MRI Relaxometry and CONTIN

Joseph P. Hornak, Ph.D.
Professor of Chemistry & Imaging Science
Rochester Institute of Technology

E-mail: jphsch@rit.edu
Voice: 585 475-2904
Office: 76-2132

MRI

Magnetic Resonance Imaging

An imaging technique used to diagnose disease in the human body.
Relaxometry

The study of the spin relaxation rates of tissues of the human body.
\[ R_1 = \text{Spin-lattice relaxation rate} \]
\[ R_2 = \text{Spin-spin relaxation rate} \]

Why study \( R_1 \) and \( R_2 \)?
Spin relaxation rates are a measure of the mobility of molecules. The mobility of molecules in a tissue changes with disease state. Therefore, \( R_1 \) and \( R_2 \) change with disease state.
CONTIN

A computer package written by Steven W. Provencher in ~ 1979 for computing the inverse Laplace transform of a function.

References

Inverse Laplace Transform (ILT)

\[ S(t) = \sum_i k_i e^{-iR_{ij}} \]
Example

\[ S(t) = 2e^{-t^2} + 4e^{-10t} + 6e^{-20t} \]

The MRI Challenge

The ILT and CONTIN are numerically intensive calculations.

- Eliminate unnecessary parts?

Magnetic resonance images can have 512x512 pixels.

- Parallel computing.

Most valuable MRI data is \( R_1 \), which is an exponential growth, and CONTIN requires exponential decays.

- Convert growth to decay.
Converting an Exponential Growth to Decay.

Available MRI Data

\[ S(t) = \sum_{i} k_i (1 - 2e^{-r_i t}) \]

Desired Data

\[ S(t) = \sum_{i} k_i e^{-r_i t} \]

Problem: Uncertainty in \( S(t) \) causes large errors in the computed \( R_i \).

Solution: Add an offset to the data which will be treated by the ILT as a short \( R_i \).
This additional \( R_i \) is outside of the range of the normal \( R_i \) values from the tissues.

The Project

Compute \( k(R_i) \) for a set of magnetic resonance images using parallel computing.

Needed Pieces:
- CONTIN Parameters
- Magnetic Resonance Images
- Mask Image

Desired Output:
- \( k(R_i) \)
Images

Symbolism:
- pixels \((i,j)\)
- \(m\) images at different \(t\)
- pixel intensity \(S(m,i,j)\)

Delay Times for Images

<table>
<thead>
<tr>
<th>File</th>
<th>Time (s)</th>
<th>File</th>
<th>Time (s)</th>
<th>File</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs-6 raw</td>
<td>0.0365</td>
<td>cs-16 raw</td>
<td>0.0025</td>
<td>cs-32 raw</td>
<td>2.5345</td>
</tr>
<tr>
<td>cs-1 raw</td>
<td>0.0265</td>
<td>cs-17 raw</td>
<td>0.0025</td>
<td>cs-33 raw</td>
<td>2.7885</td>
</tr>
<tr>
<td>cs-2 raw</td>
<td>0.1065</td>
<td>cs-18 raw</td>
<td>0.0025</td>
<td>cs-34 raw</td>
<td>2.9455</td>
</tr>
<tr>
<td>cs-3 raw</td>
<td>0.1455</td>
<td>cs-19 raw</td>
<td>0.0025</td>
<td>cs-35 raw</td>
<td>3.1545</td>
</tr>
<tr>
<td>cs-4 raw</td>
<td>0.1625</td>
<td>cs-20 raw</td>
<td>0.0025</td>
<td>cs-36 raw</td>
<td>3.3425</td>
</tr>
<tr>
<td>cs-5 raw</td>
<td>0.2205</td>
<td>cs-21 raw</td>
<td>0.0025</td>
<td>cs-37 raw</td>
<td>3.5205</td>
</tr>
<tr>
<td>cs-6 raw</td>
<td>0.2645</td>
<td>cs-22 raw</td>
<td>0.0025</td>
<td>cs-38 raw</td>
<td>3.7045</td>
</tr>
<tr>
<td>cs-7 raw</td>
<td>0.2955</td>
<td>cs-23 raw</td>
<td>0.0025</td>
<td>cs-39 raw</td>
<td>4.0055</td>
</tr>
<tr>
<td>cs-8 raw</td>
<td>0.3455</td>
<td>cs-24 raw</td>
<td>0.0025</td>
<td>cs-40 raw</td>
<td>4.2855</td>
</tr>
<tr>
<td>cs-9 raw</td>
<td>0.4025</td>
<td>cs-25 raw</td>
<td>0.0025</td>
<td>cs-41 raw</td>
<td>4.5335</td>
</tr>
<tr>
<td>cs-10 raw</td>
<td>0.4545</td>
<td>cs-26 raw</td>
<td>0.0025</td>
<td>cs-42 raw</td>
<td>4.9455</td>
</tr>
<tr>
<td>cs-11 raw</td>
<td>0.5085</td>
<td>cs-27 raw</td>
<td>0.0025</td>
<td>cs-43 raw</td>
<td>5.3205</td>
</tr>
<tr>
<td>cs-12 raw</td>
<td>0.5645</td>
<td>cs-28 raw</td>
<td>0.0025</td>
<td>cs-44 raw</td>
<td>5.7225</td>
</tr>
<tr>
<td>cs-13 raw</td>
<td>0.6245</td>
<td>cs-29 raw</td>
<td>0.0025</td>
<td>cs-45 raw</td>
<td>6.1455</td>
</tr>
<tr>
<td>cs-14 raw</td>
<td>0.6835</td>
<td>cs-30 raw</td>
<td>0.0025</td>
<td>cs-46 raw</td>
<td>6.6555</td>
</tr>
<tr>
<td>cs-15 raw</td>
<td>0.7435</td>
<td>cs-31 raw</td>
<td>0.0025</td>
<td>cs-47 raw</td>
<td>7.1655</td>
</tr>
</tbody>
</table>

