

Simulation de séquences IRM

Julien Lamy, Ph.D.

ICube (Université de Strasbourg-CNRS)

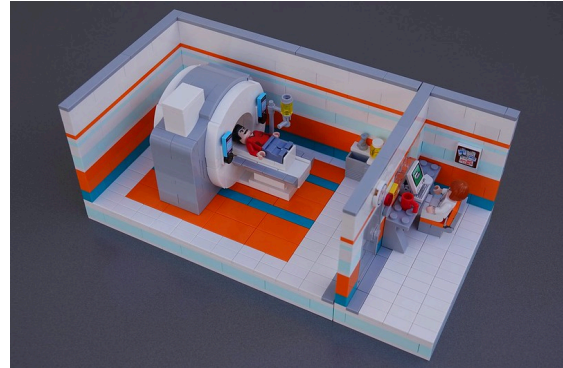
lamy@unistra.fr



Simulation de séquences IRM – Pourquoi ?

Une IRM de poche

- Pas de problème de temps machine
- Pas de limite de paramètres (échantillon ou séquence)
- Facilité d'optimisation des paramètres de séquence (contraste, SNR, etc.)



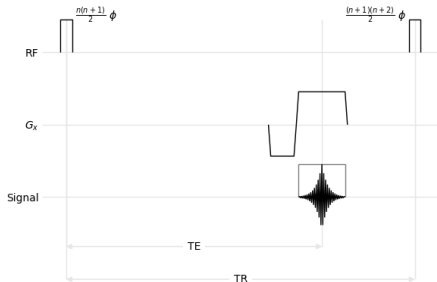
Source : Flickr

Simulation de séquences IRM – Quoi ?

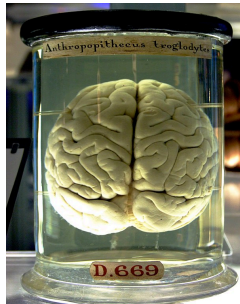
Évolution de l'aimantation au cours du temps

Effet des impulsions RF et des gradients

Prise en compte de la relaxation, de la diffusion, du mouvement, etc.



+



Source : Wikipedia

=



Simulation de séquences IRM – Comment ?

Utilisation de l'équation de Bloch (ou Bloch-Torrey ou Bloch-McConnell)

$$\frac{d\vec{M}(t)}{dt} = \vec{M}(t) \times \gamma \vec{B}_0 + \vec{M}(t) \times \gamma \vec{B}_1(t) - \begin{bmatrix} M_x(t) R_2 \\ M_y(t) R_2 \\ (M_z(t) - M_z^{\text{eq}}) R_1 \end{bmatrix}$$

1. Résolution analytique : solution exacte, uniquement possible dans des cas simples

Simulation de séquences IRM – Comment ?

Utilisation de l'équation de Bloch (ou Bloch-Torrey ou Bloch-McConnell)

$$\frac{d\vec{M}(t)}{dt} = \vec{M}(t) \times \gamma \vec{B}_0 + \vec{M}(t) \times \gamma \vec{B}_1(t) - \begin{bmatrix} M_x(t) R_2 \\ M_y(t) R_2 \\ (M_z(t) - M_z^{\text{eq}}) R_1 \end{bmatrix}$$

1. Résolution analytique : solution exacte, uniquement possible dans des cas simples
2. Résolution numérique : applicable à toutes les séquences, gourmand en calcul, risque d'instabilité numérique

Simulation de séquences IRM – Comment ?

