \end{aligned} \quad (3.80)$$

Na figura 17a temos dois sinais de indução livre de frequências distintas ($f = 100$ Hz e $f = 300$ Hz). A soma dos dois sinais pode ser vista na figura 17b.

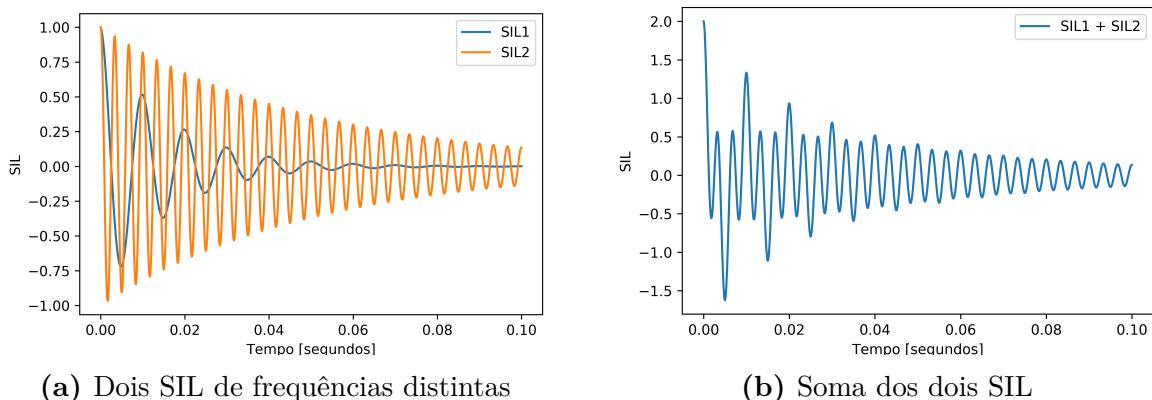


Figura 17

Ao realizar a Transformada inversa sobre a soma dos sinais, obtemos uma função no espaço de frequências como na figura 18. Nesse plot, podemos visualizar dois picos nas posições de 100 Hz e 300 Hz, como esperado.

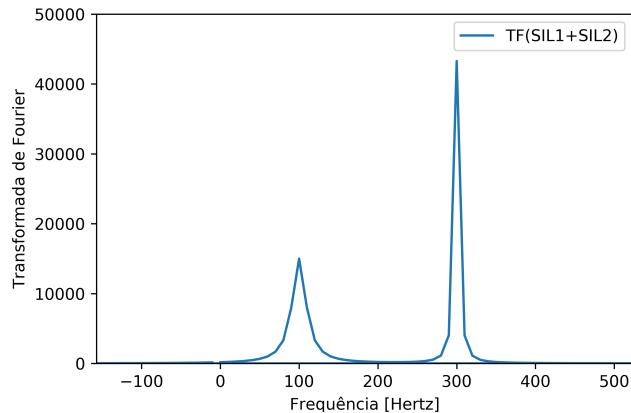


Figura 18 – Transformada de Fourier do sinal da figura 17b.

Interpretação do gráfico da Transformada de Fourier

Apresentamos aqui quais informações da amostra podem ser extraídas a partir dos picos obtidos no espaço de frequências. A solução analítica é apresentada no apêndice A, a qual nos diz que tais picos são funções Lorentzianas. Tal função pode ser reescrita em sua forma mais geral como:

$$F = 2M_0 \frac{T}{1 + (\omega - \omega_0)^2 T^2} \quad (3.81)$$

onde T é o tempo de relaxação, M_0 a amplitude e ω_0 a frequência na qual a função está centrada. O máximo (MAX) de F ocorre para $\omega = \omega_0$:

$$\text{MAX} = 2M_0T \quad (3.82)$$

Já a largura a meia altura (FWHM) é dada por

$$\text{FWHM} = \frac{2}{T} \quad (3.83)$$

a qual fornece o tempo de relaxação T .

Podemos calcular a amplitude M_0 de duas formas. A primeira consiste em inserir o valor de T obtido com 3.83 na equação 3.82. A segunda, consiste em integrar a curva em todo o domínio, obtendo

$$I = 2M_0\pi \quad (3.84)$$

de onde M_0 pode ser facilmente isolado.

3.15 Codificação de fase

Para mapear a dimensão y , utiliza-se a técnica chamada codificação de fase. Tal método também utiliza um gradiente G_y (constante em y) a fim de gerar um acúmulo de fase distinto para tecidos em diferentes coordenadas y . Esse acúmulo de fase é análogo ao apresentado pela equação 3.71:

$$\phi_{G_y}(y, t) = - \int_0^t \omega_G(y, t') dt' = -\gamma y \int_0^t G_y(t') dt' \quad (3.85)$$

onde, para G_y constante no tempo, temos

$$\phi_{G_y}(y) = -\gamma y G_y(y) t \quad (3.86)$$

Portanto, temos que para um gradiente G_y constante aplicado durante um intervalo de tempo t , existe uma relação direta entre a fase $\phi_{G_y}(y)$ acumulada e a posição y da amostra.

Ao final das codificações de fase e de frequência, cada posição da amostra possuirá uma combinação única de frequência e de fase, a qual pode ser usada para o mapeamento. No entanto, extrair essa informação espacial não é algo intuitivo como no caso da codificação de frequência, em que diferentes frequências são claramente vistas como picos distintos no espaço de frequências (figura 18). Isso ocorre pois as fases dos spins numa posição x serão diferentes, mas as frequências serão as mesmas. Nesse caso é preciso utilizar a Transformada de Fourier bidimensional. Para entender de forma mais simples a codificação de frequência, podemos analisar o espaço- k . Para isso, introduzimos primeiramente a equação de imagem multidimensional.

3.15.1 Equação de imagem multidimensional

A versão tridimensional da equação do sinal 3.73 é

$$s(\vec{k}) = \int \rho(\vec{r}') e^{-i2\pi\vec{k}\cdot\vec{r}'} d^3 r' \quad (3.87)$$

onde $\vec{k} = (k_x, k_y, k_z)$ e $\vec{r} = (x, y, z)$. Análogo ao caso unidimensional, a densidade de spin é obtida pela Transformada inversa

$$\rho(\vec{r}) = \int s(\vec{k}) e^{i2\pi\vec{k}\cdot\vec{r}} d^3 k \quad (3.88)$$

com cada componente do vetor \vec{k} obedecendo à equação

$$k_i(t) = \gamma \int_0^t G_i(t') dt' \quad (3.89)$$

com $i = x, y$ ou z e a integração sendo feita do momento em que o gradiente é ligado até um tempo t . Como esse gradientes em geral são constantes, normalmente temos simplesmente $k_i(t) = \gamma G_i t$.

Uma vez realizada a seleção do corte de espessura Δz centrado em z_0 , temos

$$s(k_x, k_y) = \int \int \left[\int_{z_0 - \frac{\Delta z}{2}}^{z_0 + \frac{\Delta z}{2}} \rho(x, y, z) dz \right] e^{-i2\pi(k_x x + k_y y)} dx dy \quad (3.90)$$

onde eliminamos a dependência em z , obtendo:

$$s(k_x, k_y) = \int \int \rho(x, y, z_0) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (3.91)$$

cuja Transformada inversa fornece

$$\rho(x, y) = \int \int s(k_x, k_y, k_{z_0}) e^{-i2\pi(k_x x + k_y y)} dk_x dk_y \quad (3.92)$$

A equação 3.92 nos mostra que podemos obter a densidade de spins no espaço a partir do sinal, desde que varramos todo o espaço- k (variáveis k_x e k_y).

Um argumento heurístico para não perdermos de vista como a fase está relacionada com a posição e o com a frequência espacial está no argumento da exponencial imaginária em 3.91, assim como na equação 3.72:

$$\begin{aligned} -2\pi(k_x x + k_y y) &= -2\pi(\gamma G_x t x + \gamma G_y t y) \\ &= -\gamma G_y t y - \gamma G_x t x \\ &= \phi_{G_y} + \phi_{G_x} \end{aligned} \quad (3.93)$$

Ou seja, a fase está diretamente relacionada com a posição e com a frequência espacial em cada dimensão.

3.15.2 O espaço- k bidimensional

Queremos varrer pontos do espaço- k a fim de reconstruir a densidade de spins pela equação 3.92. Como mencionado anteriormente, podemos obter os valores de k por meio da variação ora de G , ora de t :

$$k_i = \gamma G_i t \quad (3.94)$$

Suponha que tenhamos um espaço- k como na figura 19. A estratégia consiste em realizar diversas aquisições, onde cada uma irá varrer uma linha horizontal (k_x, k_{y_m}) .

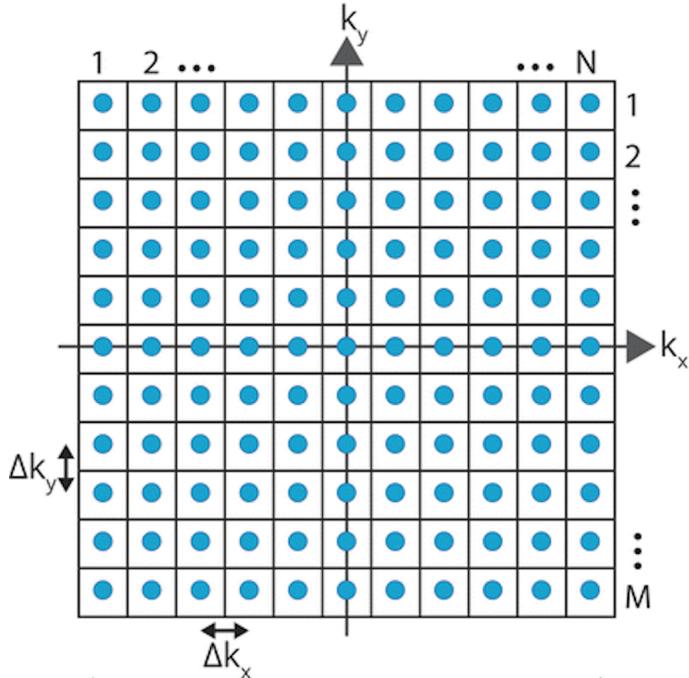


Figura 19 – Representação do espaço- k com M linhas e N colunas. Os pontos azuis correspondem a cada par ordenado (k_x, k_y) . Figura adaptada de ([LOECHER; WIEBEN, 2015](#)).

O valor de k_y numa linha m ($1 \leq m \leq M$) será dado por

$$k_y = k_{y_{max}} - (m - 1)\Delta k_y \quad (3.95)$$

onde Δk_y é obtido por meio da variação do gradiente quando este é aplicado por um tempo fixo τ :

$$\Delta k_y = \gamma \Delta G_y \tau \quad (3.96)$$

com $G_{y_m} = G_{y_{max}} - (m - 1)\Delta G_y$. Em resumo, para se obter cada linha, é preciso aplicar um gradiente G_{y_m} distinto, o que é feito em passos iguais ΔG_y . Refere-se normalmente a esse processo como *passos de codificação de fase*.

Dentro de uma determinada linha do espaço- k , é preciso varrê-la horizontalmente na dimensão k_x . Essa etapa consiste na codificação de frequência, onde a variação Δk_x é obtida aplicando um gradiente G_x constante num intervalo de tempo Δt :

$$\Delta k_x = \gamma G_x \Delta t \quad (3.97)$$

Como veremos adiante, a aquisição do sinal é feita durante cada etapa de codificação de frequência. Portanto, para um espaço- k com M linhas, serão necessárias M aquisições a fim de se obter o espaço- k completo e, assim, reconstruir a densidade de spins.

3.16 Gradiente Eco

Vimos na seção 3.11 que o método Spin Eco pode ser utilizado para compensar a defasagem que ocorre devido a diferenças locais de campo magnético. Isso resulta na geração de um eco do sinal, o qual é utilizado na aquisição. Veremos agora um segundo tipo de sequência, chamadas sequências *Gradiente Eco*, também amplamente utilizadas em aquisições de RM.

Vimos pela equação 3.75 que, para obter a distribuição de densidade dos spin da amostra, precisamos varrer muitos pontos do espaço-k. A equação 3.76 também nos mostra que os valores de k dependem de apenas duas variáveis: o gradiente aplicado G e o tempo t de aquisição do sinal.¹

A fim de varrer um maior número de pontos no espaço-k, é comum utilizar as sequências *Gradiente Echo*. Neste método, aplicam-se dois gradientes de codificação de frequência (também denominados *gradientes de leitura*) com sinais opostos e, assim, além de se gerar um eco, adquirem-se tanto valores positivos quanto negativos de k .

Com o auxílio da figura 20, podemos entender o que ocorre com o sinal perante a aplicação de ambos os gradientes. Suponha que temos um cilindro alinhado à direção z com uma distribuição de spins arbitrária. Ao aplicar apenas o campo externo, teremos a magnetização do material alinhada ao eixo z (figura 20a). Com a aplicação da RF (ainda sem gradiente), gera-se o sinal de indução livre como na figura 20b. Se aplicarmos um gradiente constante $-G_x$, o sinal passa a apresentar batimentos ao longo do tempo devido à defasagem dos spins (figura 20c). No entanto, se aplicarmos um segundo gradiente de sinal oposto após o primeiro, a defasagem dos spins será revertida, pois os mesmos passarão a acumular fase de sinal oposto de acordo com a equação 3.71. Portanto, os spins ao longo da amostra voltarão a interferir construtivamente e haverá um crescimento do sinal até um máximo no tempo $t' = 0$ (figura 20d).

Façamos uma análise mais rigorosa sobre o acúmulo de fase perante a aplicação dos gradientes. Para um gradiente $-G_x$ ($G > 0$) no intervalo $t_1 < t < t_2$, um spin na posição z no tempo t terá fase

$$\phi_G = \gamma G z (t - t_1) \quad (3.98)$$

Após a mudança do gradiente para um valor G (ligado no intervalo $t_3 < t < t_4$), a fase acumulada será

$$\phi_G = (z, t) = \gamma G z (t_2 - t_1) - \gamma G z (t - t_3) \quad (3.99)$$

¹ Não consideraremos a razão giromagnética γ como variável pois lidamos estritamente com os prótons no corpo humano.

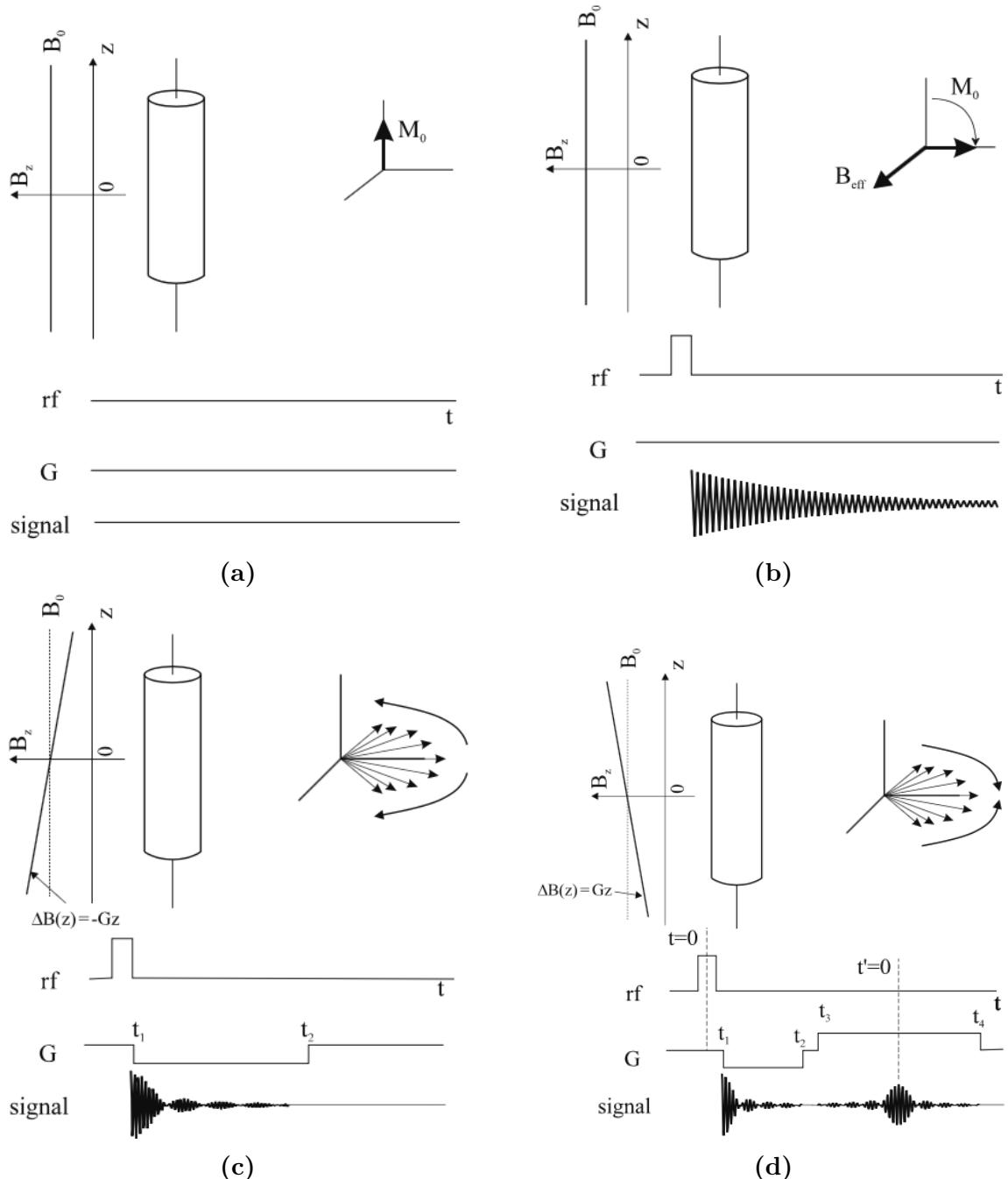


Figura 20 – Figura retirada de (BROWN et al., 2014).

A equação 3.99 indica que a fase retorna a zero para todo z quando

$$t = t_3 + t_2 - t_1 = T_E \quad (3.100)$$

onde T_E é chamada *tempo de eco*.

Uma das maiores vantagens de sequências GE é que eles podem ser mais rápidas. Isso se deve a dois fatores. Primeiro, não há necessidade de que o pulso de RF seja 90° . Ângulos de flip pequenos ($< 10^\circ$) são comumente utilizados. Assim, como a magnetização não é completamente deslocada para o plano transversal, não é necessário um grande

tempo para a relaxação por T_1 . Segundo, o fato das sequências GE não precisarem de um segundo pulso de RF 180° implica que o tempo de eco T_E pode ser muito mais curto. Consequentemente, T_R também é menor e mais aquisições são realizadas num mesmo período de tempo, se comparadas as sequências SE.

Veremos que, principalmente devido à sua agilidade, as sequências gradiente eco são utilizadas na aquisição das imagens deste trabalho.

3.16.1 Tempo de Repetição T_R

Como visto na seção 3.15, a codificação de fase requer que realizemos diversas excitações e aquisições para reconstruir a densidade de spins. Além disso, muitas vezes precisamos melhor a razão sinal-ruído (SNR, do inglês, *signal-to-noise ratio*) de uma imagem e, para tal, precisamos obter vários sinais de um mesmo corte.

A aquisição de novos sinais ocorre pela aplicação de pulsos de RF em períodos de tempo T_R , chamado *tempo de repetição*. A figura 21 mostra um esquema da magnetização transversal e longitudinal num SIL ao longo do tempo, quando diversos pulsos são aplicados.

