

SIMULAÇÃO DE SINAIS DE RMN ATRAVÉS DAS EQUAÇÕES DE BLOCH

Tiago Bueno Moraes^{a,*} e Luiz Alberto Colnago^b^aInstituto de Física de São Carlos, Universidade de São Paulo, Av. Trabalhador São-carlense 400, 13560-970 São Carlos – SP, Brasil^bEmbrapa Instrumentação, Rua XV de Novembro 1452, 13560-970 São Carlos – SP, Brasil

APÊNDICE A - Revisão de multiplicação de matrizes

Matrizes são conjuntos de elementos disposto em forma tabular e seu tamanho é definido por $(m \times n)$.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \quad (1)$$

Podemos ter vetores escritos na forma de matriz linha com tamanho $(1 \times n)$ ou uma matriz coluna, com tamanho $(n \times 1)$, da forma

$$A = (a_{11} \quad \dots \quad a_{1n}) \quad \text{ou} \quad A = \begin{pmatrix} a_{11} \\ \vdots \\ a_{n1} \end{pmatrix} \quad (2)$$

Para matrizes (3×1) , como na matriz de rotação do pulso temos a seguintes regras de multiplicação

$$\vec{M}(1) = R_\theta \vec{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

Multiplicamos a primeira linha da matriz R_θ , pela coluna do vetor \vec{M} , para obter o primeiro termo do vetor resultante

$$(1 \quad 0 \quad 0) * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 1*0 + 0*0 + 0*1 = 0 \quad (4)$$

Para o cálculo do segundo termo, multiplicamos a primeira linha da matriz R_θ pela coluna do vetor \vec{M}

$$(0 \quad 0 \quad 1) * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0*0 + 0*0 + 1*1 = 1 \quad (4)$$

E para o terceiro termo

$$(0 \quad -1 \quad 0) * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0*0 + (-1)*0 + 0*1 = 0 \quad (4)$$

Desse modo obtemos a primeira linha da matriz resultante, fazendo o mesmo procedimento para as demais linhas da matriz A, obtemos o resultado

$$\vec{M}(1) = R_\theta \vec{M} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (3)$$

No software do *Matlab* todas essas contas são feitas escrevendo apenas *R.M*, facilitando o processo de simulação dos sinais de RMN. Mais detalhes de como realizar multiplicação ou outras operações matriciais no *Matlab* podem ser encontrados na referência¹⁵.

APÊNDICE B – Pulso Simples

```
clear; clc;
%Programa Pulso Simples - (30/11/2013)

%-----%
% Este código calcula o comportamento da Magnetização quando aplicamos um
% pulso simples de ângulo theta;
% Tiago Bueno de Moraes - (tiagobuemoraes@gmail.com) - Química Nova 2013
%-----%

% === Parâmetros entrada ===
% Amostra
T1 = 600;
T2 = 400;
```

*e-mail: tiagobuemoraes@gmail.com

```

Fo = 15;           % (MHz) Posição central pico
df = 0:0.1:30;    % (MHz) Região redor pico (Isocromatas)

% Espectrômetro
theta = pi/2;      % Angulo pulso radio frequência
T = 3000;          % (ms) tempo total aquisição FID
dT = 1;            % (ms) tempo entre pontos

% Magneto
% Inomogeneidade de campo. (T22 é T2*)
T2inom = 200;  T22 = 1/((1/T2) + (1/(T2inom)));

% ===== Matrizes Rotação =====
Rtheta = [1 0 0; 0 cos(theta) sin(theta); 0 -sin(theta) cos(theta)]; % Fase x
E1 = exp(-dT/T1);  E2 = exp(-dT/T2);
E = [E2 0 0; 0 E2 0; 0 0 E1];      B = [0 0 1-E1].';

% ===== Contadores Sinal =====
N0 = round(T/dT);          % nº pontos total;      tempo total = N0*dT
M = zeros(3,N0);           % Cria vetor magnetização tamanho (3 x N0)
Ms = zeros(3,N0);          % Cria vetor magnetização Ms (armazenar sinal)
M(:,1)=[0;0;1];            % Magnetização posição inicial no eixo z

for f=1:length(df)
    phi = 2*pi*(df(f))*dT/1000;
    Rphi = [cos(phi) sin(phi) 0; -sin(phi) cos(phi) 0; 0 0 1];

    M(:,2)= Rtheta*M(:,1)+B;          % Aplicação pulso angulo theta
    for k=3:(N0-1)                    % evolução sinal
        M(:,k) = E*Rphi*M(:,k-1)+B;
    end;

    % ----- Calculo peso Inomogeneidade - T2* -----
    % Pico Lorentziano
    g = T22/(1+((2*pi*T22/1000)*(df(f)-Fo))^2);

    % Somando as componentes x y z das isocromata
    Ms = [g*M(1,:)+Ms(1,:); g*M(2,:)+Ms(2,:); g*M(3,:)+Ms(3,:)];
end;

% Normaliza sinal FID
Ms = Ms/max(Ms(:,1));

% ===== Graficando Resultados =====
tempo = [0:N0-1]*dT;  CurvaT2 = exp(-tempo/T2);
p=plot(tempo,Ms(1,:), 'b-',tempo,Ms(2,:), 'k-',tempo,Ms(3,:), 'r--',tempo,CurvaT2,'g:');
legend('M_x','M_y','M_z','T_2'); title('Pulso Simples');
xlabel('Tempo (ms)'); ylabel('Intensidade');
grid on;  set(p,'LineWidth',1.5);