Utilisation de l'équation de Bloch (ou Bloch-Torrey ou Bloch-McConnell)

$$\frac{d\vec{M}(t)}{dt} = \vec{M}(t) \times \gamma \vec{B}_0 + \vec{M}(t) \times \gamma \vec{B}_1(t) - \begin{bmatrix} M_x(t) R_2 \\ M_y(t) R_2 \\ (M_z(t) - M_z^{\text{eq}}) R_1 \end{bmatrix}$$

1. Résolution analytique : solution exacte, uniquement possible dans des cas simples
2. Résolution numérique : applicable à toutes les séquences, gourmand en calcul, risque d'instabilité numérique
3. Résolution sur un intervalle de temps court : formalisme matriciel, applicable à beaucoup de séquences, potentiellement gourmand en calcul

$$E = \begin{pmatrix} E_2 & 0 & 0 & 0 \\ 0 & E_2 & 0 & 0 \\ 0 & 0 & E_1 & M_z^{\text{eq}}(1-E_1) \\ 0 & 0 & 0 & 1 \end{pmatrix} ; \quad T = R_z(\varphi) R_x(\theta) R_z(-\varphi) ; \quad S = R_z(\Delta\omega t)$$

Simulation de séquences IRM – Comment ?

Utilisation de l'équation de Bloch (ou Bloch-Torrey ou Bloch-McConnell)

$$\frac{d\vec{M}(t)}{dt} = \vec{M}(t) \times \gamma \vec{B}_0 + \vec{M}(t) \times \gamma \vec{B}_1(t) - \begin{bmatrix} M_x(t) R_2 \\ M_y(t) R_2 \\ (M_z(t) - M_z^{\text{eq}}) R_1 \end{bmatrix}$$

1. Résolution analytique : solution exacte, uniquement possible dans des cas simples
2. Résolution numérique : applicable à toutes les séquences, gourmand en calcul, risque d'instabilité numérique
3. Résolution sur un intervalle de temps court : formalisme matriciel, applicable à beaucoup de séquences, potentiellement gourmand en calcul
4. idem, dans l'espace k (EPG) : applicable à toutes les séquences, plutôt axé mono-voxel, moins gourmand en calcul

$$E(\vec{M}) = \begin{pmatrix} E_2 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & E_1 \end{pmatrix} \vec{M} + \begin{pmatrix} 0 \\ 0 \\ M_0(1 - E_1) \end{pmatrix}; \quad T = \begin{pmatrix} \cos^2\left(\frac{\theta}{2}\right) & \exp(2i\varphi) & -i \exp(i\varphi) \sin(\theta) \\ \exp(-2i\varphi) \sin^2\left(\frac{\theta}{2}\right) & \cos^2(\theta) & i \exp(-i\varphi) \sin(\theta) \\ -\frac{i}{2} \exp(-i\varphi) \sin(\theta) & \frac{i}{2} \exp(i\varphi) \sin(\theta) & \cos(\alpha) \end{pmatrix}$$

Simulation de séquences IRM – Comment ?

Utilisation de l'équation de Bloch (ou Bloch-Torrey ou Bloch-McConnell)

$$\frac{d\vec{M}(t)}{dt} = \vec{M}(t) \times \gamma \vec{B}_0 + \vec{M}(t) \times \gamma \vec{B}_1(t) - \begin{bmatrix} M_x(t) R_2 \\ M_y(t) R_2 \\ (M_z(t) - M_z^{\text{eq}}) R_1 \end{bmatrix}$$

1. Résolution analytique : solution exacte, uniquement possible dans des cas
2. Résolution numérique : applicable à toutes les séquences, gourmand en calcul, risque d'instabilité numérique
3. Résolution sur un intervalle de temps court : formalisme matriciel, applicable à beaucoup de séquences, potentiellement gourmand en calcul
4. idem, dans l'espace k (EPG) : applicable à toutes les séquences, plutôt axé mono-voxel, moins gourmand en calcul

Au programme aujourd'hui : résolution sur un intervalle de temps court, espace physique et espace k

Équation de Bloch

Nuclear Induction, F. Bloch, Physical Review 70, 1946 :

$$\frac{d\vec{M}(t)}{dt} = \underbrace{\vec{M}(t) \times \gamma \vec{B}_0}_{\text{précession}} + \underbrace{\vec{M}(t) \times \gamma \vec{B}_1(t)}_{\text{nutation}} - \underbrace{\begin{bmatrix} M_x(t) R_2 \\ M_y(t) R_2 \\ (M_z(t) - M_z^{\text{eq}}) R_1 \end{bmatrix}}_{\text{relaxation}}$$