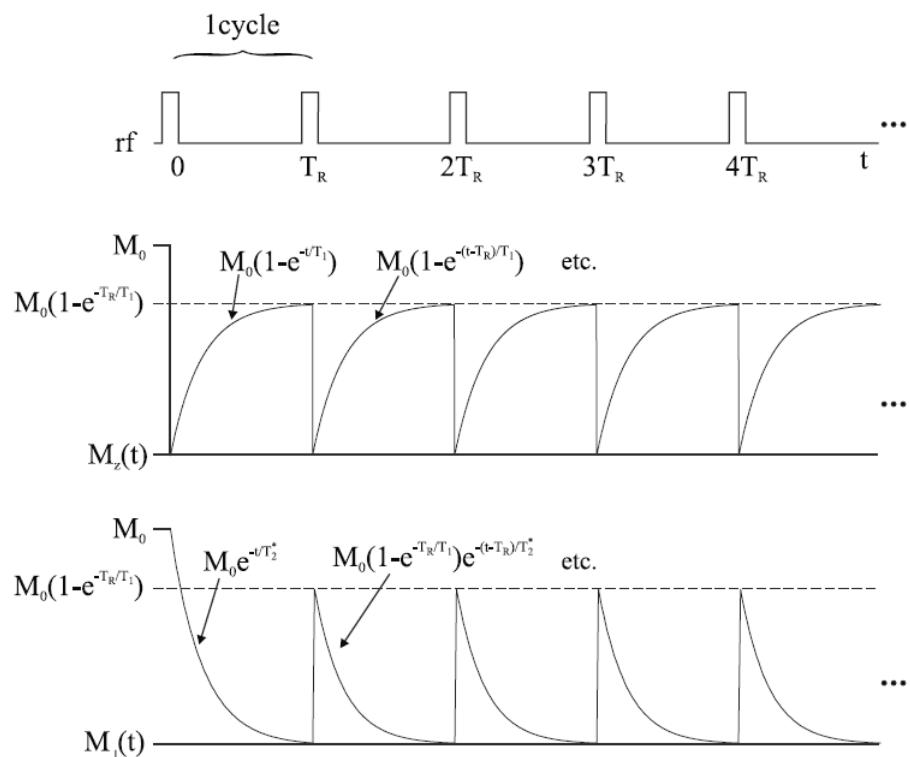


Figura 21 – Em intervalos de tempo T_R , aplica-se um pulso de RF o qual transfere toda magnetização longitudinal para o plano transversal. Figura retirada de (BROWN et al., 2014).

Introduzimos a seguir uma comparação entre sequências Spin Eco (SE) e Gradiente Eco (GE), ilustradas nas figuras 23a e 23b, respectivamente. Os diagramas de sequência mostram um único tempo de repetição.

Primeiro, comparamos a diferença no mecanismo de geração de eco entre as duas sequências com o auxílio da figura 22 para o caso simplificado de dois spins A e B. Inicialmente, ocorre uma defasagem entre os spins (figura 22a). No caso de uma sequência SE, isso ocorre devido às diferenças locais no campo externo. Já no caso GE, a defasagem obtida ocorre durante a porção negativa do gradiente de leitura.

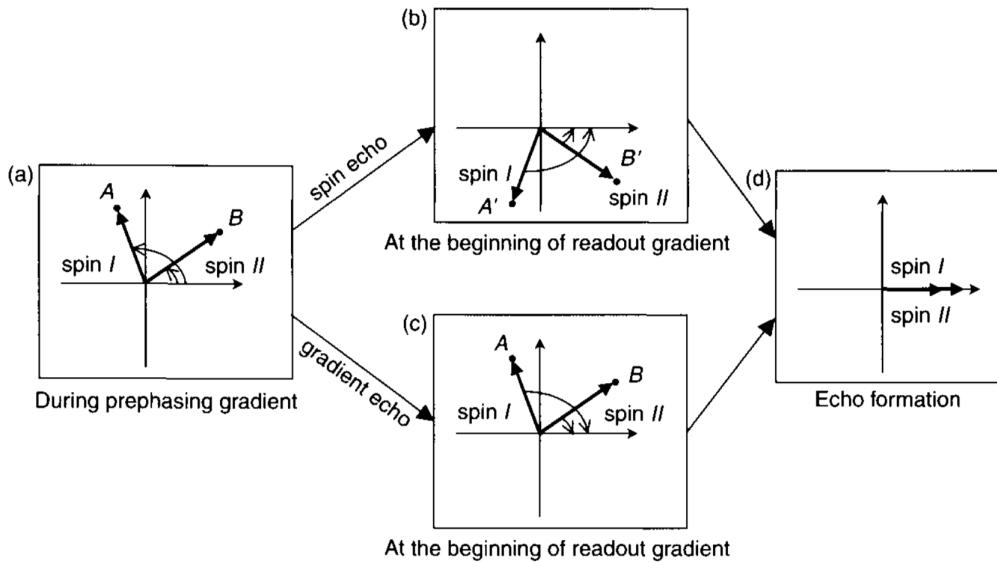


Figura 22 – Figura retirada de (BERNSTEIN; KING; ZHOU, 2004).

Após aplicação do pulso de RF 180° em $t = T_E/2$ no caso SE, a fase dos spins é invertida e eles continuam a acumular fase normalmente como até então (figura 22b), de modo que em $t = T_E$, a fase entre os spins volta a ser nula e obtemos um eco (figura 22d). Já no caso GE, a inversão do gradiente para um gradiente positivo irá apenas inverter a direção de acumulação de fase dos spins, sendo a taxa dessa acumulação mantida constante. No momento da inversão do gradiente, o spin A (figura 22c) possui maior fase acumulada em relação ao eixo horizontal do que o spin B. No entanto, como o spin A acumula fase mais rápido que o spin B, essa diferença será compensada, de modo que em $t = T_E$ teremos diferença de fase nula e, consequentemente, a formação do eco (figura 22d).

3.16.2 Comparação de sequências Spin Eco e Gradiente Eco

Por fim, descrevemos os diagramas de sequências SE e GRE, a fim de sintetizar os conceitos de formação de imagem apresentados.

Iniciamos pelo diagrama Spin Eco na figura 23a. Primeiramente, em $t = 0$ temo o pulso de RF $\pi/2$ (da forma $\text{sinc}(t)$) ligado junto de um gradiente G_z de seleção de corte (linha Slice). Imediatamente após o pulso, temos um sinal (linha Echo), já que a magnetização foi toda jogada para o plano transversal. Para $0 \leq t \leq T_E/2$, liga-se o gradiente G_y de codificação de fase (linha phase) por um tempo fixo. Repare que as linhas

pontilhadas indicam os demais valores de G_y que ocorrem nos demais tempos de repetição. Após isso temos novamente o gradiente de seleção G_z ligado, para que ocorra a excitação da amostra pelo pulso π de inversão de fase. Em seguida, inicia-se a formação do eco que é medido junto do gradiente G_x de codificação de frequência (linha Readout), sendo este gradiente centrado no tempo T_E onde ocorre o máximo do eco.

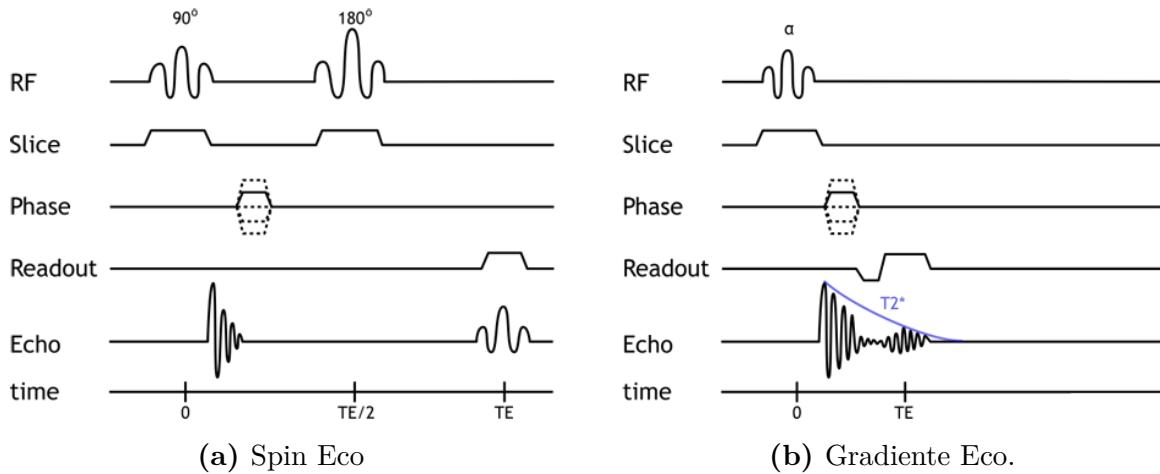


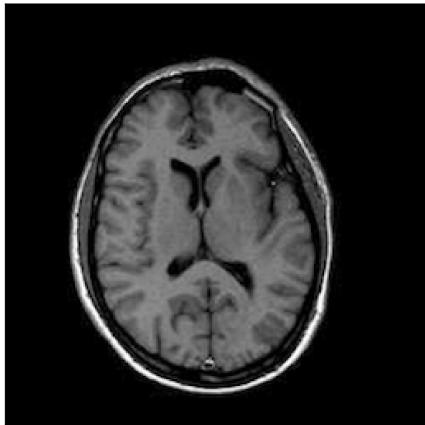
Figura 23 – Diagramas de sequência ao longo de um T_R . Figuras retiradas de ([HAMMER, 2013](#)).

Já para a sequência Gradiente Eco na figura 23b, temos inicialmente um pulso de RF ligado juntamente do gradiente de seleção de corte. O pulso, neste caso, não precisa ser $\pi/2$, uma vez que deseja-se utilizar sequências mais rápidas. Em seguida, o gradiente G_y de codificação de fase é ligado por um certo tempo, assim como no caso SE. As diferenças entre sequências SE e GE surgem a partir de então. Após o gradiente G_y , temos o gradiente de leitura G_x sendo ligado, inicialmente com valor negativo e, posteriormente, com valor positivo para gerar a refasagem e ocasionar no Eco em $t = T_E$.

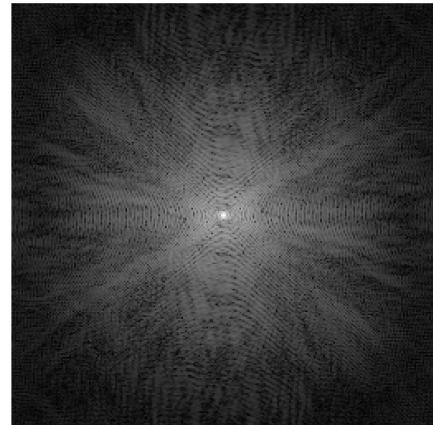
Um fato ainda não mencionado é que a defasagem devido a inhomogeneidades do campo magnético também ocorre para sequências GE e, como não há um pulso de RF 180° para corrigir tal efeito, imagens gradiente eco são ponderadas por T_2^* (como indica a linha Echo na figura 23b, e não por T_2 como as spin eco). Como $T_2^* < T_2$, a perda do sinal é mais rápida para sequências GE e, portanto, faz-se necessário o uso de T_E mais curtos.

3.17 Filtros no espaço- k : compreendendo frequências espaciais

Como visto na seção 3.15.2, podemos adquirir tantos valores positivos quanto negativos do espaço- k por meio da mudança do gradiente aplicado. A figura 24 mostra uma imagem de RM (24a) e o respectivo espaço- k dessa imagem (24a). Neste espaço, o ponto central mais claro representa o par ordenado $(k_x, k_y) = (0, 0)$, ou seja, quanto mais afastado do centro, maior o valor absoluto da frequência espacial.



(a) Figura retirada de (BROWN et al., 2014).

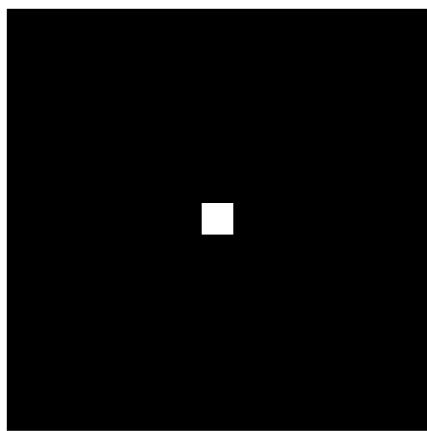


(b)

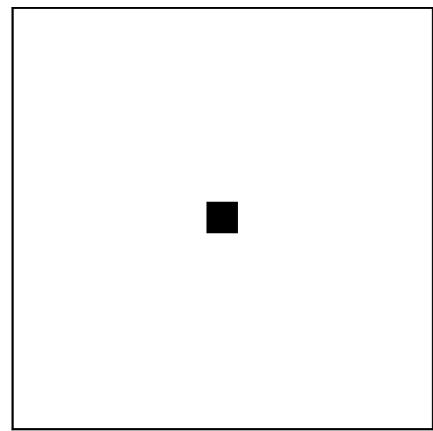
Figura 24 – Imagem de Ressonância do cérebro e do seu respectivo espaço- k .

O conceito de frequências espaciais é pouco recorrente e pode causar estranheza num primeiro contato. Os filtros de imagem aplicados ao espaço- k fornecem uma noção intuitiva sobre o conceito.

Na figura 25 apresentamos dois exemplos de filtros de frequências espaciais, onde os pixels pretos representam valor 0 e os brancos valor 1. Ao multiplicarmos os filtros das figuras 25a e 25b pelo espaço- k em 24b, obtemos os espaços- k em 26a e 27a, respectivamente.



(a) Filtro passa-baixa

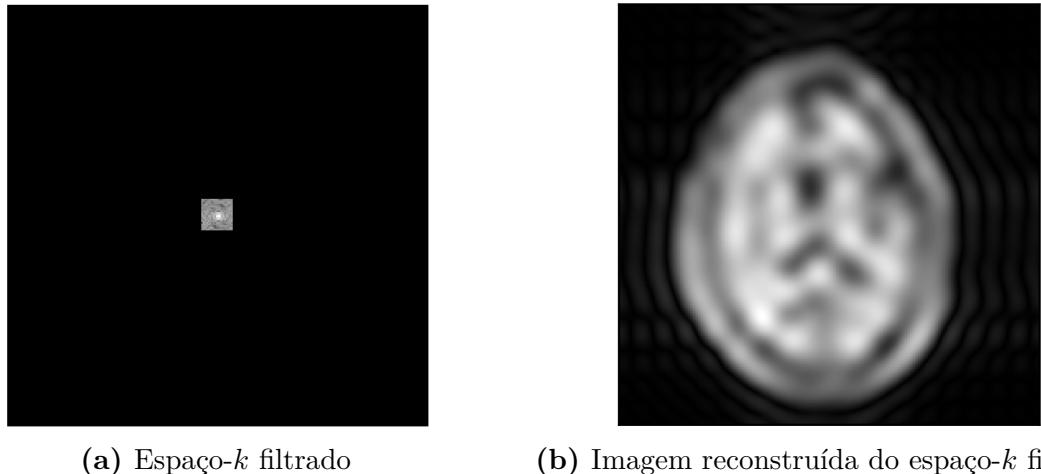
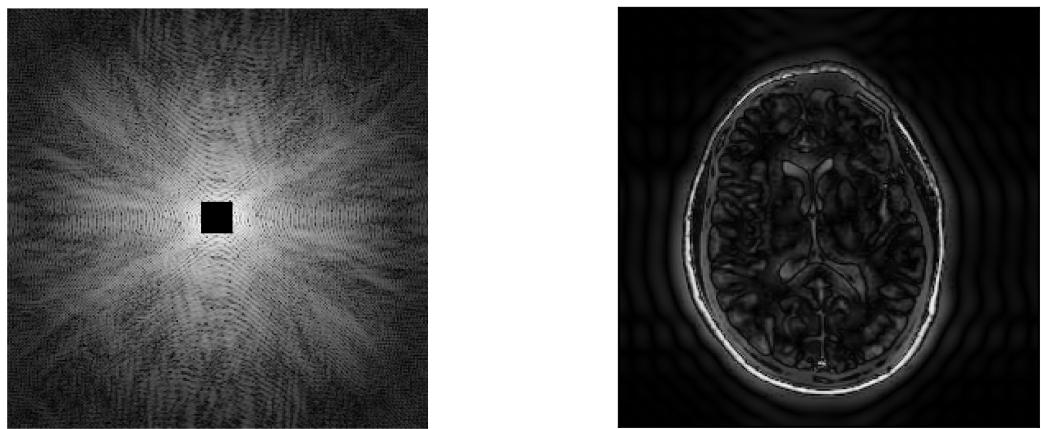


(b) Filtro passa-alta

Figura 25

No caso da figura 26a, o espaço- k filtrado retém apenas o centro do espaço original, isto é, apenas as frequências espaciais baixas. A figura 26b mostra a Transformada de Fourier inversa do espaço- k filtrado. Vemos que a imagem reconstruída manteve apenas os contornos mais grosseiros da imagem original.

Analogamente para a figura 27a, apenas componentes de alta frequência do espaço- k foram mantidas, de modo que a figura reconstruída 27b possui os detalhes finos da imagem original, e não mais os contornos grosseiros da figura 26b.

(a) Espaço- k filtrado(b) Imagem reconstruída do espaço- k filtrado**Figura 26**(a) Espaço- k filtrado(b) Imagem reconstruída do espaço- k filtrado**Figura 27**

Ao analisar o efeito dos filtros sobre o espaço- k , vemos que a formação de imagem pode ser entendida como uma sobreposição de ondas de alta e baixa frequências. Ondas de baixa frequência espacial poderão fornecer apenas contornos grosseiros devido ao seu maior comprimento de onda. Analogamente, as ondas de alta frequência espacial possuirão informações sobre os contornos finos da imagem, já que possuem pequeno comprimento de onda.

4 Materiais e Métodos

Para realizar um diagnóstico não-invasivo com imagens de Ressonância Magnética, é preciso que as imagens de um tecido saudável se diferenciem de alguma maneira das imagens de um tecido doente. Alguns trabalhos nos últimos anos (KATSUBE et al., 2011; HAIMERL et al., 2013; BESA et al., 2015; DING et al., 2015) mostraram que, após a aplicação de um agente de contraste chamado *ácido gadoxético*, os tempos de relaxação longitudinal T_1 de tecidos hepáticos doentes e saudáveis são alterados de maneira distinta. Assim, extrair T_1 de imagens adquiridas antes e após a aplicação do contraste permite avaliar se a região de interesse está doente ou não.

A seguir, introduzimos a sequência GE utilizada nas aquisições deste trabalho, bem como o modelo matemático que descreve a intensidade do sinal em função dos parâmetros da amostra e da aquisição. Em seguida, trataremos de dois métodos comumente utilizados para extrair o valor de T_1 das imagens.

4.1 Aquisição com tempo de repetição T_R curto

Obter imagens de RM da região abdominal traz algumas dificuldades extras quando comparadas à obtenção de imagens do crânio, por exemplo. Como os órgãos não podem se mover durante a aquisição, o paciente precisa prender a respiração. Isso, somado ao seu desconforto devido ao confinamento em um pequeno espaço, limita consideravelmente o tempo de apneia. Portanto, a aquisição deve ocorrer o mais rápido possível.

A escolha natural de sequências para tais aquisições são as Gradiente Eco, devido a sua rapidez. Um tipo específico dessas são as chamadas sequências FLASH (do inglês, Fast Low Angle SHot), as quais utilizam tempo de repetição T_R curto e um único pulso de RF para deslocar a magnetização por ângulos de flip pequenos (HAASE et al., 1986). Nessa sequência, após alguns T_R a magnetização longitudinal M_L atinge um regime permanente, no qual ela possui o mesmo estado inicial imediatamente antes de cada pulso de RF. Em geral, temos que $T_R < T_2$ e, portanto, não há tempo suficiente para a magnetização longitudinal M_T relaxar completamente até o próximo pulso de RF. Assim, a M_T obtida devido ao deslocamento de M_L após um pulso iria “se misturar” com M_T remanescente do ciclo anterior, gerando um sinal totalmente distinto após cada T_R . Consequentemente, aplicam-se mecanismos de *spoiling*, os quais eliminam a magnetização transversal antes de cada novo pulso de RF e, assim, a magnetização longitudinal pode ser utilizada para a aquisição de dados (BROWN et al., 2014).

