更多数学建模资料获取

B 站数学建模试看课程：

https://www.bilibili.com/video/av20238704

清风数学建模视频课程的课表：

http://note.youdao.com/noteshare?id=2e1cc342928466bb632b33304a56e368

视频录制了 45 个小时，累计 800 多页 PPT，讲解的了 20 多个常用模型，内容非常详细，并配有作业讲解。

关注微信公众号：《数学建模学习交流》可获得更多数学建模技巧和比赛信息。

怎么在手机上购买资料：在公众号后台发送“买”这个字，就可以获得一个链接，点进去后就可以购买了（购买方式和淘宝一样）

A Method for Taking Cross Sections of Three-Dimensional Gridded Data

Kelly Slater Cline
Kacee Jay Giger
Timothy O'Conner
Eastern Oregon University
LaGrande, OR 97850

Advisor: Norris Preyer

Summary

Effective three-dimensional magnetic resonance imaging (MRI) requires an accurate method for taking planar cross sections. However, if an oblique cross section is taken, the plane may not intersect any known data points. Thus, a method is needed to interpolate water density between data points.

Interpolation assumes continuity of density, but there are discontinuities in the human body at the borders of different types of tissue. Most interpolation methods try to smooth these sharp borders, blurring the data and possibly destroying useful information.

To capture qualitatively the key difficulties of this problem, we created a sequence of simulated biological data sets, such as a brain and an arm, each with some specific defect. Our data sets are cubic arrays with 100 elements on each side, for a total of one million elements, specifying water density at each point with an integer in the range $[0, 255]$. In each data set, we use differentiable functions to describe several tissue types with discontinuities between them.

To analyze these data, we created a group of algorithms, implemented in C++, and compared their effectiveness in generating accurate cross sections. We used local interpolation techniques, because the data are not continuous on a global level. Our final algorithm searches for discontinuities between tissues. If it finds one at a point, it preserves sharp edges by assigning to that point the water density of the nearest data point. If there is no discontinuity, the algorithm does a polynomial fit in three dimensions to the nearest 64 data points and interpolates the water density.

The UMAP Journal 19 (3) (1998) 211–221.

We measured the accuracy of the algorithms by finding the mean absolute difference between the interpolated water density and the actual water density at each point in the cross sections. Our final algorithm has an error 16% lower than a simple closest-point technique, 17% lower than a continuous linear interpolation, and 22% lower than a continuous polynomial interpolation without discontinuity detection.

Assumptions

- An MRI scan is an equally spaced grid of data. We take it to be a $100\times100\times100$ array.
- Each element of the array is an integer ranging from 0 to 255, representing the water density at that point.
- The resolution of the cross section to be taken is equal to the resolution of the data set. (If the array elements have a spacing of one micron, then the cross section should have the same spacing.)
- We recognize that all methods of interpolation assume continuity between the data points. Thus, we assume that the water density in living tissue can be represented as continuous differentiable functions with discontinuities between tissues.

Simulated Data Sets

Since we could find few existing data sets of three-dimensional arrays, we constructed simulated data sets. While real biological organs are extraordinarily complicated, a set of simulated organs should be able to represent qualitatively the kind of problems that an MRI scan is typically used to investigate. Although actual MRI data have much greater resolution, the characteristics that we are looking for should be the same: tumors, fractures, or general anomalies. Generally, these areas will have different water densities than surrounding tissue, generating discontinuities. We created the following mock organs with imperfections:

1. **Globules**: A continuous, repeating spherical pattern, with a density peak at the center (**Figure 1**).
2. **Arm**: Smooth tissue with two bones, one containing a small spherical hole (**Figure 2**).
3. **Generic organ**: A round shape filled with several discontinuous regions (**Figure 3**).
4. **Brain**: A dense skull, periodically varying gray matter, and a small area of different density in one lobe (**Figure 4**).

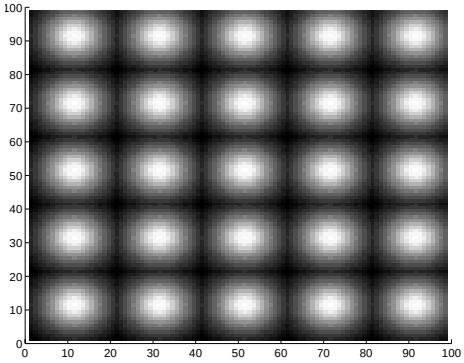

Figure 1. Globules.