Si $\vec{B}_0 = [0 ; 0 ; B_0]$ et $B_1(t) = [\cos(\omega_0 t + \varphi) ; \sin(\omega_0 t + \varphi) ; 0]$, dans le référentiel tournant :

$$\frac{d\vec{M}(t)}{dt} = \Delta\omega \begin{pmatrix} -M_y(t) \\ M_x(t) \\ 0 \end{pmatrix} + \gamma B_1 \begin{pmatrix} \sin(\theta) M_z(t) \\ -\cos(\theta) M_z(t) \\ M_y(t) \cos(\theta) - M_x(t) \sin(\theta) \end{pmatrix} - \begin{pmatrix} R_2 M_x(t) \\ R_2 M_y(t) \\ R_1 (M_z(t) - M_z^{\text{eq}}) \end{pmatrix}$$

Décrit le comportement d'un **isochromat** : ensemble de noyaux ayant le même T_1 , T_2 , $\Delta\omega$, position

Hypothèses : pas de diffusion, pas de flux, pas d'échange

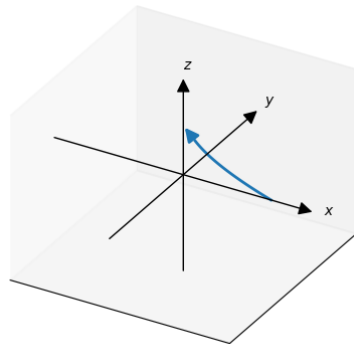
Relaxation

En l'absence d'excitation ($B_1 = 0$), en résonance ($\Delta\omega = 0$),

$$\frac{d\vec{M}(t)}{dt} = - \begin{pmatrix} R_2 M_x(t) \\ R_2 M_y(t) \\ R_1 (M_z(t) - M_z^{\text{eq}}) \end{pmatrix}$$

Soit

$$\begin{bmatrix} M_x(t) \\ M_y(t) \\ M_z(t) \end{bmatrix} = \begin{bmatrix} \exp(-R_2 t) M_x(0) \\ \exp(-R_2 t) M_y(0) \\ M_z^{\text{eq}} - \exp(-R_1 t) (M_z^{\text{eq}} - M_z(0)) \end{bmatrix}$$



Relaxation

Soient $E_1(t) = \exp(-R_1 t)$ et $E_2(t) = \exp(-R_2 t)$

Forme matrice-vecteur

$$\vec{M}(t) = \begin{pmatrix} E_2(t) & 0 & 0 \\ 0 & E_2(t) & 0 \\ 0 & 0 & E_1(t) \end{pmatrix} \vec{M}(0) + \begin{bmatrix} 0 \\ 0 \\ M_z^{\text{eq}} (1 - E_1(t)) \end{bmatrix}$$

Forme matricielle (coordonnées projectives)

$$\begin{bmatrix} M_x(t) \\ M_y(t) \\ M_z(t) \\ 1 \end{bmatrix} = \begin{pmatrix} E_2(t) & 0 & 0 & 0 \\ 0 & E_2(t) & 0 & 0 \\ 0 & 0 & E_1(t) & M_z^{\text{eq}} (1 - E_1(t)) \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} M_x(0) \\ M_y(0) \\ M_z(0) \\ 1 \end{bmatrix} = E(R_1, R_2, t, M_z^{\text{eq}}) \vec{M}(0)$$

$E(R_1, R_2, t, M_z^{\text{eq}})$ est l'opérateur de relaxation

Impulsion RF

En résonance ($\Delta\omega = 0$), pour un temps court ($E_1, E_2 \rightarrow 0$),

$$\frac{d\vec{M}(t)}{dt} = \gamma B_1 \begin{pmatrix} \sin(\theta) M_z(t) \\ -\cos(\theta) M_z(t) \\ M_y(t) \cos(\theta) - M_x(t) \sin(\theta) \end{pmatrix}$$