Numa sequência FLASH, o sinal S é proporcional à densidade de spins em um

voxel ρ , sendo esta dada pela equação (BROWN et al., 2014):

$$\rho = \rho_0 \sin(\theta) \frac{1 - e^{-T_R/T_1}}{1 - e^{-T_R/T_1} \cos(\theta)} e^{-T_E/T_2} \quad (4.1)$$

onde

- ρ_0 : amplitude
- T_E : tempo de eco
- θ : ângulo de flip
- T_1 : tempo de relaxação longitudinal
- T_R : tempo de repetição
- T_2 : tempo de relaxação transversal

Assim, temos

$$S = S_0 \sin(\theta) \frac{1 - e^{-T_R/T_1}}{1 - e^{-T_R/T_1} \cos(\theta)} e^{-T_E/T_2} \quad (4.2)$$

onde S_0 é proporcional a ρ_0 e aos demais fatores de amplificação A do equipamento, de modo que $S_0 = A\rho_0$.

A forma qualitativa dessa função pode ser vista na figura 28, onde vemos que a função apresenta um máximo. Esse máximo ocorre no chamado ângulo de Ernst:

$$\theta_E = \cos^{-1}(e^{-T_R/T_1}) \quad (4.3)$$

Uma forma padrão de se extrair o valor de T_1 consiste em ajustar pontos experimentais por meio da função 4.2. Para tal, é necessário realizar aquisições com T_R , T_E e T_2 fixos e variar θ . Esse método é normalmente chamado de “VFA” (*Variable Flip Angle*) e tem sido amplamente utilizado e aprimorado com o passar dos anos (GUPTA, 1977; WANG; RIEDERER; LEE, 1987; FRAM et al., 1987; PARKER; BARKER; TOFTS, 2001; DEONI; PETERS; RUTT, 2004; CHENG; WRIGHT, 2006; LI et al., 2016; HEULE; BIERI, 2017). Lamentavelmente, máquinas comerciais de RM não fornecem sequências do tipo FLASH em que possamos variar θ numa mesma aquisição. Assim, torna-se necessário realizar aquisições distintas para se obter diversos pontos (θ, S) de um mesmo tecido.

Como mencionado anteriormente, é vital realizar a aquisição o mais rápido possível, devido à limitação físico-biológica do paciente em prender a respiração por longos períodos de tempo. Portanto, a maneira ideal de se obter T_1 seria por meio de apenas duas aquisições, isto é, dois pontos (θ_1, S_1) e (θ_2, S_2) . Obviamente, é inviável realizar o ajuste de uma função não-linear como a 4.2 com apenas dois pontos. Felizmente, algumas técnicas permitem tal feito, ainda que ocasionando em algumas limitações para os parâmetros da aquisição. Apresentamos a seguir os dois principais métodos encontrados na literatura para obtenção de T_1 neste cenário.

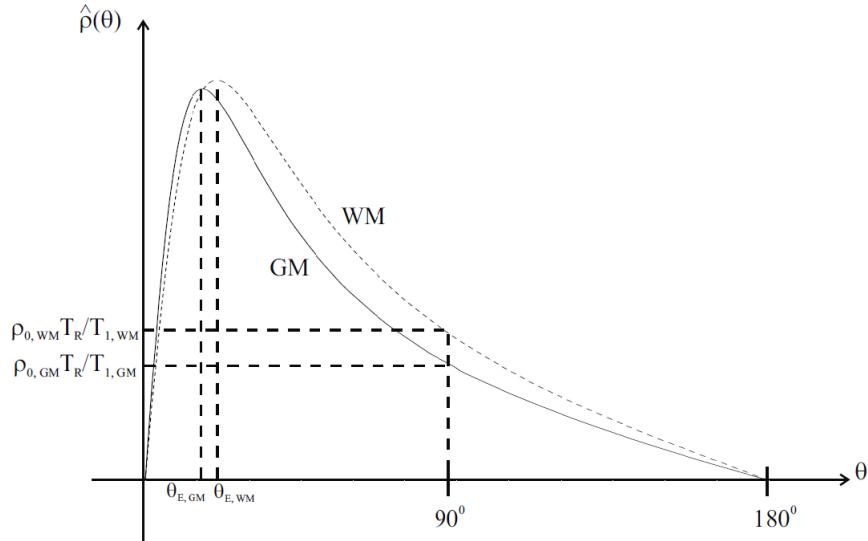


Figura 28 – Plot da densidade de spins ρ em função do ângulo de flip θ para massa branca e massa cinzenta. A função pode ser aproximada por uma reta para $\theta \ll \theta_E$. Para aplicações com tempo de repetição curto e ângulos pequenos, costuma-se se operar nessa região linear. Além disso, o contraste (diferença entre as curvas) torna-se maior para $\theta > \theta_E$. (BROWN et al., 2014)

4.2 Método 1: Linearização

O primeiro método encontrado (GUPTA, 1977) consiste em trabalhar a equação a fim de obter uma função linear da forma $Y = \alpha \times X + \beta$:

$$\frac{S}{\sin(\theta)} = e^{-T_R/T_1} \times \frac{S}{\tan(\theta)} + S_0 e^{-T_E/T_2^*} (1 - e^{-T_R/T_1}) \quad (4.4)$$

onde $S/\sin(\theta)$ está para Y , assim como $S/\tan(\theta)$ está para X .

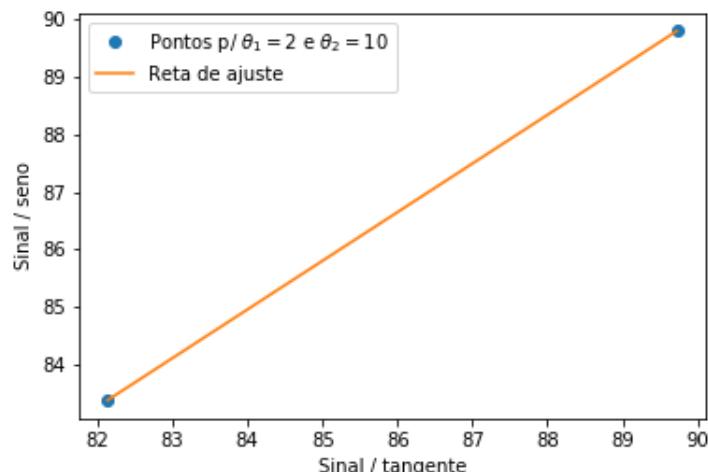


Figura 29

Ao plotar os valores de $S/\sin(\theta)$ em função de $S/\tan(\theta)$ como na figura 29, podemos realizar um ajuste linear pelo método dos mínimos quadrados e obter os coeficientes angular α e linear β da reta. O valor de T_1 pode então ser extraído por:

$$\alpha = e^{-T_R/T_1} \Rightarrow T_1 = -\frac{T_R}{\log(\alpha)} \quad (4.5)$$

Em posse de T_1 , podemos utilizar β para obter a amplitude S_0 :

$$S_0 = \frac{\beta}{e^{-T_E/T_2^*}(1 - e^{-T_R/T_1})} \quad (4.6)$$

4.3 Método 2: Razão entre os sinais

Na equação da intensidade do sinal resultante de uma sequência FLASH

$$S = S_0 \sin(\theta) \frac{1 - e^{-T_R/T_1}}{1 - e^{-T_R/T_1} \cos \theta} e^{-T_E/T_2} \quad (4.7)$$

temos que T_1 , T_2 e S_0 são parâmetros da amostra, enquanto que T_R , T_E e θ são parâmetros da aquisição. Tendo isso em vista, alguns autores propuseram um método mais simples para se obter T_1 (PARKER; BARKER; TOFTS, 2001). Se realizarmos aquisições com ângulos de flip distintos, obteremos sinais S_1 e S_2 . Ao dividirmos a expressão 4.7 para ambos os sinais, obtemos

$$\frac{S_2}{S_1} = \frac{\sin \theta_2}{\sin \theta_1} \frac{1 - e^{-T_R/T_1} \cos \theta_1}{1 - e^{-T_R/T_1} \cos \theta_2} \quad (4.8)$$

Após a divisão, ocorre o cancelamento de alguns termos. Observe que, como a densidade de spins é a mesma para ambas aquisições, exclui-se S_0 em 4.8. Assim, após alguma álgebra, podemos isolar T_1 :

$$T_1 = \frac{-T_R}{\ln \left(\frac{\gamma - 1}{\gamma \cos \theta_2 - \cos \theta_1} \right)} \quad (4.9)$$

onde

$$\gamma = \frac{S_2 \sin \theta_1}{S_1 \sin \theta_2} \quad (4.10)$$

Diferente do método da linearização, esta equação fornece uma forma mais simples e rápida de se obter T_1 . No entanto, não há como obter o valor de S_0 .

Repare ainda que, a equação 4.9 possui exatamente a mesma forma da equação 4.5. De fato, ao utilizar os mesmos dados em ambos os métodos, obtém-se resultados idênticos,

indicando que os métodos são equivalentes e que a expressão analítica para o coeficiente angular no Método 1 é

$$\alpha = \ln \left(\frac{\gamma - 1}{\gamma \cos \theta_2 - \cos \theta_1} \right) \quad (4.11)$$

4.3.1 Correção para aquisições distintas

Como mencionado anteriormente, $S_0 = A\rho_0$, sendo ρ_0 densidade de spins e A um termo devido a demais fatores de amplificação do equipamento. Dessa forma, reescrevemos a equação 4.7 como:

$$S = A\rho_0 \sin(\theta) \frac{1 - e^{-T_R/T_1}}{1 - e^{-T_R/T_1} \cos \theta} e^{-T_E/T_2} \quad (4.12)$$

Uma vez que são realizadas aquisições distintas para variar o ângulo θ , o fator de amplificação A do equipamento pode ser diferente em cada aquisição. Assim, ao dividir uma equação pela outra como em 4.8, apenas ρ_0 é cancelado. Nessa caso, temos:

$$\frac{S_2}{S_1} = \frac{A_2 \sin \theta_2}{A_1 \sin \theta_1} \frac{1 - e^{-T_R/T_1} \cos \theta_1}{1 - e^{-T_R/T_1} \cos \theta_2} \quad (4.13)$$

Ao isolar T_1 , obtemos novamente

$$T_1 = \frac{-T_R}{\ln \left(\frac{\gamma - 1}{\gamma \cos \theta_2 - \cos \theta_1} \right)} \quad (4.14)$$

mas com

$$\gamma = \frac{S_2}{S_1} \frac{A_1 \sin \theta_1}{A_2 \sin \theta_2} \quad (4.15)$$

onde denominamos a razão $\frac{A_1}{A_2} = F$ como “fator de normalização”.

Normalmente, assume-se $A_1 = A_2$. No entanto, veremos que tal igualdade nem sempre é válida, o que pode causar problemas no processamento.

4.4 Limitações dos Métodos

Os dois métodos para obtenção de T_1 aqui apresentados simplificam consideravelmente a tarefa. Obtemos T_1 com apenas dois pontos, o que seria inviável se fôssemos realizar o ajuste de uma equação não-linear como a 4.2.

No entanto, existe um custo atrelado a tal simplificação. Veremos a seguir que algumas limitações surgem para garantir a eficácia do método VFA quando utilizamos apenas dois ângulos.

4.4.1 Ângulos ideais

Para se obter um resultado confiável pelos métodos apresentados, um fator de fundamental importância é a seleção correta dos ângulos de flip. A equação 4.14 mostra que o cálculo de T_1 depende de um logaritmo. Ao analisar o resultado de tal equação utilizando diferentes pares de ângulos, percebe-se que a equação é extremamente sensível a pequenas variações em S_1 e S_2 quando os ângulos não são escolhidos apropriadamente, uma vez que o argumento do logaritmo é normalmente muito próximo de 1, podendo inclusive gerar resultados absurdos devido a mudança de sinal no entorno de $\log(1) = 0$.

Alguns trabalhos mostram que existem valores ótimos de θ de modo a minimizar a sensibilidade do método e, portanto, a incerteza no valor obtido de T_1 (WANG; RIEDERER; LEE, 1987; DEONI; RUTT; PETERS, 2003). Com os ângulos ótimos, os resultados são equiparáveis ou até melhores do que ao utilizar muitos ângulos equidistantes entre si (WANG; RIEDERER; LEE, 1987). Uma regra simples, ainda que aproximada, para obtenção dos ângulos ótimos consiste em utilizar os ângulos para os quais o sinal é 71% do valor máximo (DEONI; PETERS; RUTT, 2004):

$$S_{\text{ideal}}(\theta_{\text{ideal}}) = 0.71 \times S(\theta_E) \quad (4.16)$$

onde θ_E é o ângulo de Ernst.

4.4.2 Incertezas no ângulo θ

Um segundo fator que influencia consideravelmente os resultados são incertezas no valor do ângulo de flip. Vimos na seção 3.9.1 que, inevitavelmente, ocorre uma excitação não uniforme dos spins devido ao truncamento temporal do pulso de excitação. Além disso, as bobinas utilizadas nos experimentos não são ideias e, portanto, o perfil de excitação gerado por elas será inhomogêneo. Como consequência, os ângulos de flip ao longo da região excitada não serão uniformes. Essas pequenas inhomogeneidades do ângulo de flip dentro de uma região de interesse tem influência considerável no valor final de T_1 obtido pelo método VFA (PARKER; BARKER; TOFTS, 2001; CHENG; WRIGHT, 2006).

O gráfico na figura 30 mostra o resultado de simulações ilustrando o efeitos das inhomogeneidades no ângulo de flip. No eixo vertical temos a razão entre o valor obtido de T_1 pelo método VFA e o valor verdadeiro. Já o eixo horizontal mostra a razão entre o valor verdadeiro do ângulo de flip e o valor nominal devido às inhomogeneidades. A linha

menos espessa mostra os valores teóricos, enquanto a mais espessa mostra os resultados da simulação. Temos que, quanto maior a diferença entre os ângulos nominal e verdadeiro, maior é o erro no valor obtido de T_1 . As duas linhas concordam relativamente bem quando o erro relativo no ângulo de flip é de até 15% (CHENG; WRIGHT, 2006).

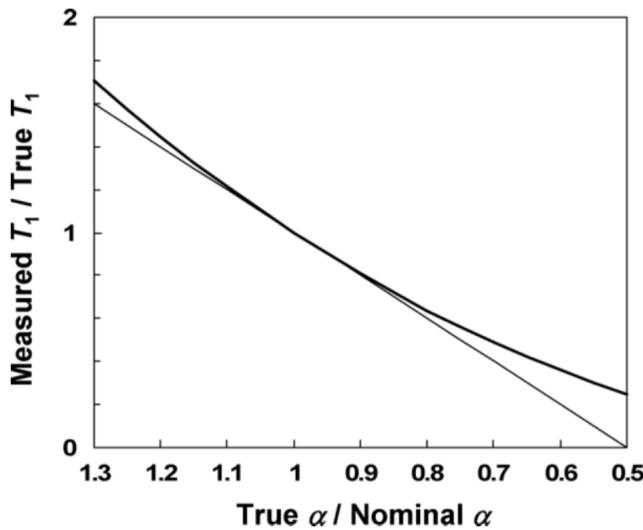


Figura 30 – O erro no valor obtido de T_1 pela simulação aumenta conforme o ângulo de flip α se afasta do valor verdadeiro. Retirado de (CHENG; WRIGHT, 2006).

Uma forma de contornar o problema das inhomogeneidades consiste em criar um mapa do campo de excitação B_1 . Para tal, utiliza-se um fantoma de composição uniforme e T_1 conhecido a fim de mapear espacialmente a distribuição do campo B_1 . Esse mapa permite inferir o valor do ângulo de flip gerado ponto a ponto. Posteriormente, essa informação é utilizada para corrigir os valores de T_1 obtidos.

O gráfico da figura 31 compara os resultados de experimentos com e sem a correção dos ângulos de flip. A linha contínua indica o valor correto de T_1 . É evidente que os pontos corrigidos concordam com os valores numéricos esperados, enquanto que os valores não corrigidos se distanciam do ideal quanto maior for T_1 .

4.5 Aquisição das imagens de Ressonância Magnética

As imagens de Ressonância Magnética utilizadas neste trabalho foram adquiridas em parceria com o Instituto D'Or de Pesquisa e Ensino (IDOR)¹.

O equipamento utilizado foi o Philips Achieva 1.5 Tesla. Foram coletados dados de 66 pacientes com diferentes graus de obesidade, tanto antes quanto após a injeção do agente de contraste “ácido gadoxético”. As imagens pós-contraste foram adquiridas em quatro momentos: 5, 10, 15 e 20 minutos após a aplicação do agente. Exames histopatológicos

¹ Instituto D'Or de Pesquisa e Ensino (IDOR). Rua Diniz Cordeiro, 30 22281-100 - Botafogo-RJ

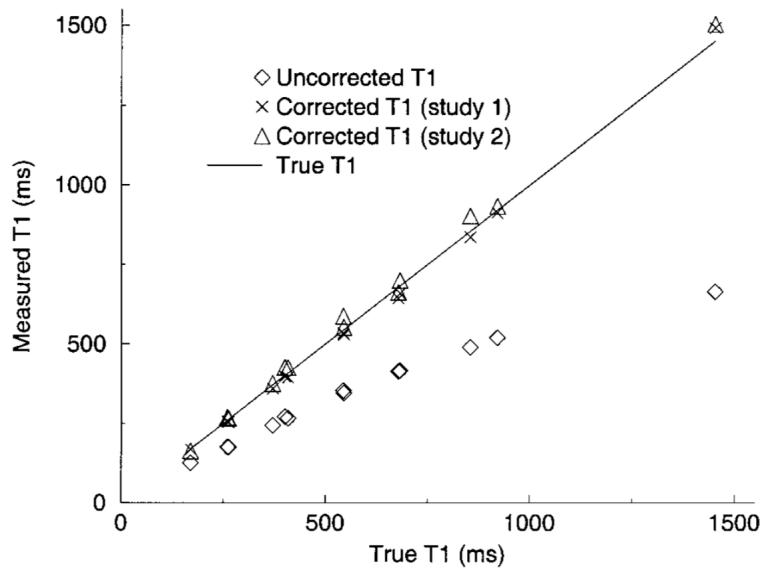


Figura 31 – Pontos marcados por X e por um triângulo indicam valores corrigidos, em dois estudos distintos. Quanto maior o valor de T_1 do tecido, maior é o erro do resultado quando não se realiza a correção. Retirado de (PARKER; BARKER; TOFTS, 2001).

do fígado foram realizados para diagnóstico e posterior comparação com o processamento de imagem. Os diagnósticos foram diversos e classificados em 3 grupos, dos quais dois possuem ainda sub-classificações:

- Cirrose (15 pacientes)
- Esteatose (10 pacientes)
 - Esteatose hepática leve. (8 pacientes)
 - Esteatose hepática moderada. (2 pacientes)
- Esteato-hepatite (41 pacientes)
 - Com atividade inflamatória leve e sem fibrose. (4 pacientes)
 - Com atividade inflamatória leve e fibrose leve. (18 pacientes)
 - Com atividade inflamatória leve e fibrose moderada. (1 paciente)
 - Com atividade inflamatória leve e fibrose moderada a acentuada. (1 paciente)
 - Com atividade inflamatória leve e fibrose acentuada. (1 paciente)
 - Com atividade inflamatória moderada e fibrose leve. (8 pacientes)
 - Com atividade inflamatória moderada e fibrose moderada. (5 paciente)
 - Com atividade inflamatória acentuada e fibrose moderada a acentuada. (1 paciente)
 - Com atividade inflamatória moderada e fibrose acentuada. (2 pacientes)

Para o escopo deste trabalho, lidamos apenas com a classificação dos 3 grupos principais: cirrose, esteatose e esteato-hepatite.

Os parâmetros da aquisições foram escolhidos de modo a otimizar tempo de aquisição e obter o número de imagens desejável por paciente. Os parâmetros escolhidos foram $T_R = 3.789\text{ ms}$, $T_E = 1.813\text{ ms}$ e ângulos de flip $\theta_1 = 2^\circ$ e $\theta_2 = 10^\circ$ (cada ângulo utilizado em diferentes aquisições).

Tomando um valor típico de $T_1 = 613.9\text{ ms}$ para o fígado saudável (BESA et al., 2015), os parâmetros aplicados à equação 4.2 fornecem um ângulo de Ernst $\theta_E = 6.05^\circ$ e ângulos ótimos para aquisição de aproximadamente 2.5° e 16° .

4.6 Interface Gráfica

Para criar a Interface Gráfica do Utilizador (GUI), utilizamos a linguagem Python, versão 3.6.5 (Anaconda Software Distribution, 2018). O código pode ser encontrado no apêndice C.