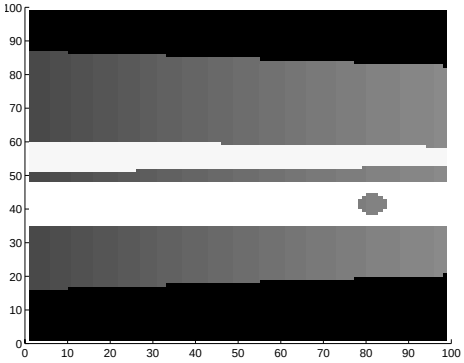

Figure 2. Arm.

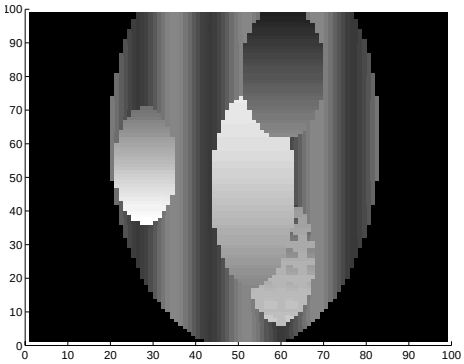

Figure 3. Generic organ.

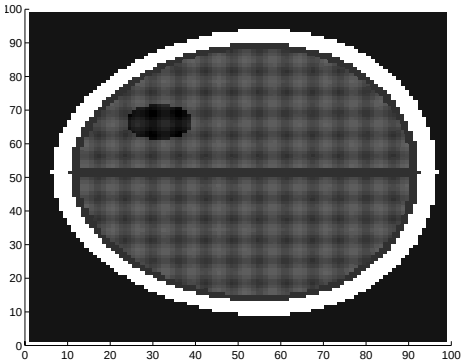

Figure 4. Brain.

Coordinate Systems and Definitions

The existing array of data imposes a Cartesian coordinate system on the problem. If there are n data points in each direction, then the coordinates range from $(0, 0, 0)$ to (n, n, n). We define a cross section by picking a point in this coordinate system (x_0, y_0, z_0) and two angles (θ, ϕ) representing the angles that the plane makes with the positive x-axis and the positive y-axis. This point becomes the origin of our plane, with the new x-axis being the projection of the x-axis onto this plane in the z-direction, so the unit vector is

$$\hat{x}' = \hat{x}\cos\theta + \hat{z}\sin\theta.$$

We can solve for the unit vector $\hat{y}'$ if we require it to be orthogonal to $\hat{x}'$, to make an angle ϕ with the unit vector $\hat{y}$, and to be of unit length, so

$$\hat{y}' = -\hat{x}\sin\phi\sin\theta + \hat{y}\cos\phi + \hat{z}\sin\phi\cos\theta.$$

Thus, we can convert from the (x', y') coordinate system back to the array system as:

$$\begin{aligned} x &= x_0 + x' \cos\theta - y' \sin\phi \sin\theta \\ y &= y_0 \qquad\qquad + y' \cos\phi \\ z &= z_0 + x' \sin\theta + y' \sin\phi \cos\theta \end{aligned}$$

We call the known points (x, y, z) *data points* and the unknown points (x', y') *plane points*.

Interpolation Algorithms

The plane points do not generally match existing data points. We know the water density surrounding each plane point, so we must interpolate to estimate plane point density.

There are two major classes of interpolation techniques:

- global methods, which use every data point in the set to estimate the density at each plane point, and
- local methods, which only use a small subset of the data points.

Because interpolation methods assume continuity, global methods are inappropriate to this problem. We know that the organs are only piecewise continuous and differentiable with discontinuities between tissues. Thus, all of our algorithms use local interpolation techniques.

Proximity

This algorithm assigns to the plane point the density of the data point that is closest to the plane point. This method seems naive, but it should preserve sharp edges without blurring. It looks at each point in the plane (x', y'), calculates x, y, and z in the original array, and rounds them to integer values (X, Y, Z), thus giving the closest data point.

Density Mean

This method uses more information to estimate the water density at each point. We can visualize every plane point as being inside a cube, with data points at the corners. To estimate the value inside, we take the arithmetic mean of the density from the surrounding eight points. Despite the use of more information, this method blurs the edges of discontinuities.

Trilinear Interpolation

This algorithm uses the same eight points as the density mean method but does a weighted average of the density (ρ) values. This method assumes that the slope $d\rho/dx$ is constant between the data points. With a low-resolution dataset, this approach will create inaccuracies; but as the resolution increases, the slope will appear to be more constant, since any differentiable function appears linear when examined on a small enough scale. The formula for linear interpolation [Press 1988, 104–105] is

$$\rho(x') = \sum_{i=1..2} (1 - T_i)\rho_i,$$

where $T_i = |x' - x_i|$.