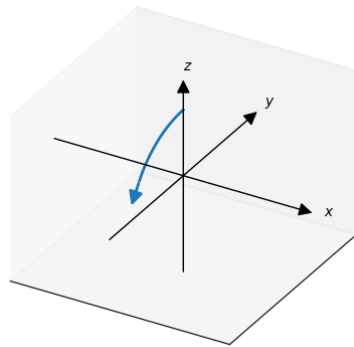
Soit une rotation autour de l'axe $[\cos(\varphi) ; \sin(\varphi) ; 0]$ d'un angle $\theta = \gamma B_1 t$

Si $\varphi = 0$, on obtient une rotation autour de l'axe x , et, dans le cas général, la composition de trois rotations

$$\vec{M}(t) = R_z(\varphi) R_x(\theta) R_z(-\varphi) \vec{M}(0) = T(\theta, \varphi) \vec{M}(0)$$

$T(\theta, \varphi)$ est l'opérateur d'impulsion RF

Pour des impulsions longues, différents opérateurs sont enchaînés (multiplication de matrices)



Précession libre et application de gradients

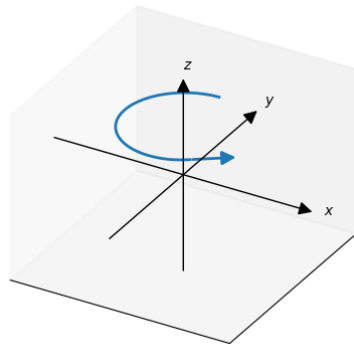
En l'absence d'excitation ($B_1 = 0$), pour un temps court ($E_1, E_2 \rightarrow 0$),

$$\frac{d\vec{M}(t)}{dt} = \Delta\omega \begin{pmatrix} -M_y(t) \\ M_x(t) \\ 0 \end{pmatrix}$$

Soit une rotation autour de l'axe z

$$\vec{M}(t) = R_z(\Delta\omega t) \vec{M}(0) = S(\Delta\omega, t) \vec{M}(0)$$

$S(\Delta\omega, t)$ est l'opérateur de précession



Simulation par opérateurs matriciels

$$\frac{d\vec{M}(t)}{dt} = \Delta\omega \begin{pmatrix} -M_y(t) \\ M_x(t) \\ 0 \end{pmatrix} + \gamma B_1 \begin{pmatrix} \sin(\theta) M_z(t) \\ -\cos(\theta) M_z(t) \\ M_y(t) \cos(\theta) - M_x(t) \sin(\theta) \end{pmatrix} - \begin{pmatrix} R_2 M_x(t) \\ R_2 M_y(t) \\ R_1 (M_z(t) - M_z^{\text{eq}}) \end{pmatrix}$$

Relaxation : $E(R_1, R_2, t)$

Impulsion RF : $T(\theta, \varphi)$

Précession : $S(\Delta\omega, t)$

Une séquence IRM est une suite d'opérations (impulsions RF, gradients, temps morts)

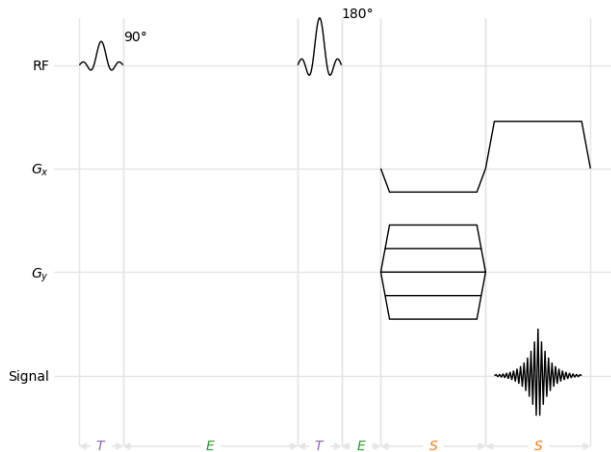
Donc une suite d'opérateurs

Donc une suite de produits de matrices

Simulation par opérateurs matriciels

Pour une séquence en écho de spin, on a donc

- Deux opérateurs d'impulsion RF+relaxation (excitation et refocalisation)
- Deux opérateurs de relaxation (temps morts)
- Deux opérateurs de déphasage+relaxation (codage spatial et lecture)