O objetivo foi criar uma GUI simples, na qual é possível carregar imagens de ressonância e selecionar determinada região de interesse (ROI) na imagem, a fim de ter um diagnóstico sobre a saúde da ROI. A GUI pode ser visualizada na figura 32. Descrevemos a seguir como o usuário deve utilizar essa ferramenta.

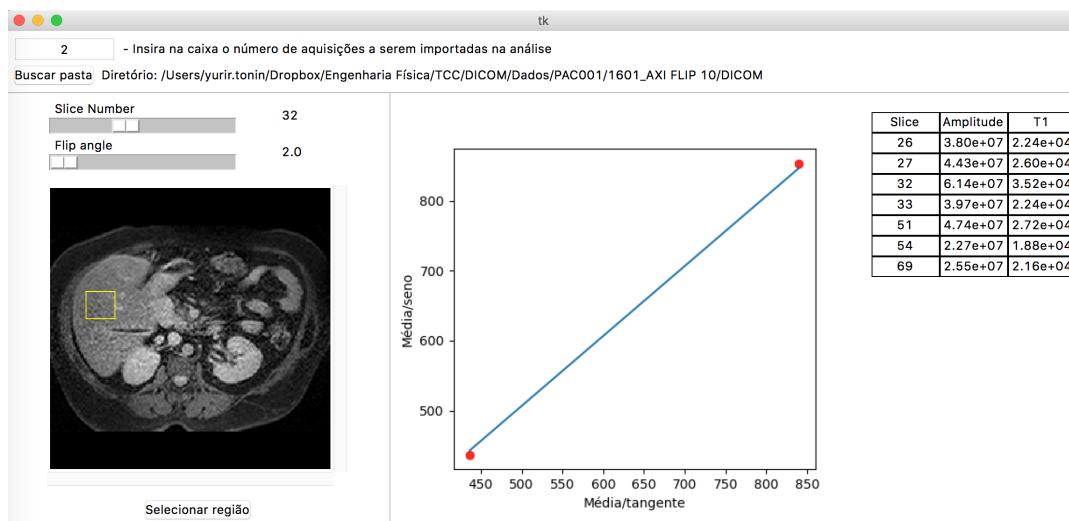


Figura 32 – Interface Gráfica desenvolvida em Python para processamento de imagem.

Ao iniciar a GUI, deve-se primeiramente selecionar quantas aquisições com ângulos de flip distintos deseja-se utilizar no processamento. Para tal, basta digitar o número N de aquisições na caixa superior da GUI. O usuário deve clicar no botão “Buscar Pasta” para selecionar onde em seu computador se encontram os arquivos da aquisição. A GUI automaticamente pede para que o usuário selecione N pastas, uma após a outra, de modo a selecionar todas as aquisições para análise.

Após selecionada a última pasta, a Interface carrega os dados, mostra as barras de seleção e exibe a imagem, como pode ser visto na zona à esquerda da figura 32. Uma barra é utilizada para selecionar a aquisição, isto é, qual conjunto de imagens com determinado ângulo de flip se deseja visualizar. Com a segundo barra, o usuário escolhe o corte de interesse. Conforme se movem as barras, a imagem é atualizada em tempo real no visualizador abaixo das mesmas.

Dessa maneira, o usuário pode encontrar o corte que mostra a região de interesse do fígado e, clicando e arrastando o mouse, selecionar uma região do órgão sobre a qual ele deseja realizar o processamento e obter um diagnóstico. Após isso, basta clicar no botão “Selecionar região”. Surge então um gráfico à direita da GUI, que nada mais é que um plot do método da “Linearização”, como o da figura 29.

O ajuste linear do gráfico é utilizado para extrair o coeficiente angular da reta que, por sua vez, fornece o valor de T_1 por meio da equação 4.5. A direita do gráfico, uma tabela mostra o valor da amplitude S_0 e do tempo de relaxação T_1 para a cada um dos cortes selecionados sobre a imagem na interface dinâmica.

5 Resultados

Apresentamos os resultados e análises na ordem cronológica em que foram realizados. Primeiramente, tratamos do processamento utilizando a Interface Gráfica desenvolvida. Essa primeira análise não forneceu os resultados esperados. Realizamos então simulações a fim de esclarecer a fonte do problema. Por fim, fizemos uma análise numérica dos dados diretamente por Excel para contornar os problemas encontrados ao utilizar a GUI.

5.1 Análise com a Interface Gráfica

Imagens do fígado de diversos pacientes foram processadas com a Interface Gráfica. Isso foi feito tanto para aquisições pré quanto pós-contraste. No entanto, os valores obtidos para o tempo de relaxação T_1 foram maiores do que o esperado. Para o fígado (doente ou saudável) os valores esperados são da ordem de centenas de milissegundos ([BESA et al., 2015](#); [KATSUBE et al., 2011](#)), enquanto que os valores obtido pela GUI foram da ordem de 10^4 segundos, como mostra a figura [32](#).

Num primeiro momento, assumimos que havia algum erro de cálculo no código o que, após uma extensa análise, foi descartado. A fim de reforçar de que o código estava livre de erros, utilizamos um software livre de processamento de imagens médicas chamado MIPAV ([National Institutes of Health Center for Information Technology \(CIT\), 2017](#)), a fim de avaliar se a média da intensidade do sinal numa ROI era a mesma obtido pela GUI e pelo MIPAV. Isso se confirmou verdadeiro e descartamos definitivamente a hipótese de erros no código.

5.2 Simulações

Após uma primeira tentativa sem sucesso, buscamos esclarecer o comportamento inesperado do processamento por meio de simulações. Como o sinal possui certo ruído e o método pode ser extremamente sensível a variações no sinal, conjecturamos que aquele poderia ser a fonte de erro.

Os parâmetros utilizados na simulação foram os mesmos da aquisição. Tomamos $T_1 = 613.9$ ms, um valor típico para o fígado saudável ([BESA et al., 2015](#)). A curva amarela na figura [33](#) mostra o sinal ideal. As demais curvas indicam sinais com diferentes níveis de ruído. Definimos o ruído como valores aleatórios no intervalo $[0, \sigma]$, sendo σ o desvio padrão do sinal em uma região do fígado.

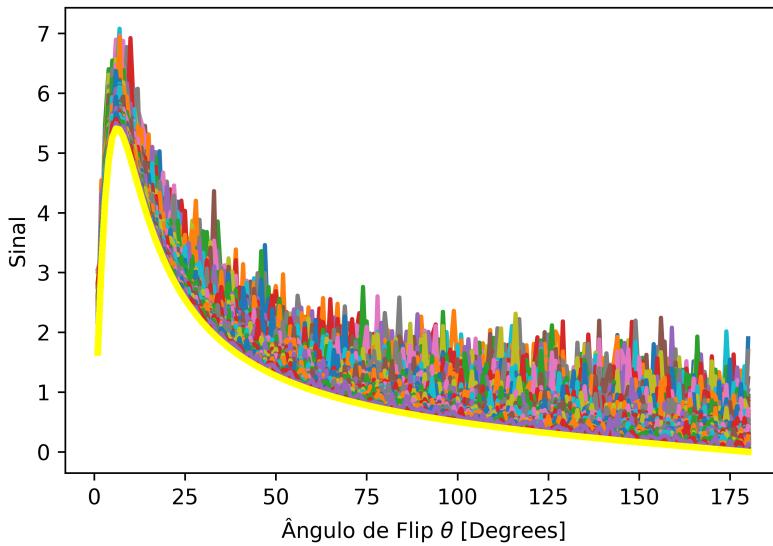


Figura 33 – Simulação da equação 4.2 para $S_0 = 100$, $T_1 = 613.9$ ms, $T_R = 3.789$ ms, $T_E = 1.813$ ms, $\theta_1 = 2^\circ$ e $\theta_2 = 10^\circ$.

Na figura 34 temos o conjunto do pontos linearizados pelo método 1, extraídos dos sinais da figura 33. Vemos que os pontos se aglomeram em duas regiões: no canto inferior, temos aqueles referentes ao sinal para o ângulo de 10° ; à direita, para o ângulo de 2° . Lembre-se que o valor de T_1 é extraído pelo coeficiente angular da reta formada pelos dois pontos. Como os dois aglomerados de pontos possuem a mesma inclinação, uma primeira análise parece indicar que a as retas de ajuste também a terão e que, portanto, o coeficiente angular das retas será aproximadamente o mesmo. No entanto, veremos a seguir que, mesmo neste caso, a variação final no valor T_1 não é insignificante, comprovando a grande sensibilidade do método aos demais parâmetros da aquisição.

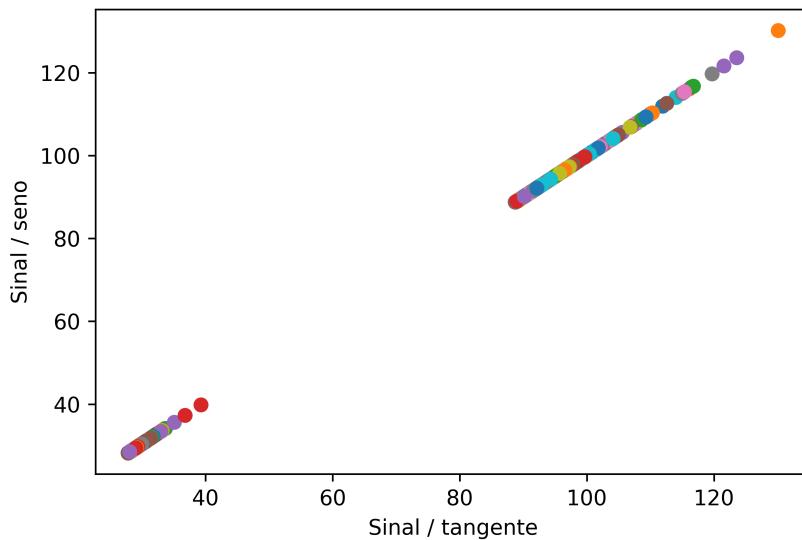


Figura 34 – Conjunto dos pontos linearizados pelo método 1 a partir dos sinais na figura 33.

Definindo a razão sinal-ruído (SNR) como a razão entre a média do sinal e seu

desvio padrão, obtivemos para nossas imagens com $\theta = 2^\circ$:

$$\text{SNR} = \frac{S}{\sigma} \approx 15 \quad (5.1)$$

Para analisar os efeitos de diferentes SNR sobre o valor obtido de T_1 , calculamos a variação percentual $\Delta\%$ do tempo de relaxação antes (T_1) e após ($T_{1\text{ruído}}$) adicionar o ruído aleatório:

$$\Delta\%T_1 = \left| \frac{T_{1\text{ruído}} - T_1}{T_1} \right| \quad (5.2)$$

Plotamos os valores de $\Delta\%T_1$ em função da SNR na figura 35. Como esperado, vemos que $\Delta\%T_1$ diminui conforme cresce a SNR, isto é, quanto menor o ruído, mais preciso é o resultado. O SNR = 15 de nossas imagens é marcado pela linha verde.

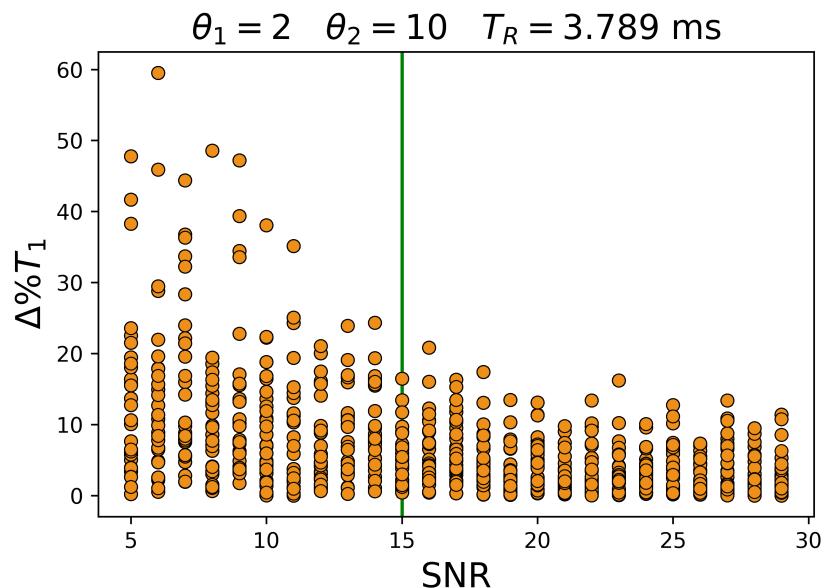


Figura 35 – Variação percentual do valor de T_1 obtido pelo método da linearização em relação ao valor correto.

Vemos que a variação percentual esperada para nossas imagens é de aproximadamente 20%. Se $T_1 = 613.9 \text{ ms}$, tal variação indica que obteríamos um tempo de relaxação de até $1.2 \times 613.9 \text{ ms} = 736.7 \text{ ms}$ caso o único problema com as imagens decorresse do ruído. Tal valor ainda é muito menor do que o obtido pelo processamento na GUI ($T_1 \approx 10^4$), sugerindo a existência de uma outra fonte de erro, muito mais significativa.

A figura 35 reforça também a sensibilidade do método. Mesmo com o coeficiente angular das retas variando pouco em relação aos sinais com baixo e com alto SNR (figura 34), isso implica numa diferença de até 50% em $\Delta\%T_1$.

5.3 Fonte do erro

Após excluirmos a possibilidade de que o erro provinha do processamento, passamos a investigar a possibilidade de problemas nos próprios dados. Um indicativo de que o problema estava de fato nas imagens surgiu quando comparamos os valores da intensidade de sinal extraídos pela GUI para 2° e 10°

Ao simular o sinal com os parâmetros de nossas aquisições, obtemos que a razão ideal entre a intensidade do sinal para os ângulos de 2° e 10° é $S_{10^\circ}/S_{2^\circ} = 1.58$ para imagens pré-contraste (figura 36a) e $S_{10^\circ}/S_{2^\circ} = 2.86$ para imagens pós-contraste¹ (figura 36b). No entanto, ao realizar essa mesma análise para os valores de intensidade do sinal de nossas imagens, obtemos razões significativamente diferentes e até mesmo razões menores que 1.

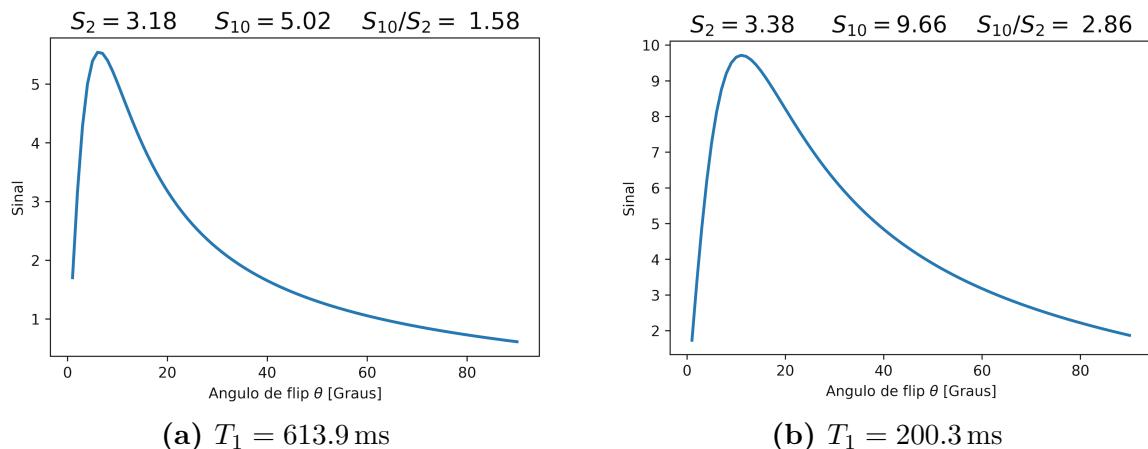


Figura 36 – Sinal FLASH simulado para avaliação das razões entre os sinais em 2° e 10°.

Tal fato nos levou a considerar que as imagens, por serem obtidas em aquisições distintas, apresentavam diferentes fatores de normalização A_1 e A_2 e, portanto, a intensidade relativa entre os sinais apresentava completa inconsistência.

Para resolver tal problema seria necessário conhecer os valores de A_1 e A_2 para corrigir o γ com o fator de normalização F , como apresentado na seção 4.3.1. No entanto, esses valores são mecanismos internos do equipamento de ressonância, os quais não são fornecidos ao usuário. Desenvolvemos então uma estratégia encontrada para corrigir a intensidade de sinal e obter um melhor resultado, como veremos a seguir.

5.4 Obtensão do fator “F” de Normalização

Com o apoio do Instituto D'Or de Pesquisa e Ensino, extraímos os valores de intensidade do sinal diretamente pelo software da Philips, a fim de garantir que erros no

¹ Assumimos como tempo de relaxação pós-contraste $T_1 \equiv 200.3\text{ ms}$ (BESA et al., 2015).

sinal seriam oriundos apenas do ruído e dos diferentes fatores de amplificação.

Para todos os pacientes foram escolhidas sete ROIs², similares às mostradas pelo círculos verde na figura 37. Três foram selecionadas no fígado, duas na musculatura paravertebral, uma no baço e uma na região exterior ao paciente (ruído).

O plano para correção dos valores foi utilizar a ROI do músculo ou do baço para ajustar os valores das ROIs de interesse, uma vez que músculo e baço, ao contrário do fígado, apresentam pouca ou nenhuma absorção do agente de contraste. Assim, poderíamos assumir que o valor do sinal antes e após a injeção do agente deveria permanecer constante e, portanto, teríamos sempre a mesma razão S_{10°/S_{2° para esses tecidos.

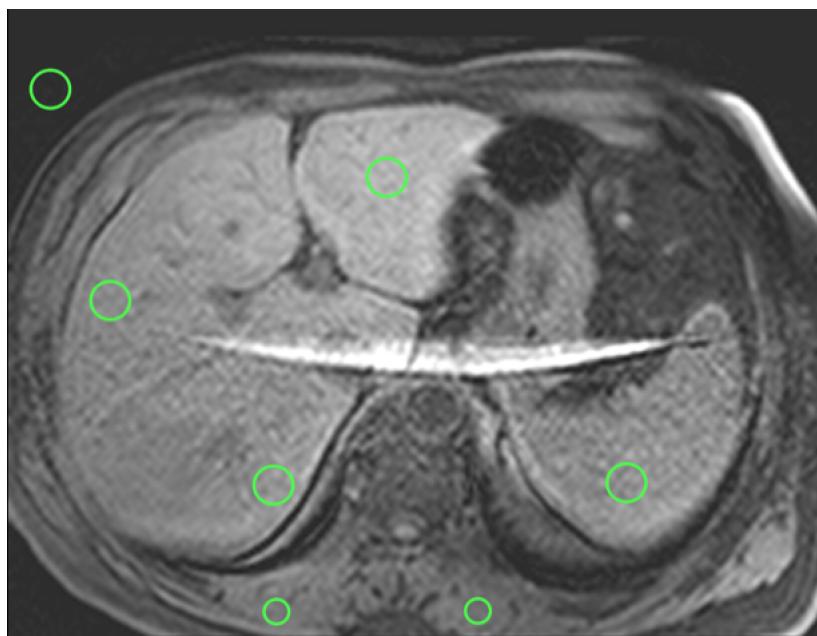


Figura 37 – Exemplo da seleção de regiões de interesse (ROIs) utilizando o software da Philips.

Primeiramente, simulamos qual deveria ser a razão ideal S_{10°/S_{2° para o músculo, utilizando um valor típico de $T_1 = 900$ ms para a musculatura paravertebral (BAZELAIRE et al., 2004). Obtivemos $S_{10^\circ}/S_{2^\circ} = 1.236$.

Realizamos então uma análise numérica dos dados utilizando Microsoft Excel. Uma parte da planilha construída na análise pode ser vista na figura 38. Extraímos os sinais do primeiro tecido muscular (MSK1) para todas as aquisições, os quais podem ser vistos na tabela em azul. Temos que as razões S_{10°/S_{2° calculadas são de fato discrepantes dos valores esperados e são, em sua maioria, menores que 1.