```

APÊNDICE C – Inversão Recuperação

```

clear; clc;
%Programa Inversão Recuperação - (30/11/2013)

%-----%
% Este programa calcula a Magnetização resultante após a aplicação da seq.
% de Inversão Recuperação      180x_t_90x_Fid
% Tiago Bueno de Moraes - (tiagobuemoraes@gmail.com) - Química Nova 2013
%-----%

```

```

% === Parâmetros entrada ===
% Amostra
T1 = 600;
T2 = 400;
Fo = 15;           % (MHz) Posição central pico
df = 0:0.1:30;     % (MHz) Região redor pico (Isocromatas)

% Espectrômetro
theta = pi;         % Pulso 180
theta2 = pi/2;      % Pulso 90
T = 1600;           % (ms) tempo total aquisição FID
dT = 1;             % (ms) tempo entre pontos

% Magneto
% Inomogeneidade de campo. (T22 é T2*)
T2inom = 200;  T22 = 1/((1/T2) + (1/(T2inom)));

% ===== Matrizes Rotação =====
Rtheta = [1 0 0; 0 cos(theta) sin(theta); 0 -sin(theta) cos(theta)]; % Fase x
Rtheta2 = [1 0 0; 0 cos(theta2) sin(theta2); 0 -sin(theta2) cos(theta2)]; % Fase x
E1 = exp(-dT/T1);  E2 = exp(-dT/T22);
E = [E2 0 0; 0 E2 0; 0 0 E1];      B = [0 0 1-E1]';

Tn1 = 200;      % tempo até primeiro pulso

% ===== Contadores Sinal =====
N0 = round(T/dT);      % n° pontos total;      tempo total = N0*dT
N1 = round(Tn1/dT);    % n° pontos até primeiro pulso
N2 = N0 - N1;          % n° pontos depois do ultimo pulso

M = zeros(3,N0);       % Criar vetor magnetização Msignal
Ms = zeros(3,N0);      % Criar vetor magnetização Msignal
M(:,1)=[0;0;1];        % Magnetização inicial

for f=1:length(df)
    phi = 2*pi*df(f)*dT/1000;
    Rphi = [cos(phi) sin(phi) 0; -sin(phi) cos(phi) 0; 0 0 1];

    M(:,2)= Rtheta*M(:,1)+B;          % Pulso 180°x
    for k=3:(N1+1)                    % evolução sinal
        M(:,k) = E*Rphi*M(:,k-1)+B;
    end;

    M(:,N1+2)=Rtheta2*M(:,N1+1)+B;    % pulso de 90°x
    for k=2:N2-1                      % evolução sinal
        M(:,k+N1+1) = E*Rphi*M(:,k+N1)+B;
    end;

    % ----- Calculo peso Inomogeneidade - T2* -----
    % Pico Lorentziano
    g = T22/(1+((2*pi*T22/1000)*(df(f)-Fo))^2);

    % Somando as componentes x y z das isocromata
    Ms = [g*M(1,:)+Ms(1,:); g*M(2,:)+Ms(2,:); g*M(3,:)+Ms(3,:)];

end;

% Normaliza sinal FID
Ms = Ms/max(Ms(:,1));