Simulation par opérateurs matriciels – Cas du *spoiling*

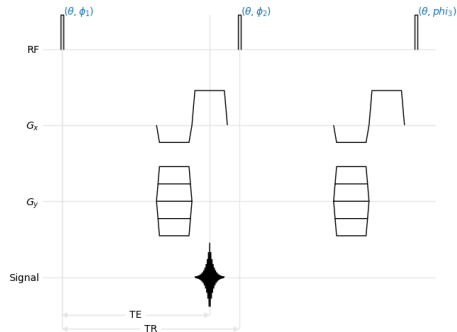
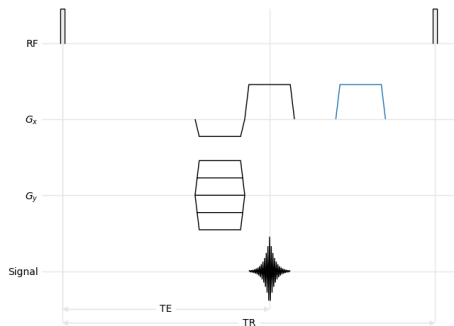
Pour que l'aimantation transversale soit nulle :

- Attendre une relaxation complète ($\approx 5 T_2$)
- Utiliser une technique de *spoiling* : l'aimantation au sein d'un voxel est déphasée sans relaxation complète

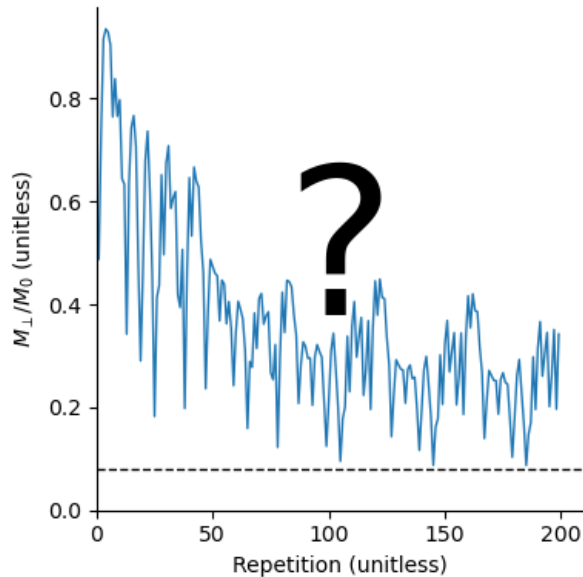
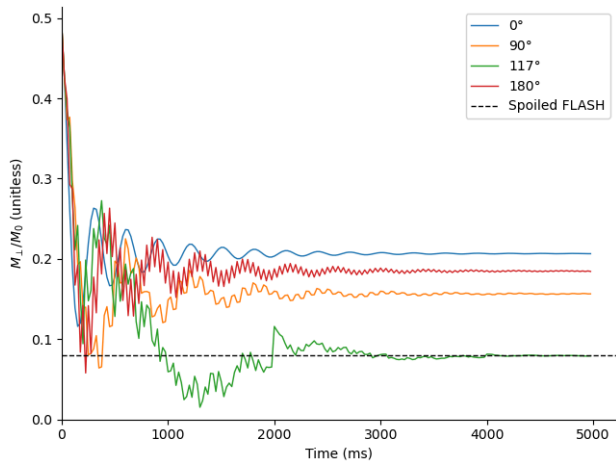
Deux techniques existent

- Utilisation d'un lobe de gradient pour créer un déphasage de 2π au sein d'un voxel
- Utilisation de la phase des impulsions RF

Spoiling of transverse magnetization in steady-state sequences, Zur et al. Magnetic Resonance in Medicine 21(2). 1991