Na tabela cinza, temos o fator F que deve ser multiplicado pela razão do sinal a fim de se obter seu valor ideal $S_{10^\circ}/S_{2^\circ} = 1.236$. A terceira e última tabela (em verde) mostra o termo γ corrigido pelo fator de normalização.

² As áreas das ROIs foram de 1.037 cm^2 para a musculatura e 2.506 cm^2 para os demais tecidos.

Razão entre sinal em 10 e 2 graus para o Músculo						
Paciente	S2/S1 - MSK1 - PRE	S2/S1 - MSK1 - 5'	S2/S1 - MSK1 - 10'	S2/S1 - MSK1 - 15'	S2/S1 - MSK1 - 20'	
SUBJ003	1.230	0.932	0.893	0.834	0.882	
SUBJ004	1.420	0.920	0.904	0.538	0.502	
SUBJ005	1.038	0.789	0.522	0.684	0.414	
SUBJ006	1.101	1.101	0.846	0.671	0.532	
SUBJ007	0.981	0.800	0.789	0.632	0.523	
SUBJ008	0.801	0.520	0.419	0.402	0.344	

Fator "F" para normalização						
Paciente	F - MSK1 - PRE	F - MSK1 - 5'	F - MSK1 - 10'	F - MSK1 - 15'	F - MSK1 - 20'	
SUBJ003	1.005	1.326	1.384	1.482	1.401	
SUBJ004	0.871	1.343	1.368	2.296	2.464	
SUBJ005	1.190	1.566	2.366	1.806	2.988	
SUBJ006	1.123	1.123	1.461	1.842	2.323	
SUBJ007	1.260	1.545	1.566	1.956	2.363	
SUBJ008	1.544	2.375	2.947	3.075	3.592	
SUBJ010	0.570	0.504	0.043	0.535	0.600	

Fator gamma multiplicado pelo fator F de normalização						
Paciente	γ - HEP1 - PRE	γ - HEP1 - 5'	γ - HEP1 - 10'	γ - HEP1 - 15'	γ - HEP1 - 20'	
SUBJ003	0.276	0.365	0.393	0.388	0.378	
SUBJ004	0.206	0.348	0.381	0.397	0.362	
SUBJ005	0.293	0.397	0.425	0.414	0.470	
SUBJ006	0.312	0.323	0.405	0.359	0.406	
SUBJ007	0.269	0.408	0.380	0.398	0.401	
SUBJ008	0.325	0.413	0.372	0.355	0.406	
SUBJ010	0.263	0.309	0.030	0.352	0.385	

Figura 38 – Parte da planilha criada para o correção dos valores de intensidade do sinal para ROIs do fígado. “PRE” indica a aquisição pré-contraste, enquanto 5’, 10’, 15’ e 20’ indicam as aquisições pós-contraste para os respectivos tempos após injeção do agente.

Em posse dos valores corrigidos de γ , calculamos T_1 e obtivemos valores coerentes com o esperado, da ordem de centenas de mili segundos, como vemos na figura 39.

Paciente	T1 - HEP1 - PRE	T1 - HEP1 - 5'	T1 - HEP1 - 10'	T1 - HEP1 - 15'	T1 - HEP1 - 20'
SUBJ003	0.764	0.489	0.430	0.441	0.460
SUBJ004	1.193	0.530	0.453	0.421	0.494
SUBJ005	0.696	0.423	0.374	0.391	0.309
SUBJ006	0.630	0.596	0.408	0.502	0.406
SUBJ007	0.796	0.402	0.456	0.420	0.415
SUBJ008	0.593	0.393	0.473	0.511	0.406
SUBJ011	0.721	0.553	0.349	0.309	0.348
SUBJ012	0.988	0.570	0.368	0.425	0.457
SUBJ013	0.740	0.479	0.397	0.436	0.401
SUBJ014	0.703	0.336	0.509	0.467	0.381
SUBJ015	0.792	0.436	0.458	0.423	0.419

Figura 39 – Valores de T_1 obtidos para alguns pacientes após a normalização.

De acordo com outros trabalhos, o agente de contraste deve reduzir o valor de T_1 (KATSUBE et al., 2011; HAIMERL et al., 2013; BESA et al., 2015). Assim, calculamos a variação percentual de T_1 entre o valor pré-contraste e os diferentes valores pós-contraste. Plotamos esses dados para cada um dos tempos de aquisição, bem como o desvio padrão dos mesmos, para as três regiões do fígado. Esses gráficos podem ser vistos nas figuras 40 a 42. Como esperado, os diferentes diagnósticos apresentam, na média, uma variação distinta do valor de T_1 ao longo do tempo. Discutiremos os resultados em detalhe no capítulo seguinte.

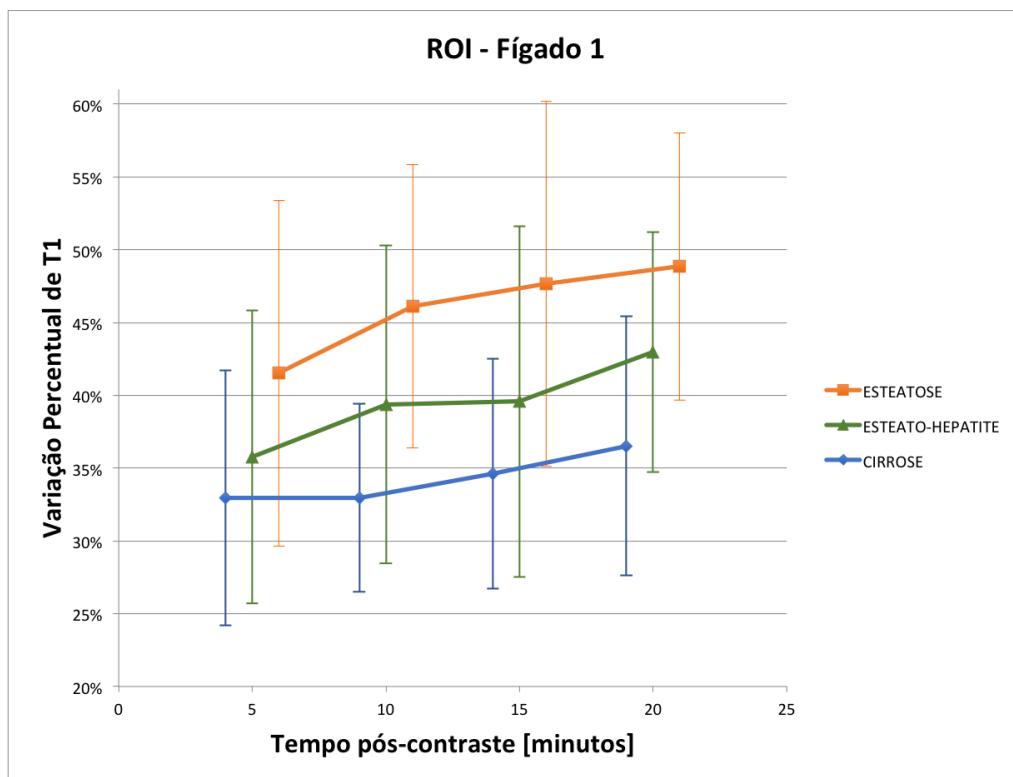


Figura 40 – Variação percentual de T_1 para a ROI 1 do fígado.

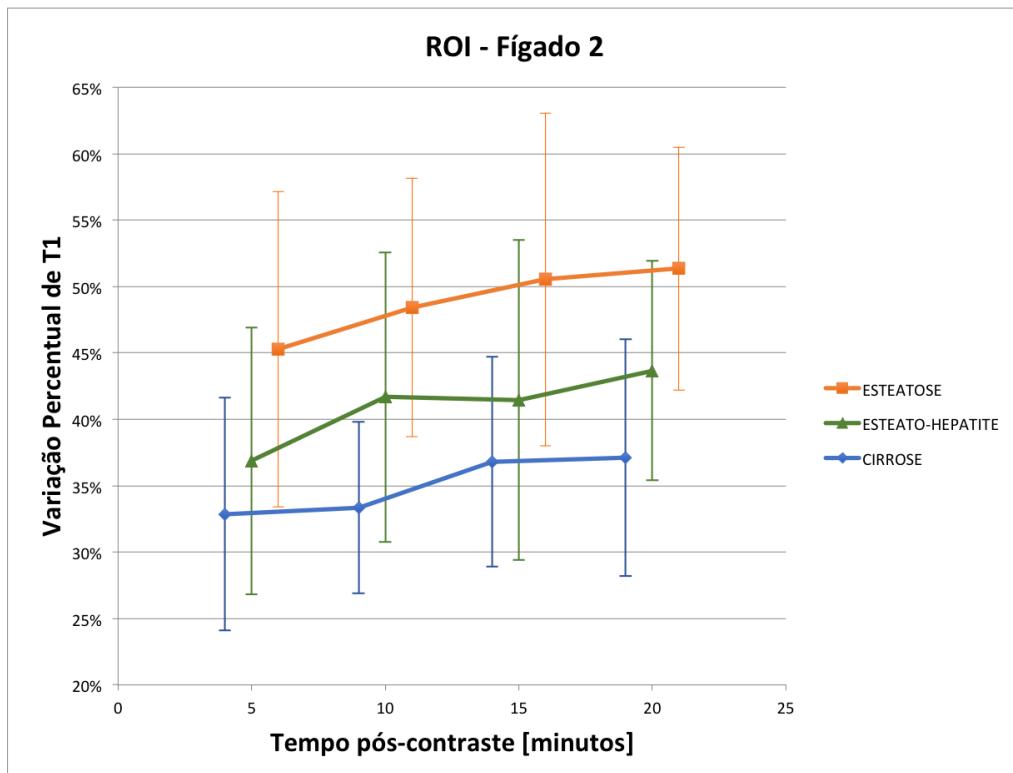


Figura 41 – Variação percentual de T_1 para a ROI 2 do fígado.

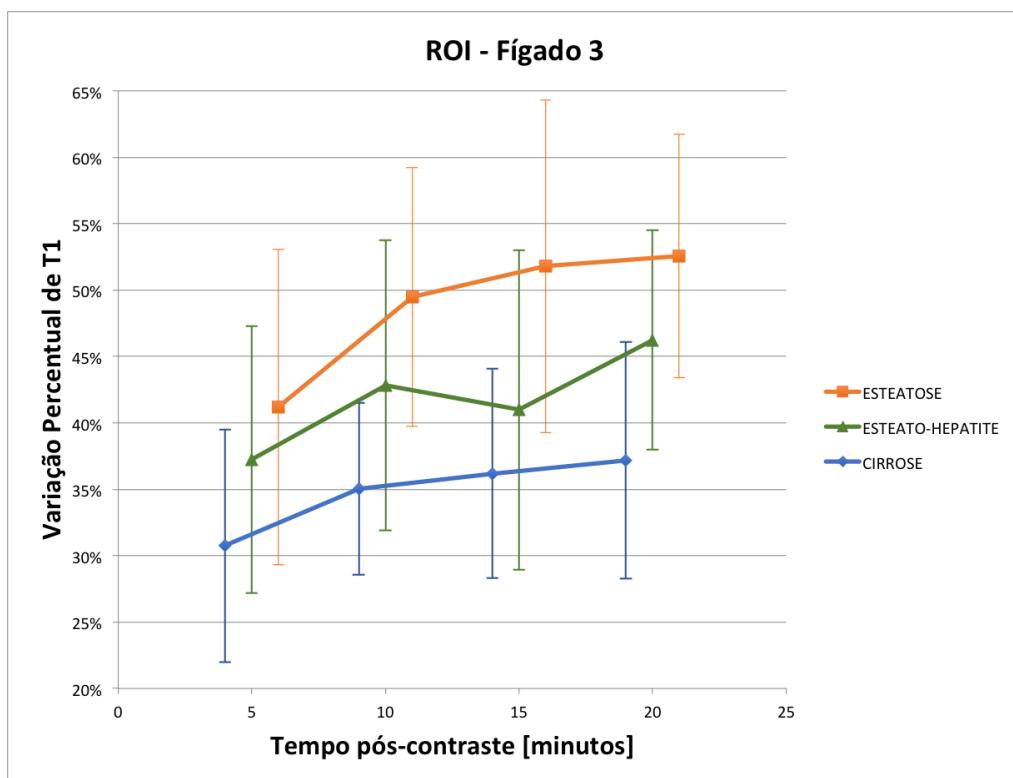


Figura 42 – Variação percentual de T_1 para a ROI 3 do fígado.

6 Discussão

O processamento de imagem com a Interface Gráfica não resultou em valores razoáveis de T_1 . Os problemas não ocorreram devido ao processamento em si, mas sim como consequência do procedimento de aquisição, o qual requer aquisições distintas para se obter mais de um ângulo de flip. Daí decorre que os fatores de amplificação do equipamento em cada aquisição não são iguais. Isso implica que as interpretações do equipamento sobre uma amostra com as mesmas condições físicas serão distintas ,isto é, serão representados por valores numéricos diferentes). Portanto, torna-se impraticável utilizar esses dados diretamente no processamento.

A solução encontrada para se obter um resultado coerente foi a normalização dos valores por meio do sinal do músculo paravertebral. Como T_1 para esse tecido não deve ser influenciado pelo agente de contraste, conseguimos normalizar os pontos de acordo com seus respectivos fatores “F”, que foram extraídos a partir da razão S_{10°/S_{2° do músculo.

Os resultados nos gráficos 40 a 42 concordam com outros trabalhos ([BESA et al., 2015](#); [KATSUBE et al., 2011](#)) e indicam que o método pode auxiliar no diagnóstico de função hepática.

Um primeiro indicativo da coerência dos resultados encontra-se na ordem de grandeza dos valores obtidos para T_1 (centenas de milissegundos). Além disso, temos que todas as curvas são ascendentes. Isso indica a redução do tempo de relaxação entre 5 e 20 minutos após aplicação do contraste, em concordância com a literatura.

A eficácia do método em obter bons valores de T_1 não garante, no entanto, que o método pode ser utilizado no diagnóstico de função hepática. Para tal, é preciso que os três grupos analisados (cirrose, esteatose e esteato-hepatite) possam ser diferenciados entre si pelo valor de T_1 . Isso também se confirma, uma vez que as curvas não se cruzam. Porém, é preciso ressaltar e discutir o grande desvio-padrão observado para todos os pontos. Como há a sobreposição parcial do desvio padrão das três curvas, *a priori* não podemos atribuir valores $\Delta\%T_1$ característicos a cada uma das doenças em análise. Todavia, existem motivos que justificam essa grande dispersão dos dados, como discutido a seguir.

Em nossa análise não fizemos nenhuma correção devido a incertezas do ângulo de flip, o que tem influência considerável na precisão quantitativa do método VFA, como visto na seção 4.4.2. Para realizar a correção, tivemos de assumir o mesmo valor da razão do sinal para o tecido muscular em todos as imagens analisadas. Tal hipótese já levará a erros quantitativos no resultado por alguns fatores. Primeiro, não sabemos exatamente qual o T_1 do músculo paravertebral de cada paciente e esse valor pode variar consideravelmente entre os mesmos. Isto, somado a incertezas no ângulo de flip, terá considerável influência

quantitativa no processamento e, portanto, espera-se que T_1 varie mais que o esperado em imagens de diferentes pacientes para um mesmo tempo pós-contraste, ou até mesmo para um mesmo paciente nas diferentes aquisições.

Um outro problema quantitativo encontra-se na amostragem. Como evidenciado na seção 4.5, o número de pacientes não está bem distribuído entre os 3 grupos sob análise: cirrose (15 pacientes), esteatose (10 pacientes) e esteato-hepatite (41 pacientes). Além disso, há diversas sub-divisões nos grupos esteatose e esteato-hepatite, nas quais os pacientes também não estão uniformemente distribuídos. Todos esses fatores influenciam na distribuição de valores de T_1 e, naturalmente, na variação percentual deste ao longo do tempo.

Por fim, temos ainda possíveis inconsistências entre os exames histopatológicos (que nos permitiram dividir os pacientes nos 3 grupos citados) e as regiões do fígado utilizadas no processamento. Nestes exames, é feita a biópsia de uma pequena amostra do fígado do paciente. Essa amostra não necessariamente será da mesma região selecionada para o processamento. Como existem variações no estado de saúde do fígado dentro do próprio órgão, podemos esperar valores distintos de T_1 num mesmo paciente, o que naturalmente resultará num maior desvio-padrão.

Em suma, os resultados mostram que o método desenvolvido pode, de fato, auxiliar no diagnóstico de função hepática. A menor precisão quantitativa obtida até aqui é justificável devido às limitações do processamento e da amostragem. No entanto, tais limitações podem ser contornadas no futuro do projeto, ao custo de se aumentar a complexidade do procedimento como um todo. Ademais, a Interface Gráfica desenvolvida para este trabalho pode ser aprimorada, a fim de permitir que a normalização dos dados pelo tecido muscular paravertebral ocorra dentro da própria Interface. Com isso, teremos uma GUI funcional que poderá fornecer um pré-diagnóstico imediatamente após o exame de Ressonância Magnética.

7 Conclusão

Este trabalho teve como objetivo inicial a implementação de um método de processamento de imagens de Ressonância Magnética para diagnósticos hepáticos. O plano inicial consistia em criar uma Interface Gráfica de processamento que pudesse ser utilizada pelo usuário imediatamente após a aquisição das imagens e fornecer um diagnóstico rápido e eficaz.

Foram encontrados alguns problemas durante o processamento utilizando a Interface Gráfica. A fim de esclarecer a origem de tais problemas, realizamos diversas simulações, análises e uma extensa pesquisa literária. A conclusão foi que o método em si era funcional, mas havia um problema inerente ao processo de aquisição. Conseguimos contornar as dificuldades e obter resultados satisfatórios ao realizar uma normalização dos dados. Para isso, foi necessária a hipótese de que o tempo de relaxação longitudinal do tecido muscular paravertebral de todos os pacientes era o mesmo.

O resultado final do processamento careceu de precisão quantitativa devido a suposição feita e a demais fatores de aquisição e amostragem. No entanto, os gráficos obtidos para a variação percentual de T_1 ao longo do tempo sustentam, ainda que qualitativamente, a tese de que o método VFA pode ser utilizado de maneira eficaz no diagnóstico de função hepática. Com a continuidade do projeto, podemos aprimorar a precisão numérica do processamento e complementar as funcionalidades da Interface Gráfica. Se bem sucedidas, acreditamos que o método fornecerá uma forma rápida e eficaz de diagnóstico, podendo até mesmo dispensar diagnósticos histopatológicos no futuro.

8 Avaliação do Orientador

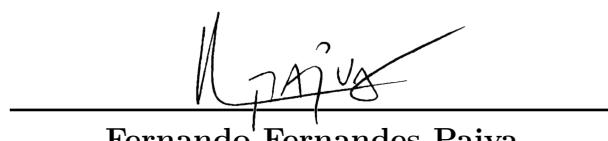
Este trabalho apresenta a implementação de uma ferramenta que possibilita o processamento clínico de dados de imagens por Ressonância Magnética visando a obtenção dos tempos de relaxação longitudinal T1. Esse método encontra aplicação no diagnóstico de doenças hepáticas, como cirrose, fibrose, esteatose e esteato-hepatite. O texto produzido pelo aluno está bem escrito, claro e descreve muito bem as motivações, as etapas da realização e os resultados do trabalho. Além disso, fica claro o rigor científico adotado pelo mesmo nas diversas etapas do trabalho, o que valoriza o trabalho e dá solidez aos resultados apresentados.

Quando de sua proposição, esse projeto possuía um desafio grande do ponto de vista de processamento de imagens, uma vez que os dados haviam sido obtidos em duas aquisições distintas. Isso deixa os dados susceptíveis aos ajustes realizados automaticamente pelo equipamento de imagens durante a aquisição, o que impõe uma dificuldade para a quantificação. Além disso, pretendíamos ter uma interface gráfica que fosse simples e amigável, permitindo sua utilização por médicos radiologistas em rotina clínica.