J. Hornak, March 2004
**Mask Image**

Format:
- Binary
- 64x64 pixels
- 16 bits per pixel
- Values of 1 and 0

Symbolism:
- mask(i,j)

Compute CONTIN output only for pixels with value of mask(i,j)=1.

---

**Input Parameters**

Test Data - Charles Springer’s Images

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMNMX</td>
<td>1</td>
</tr>
<tr>
<td>GMNMX</td>
<td>2</td>
</tr>
<tr>
<td>NG</td>
<td>300</td>
</tr>
<tr>
<td>NINTT</td>
<td>0</td>
</tr>
<tr>
<td>IFORMT</td>
<td>(F9.6)</td>
</tr>
<tr>
<td>IFORMY</td>
<td>(F8.6)</td>
</tr>
<tr>
<td>IUSER</td>
<td>10</td>
</tr>
<tr>
<td>NONNEG</td>
<td>1</td>
</tr>
<tr>
<td>NY</td>
<td>64</td>
</tr>
</tbody>
</table>

Title
- Minimum \( R_1 \) Range
- Maximum \( R_1 \) Range
- Points between \( R_{1_{\text{min}}} \) and \( R_{1_{\text{max}}} \)
- Unequal spaced points in time
- Input format for time values
- Input format for signal values
- Inverse Laplace transform
- Constrained

Number of input points
Output Detail

CONTIN output \( k(R_i) \) is on a log scale with equal increments in \( \log(R_i) \).

We desire \( k(R_i) \) on a scale linear in \( R_i \).

**Linear Conversion:**
\[
k(R_i) = \delta R_i \, k(R_i)
\]
\[
\delta R_i = R_{\text{in}} - R_{\text{in-1}}
\]

Output also needs to be scaled to a 16 bit integer.

**Scaling:**
\[
k(R_{i,j}) = 32768 \times \frac{k(R_{i,j})}{\text{Max}(k(R_{i,j}))}
\]

Read In Parameters

Read in \( cs-m.raw \) images as \( S(m,i,j) \)

If \( S(m,1,1) \neq 1 \)
    swap bytes of image

Read in mask\((i,j)\)

If mask\((i,j)\) EQ 1
    pass \( S(m) \) and parameters for each \((i,j)\) to CONTIN

Input \( k(R_i) \) for \((i,j)\) from CONTIN

Test Data -- Charles Springer's Images

<table>
<thead>
<tr>
<th>GMNMX</th>
<th>1</th>
<th>1.0E-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMNMX</td>
<td>2</td>
<td>1.0E+4</td>
</tr>
<tr>
<td>NG</td>
<td>300.</td>
<td></td>
</tr>
<tr>
<td>NINTT</td>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>IFORMT</td>
<td>(F9.6)</td>
<td></td>
</tr>
<tr>
<td>IFORMY</td>
<td>(F8.6)</td>
<td></td>
</tr>
<tr>
<td>IUSER</td>
<td>10.</td>
<td></td>
</tr>
<tr>
<td>NONNEG</td>
<td>1.</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NY</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>

See Next Page
Discard $k(R_{i,j})$ for $R_{i,j} \lt 0.1$ (n=1 to 150)

Convert $k(R_{i,j})$ to non log scale

Find max $k(R_{i,j})$

Scale: $k(R_{i,j}) = 32768 \times k(R_{i,j}) / \max$

Set (1,1) pixel of each image to 1

Parallel Option
Read In Parameters

Read in cs-m.raw images as S(m,i,j)

If S(m,1,1) NE 1 swap bytes of image

Read in mask(i,j)

Partition image space for mask(i,j) EQ 1 and pass S(m) and parameters for each (i,j) to CONTIN on machine #n

CONTIN on #1

CONTIN on #1

CONTIN on #n

CONTIN on #1

CONTIN on #1

CONTIN on #1

CONTIN on #1

-input k(R_1) for all (i,j)

Save

From Previous Page

Find max k(R_1,i,j)

Scale: k(R_1,i,j) = 32768 * k(R_1,i,j) /max

Set (1,1) pixel of each image to 1

Save
A few more details…

Image size: \((i,j) = (64,64)\)
Number of input images: \(m = 64\)  
Number of saved \(R\) images: 150  
CONTIN Variable = NY  
\([R_i = 0.102 \text{ to } 100 \text{ s}^{-1}]\)  
\([\text{CONTIN NG = 151 to 300}]\)

CIS Cluster
36 node dedicated MPI (condor) cluster of Sun blade 100’s,  
HPC 5.0, Solaris 9, Forte 7