Simulation par opérateurs matriciels – Cas du *RF Spoiling*

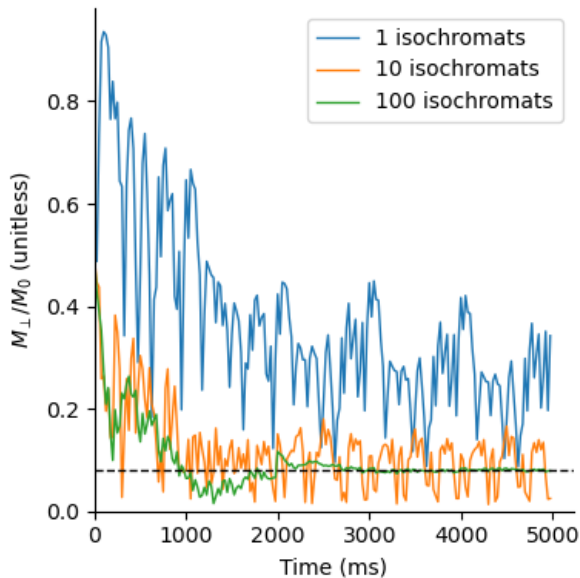


Simulation par opérateurs matriciels – Cas du *RF Spoiling*

Le *RF spoiling* est un comportement d'ensemble

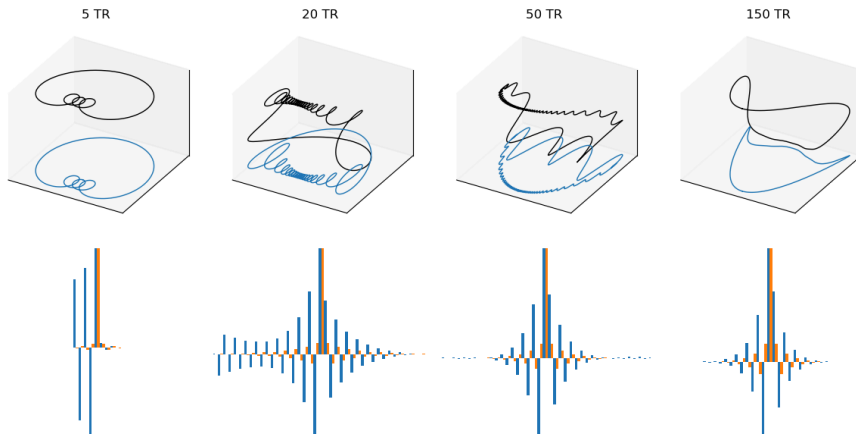
Simuler un seul isochromat ne suffit pas

Il faut simuler une distribution spatiale,
donc « beaucoup » d'isochromats



Distribution spatiale de l'aimantation

La distribution de l'aimantation est régulière : sa transformée de Fourier est creuse



Une brève histoire d'EPG

Proto-histoire : « The basic idea of the method is to follow the phase shift accumulated »

Diffusion in Spin-Echo Experiments, Woessner, 1961

Pré-histoire : « partition of the observable signal amplitude into its contributions from various echo paths »

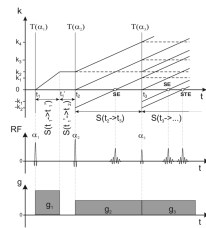
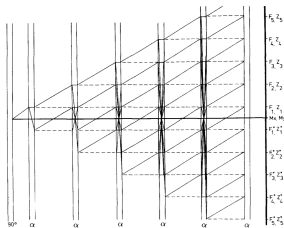
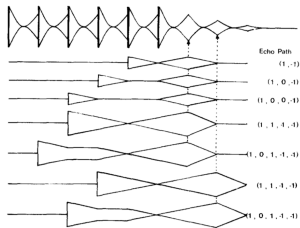
NMR Fourier spectroscopy, Kaiser et al., 1974

Renaissance : « observation of symmetries in the phase evolution of the spin system »

Imaging with Low-Angle Refocusing, Hennig, 1988

Temps modernes : « They represent the general continuous form of the EPG. »