```

```
% ===== Graficando Resultados =====
tempo = [0:N0-1]*dT;
p=plot(tempo,Ms(1,:), 'b-', tempo,Ms(2,:), 'k-', tempo,Ms(3,:), 'r--');
legend('M_x', 'M_y', 'M_z'); title('Inversão Recuperação');
xlabel('Tempo (ms)'); ylabel('Intensidade');
grid on; set(p, 'LineWidth', 1.5);
```

APÊNDICE D – Spin Eco

```
clear; clc;
%Programa Spin Eco - (30/11/2013)

%-----%
%Este programa calcula a Magnetização resultante após a aplicação da seq.
%Spin Eco de NMR.      90y_t_180x_t_eco
% Tiago Bueno de Moraes - (tiagobuemoraes@gmail.com) - Química Nova 2013
%-----%

% === Parâmetros entrada ===
% Amostra
T1 = 600;
T2 = 400;
Fo = 15;           % (MHz) Posição central pico
df = 0:0.1:30;     % (MHz) Região redor pico (Isocromatas)

% Espectrômetro
theta = pi/2;      % Pulso 90
theta2 = pi;       % Pulso 180
T = 2000;          % (ms) tempo total aquisição FID
dT = 1;            % (ms) tempo entre pontos

% Magneto
% Inomogeneidade de campo. (T22 é T2*)
T2inom = 100; T22 = 1/((1/T2) + (1/(T2inom)));

% ===== Matrizes Rotação =====
Rtheta = [cos(theta) 0 sin(theta); 0 1 0; -sin(theta) 0 cos(theta)]; % Fase y
Rtheta2 = [1 0 0; 0 cos(theta2) -sin(theta2); 0 sin(theta2) cos(theta2)]; % Fase x

E1 = exp(-dT/T1); E2 = exp(-dT/T2);
E = [E2 0 0; 0 E2 0; 0 0 E1]; B = [0 0 1-E1].';

Tn1 = 400; % tempo até segundo pulso

% ===== Contadores Sinal =====
N0 = round(T/dT); % n° pontos total; tempo total = N0*dT
N1 = round(Tn1/dT); % tempo até primeiro pulso
N2 = N0 - N1; % n° pontos depois do ultimo pulso

M = zeros(3,N0); % Criar vetor magnetização Msignal
Ms = zeros(3,N0); % Criar vetor magnetização Msignal
M(:,1)=[0;0;1]; % Magnetização inicial

for f=1:length(df)
    phi = 2*pi*df(f)*dT/1000;
    Rphi = [cos(phi) sin(phi) 0; -sin(phi) cos(phi) 0; 0 0 1];

    M(:,2)= Rtheta*M(:,1)+B; % Pulso 90°
    for k=3:(N1+1) % evolução sinal
        M(:,k) = E*Rphi*M(:,k-1)+B;
    end;
```

```

M(:,N1+2)= Rtheta2*M(:,N1+1)+B;           % pulso de 180°x
for k=2:(N2-1)                             % evolução sinal
    M(:,k+N1+1) = E*Rphi*M(:,k+N1)+B;
end;

% ----- Calculo peso Inomogeneidade - T2* -----
% Pico Lorentziano
g = T22/(1+((2*pi*T22/1000)*(df(f)-Fo))^2);

% Somando as componentes x y z das isocromata
Ms = [g*M(1,:)+Ms(1,:) ; g*M(2,:)+Ms(2,:) ; g*M(3,:)+Ms(3,:)];

end;

% Normaliza sinal FID
Ms = Ms/max(Ms(:,1));

% ===== Graficando Resultados =====
tempo = [0:N0-1]*dT; CurvaT2 = exp(-tempo/T2);
p=plot(tempo,Ms(1,:), 'b-',tempo,Ms(2,:), 'k-',tempo,Ms(3,:), 'r--',tempo, CurvaT2, 'g:');
legend('M_x','M_y','M_z','T_2'); title('Spin Eco');
xlabel('Tempo (ms)'); ylabel('Intensidade');
grid on; set(p, 'LineWidth', 1.5);