Esses desafios foram enfrentados pelo aluno de maneira profissional e com competência e dedicação que merecem destaque. O aluno deu início ao projeto sem conhecimento algum sobre o fenômeno de Ressonância Magnética e, ao longo desse período, foi construindo uma base sólida e destacável quando comparado a alunos de graduação. Esses conhecimentos, descritos de maneira rigorosa e didática na seção introdutória do presente texto, foram cruciais para o desenvolvimento do projeto e para a participação ativa do aluno nas discussões sobre as alternativas para tentar contornar o problema gerado pela necessidade de duas aquisições.

Do ponto de vista dos resultados, é importante ressaltar a importância clínica que os mesmos apresentam já no corrente estágio. Tanto o método de processamento desenvolvido e implementado aqui, quanto a ferramenta que será disponibilizada, viabilizarão a próxima etapa do trabalho que se refere à avaliação clínica da extensão da aplicabilidade desse método não invasivo para avaliação de doenças hepáticas.

Com base nessas considerações, entendo que o trabalho apresentado possui mérito e relevância que vai além do esperado para a fase que o aluno se encontra e, portanto, aprovo-o sem restrições, atribuindo nota 10,0 para o mesmo.



Fernando Fernandes Paiva
Orientador

Referências

- Anaconda Software Distribution. *Anaconda*. 2018. Disponível em: <<https://anaconda.com>>. Citado na página 59.
- BAZELAIRE, C. M. J. de et al. Mr imaging relaxation times of abdominal and pelvic tissues measured in vivo at 3.0 t: Preliminary results. *Radiology*, v. 230, n. 3, p. 652–659, 2004. PMID: 14990831. Disponível em: <<https://doi.org/10.1148/radiol.2303021331>>. Citado na página 65.
- BEDOSSA, P.; CARRAT, F. Liver biopsy: The best, not the gold standard. *Journal of Hepatology*, Elsevier, v. 50, n. 1, p. 1–3, 2009. Disponível em: <<https://doi.org/10.1016/j.jhep.2008.10.014>>. Citado na página 7.
- BERNSTEIN, M. A.; KING, K. F.; ZHOU, X. J. Chapter 3 - basic radiofrequency pulse functions. In: BERNSTEIN, M. A.; KING, K. F.; ZHOU, X. J. (Ed.). *Handbook of MRI Pulse Sequences*. Burlington: Academic Press, 2004. p. 67 – 95. ISBN 978-0-12-092861-3. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780120928613500078>>. Citado 2 vezes nas páginas 23 e 46.
- BESA, C. et al. 3d t1 relaxometry pre and post gadoxetic acid injection for the assessment of liver cirrhosis and liver function. *Magnetic Resonance Imaging*, Elsevier, v. 33, n. 9, p. 1075–1082, 2015. Citado 7 vezes nas páginas 7, 51, 59, 61, 64, 66 e 69.
- BROWN, R. et al. *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. [S.l.]: Wiley, 2014. Citado 20 vezes nas páginas 11, 13, 15, 16, 21, 24, 25, 26, 27, 28, 30, 31, 34, 36, 44, 45, 48, 51, 52 e 53.
- CHENG, H.-L. M.; WRIGHT, G. A. Rapid high-resolution t1 mapping by variable flip angles: Accurate and precise measurements in the presence of radiofrequency field inhomogeneity. *Magnetic Resonance in Medicine*, v. 55, n. 3, p. 566–574, 2006. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.20791>>. Citado 3 vezes nas páginas 52, 56 e 57.
- DEONI, S. C.; PETERS, T. M.; RUTT, B. K. Determination of optimal angles for variable nutation proton magnetic spin-lattice, t1, and spin-spin, t2, relaxation times measurement. *Magnetic Resonance in Medicine*, v. 51, n. 1, p. 194–199, 2004. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.10661>>. Citado 2 vezes nas páginas 52 e 56.
- DEONI, S. C.; RUTT, B. K.; PETERS, T. M. Rapid combined t1 and t2 mapping using gradient recalled acquisition in the steady state. *Magnetic Resonance in Medicine*, v. 49, n. 3, p. 515–526, 2003. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.10407>>. Citado na página 56.
- DING, Y. et al. Assessing liver function in patients with hbv-related hcc: a comparison of t1 mapping on gd-eob-dtpa-enhanced mr imaging with dwi. *European Radiology*, v. 25, n. 5, p. 1392–1398, May 2015. ISSN 1432-1084. Disponível em: <<https://doi.org/10.1007/s00330-014-3542-x>>. Citado na página 51.

FRAM, E. K. et al. Rapid calculation of t1 using variable flip angle gradient refocused imaging. *Magnetic Resonance Imaging*, v. 5, n. 3, p. 201 – 208, 1987. ISSN 0730-725X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0730725X8790021X>>. Citado na página 52.

GRIFFITHS, D. J. *Introduction to Quantum Mechanics*. 2. ed. [S.l.]: Pearson Prentice Hall, 2005. (Pearson International Edition). Citado na página 19.

GRIFFITHS, D. J. *Introduction to Electrodynamics*. 4. ed. [S.l.]: PHI Learning Private Limited, 2014. Citado 2 vezes nas páginas 12 e 13.

GUPTA, R. K. A new look at the method of variable nutation angle for the measurement of spin-lattice relaxation times using fourier transform nmr. *Journal of Magnetic Resonance (1969)*, v. 25, n. 1, p. 231 – 235, 1977. ISSN 0022-2364. Disponível em: <<http://www.sciencedirect.com/science/article/pii/002223647790138X>>. Citado 2 vezes nas páginas 52 e 53.

HAASE, A. et al. Flash imaging. rapid nmr imaging using low flip-angle pulses. *Journal of Magnetic Resonance (1969)*, v. 67, n. 2, p. 258 – 266, 1986. ISSN 0022-2364. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0022236486904336>>. Citado na página 51.

HAIMERL, M. et al. Assessment of clinical signs of liver cirrhosis using t1 mapping on gd-eob-dtpa-enhanced 3t mri. *PLoS ONE*, v. 8, n. 12, p. e85658, 2013. Citado 3 vezes nas páginas 7, 51 e 66.

HAMMER, M. MRI Physics: Pulse Sequences. 2013. <<http://xrayphysics.com/sequences.html>>. Citado na página 47.

HEULE, R.; BIERI, O. Rapid and robust variable flip angle t1 mapping using interleaved two-dimensional multislice spoiled gradient echo imaging. *Magnetic Resonance in Medicine*, v. 77, n. 4, p. 1606–1611, 2017. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26246>>. Citado na página 52.

Information on MRI Technique. 2005. Disponível em: <<http://www.nevusnetwork.org/mritech.htm>>. Acesso em: 21 de Outubro 2017. Citado na página 19.

KATSUBE, T. et al. Estimation of liver function using t1 mapping on gd-eob-dtpa-enhanced magnetic resonance imaging. *Investigative Radiology*, v. 46, n. 4, 2011. Citado 5 vezes nas páginas 7, 51, 61, 66 e 69.

LI, Z. et al. Assessment of liver fibrosis by variable flip angle t1 mapping at 3.0t. *Journal of Magnetic Resonance Imaging*, v. 43, n. 3, p. 698–703, 2016. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.25030>>. Citado na página 52.

LOECHER, M.; WIEBEN, O. k-space. In: _____. *Basic Principles of Cardiovascular MRI: Physics and Imaging Technique*. Cham: Springer International Publishing, 2015. p. 13–23. ISBN 978-3-319-22141-0. Disponível em: <https://doi.org/10.1007/978-3-319-22141-0_2>. Citado na página 42.

National Institutes of Health Center for Information Technology (CIT). *MIPAV*. 2017. Disponível em: <<https://mipav.cit.nih.gov/>>. Citado na página 61.

PARKER, G. J.; BARKER, G. J.; TOFTS, P. S. Accurate multislice gradient echo t1 measurement in the presence of non-ideal rf pulse shape and rf field nonuniformity. *Magnetic Resonance in Medicine*, v. 45, n. 5, p. 838–845, 2001. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.1112>>. Citado 4 vezes nas páginas 52, 54, 56 e 58.

VLAARDINGERBROEK, M. T.; BOER, J. A. *Magnetic Resonance Imaging. Theory and Practice*. 3. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2003. Citado na página 11.

WANG, H. Z.; RIEDERER, S. J.; LEE, J. N. Optimizing the precision in t1 relaxation estimation using limited flip angles. *Magnetic Resonance in Medicine*, Wiley Subscription Services, Inc., A Wiley Company, v. 5, n. 5, p. 399–416, 1987. ISSN 1522-2594. Disponível em: <<http://dx.doi.org/10.1002/mrm.1910050502>>. Citado 2 vezes nas páginas 52 e 56.

Apêndices

APÊNDICE A – Resolução analítica da TF do sinal

A fim de entender melhor picos obtidos na figura 18, resolvemos a seguir a transformada de Fourier de $e^{-|t|/T}$ analiticamente. Para realizar essa análise, é necessário generalizar a função, incluindo também os valores negativos de t , fato esse indiferente para as medidas experimentais.

$$F = \int_{-\infty}^{\infty} M_0 e^{-|t|/T} e^{-i2\pi ft} dt \quad (\text{A.1})$$

Dividimos essa integral em duas

$$F = \int_{-\infty}^0 M_0 e^{t/T} e^{-i2\pi ft} dt + \int_0^{\infty} M_0 e^{-t/T} e^{-i2\pi ft} dt \quad (\text{A.2})$$

e fazemos uma mudança de variável $t = -u$, $dt = -du$ e invertemos os limites de integração na primeira integral.

$$F = - \int_0^{\infty} M_0 e^{-u/T} e^{i2\pi fu} - du + \int_0^{\infty} M_0 e^{-t/T} e^{-i2\pi ft} dt \quad (\text{A.3})$$

Expandindo a exponencial imaginária em senos e cossenos:

$$F = \int_0^{\infty} M_0 e^{-u/T} (\cos(2\pi fu) + i \sin(2\pi fu)) du + \int_0^{\infty} M_0 e^{-t/T} (\cos(2\pi ft) - i \sin(2\pi ft)) dt \quad (\text{A.4})$$

Como as duas integrais possuem o mesmo limite de integração podemos juntá-las, substituindo a variável muda u por t :

$$F = \int_0^{\infty} M_0 e^{-t/T} (\cos(2\pi ft) - i \sin(2\pi ft)) + M_0 e^{-t/T} (\cos(2\pi ft) + i \sin(2\pi ft)) dt \quad (\text{A.5})$$

Simplificando, obtemos

$$F = 2M_0 \int_0^{\infty} e^{-t/T} \cos(2\pi ft) dt \quad (\text{A.6})$$

Realizamos um integração por partes para obtermos:

$$F = 2M_0 \frac{T}{1 + \omega^2 T^2} , \quad \omega = 2\pi f \quad (\text{A.7})$$

que é a função Lorentziana. De maneira mais geral, podemos reescrever tal função centrada na frequência ω_0

$$F = 2M_0 \frac{T}{1 + (\omega - \omega_0)^2 T^2} \quad (\text{A.8})$$

cujo máximo (MAX) ocorre para $\omega = \omega_0$:

$$\text{MAX} = 2M_0 T \quad (\text{A.9})$$

com largura a meia altura (FWHM)

$$\text{FWHM} = \frac{2}{T} \quad (\text{A.10})$$

As equações A.9 e A.10 comprovam o comportamento observado. Ao aumentar a magnetização M_0 a amplitude cresce, enquanto que ao aumentar T cresce a amplitude mas diminui-se a largura.

Podemos realizar ainda outra análise na equação A.8. Primeiramente, utilizamos do seguinte resultado:

$$\int_{-\infty}^{+\infty} \frac{1}{1+x^2} dx = \tan^{-1}(x) \Big|_{-\infty}^{+\infty} = \pi \quad (\text{A.11})$$

Realizando o procedimento análogo para a equação A.8, obtemos

$$I = 2M_0 T \int_{-\infty}^{+\infty} \frac{1}{1 + ((\omega - \omega_0)T)^2} dt = 2M_0 T * \frac{\pi}{T} \quad (\text{A.12})$$

Portanto, obtemos que a integral em todo o domínio é apenas proporcional à magnetização:

$$I = 2M_0 \pi \quad (\text{A.13})$$

APÊNDICE B – Código da Simulação em Python

```

1  # -*- coding: utf-8 -*-
3 """
4     Created on Thu Sep 21 12:10:29 2017
5
6     @author: Yuri Tonin
7 """
8
9 import numpy as np
10 import matplotlib.pyplot as plt
11 from lmfit import minimize, Parameters # lmfit is the package for fitting
12
13 #from matplotlib import rc
14 #rc('font',**{'family':'sans-serif','serif':['Computer Modern Roman']})
15 #rc('text', usetex=True)
16
17 """
18     This programm calculates the two points of the linearization method for
19     many
20     runs with the same T1 value but with different noise. The Ernst equation is
21     plotted
22     without any noise on top of all the equations+random_noise. A second plot
23     shows the distribution
24     of the pair of points of the linearization method in the same plot.
25     ...
26
27 class main():
28
29     def __init__(self):
30
31         self.n_points = 180 #number of angles
32         self.theta = np.linspace(1, self.n_points, self.n_points) #array with
33         angle values
34         self.sin_theta = np.sin(self.theta*np.pi/180) #array with sine of
35         angles
36         self.tan_theta = np.tan(self.theta*np.pi/180) #array with tangent
37         of angles
38
39         self.index0 = 1 #index to select first angle from "theta" array
40         self.index1 = 9 #index to select second angle

```

```

35
36     self.TR = 3.789*10**-3
37     self.TE = 1.813*10**-3
38     self.T2 = 72*10**-3
39     self.T1 = 613.9*10**-3 # Besa values: T1 = 613.9 ms (pre) and T1 =
40     200.3 ms (post)

41     def residual(self, params, x, data):
42         # Parameters to be fitted must be declared below
43         amplitude = params['amplitude']
44         T1 = params['T1']
45         # T2 = params['T2']
46         model = self.model_equation(amplitude, self.TE, self.TR, T1, self.T2, x)
47         return (data - model)

48
49     def model_equation(self, a, TE, TR, T1, T2, x): # x is the flip angle
50         E1 = np.exp(-TR/T1)
51         b = a*(1-E1)#*np.exp(-TE/T2)
52         return E1*x+b

53
54     def calculations(self):
55
56         SNR_list = []
57         self.difference_list = []

58         self.X0_array = np.empty(0) #declaring empty arrays to save (x,y)
59         values of each point
60         self.X1_array = np.empty(0)
61         self.Y0_array = np.empty(0)
62         self.Y1_array = np.empty(0)

63
64         for j in range(5,30): #vary SNR
65             for i in range(1,30): #multiple runs with same T1
66                 print('SNR = {0:.0f}; Run #{1:0.0f}'.format(j, i))

67
68                 E1 = np.exp(-self.TR / self.T1)
69                 thetaErnst = np.arccos(E1)*180/np.pi

70
71                 print('\n===== // ===== // ===== // ===== // ===== //')
72                 print('T1 = {0:.2e}'.format(self.T1))
73                 print('TR = {0:.2e}'.format(self.TR))
74                 print('Ernst_angle = {0:.2f}'.format(thetaErnst))

75
76                 rho0 = 100
77                 rho = rho0 * np.sin(self.theta*np.pi/180)*(1-E1)*np.exp(-
78                     self.TE / self.T2)/(1-E1*np.cos(self.theta*np.pi/180))

```

```

79             rho_without_noise = rho
# rho2 is a second 'ideal' Ernst curve, in case you want to compare two
# different ones
81             T12 = 1000*10**-3
82             E12 = np.exp(-self.TR/T12)
83             rho2 = rho0 * np.sin(self.theta*np.pi/180)*(1-E12)*np.exp(-
84                 self.TE/self.T2)/(1-E12*np.cos(self.theta*np.pi/180))
# == # == # == # == # == # == # == # == # == # == #
# == #
85
#   Find best angles according to Deoni 2003:
86             maximum = npamax(rho)
87             rho_at_best_angles = 0.71*maximum
88             find_rho = rho
89             best_angle = (np.abs(find_rho-rho_at_best_angles)).argmin()
# returns the minimum values along the axis
90             find_rho = np.delete(find_rho,best_angle) #delete first
best angle, so the function can find the next with the subsequent .
argmin() command
91             for i in range(1,2):
92                 find_rho = np.delete(find_rho,best_angle+i)
#                 find_rho = np.delete(find_rho,best_angle-i)
93             best_angle2 = (np.abs(find_rho-rho_at_best_angles)).argmin()
()
94             print('Ideal angles are: {0:.0f} and {1:.0f}\n'.format(
95                 best_angle2,best_angle))
# == # == # == # == # == # == # == # == # == #
# == #

96
97             SNR = j
98             SNR_list.append(SNR)
100            StDev = rho[self.index0]/SNR #defining standard deviation
101            noise = np.random.normal(0,StDev,self.n_points) #noise is a
gaussian distribution
102            rho = rho + np.abs(noise) #signal with noise
#             print('\nNoise is at most the Standard Deviation = {0:.5f
103 }'.format(StDev))

104            rho_sin = np.divide(rho, self.sin_theta) #signal/sine for
linearization method
105            rho_tan = np.divide(rho, self.tan_theta) #signal/tan for
linearization method

106            Linearization Y = alpha*X + beta for theta = 2 and 10 degrees
107            Y = np.empty(2)
108            X = np.empty(2)

```

```

113         Y[0] = rho[ self .index0]/ self .sin_theta[ self .index0]
114         Y[1] = rho[ self .index1]/ self .sin_theta[ self .index1]
115         X[0] = rho[ self .index0]/ self .tan_theta[ self .index0]
116         X[1] = rho[ self .index1]/ self .tan_theta[ self .index1]

117         self .X0_array = np.append( self .X0_array,X[0])
118         self .X1_array = np.append( self .X1_array,X[1])
119         self .Y0_array = np.append( self .Y0_array,Y[0])
120         self .Y1_array = np.append( self .Y1_array,Y[1])

121         coef_angular = (Y[0]-Y[1])/(X[0]-X[1]) #manual calculation
122         of angular coefficient of the line that connects pair of points (X0,Y0)
123         and (X1,Y1)
124         #
125         print( '\nHand angular coef = {0:.6e}'.format(coef_angular))
126         )
127         if coef_angular < 0:
128             print( '\nA negative angular coefficient was calculated!
129             This indicates noise that is too high!')
130             T1_hand = - self .TR/np.log(coef_angular)

# Linear fit
131         params = Parameters()
132         params.add('amplitude', value=100) #value is the initial
133         value for fitting
134         params.add('T1', value = 0.05, min = 0, max=2)

135         fitting = minimize(self .residual, params, args=(X, Y))

136         if fitting.success == False:
137             print( '\nFitting was NOT successful')
138         else:
139             #
140             print( '\nFitting was successful:')

self .fitted_amplitude = fitting.params['amplitude'].value
141             self .fitted_T1 = fitting.params['T1'].value
142             #
143             print( '    Fitted Amplitude = {0:.2e}'.format(self .
144             fitted_amplitude))
145             print( '    Fitted T1      = {0:.2e}'.format(self .
146             fitted_T1))
147             #
148             print( '    Hand T1      = {0:.2e}'.format(T1_hand))
149             )

if np.abs(self .fitted_T1) > 100:
150     Fitting does not converge sometimes. It gives |T1| >
151     1000. We are desconsidering these points in this conditional.