Extended phase graphs with anisotropic diffusion, Weigel et al. 2010



EPG en trois étapes

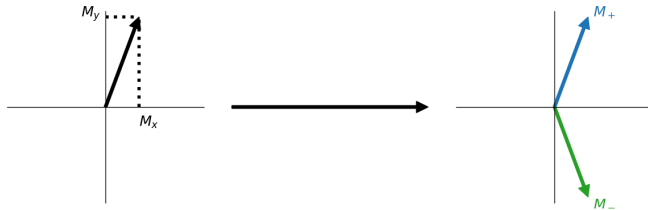
1. Ré-écrire l'aimantation sous **forme complexe**

$$M_+ = M_x + iM_y, M_- = M_x - iM_y \quad [M_x ; M_y ; M_z] \rightarrow [M_+ ; M_- ; M_z]$$

2. Calculer la **transformée de Fourier** pour obtenir les états $F_+(\vec{k})$, $F_-(\vec{k})$ et $Z(\vec{k})$ à l'ordre k

$$\begin{bmatrix} F_+(\vec{k}) \\ F_-(\vec{k}) \\ Z(\vec{k}) \end{bmatrix} = \int \begin{bmatrix} M_+(\vec{r}) \\ M_-(\vec{r}) \\ M_z(\vec{r}) \end{bmatrix} \exp(-i\vec{k} \cdot \vec{r}) d\vec{r}$$

3. Ré-écrire les opérateurs matriciels dans cet espace k



EPG – Opérateur de relaxation

Égal à la forme matrice-vecteur de l'opérateur montré précédemment

$$\begin{bmatrix} F_+(k) \\ F_-(k) \\ Z(k) \end{bmatrix} \xrightarrow{E} \begin{pmatrix} E_2(t) & 0 & 0 \\ 0 & E_2(t) & 0 \\ 0 & 0 & E_1(t) \end{pmatrix} \begin{bmatrix} F_+(k) \\ F_-(k) \\ Z(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M_2^{\text{eq}} (1 - E_1(t)) \end{bmatrix}$$

Dans le formalisme EPG, les opérateurs sont vus comme une **transformation** des ordres et des états
Certains n'ont pas de forme matricielle

EPG – Opérateur d'impulsion RF

Le passage à l'aimantation sous forme complexe rend l'opérateur un peu plus complexe

$$\begin{bmatrix} F_+(k) \\ F_-(k) \\ Z(k) \end{bmatrix} \xrightarrow{T} \begin{pmatrix} \cos^2\left(\frac{\theta}{2}\right) & \exp(2i\varphi) & -i \exp(i\varphi) \sin(\theta) \\ \exp(-2i\varphi) \sin^2\left(\frac{\theta}{2}\right) & \cos^2(\theta) & i \exp(-i\varphi) \sin(\theta) \\ -\frac{i}{2} \exp(-i\varphi) \sin(\theta) & \frac{i}{2} \exp(i\varphi) \sin(\theta) & \cos(\alpha) \end{pmatrix} \begin{bmatrix} F_+(k) \\ F_-(k) \\ Z(k) \end{bmatrix}$$

EPG – Opérateur de déphasage

L'application d'un gradient déplace les ordres dans l'espace k

$$\begin{bmatrix} F_+(k) \\ F_-(k) \\ Z(k) \end{bmatrix} \xrightarrow{s} \begin{bmatrix} F_+(k + \delta_k) \\ F_-(k + \delta_k) \\ Z(k) \end{bmatrix}$$

Seuls les états transversaux (F_+ et F_-) sont affectés par le déphasage

L'état transversal Z reste constant

Cet opérateur n'a pas de forme matricielle, il décale les ordres

Il est le seul à créer de nouveaux états

Diffusion

Diffusion anistrophe avec un propagateur gaussien

Les populations des états sont atténuées

EPG with anisotropic diffusion, Weigel et al.
Journal of Magnetic Resonance 205(2). 2010

Relaxation-échange

Deux compartiments avec échanges chimiques

La relaxation et l'échange sont simultanés

EPG for systems with MT & exchange, Malik et al.
Magnetic Resonance in Medicine 80(2). 2018

EPG – Rôle et stockage des états

Les états transversaux F_+ et F_- permettent de revenir à l'aimantation transversale

Lorsqu'une séquence est au centre de l'espace k (présence d'un écho), $M_{\perp} = F_+(0) = F_-(0)$