```

APÊNDICE E – CPMG

```

clear; clc;
%Programa CPMG (30/11/2013)

%-----%
%Este programa calcula a Magnetização resultante após a aplicação da seq.
%CPMG de NMR.          90y_t_[180x_t_eco]
% Tiago Bueno de Moraes - (tiagobuemoraes@gmail.com) - Química Nova 2013
%-----%

% === Parâmetros entrada ===
% Amostra
T1 = 600;
T2 = 400;
Fo = 15;           % (MHz) Posição central pico
df = 0:0.1:30;     % (MHz) Região redor pico (Isocromatas)

% Espectrômetro
theta = pi/2;      % Pulso 90
theta2 = pi;       % Pulso 180
T = 2000;          % (ms) tempo total aquisição FID
dT = 1;            % (ms) tempo entre pontos

% Magneto
% Inomogeneidade de campo. (T22 é T2*)
T2inom = 100; T22 = 1/((1/T2) + (1/(T2inom)));

% ===== Matrizes Rotação =====
Rtheta = [cos(theta) 0 sin(theta); 0 1 0; -sin(theta) 0 cos(theta)]; % Fase y
Rtheta2 = [1 0 0; 0 cos(theta2) -sin(theta2); 0 sin(theta2) cos(theta2)]; % Fase x

E1 = exp(-dT/T1); E2 = exp(-dT/T2);
E = [E2 0 0; 0 E2 0; 0 0 E1]; B = [0 0 1-E1]';

```

```

Tn1 = 200; % tempo até primeiro pulso
Tn2 = 2*Tn1; % tempo entre pulsos
Np = 4; % numero de pulsos

% ===== Contadores Sinal =====
N0 = round(T/dT); % n° pontos total; tempo total = N0*dT
N1 = round(Tn1/dT); N1c = N1; % tempo até primeiro pulso
N2 = round(Tn2/dT); % tempo entre pulsos
N3 = N0 - (N1+(Np-1)*(N2)); % n° pontos depois do ultimo pulso

M = zeros(3,N0); % Criar vetor magnetização Msignal
Ms = zeros(3,N0); % Criar vetor magnetização Msignal
M(:,1)=[0;0;1]; % Magnetização inicial

for f=1:length(df)
    phi = 2*pi*df(f)*dT/1000;
    Rphi = [cos(phi) sin(phi) 0; -sin(phi) cos(phi) 0; 0 0 1];

    M(:,2)= Rtheta*M(:,1)+B; % pulso de 90°y
    for k=3:(N1c+1) % evolução até TE/2
        M(:,k) = E*Rphi*M(:,k-1)+B;
    end;

    for n=1:Np-1
        M(:,N1c+2) = Rtheta2*M(:,N1c+1)+B; % pulso de 180°x no instante TE/2
        for k=2:N2-1 % evolução até o final
            M(:,k+N1c+1) = E*Rphi*M(:,k+N1c)+B;
        end;
        N1c=N1c+N2-1;
    end;

    %Ultimo pulso
    M(:,N1c+2)= Rtheta2*M(:,N1c+1)+B; % pulso de 180°x no instante TE/2
    for k=2:N3-1 % evolução até o final
        M(:,k+N1c+1) = E*Rphi*M(:,k+N1c)+B;
    end;
    N1c=N1; % fazendo N1c voltar ao valor inicial

% ----- Calculo peso Inomogeneidade - T2* -----
% Pico Lorentziano
g = T22/(1+((2*pi*T22/1000)*(df(f)-Fo))^2);

% Somando as componentes x y z das isocromata
Ms = [g*M(1,:)+Ms(1,:) ; g*M(2,:)+Ms(2,:) ; g*M(3,:)+Ms(3,:)];

end;

% Normaliza sinal FID
Ms = Ms/max(Ms(:,1));

% ===== Graficando Resultados =====
tempo = [0:N0-1]*dT; CurvaT2 = exp(-tempo/T2);
p=plot(tempo,Ms(1,:), 'b-',tempo,Ms(2,:), 'k-',tempo,Ms(3,:), 'r--',tempo, CurvaT2, 'g:');
legend('M_x','M_y','M_z','T_2'); title('CPMG');
xlabel('Tempo (ms)'); ylabel('Intensidade');
grid on; set(p, 'LineWidth', 1.5);

```