```

```

                print('Fitting result in a value too high for T1
(>100). This point will be disregarded')
149         del SNR_list[-1] #removes last element of list
150         continue          #skips current iteration

151         difference = np.abs(self.fitted_T1 - self.T1)/self.T1 #
152         percentual difference between T1 from ideal and from noisy signal
153         self.difference_list.append(difference*100)
154         print('Difference between fitted and simulated T1 is
{0:.5f}%'.format(difference*100))

155 # Following declarations are used to plot the fitting line between
156 # points ahead
157         self.fit_x_points = np.linspace(X[0],X[1],1000)
158         self.fitted_plot = self.model_equation(self.
159         fitted_amplitude, self.TE, self.TR, self.fitted_T1, self.T2, self.
160         fit_x_points)

161 #PLOT OF ERNST EQUATION WITHOUT NOISE ON TOP OF ERNST EQUATIONS WITH NOISE
162         a = plt.figure(4)
163         graph = plt.axhline(y=rho_at_best_angles, color='y',
164         linestyle='-',label ='71% of A maximum')
165         graph = plt.axvline(x=thetaErnst, color='blue',
166         linestyle='--')
167         graph = plt.plot(self.theta,rho, label='Signal A +
168         noise 'r'$\sigma$', linewidth=2)
169         graph = plt.plot(self.theta,rho_without_noise, label=
170         'Signal A', linewidth=3,color='yellow')
171         graph = plt.plot(self.theta,rho2, label='Signal B',
172         linewidth=2)
173         plt.xlabel('Ángulo de Flip 'r'$\theta$ [Degrees]')
174         plt.ylabel('Sinal')
175         plt.legend()
176         if j == 29 & i==29: plt.savefig('signaltheta.png',dpi
177         =600)

178 #LINEARIZATION POINTS FOR ALL THE ERNST EQUATIONS THAT WERE CALCULATED
179         plt.figure(5)
180         graph1 = plt.plot(rho_tan, self.fitted_plot)
181         graph1 = plt.plot(X,Y,'o')
182         graph1 = plt.plot(self.fit_x_points, self.fitted_plot)#
183         label ='Reta de ajuste')
184         graph1 = plt.plot(X,Y,'o')#,label='Pontos p/ 'r'$\theta_1 = 2$ e $\theta_2 = 10$')
185         plt.xlabel('Sinal / tangente')
186         plt.ylabel('Sinal / seno')
187         plt.legend()

```

```
181             if j == 29 & i==29: plt.savefig('linear2.png',dpi=600)

183 # PERCENTUAL VARIATION OF T1 (COMPARING TO IDEAL) vs SNR
185     plt.figure(1)
186     plt.axvline(x=15, color='g', linestyle='--')
187     #plt.axhline(y=50, color='black', linestyle=':')
188     #plt.axhline(y=100, color='red', linestyle=':')
189     plt.plot(SNR_list, self.difference_list, 'o', c='#ED8F19',
190             markeredgecolor='black', markeredgewidth=0.5)
191     plt.xlabel('SNR', fontsize=16)
192     plt.ylabel(r'$\Delta \% T_1$', fontsize=16)
193     #plt.ylim(0,120)
194     plt.title(r'$\theta_1 = {0:.0f} \quad \theta_2 = {1:.0f} \quad T_R = {2:.3f} ms'.format(self.theta[self.index0], self.
195         theta[self.index1], self.TR*1000), fontsize=16)
196     if j == 29 & i==29: plt.savefig('uncertainty.png', dpi
197 =600)

195
196 a = main()
197 a.calculations()
```

APÊNDICE C – Código da Interface Gráfica em Python

```

2 import matplotlib # these 3 lines need to exist in this way, otherwise
      #matplotlib crashes with tkinter
3 matplotlib.use("TkAgg") #backend of matplotlib
4
5 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
      NavigationToolbar2TkAgg
6 from matplotlib.figure import Figure
7 from matplotlib import pyplot as plt
8
9 import dicom
10 import os
11 from os import listdir
12 from os.path import isfile , join
13
14 from tkinter import *
15 from tkinter import ttk
16 from PIL import Image, ImageTk
17
18 from tkinter import filedialog
19
20 import numpy as np
21 # from scipy import optimize
22 from lmfit import minimize, Parameters # lmfit is the package for fitting
23
24
25 # ATTENTION: This program was first implemented for an analysis of slice
26 #             numbers and echo times. Only later it was
27 # switched to analyze slice number and flip angle. Therefore, lost of
28 #             arrays and variables contain "echo" in their
29 #             names, but they actually posses FLIP ANGLE values!
30
31 class SecondaryFrames(Frame):
32
33     # This class declares frames to divide de main window in regions for
34     # different widgets
35
36     def __init__(self, parent):
37
38         self.top_frame = Frame(parent)

```

```

36     self.left_frame = Frame(parent, width=400)
37     self.right_frame = Frame(parent, width=500)
38     self.right_frame2 = Frame(parent)

40     self.horizontal_line = ttk.Separator(parent, orient=HORIZONTAL)
41     self.vertical_line = ttk.Separator(parent, orient=VERTICAL)
42

44 class SearchFolder(Frame):

46     # This class is responsible for selecting the folders where acquisition
47     # files area and, after that is done, placing
48     # the scales and canvas used for selecting a specific image and drawing
49     # a region of interest (ROI) on it. The last
50     # action performed by this class is to select the ROI through a button,
51     # so that data is exported to a plot.

52     def __init__(self, parent, left_frame, right_frame, right_frame2, first_run,
53                  first_plot):
54
55         self.first_run      = first_run
56         self.first_plot     = first_plot
57         self.parent         = parent
58         self.left_frame     = left_frame
59         self.right_frame    = right_frame
60         self.right_frame2   = right_frame2

61
62         self.entry_frame     = Frame(self.parent)
63         self.label_acquisition = Label(self.entry_frame, text='— Insira
64                                         na caixa o número de aquisições a serem importadas na análise')
65         self.acquisition_number = IntVar()
66         self.acquisition_number = Entry(self.entry_frame, width=11, justify=CENTER)
67
68         self.acquisition_number.delete(0, END)
69         self.acquisition_number.insert(0, '2')
70         self.acquisition_number.pack(padx=2, pady=1, anchor="w", side=LEFT)
71         self.label_acquisition.pack(padx=2, pady=1, anchor="w", side=LEFT)

72
73         self.search_button = Button(self.parent, text="Buscar pasta",
74                                     command=self.get_images) # se colocar search_folder() - com () - ele
75                                     # chama a função automaticamente!
76         self.path_box      = Label(self.parent, text='— Selecione a pasta
77                                     com as imagens DICOM', relief=FLAT)

78
79         # self.dirname = ''

```

```

74     def atoi(self, text):
75         return int(text) if text.isdigit() else text
76
77     def natural_keys(self, text):
78         """
79             alist.sort(key=natural_keys) sorts in human order
80             http://nedbatchelder.com/blog/200712/human_sorting.html
81             (See Toothy's implementation in the comments)
82         """
83
84         return [self atoi(c) for c in re.split('(\d+)', text)]
85
86     def get_images(self):
87
88         # self.dirname          = [] #comment this line if you insert
89         # folder path by hand in the conditional "if j == 0" a few lines below
90
91         self.dcm_folder        = []
92         self.dcm_files         = []
93         self.slice_thickness   = []
94         self.echo_time         = []
95         self.repetition_time  = []
96         self.slice_number      = []
97         self.flip_angle        = []
98         self.rescale_intercept = []
99         self.rescale_slope     = []
100        self.slice_and_flip    = []
101        self.FOVx              = []
102        self.FOVy              = []
103        self.acquisition_time = []
104        self.gradient_echo_train_length = []
105        self.slice_and_echo    = np.array([], dtype=np.int64).reshape(0,2)
106
107
108        for j in range(0,np.int(self.acquisition_number.get())):
109            pass
110
111            if self.first_run == False:
112                # Destroys all widgets inside of self.left_frame if this is
113                # not the first time the widget is created.
114                # This is done so that the widget can be "refreshed"
115                # without a new one appearing on the screen
116                for child in self.left_frame.winfo_children():
117                    child.destroy()
118
119
120            # if j == 0:
121            # A conditional for selecting all the folders. This is only
122            # needed if you want to declare by hand

```

```

116      # the folder with the dicom files. If you want the interface to
117      # search the folder to appear when you click
118      # the "Search Folder" button, simply remove this conditional
119      # leaving only the command self.dirname.append(filedialog.askdirectory(
120      # parent=root, initialdir="/", title='Selecione uma pasta'))
121      # self.dirname.append(filedialog.askdirectory(parent=root,
122      # initialdir="/", title='Selecione uma pasta'))
123      # self.dirname = '/Users/yurir.tonin/Dropbox/TCC/DICOM/
124      # Dados/PAC001/901_AXI FLIP 2/DICOM'
125      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/PAC002
126      # /901_AXI FLIP 2/DICOM'
127      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
128      # Pacientes/SUBJ001/901_AXI FLIP 2/DICOM'
129      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
130      # Pacientes/SUBJ002/1201_AXI FLIP 2/DICOM'
131      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
132      # Pacientes/SUBJ003/801_AXI FLIP 2/DICOM'
133      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
134      # Pacientes/SUBJ004/1201_AXI FLIP 2/DICOM'
135      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
136      # Pacientes/SUBJ005/1301-MR1_AXI FLIP 2/DICOM'
137      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
138      # Pacientes/SUBJ006/1001_AXI FLIP 2/DICOM'

139
140      # else:
141      # self.dirname.append(filedialog.askdirectory(parent=root,
142      # initialdir="/", title='Selecione uma pasta'))
143      # self.dirname = '/Users/yurir.tonin/Dropbox/TCC/DICOM/
144      # Dados/PAC001/1001_AXI FLIP 10/DICOM'
145      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/PAC002
146      # /1001_AXI FLIP 10/DICOM'
147      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
148      # Pacientes/SUBJ001/1001_AXI FLIP 10/DICOM'
149      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
150      # Pacientes/SUBJ002/1301_AXI FLIP 10/DICOM'
151      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
152      # Pacientes/SUBJ003/901_AXI FLIP 10/DICOM'
153      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
154      # Pacientes/SUBJ004/1301_AXI FLIP 10/DICOM'
155      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
156      # Pacientes/SUBJ005/1401-MR1_AXI FLIP 10/DICOM'
157      # self.dirname = 'C:/ Users/Yuri Tonin/Desktop/Dados/
158      # Pacientes/SUBJ006/1101_AXI FLIP 10/DICOM'

159
160      self.dirname = (filedialog.askdirectory(parent=root, initialdir=
161      # "/",
162      # title='Selecione uma pasta'))
```

```

    print( self . dirname )
142
        self . path_box . configure( text='Diretório: {0:s}' . format( self .
dirname )) #Changes the name of the label to show the last selected
        folder
144
            self . dcm_folder . append( self . dirname ) # Saves the folder path
with dcm files to a list

146
            self . dcm_files . append( [ f for f in listdir( self . dcm_folder [ j ]) ]
if isfile( join( self . dcm_folder [ j ] , f ))) # create list with the names of
the files in the current selected folder
            self . dcm_files [ j ]. sort( key= self . natural_keys ) # order files by
their number in file name
148
            # Declare empty arrays to import parameters later
150
            self . slice_thickness . append( np . zeros( len( self . dcm_files [ j ])))
            self . echo_time . append( np . zeros( len( self . dcm_files [ j ])))
152
            self . repetition_time . append( np . zeros( len( self . dcm_files [ j ])))
            self . gradient_echo_train_length . append( np . zeros( len( self .
dcm_files [ j ])))
154
            self . slice_number . append( np . zeros( len( self . dcm_files [ j ])))
            self . rescale_slope . append( np . zeros( len( self . dcm_files [ j ])))
156
            self . rescale_intercept . append( np . zeros( len( self . dcm_files [ j ])))
        )
158
            self . flip_angle . append( np . zeros( len( self . dcm_files [ j ])))
            self . acquisition_time . append( np . zeros( len( self . dcm_files [ j ])))

160
            # Read parameters from all files and store them in their
respective arrays
162
            for i in range( len( self . dcm_files [ j ])):
                file_path = os . path . join( self . dcm_folder [ j ] , self . dcm_files
[j ][ i ])
164
                #
=====

166
                # Read file and tags
                #
=====

168
                dcm_read = dicom . read_file( file_path )

170
                self . slice_number [ j ][ i ] = dcm_read [ 0x2001 , 0x100A ]. value
                self . slice_thickness [ j ][ i ] = dcm_read [ 0x18 , 0x50 ]. value
172
                self . echo_time [ j ][ i ] = dcm_read [ 0x18 , 0x81 ]. value

```

```

174         self.rescale_intercept[j][i] = dcm_read[0x28, 0x1052].value
175         self.rescale_slope[j][i] = dcm_read[0x28, 0x1053].value
176     self.repetition_time[j][i] = dcm_read[0x18, 0x80].value
177     self.gradient_echo_train_length[j][i] = dcm_read[0x18, 0x91].value
178     # magnetic_field_strength = dcm_read[0x18, 0x87].value
179     # gradient = dcm_read[0x18, 0x1318].value #dB
180     /dt
181     self.flip_angle[j][i] = dcm_read[0x18, 0x1314].value
182
183     # Single value for all measurements
184     self.rows = dcm_read[0x28, 0x10].value
185     self.columns = dcm_read[0x28, 0x11].value
186     self.pixel_spacing = dcm_read[0x28, 0x30].value
187     self.number_averages = dcm_read[0x18, 0x83].value
188
189     self.acquisition_time[j][i] = dcm_read[0x8, 0x32].value
190
191     self.slice_and_flip.append(np.column_stack((self.slice_number[j][:], self.flip_angle[j][:])))
192
193     self.FOVx.append(self.rows * self.pixel_spacing)
194     self.FOVy.append(self.columns * self.pixel_spacing)
195
196     self.slice_and_echo = np.vstack([self.slice_and_echo, self.
197 slice_and_flip[j]]). #THE NAME "SLICE AND ECHO" IS NOT SUITABLE! IT
198 SHOULD ACTUALLY BE SLICE AND FLIP. MANTAINED IT LIKE THIS BECAUSE OF
199 OLDER IMPLEMENTATION WAS CREATED FOR ECHO TIMES!
200
201     print('Rescale2: A = {0:.2f} B = {1:.1f}'.format(self.
202 rescale_slope[0][0], self.rescale_intercept[0][0]))
203     print('Rescale10: A = {0:.2f} B = {1:.1f}'.format(self.
204 rescale_slope[1][0], self.rescale_intercept[1][0]))
205
206     print(dcm_read.dir()) # All Dicom tag available
207     print(dcm_read[0x8, 0x1090].value) #manufac model name
208     print(dcm_read[0x8, 0x70].value) #manufacturer
209     print(dcm_read[0x8, 0x20].value) #scanning sequence
210
211     # print('Acquisition time = {0:.0f}'.format(self.acquisition_time
212 [0][1]))
213     print('Echo Train Length = {0:.0f}'.format(self.
214 gradient_echo_train_length[0][0]))

```

```

208     self.total_repetition_time = np.multiply(self.repetition_time, self
209       .gradient_echo_train_length)
210     self.total_repetition_time = np.multiply(self.repetition_time,1)
211     # print(self.total_repetition_time)

212     print('Number of averages = {0:.0f}'.format(self.number_averages))

214     self.TR = self.total_repetition_time[0][0]*10**-3 # Values here
215     work just like for echo time
216     self.T1 = 678*10**-3                                # Value in
217     seconds. T1 and T2 values are for 1.5 T
218     self.T2 = 72*10**-3                                # Obtained from
219     https://www.ncbi.nlm.nih.gov/pubmed/22302503
220     self.TE = self.echo_time[0][0]*10**-3                # Value in
221     seconds. Echo time should be the same for all elements of
222     # the array if we
223     are dealing with the case for different flip angles

224     self.Ernst_angle = np.arccos(np.exp(-1*self.TR/self.T1))*180/np.pi
#678ms obtained in literature for healthy liver

225     print('Echo time      = {0:.5e} seconds'.format(self.TE))
226     print('Repetition time = {0:.5e} seconds'.format(self.TR))
227     print('Ernst angle     = {0:.2f} Degrees'.format(self.Ernst_angle))

228     self.dcm_files = [item for sublist in self.dcm_files for item in
229       sublist] # Makes a flat list (remove separation of interior lists,
230       creating one huge list)

231     self.unique_sorted_echoes = np.array(np.unique(self.slice_and_echo
232       [:,1])) #Removing repetitions and ordering values
233     self.unique_sorted_slices = np.array(np.unique(self.slice_and_echo
234       [:,0])) #Removing repetitions and ordering values

235     self.place_scales_and_canvas() # Puts everything on canvas

236     self.first_run = False          # Declares that the first run has
237     already been completed

238   def place_scales_and_canvas(self):
239
240     # Creating scale objects from the class
241     self.slice_number_scale = Scales(self.left_frame,"Slice Number",
242       self.unique_sorted_slices[0],self.unique_sorted_slices[-1])
243     self.echo_time_scale    = Scales(self.left_frame,"Flip angle",0,
244       self.unique_sorted_echoes.size-1)

```

```

#Creating interactive canvas object where images are shown and ROI
selected
242     self.interactive_canvas = InteractiveCanvas(self.left_frame)

244     self.slice_number_scale.scales_frame.pack(anchor='w')
245     self.echo_time_scale.scales_frame.pack(anchor='w')

246     self.interactive_canvas.pack(pady=15)
247

248     self.slice_number_scale.variable_name.trace("w", self.
call_change_image) #When scale is changed, calls self.call_change_image
249     self.echo_time_scale.variable_name.trace("w", self.
call_change_image)

250

251     # Creates plot object to visualize data and fitting
252     self.plot = Plot(self.slice_and_echo, self.first_plot, self.
right_frame, self.right_frame2, self.TE, self.T1, self.T2, self.TR)

253

254     # Creates button to select ROI after it is drawn.
255     self.select_button = Button(self.left_frame, text='Selecionar região',
,
256                               command= lambda: self.plot.ROI_average(
257                                   self.dcm_all_echoes, self.interactive_canvas.get_ROI(),
self.
258                                   echo_time_scale.scale_name.get(),
self.slice_number_scale.scale_name.get()
() )
259

260     self.select_button.pack()

261

262     def call_change_image(self,*args): #need *args to take arguments that
263     are automatically passed by the tracing mechanism

264

265     # Obtains the scale value to that this values can be passed to the
interactive_canvas class, where they will be used
266     # for the displayed image to be changed to an image corresponding to
the new values

267

268     self.scale1_value = self.slice_number_scale.scale_name.get()
269     self.scale2_value = self.echo_time_scale.scale_name.get()

270

271     self.scale1_label = self.slice_number_scale.scale_label
272     self.scale2_label = self.echo_time_scale.scale_label

273

274     self.dcm_all_echoes = self.interactive_canvas.change_image( self.
scale1_value, self.scale2_value, self.scale1_label, self.scale2_label, self.
slice_and_echo
self.dcm_folder, self.dcm_files, self.left_frame, self.
interactive_canvas, self.rescale_intercept, self.rescale_slope)

```

```

274 class Scales(Frame):
275
276 # This class simply declares the characteristics of the Scales used for
277 # selecting images. They are declared here, but
278 # only created (packed) in the GUI at SearchFolder class, inside of
279 # place_scales_and_canvas method.
280
281
282     def __init__(self, parent, label_text, initial_value, final_value, *args):
283         self.parent = parent
284         self.bar_length = 200
285         self.label_text = label_text
286         self.initial_value = initial_value
287         self.final_value = final_value
288         self.variable_name = DoubleVar()
289
290
291         self.scales_frame = Frame(self.parent)
292
293         self.scale_name = Scale(self.scales_frame,
294                                variable=self.variable_name,
295                                orient=HORIZONTAL,
296                                from_=self.initial_value,
297                                to=self.final_value,
298                                length=self.bar_length,
299                                cursor="hand2",
300                                label=self.label_text,
301                                showvalue=0)
302
303
304         self.scale_label = Label(self.scales_frame)##,textvariable=self.
305         variable_name)
306
307         self.scale_name.grid(row=0, column=0, padx=40)
308         self.scale_label.grid(row=0, column=1, padx=5)
309
310
311     def get_scale_value(self):
312         self.variable_name.get()
313
314 class InteractiveCanvas(Frame):
315
316 # In this class all the functionality for displaying the proper image on
317 # canvas and drawing the ROI on top of the imaged
318 # is created. There is also a function that returns the ROI coordinates.
319
320     def __init__(self, parent):
321         Frame.__init__(self, parent)
322         self.x = self.y = 0