En simulation numérique, l'espace k du modèle EPG est discrétisé : les ordres k de $F(k)$ et $Z(k)$ sont discrétisés

⇒ Les aires des gradients simulés sont discrétisées

	EPG régulier	EPG discret
Ordres stockés	Tous	Population non-nulle
Stockage	Simple (tableau)	Complexe (table de hachage)
Utilisation principale	Gradients multiples les uns des autres	Toutes les situations

Simulation par intervalle de temps court – Spatial vs. EPG

Spatial

Facile à comprendre, facile à coder

Nombre explicite de positions simulées

Adapté à une explication de l'équation de Bloch

EPG

Facile d'utilisation

Nombre implicite d'états (dépend de la séquence)

Très flexible

Outils

De nombreux logiciels sont disponibles

Focus sur quatre, licences libres et biens documentés

	Modèle	Langage	Diffusion	Échange
Jemris	Éq. diff.	MATLAB (interface) C++ (calcul)	Non	Non
MRZero	Temps court (EPG)	Python (interface) C++ (PyTorch, calcul)	Isotrope uniquement	Non
EPG-X	Temps court (EPG)	MATLAB	Non	Oui
Sycomore	Temps court (spatial, EPG)	Python (interface) C++ (interface & calcul)	Oui	Oui

Exemple avec Sycomore

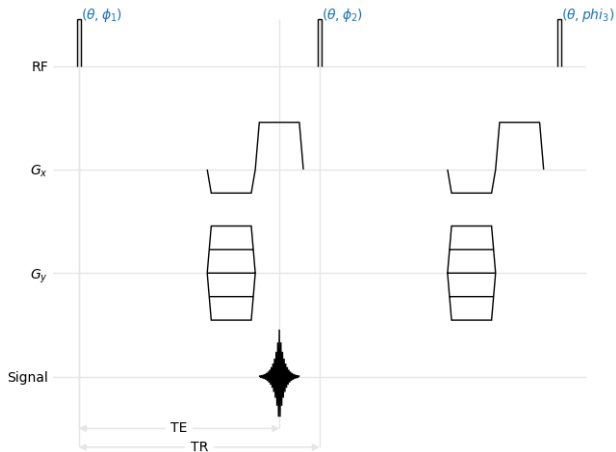
```
import numpy
import sycomore
from sycomore.units import *

species = sycomore.Species(1000*ms, 100*ms)
TE, TR = 20*ms, 25*ms

model = sycomore.epg.Discrete(species)
echo = numpy.empty(repetitions, complex)
for r in range(repetitions):
    model.apply_pulse(flip_angle, phase)
    model.apply_time_interval(before_readout)
    model.apply_time_interval(to_begin)
    model.apply_time_interval(half_readout)

    echo[r] = model.echo

    model.apply_time_interval(half_readout)
    model.apply_time_interval(after_readout)
```



Conclusion

Modèles de simulation

- Zoom sur les modèles par intervalle de temps court (spatial, EPG)
- D'autres existent, par exemple *Configuration Model* (C. Ganter), PDG (MRZero)

Références

- Formes matricielles : *Transient Response in Steady-State MR Imaging*, Hargreaves et al. *Magnetic Resonance in Medicine* 46(1), 2001.
- EPG : *Extended phase graphs*, Weigel. *Journal of Magnetic Resonance Imaging* 41(2), 2015.
- PDG : *Phase distribution graphs for fast, differentiable, and spatially encoded Bloch simulations of arbitrary MRI sequences*, Endres et al. *Magnetic Resonance in Medicine* 92(3), 2024
- Approches « historiques »
 - Effects of Diffusion in Nuclear Magnetic Resonance Spin-Echo Experiments*, Woessner, *J. Chem. Phys.*, 1961
 - Diffusion and field-gradient effects in NMR Fourier spectroscopy*, Kaiser et al. *J. Chem. Phys.*, 1974
 - Echoes—how to generate, recognize, use or avoid them in MR-imaging sequences*, Hennig, *Concepts in MR*, 1991