```

```
316     self.canvas = Canvas(self, width=300, height = 300, bg="red", cursor="cross")  
  
318     self.sbarv=Scrollbar(self, orient=VERTICAL)  
319     self.sbarh=Scrollbar(self, orient=HORIZONTAL)  
320     self.sbarv.config(command=self.canvas.yview)  
321     self.sbarh.config(command=self.canvas.xview)  
  
322     self.canvas.config(yscrollcommand=self.sbarv.set)  
323     self.canvas.config(xscrollcommand=self.sbarh.set)  
  
326     self.canvas.grid(row=0,column=0,sticky=N+S+E+W)  
327     self.sbarv.grid(row=0,column=1,stick=N+S)  
328     self.sbarh.grid(row=1,column=0,sticky=E+W)  
  
330     self.canvas.bind("<ButtonPress-1>", self.on_button_press)  
331     self.canvas.bind("<B1-Motion>", self.on_move_press)  
332     self.canvas.bind("<ButtonRelease-1>", self.on_button_release)  
  
334     self.rect = None  
  
336     self.start_x = None  
337     self.start_y = None  
  
338     self.im = Image.open('greyscale.png')  
339     self.wazil, self.lard=self.im.size  
340     self.canvas.config(scrollregion=(0,0, self.wazil, self.lard))  
341     self.tk_im = ImageTk.PhotoImage(self.im)  
342     self.image_on_canvas = self.canvas.create_image(0,0, anchor="nw",  
343         image=self.tk_im) #create image on canvas  
  
344     self.coordenadas = [0,0,0,0]  
  
346     def on_button_press(self, event):  
347         # save mouse drag start position  
348         self.start_x = self.canvas.canvasx(event.x)  
349         self.start_y = self.canvas.canvasy(event.y)  
  
352         # create rectangle if not yet exist  
353         if not self.rect:  
354             self.rect = self.canvas.create_rectangle(self.x, self.y, 1, 1,  
355                 outline='yellow')  
  
356     def on_move_press(self, event):  
357         curX = self.canvas.canvasx(event.x)  
358         curY = self.canvas.canvasy(event.y)
```

```

360     w, h = self.canvas.winfo_width(), self.canvas.winfo_height()
361     if event.x > 0.9*w:
362         self.canvas.xview_scroll(1, 'units')
363     elif event.x < 0.1*w:
364         self.canvas.xview_scroll(-1, 'units')
365     if event.y > 0.9*h:
366         self.canvas.yview_scroll(1, 'units')
367     elif event.y < 0.1*h:
368         self.canvas.yview_scroll(-1, 'units')

370     # expand rectangle as you drag the mouse
371     self.canvas.coords(self.rect, self.start_x, self.start_y, curX,
372 curY)
373
374     self.coordenadas = self.canvas.coords(self.rect)
375     print(self.coordenadas)

376 def on_button_release(self, event):
377     pass

378 def change_image(self, user_slice_number, user_echo_time, scale1_label,
379 scale2_label, slice_and_echo, dcm_folder, dcm_files, left_frame,
380 interactive_canvas, rescale_intercept, rescale_slope):
381
382     scale1_label.configure(text=user_slice_number) # Displays value of
383     the slice number
384
385     self.unique_sorted_echoes = np.array(np.unique(slice_and_echo[:,1]))
386     scale2_label.configure(text=self.unique_sorted_echoes[
387 user_echo_time]) # Displays value of the flip angle
388
389     self.user_echo_time_index = user_echo_time # Declares the value
390     from the scale (that may have the same numerical value as the echo) as
391     an index
392     user_echo_time = self.unique_sorted_echoes[self.
393 user_echo_time_index] #Selects echo time from the array of echo time
394     values according to the index declared above
395
396     indexes = np.where(slice_and_echo[:, 0] == user_slice_number)[0] # indexes of the elements where the user input match the slice number
397
398     self.dcm_all_echoes = []
399
400     for i in indexes: # Go through index values. Check and record
401     those in which echo time (element) matches user input

```

```

        #get array of arrays with pixel values for all echoes of a
slice
396         self.which_folder = 0 # used as index to select the right
rescaling for the pixel array
            for n in range(len(dcm_folder)): #the range of elements of
dcm_folder should match the number of different flip angles/echo times
398                 # Will evaluate if the i_th file dcm_files[i] exists for
the folder we have.
                    # It will match for 1 folder only. When it does, isfile
returns True, and we
400                     # record the pixel array for that flip_angle/echo_time in
dcm_all_echoes
                        file_path = os.path.join(dcm_folder[n], dcm_files[i]) #
path of the file whose index match user input
402                     if os.path.isfile(file_path):
                        dcm_read = dicom.read_file(file_path) # read file user
wants
                            # dcm_pixels = (dcm_read.pixel_array -
rescale_intercept[self.user_echo_time_index][0]) / rescale_slope[self.
user_echo_time_index][0]
                            dcm_pixels = (dcm_read.pixel_array - rescale_intercept[
self.which_folder][0]) / rescale_slope[self.which_folder][0]
                            # dcm_pixels = dcm_pixels/np.max(dcm_pixels)
                            # dcm_pixels = dcm_read.pixel_array
408                     self.dcm_all_echoes.append(dcm_pixels) # extract pixel
values

410                     self.which_folder += 1 # used as index to select the
right rescaling for the pixel array

412             if slice_and_echo[i, 1] == user_echo_time: index = i #Finds
index for specific echo_time

414
415             file_path = os.path.join(dcm_folder[self.user_echo_time_index],
dcm_files[index]) # path of the file whose index match user input
416             dcm_read = dicom.read_file(file_path) # read file user wants

418             dcm_pixels = dcm_read.pixel_array      #NOT USING RESCALE HERE
BECAUSE THIS IS USED FOR USER VISUALIZATION ONLY. Rescaling is done only
in the previous for loop, since that data is the one used for plotting/
fitting
419             dcm_pixel_values = dcm_pixels # extract pixel values
420             dcm_image = Image.fromarray(dcm_pixel_values).resize((300, 300)) #
Convert array to image #CHECAR: A SELEÇÃO VAI SER NA IMAGEM ORIGINAL OU
RESIZED? PODE SER QUE A MÉDIA MUDE CASO SEJA FEITA EM UMA OU EM OUTRA!

```

```

#           dcm_image = Image.fromarray(dcm_pixel_values)
# Convert array to image #CHECAR: A SELEÇÃO VAI SER NA IMAGEM ORIGINAL
# OU RESIZED? PODE SER QUE A MÉDIA MUDE CASO SEJA FEITA EM UMA OU EM
# OUTRA!
422         new_tk_image = ImageTk.PhotoImage(dcm_image) # crete tk image
424
        imagem = Label(left_frame, image='') # Create empty image to keep
        a reference of image. That way, GarbageCollector won't throw away image
        (http://effbot.org/pyfaq/why-do-my-tkinter-images-not-place\_interactive\_canvasear.htm)
426         imagem.configure(image=new_tk_image) # Used to keep reference (go
        to imagem declaration to understand)
        imagem.image = new_tk_image # Used to keep reference (go
        to imagem declaration to understand)
428         interactive_canvas.canvas.itemconfig(interactive_canvas.
        image_on_canvas, image=new_tk_image) # update image on canvas
430
        return self.dcm_all_echoes
432     def get_ROI(self):
        return self.coordenadas
434
class Plot():
436
    # Creates the plot. Calculates average of each ROI and performs fitting
    # algorithm
438
    def __init__(self, slice_and_echo, first_plot, right_frame, right_frame2, TE,
                 T1, T2, TR):
440
        self.slice_and_echo = slice_and_echo
442        self.first_plot = first_plot
        self.right_frame = right_frame
444        self.right_frame2 = right_frame2
446
        self.slices_list = []
        self.amplitudes_list = []
448        # self.decay_coefs_list = []
        self.T1_list = []
450
        self.TE = TE
        self.TR = TR
        # self.T1 = T1
454        self.T2 = T2

```

```

456     def ROI_average(self ,dcm_all_echoes ,coordinates ,echo_time_scale_value ,
457                     slice_scale_value):
458
459         self.dcm_all_echoes          = dcm_all_echoes
460         self.coordinates            = coordinates
461         self.echo_time_scale_value = echo_time_scale_value
462         self.slice_scale_value     = slice_scale_value
463
464         self.average    = np.empty(len(self.dcm_all_echoes))
465         self.stdev      = np.empty(len(self.dcm_all_echoes))
466         self.maximo     = np.empty(len(self.dcm_all_echoes))
467
468
469         for i in range(len(self.dcm_all_echoes)): # length o
470             dcm_all_echoes should be the same number of distinct angles we have ,
471             probably 2 angles
472
473             # Calculates the average of the ROI for each echo_time/
474             flip_angle with that slice number
475
476             print('Image shape = {0}'.format(np.shape(self.dcm_all_echoes[i])))
477             # size (224,224) < (300,300)
478
479             # dcm_image = Image.fromarray(self.dcm_all_echoes[i]).resize
480             ((300, 300)) # CAREFUL. SIZE OF THE OPENED IMAGE MUST BE THE SAME AS THE
481             ONE IN THE GUI BECAUSE COORDINATES ARE GIVEN IN THE GUI IMAGE
482             dcm_image = Image.fromarray(self.dcm_all_echoes[i]) # CAREFUL.
483             SIZE OF THE OPENED IMAGE MUST BE THE SAME AS THE ONE IN THE GUI BECAUSE
484             COORDINATES ARE GIVEN IN THE GUI IMAGE
485
486             croped_dcm_image = dcm_image.crop(coordinates)
487             croped_pixel_array = np.array(croped_dcm_image)
488             # croped_pixel_array = croped_pixel_array/np.max(
489             croped_pixel_array) #normalize
490             print('ROI shape = {0}'.format(np.shape(croped_pixel_array)))
491
492             self.maximo[i] = np.max(croped_pixel_array)
493
494             self.average[i] = np.average(croped_pixel_array)
495             self.stdev[i]   = np.std(croped_pixel_array)
496             if i == 0: np.savetxt('croped_pixel_array2.txt',
497             croped_pixel_array,fmt='%.10f')
498             else: np.savetxt('croped_pixel_array10.txt',croped_pixel_array,
499             fmt='%.10f')
500
501             # print(self.average)
502             print(self.average[1]/self.average[0])

```

```

# print(self.echoes)
492    self.echoes = np.array(np.unique(self.slice_and_echo[:,1])) # clean
array to have only unique angle values
        self.average_by_sin = np.divide(self.average,np.sin(self.echoes*np.
pi/180))
        self.average_by_tan = np.divide(self.average,np.tan(self.echoes*np.
pi/180))

496    # teste. testando aproximação do eq de Ernst para o caso theta <<
thetaErnst
        # self.average_by_sin = self.average
498    # self.average_by_tan = np.sin(self.echoes * np.pi / 180)

500    self.create_plot()

502    # print(np.sin(self.echoes*np.pi/180))
      print(self.average)
504    # print(self.echoes)
      # print(np.sin(self.echoes*np.pi/180))
506    # print(np.tan(self.echoes*np.pi/180))
      print('Avg/Sin = {0}'.format(self.average_by_sin))
508    print('Avg/Tan = {0}'.format(self.average_by_tan))

510    # print('Average @ 2 degrees = {0:.2e}'.format(self.average[0]))
      # print('StDev @ 2 degrees = {0:.2e}'.format(self.stdev[0]))
      self.SNR2 = self.average[0]/self.stdev[0]
      # print('SNR_2 = {0:.2f}'.format(self.SNR2))

514    # print('Average @ 10 degrees = {0:.2e}'.format(self.average[1]))
      # print('StDev @ 10 degrees = {0:.2e}'.format(self.stdev[1]))
      self.SNR10 = self.average[1]/self.stdev[1]
      # print('SNR_10 = {0:.2f}'.format(self.SNR10))

520    a=(self.average_by_sin[1] - self.average_by_sin[0]) / (self.
average_by_tan[1] - self.average_by_tan[0])
      print('Angular coeff. = {0:.7f}'.format(a))
522    print('T1_calculated = {0:.3e}\n'.format(-self.TR/np.log(a)))

524    def residual(self,params, x, data):
      # Parameters to be fitted must be declared below
526    amplitude = params['amplitude']
      T1 = params['T1']
528    # T2 = params['T2']

530    model = self.model_equation(amplitude, self.TE, self.TR,T1, self.T2,x)

532    return (data - model)

```

```

534     def model_equation(self, a, TE, TR, T1, T2, x): # x is the flip angle
535         #This is the equation to be fitted. If you change input parameters,
536         #remember to also change them when this method is called
537         E1 = np.exp(-TR/T1)
538         b = a*(1-E1)#*np.exp(-TE/T2)
539         # return a*np.sin(x*np.pi/180)*(1-E1)*np.exp(-TE/T2)/(1-E1*np.cos(x
540         *np.pi/180))
541         # return a*x/(1+0.5*E1*x**2/(1-E1)) #approximation
542         for flip_angle << Ernst angle
543             return E1*x+b
544
545     def create_plot(self):
546
547         if self.first_plot == False:
548             self.the_plot.clear()
549         else:
550             self.plot_figure = Figure(figsize=(10, 10), dpi=100)
551             self.the_plot = self.plot_figure.add_subplot(111)
552
553
554         self.the_plot.plot(self.average_by_tan, self.average_by_sin, 'ro')
555         self.the_plot.get_yaxis().set_major_formatter(plt.LogFormatter(10,
556         labelOnlyBase=False))
557
558         params = Parameters()
559         params.add('amplitude', value=100) #value is the initial value for
560         fitting
561         params.add('T1', value = 0.5)
562
563         fitting = minimize(self.residual, params, args=(self.average_by_tan
564         , self.average_by_sin))
565
566         self.fitted_amplitude = fitting.params['amplitude'].value
567         self.fitted_T1 = fitting.params['T1'].value
568         # self.T2 = fitting.params['T2'].value
569
570         E1 = np.exp(-self.TR/self.fitted_T1)
571         b = self.fitted_amplitude*np.exp(-self.TE/self.T2)*(1-E1)
572         # print('b = {0:.2e}'.format(b))
573         # print('amplitude fitted = {0:.2e}'.format(self.fitted_amplitude))
574         print('\nT1 fitted = {0:.3e}'.format(self.fitted_T1))
575         # print('TR = {0:.2e}'.format(self.TR))
576         # print('E1 = {0:.2e}'.format(E1))

```

```

574     if fitting.success: self.save_fit_params() # if fitting is
succesfull , save the parrameters
576
576     self.fit_x_points = np.linspace(self.average_by_tan[0],self.
average_by_tan[-1],1000)
# self.fit_x_points = np.linspace(0,self.average_by_tan[-1],1000)
578
578     self.fitted_plot = self.model_equation(self.fitted_amplitude, self.
TE, self.TR, self.fitted_T1, self.T2, self.fit_x_points)
580     # print(self.model_equation(self.fitted_amplitude, self.TE, self.TR,
self.fitted_T1, self.T2,0))
582
582     self.the_plot.plot(self.fit_x_points, self.fitted_plot)
584
584     # self.the_plot.set_title('ROI average X Echo Time')
586     self.the_plot.set_xlabel('Média/tangente')
586     self.the_plot.set_ylabel('Média/seno')
588
588     if self.first_plot == False:
590         self.canvas.draw()
590     else:
592         self.canvas = FigureCanvasTkAgg(self.plot_figure, self.
right_frame)
592         self.canvas.show()
594         self.canvas.get_tk_widget().pack()
596
596     self.first_plot = False
598
598 def save_fit_params(self):
600
600     #Save fit params and shows them at a table
602
602     for i in range(len(self.slices_list)): # Guarantees that only 1
line per slice number fit will appear at the table
604     if self.slices_list[i] == self.slice_scale_value:
604         self.repetition_message = "Você refez o fitting de um slice
. A ocorrência anterior foi apagada."
604         print(self.repetition_message)
604         del self.slices_list[i]; del self.T1_list[i]; del self.
amplitudes_list[i] #deletes the previous occurence of that slice number
in the list
608         break
608
610
610     self.slices_list.append(self.slice_scale_value)
610     self.T1_list.append(self.fitted_T1)

```

```

612         self.amplitudes_list.append(self.fitted_amplitude)
614
615         self.table_values = np.column_stack((self.slices_list, self.
616         amplitudes_list, self.T1_list)) # Create a matrix with 3 columns, each
617         column being one of the lists declared above
618         self.table_values = self.table_values[np.lexsort((self.table_values
619         [:, 0],))] #sort by 1st column keeping respective 2nd and 3rd columns
620
621         self.table_shape = self.table_values.shape
622
623         if self.first_plot == False: #Destroys previous table and create a
624         new one with refreshed values
625             for child in self.right_frame2.winfo_children(): # destroys
626             all widgets inside of self.right_frame2
627                 child.destroy()
628
629             t = SimpleTable(self.right_frame2, len(self.slices_list)+1,3)
630             t.pack(side="top", fill="x")
631             t.set(0,0,"Slice"); t.set(0,1,"Amplitude"); t.set
632             (0,2,"T1")
633
634             for m in range(self.table_shape[0]):
635                 for n in range(self.table_shape[1]):
636                     if n == 0: # This conditional is used only for the
637                     formating of the first column numbers to be different from the formating
638                     of the other columns
639                         t.set(m+1, n, '{0:.0f}'.format(self.table_values[m][n]))
640                     )
641                     else:
642                         t.set(m+1,n, '{0:.2e}'.format(self.table_values[m][n]))
643
644         class SimpleTable(Frame):
645
646             #Simply formats a table to display fit parameters values.
647
648             def __init__(self, parent, rows=2, columns=2):
649                 # use black background so it "peeks through" to
650                 # form grid lines
651                 Frame.__init__(self, parent, background="black")
652                 self._widgets = []
653                 for row in range(rows):
654                     current_row = []
655                     for column in range(columns):
656                         label = Label(self, text="%s/%s" % (row, column),
657                                     borderwidth=0, width=10)
658                         label.grid(row=row, column=column, sticky="nsew", padx=1,
659                         pady=1)

```

```

650         current_row.append(label)
651         self._widgets.append(current_row)

652     for column in range(columns):
653         self.grid_columnconfigure(column, weight=1)
654

656     def set(self, row, column, value):
657         widget = self._widgets[row][column]
658         widget.configure(text=value)

660 class MainApplication(Frame):

662     # This is the main application that contains the core of the program.
663     Places secondary frames and the
664     # search folder objects that are used to initialize the rest of the
665     program

666     def __init__(self, master):#, *args, **kwargs):
667         Frame.__init__(self, master)#, *args, **kwargs)

668         self.parent = master # Master will be the parent for widgets to
669         come

670         # print(Tk().eval('info patchlevel'))

672         self.first_run = True
673         self.first_plot = True

674         self.secondary_frames = SecondaryFrames(self.parent)

676         self.top_frame      = self.secondary_frames.top_frame
677         self.left_frame     = self.secondary_frames.left_frame
678         self.right_frame    = self.secondary_frames.right_frame
679         self.right_frame2   = self.secondary_frames.right_frame2

682         self.secondary_frames.top_frame.pack(padx=5, pady=5, side=TOP, fill
683         =X)
684             self.secondary_frames.horizontal_line.pack(side=TOP, fill=X)
685             self.secondary_frames.left_frame.pack_propagate(0)
686             self.secondary_frames.left_frame.pack(padx=5, pady=5, side=LEFT,
687             fill=Y)
688                 self.secondary_frames.vertical_line.pack(side=LEFT, fill=Y)
689                 self.secondary_frames.right_frame.pack_propagate(0)
690                 self.secondary_frames.right_frame.pack(padx=5, pady=5, side=LEFT,
691                 fill=Y)
692                     self.secondary_frames.vertical_line.pack(side=LEFT, fill=Y)

```

```
690     self.secondary_frames.right_frame2.pack(padx=5, pady=20, side=LEFT,
691                                               fill=Y)
692
693     self.search_folder_objects = SearchFolder(self.top_frame, self.
694                                               left_frame, self.right_frame, self.right_frame2, self.first_run, self.
695                                               first_plot)
696
697     self.search_folder_objects.entry_frame.pack(anchor='w')
698
699     self.search_folder_objects.search_button.pack(padx=2, pady=1,
700                                               anchor="w", side=LEFT)
701     self.search_folder_objects.path_box.pack(padx=2, pady=1, side=LEFT)
702
703 if __name__ == '__main__':
704     root = Tk()
705     root.geometry("1150x530")
706     MainApplication(root)
707     root.mainloop()
```