The weight that we give to the value ρ_i is equivalent to the distance to the opposite point. Here, every pair of consecutive data points has a distance of 1 unit between them, so the distance to the opposite point is $1 - T_i$. For trilinear interpolation, we extend this sum over all the points, so that

$$\rho(x', y', z') = \sum_{i=1..2} \sum_{j=1..2} \sum_{k=1..2} (1 - T_i)(1 - U_j)(1 - V_k)\rho_{ijk},$$

where $T_i = |x' - x_i|$, $U_j = |y' - y_j|$, and $V_k = |z' - z_k|$.

Polynomial Interpolation

To estimate better the water density function, the polynomial interpolation method uses even more data. We expand the surrounding cube of eight points in every direction to make a cube with four points on each side, getting the 64 nearest points. Polynomials can fit differentiable functions better because they have more derivatives and can incorporate larger trends in the function. Recall that two points determine a unique line, three points determine a unique quadratic, and four points determine a unique cubic. Making use of this, we can develop a method for fitting functions in three dimensions. By doing a sequence of fits in the x, y, and z directions, we can synthesize these into a density estimate for a point in space. Thus, we break the problem into a series of one-dimensional interpolations.

Let (x, y, z) be the plane point, and let the data points be $(x_{1..4}, y_{1..4}, z_{1..4})$. First, we fix x_1 and y_1, fit the four points $(x_1, y_1, z_{1..4})$ to a cubic, and interpolate the density at (x_1, y_1, z). We increment to x_1, y_2 and do the same until we have the densities at the points (x_1, y_1, z), (x_1, y_2, z), (x_1, y_3, z), (x_1, y_4, z). Then we fit a polynomial in the y-direction to these four points and interpolate the density at (x_1, y, z). We repeat this whole process to find (x_2, y, z), (x_3, y, z), and (x_4, y, z), and then perform one last polynomial fit to these points to interpolate the density at (x, y, z).

There are many techniques for doing polynomial fits. We used the Lagrange formula [Acton 1990, 96] because it is the least computationally intensive:

$$\rho(x') = \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)}\,\rho_1 + \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)}\,\rho_2 + \frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)}\,\rho_3 + \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)}\,\rho_4,$$

where ρ_i is the density at x_i.

This method will blur edges but should do very well over regions described by differentiable functions.

Hybrid Algorithms

All of the above methods have strengths and weaknesses. The methods that are strongest on the differentiable regions (trilinear, polynomial) are weakest at discontinuities, because they try to smooth out the sharp borders. The method that most closely preserves discontinuities (proximity) is weakest at identifying smooth trends in the functions.

To capitalize on the strengths of both approaches, we created a hybrid algorithm. Before interpolating between a group of points, the hybrid looks for discontinuities within them; it uses the proximity method if it finds any and a continuous method otherwise. This hybrid algorithm locates discontinuities by measuring the difference in density ($\Delta\rho$) between each pair of extreme opposite points surrounding the plane point. If there is a discontinuity, then $\Delta\rho$ will be large and we use the proximity method. If not, $\Delta\rho$ will be small and we use a continuous method, either trilinear or polynomial. To distinguish between the two cases, we set the threshold value of $\Delta\rho_0$. Thus, the hybrid algorithm allows us to use each method where it is strongest.

Testing and Results

Because we have defined precise water density functions for each of our four simulated data sets, we can compare the interpolation value with the actual density and find the residual. To measure the accuracy of a cross section, we calculate the mean absolute residual over all of the plane points (i.e., the average of the absolute values of the residuals).

To compare the algorithms, we took 12 cross sections through each simulated data set, at different angles, points, and discontinuity threshold levels. We generally selected points near the center of the data so as to generate a large planar region.

Table 1 shows the mean absolute residual for each algorithm applied to each data set and averaged over all data sets. We discuss the results on each of the data sets (columns in **Table 1**) in turn.

Table 1.
Mean absolute residuals for interpolation methods applied to the data sets.

Algorithm	Data Set				
	Globules	Arm	Generic organ	Brain	Combined
Proximity	1.82	0.97	0.89	3.25	1.73
Density mean	1.39	1.89	1.34	5.52	2.53
Trilinear interpolation	1.54	1.27	0.70	3.49	1.75
Polynomial interpolation	1.55	1.48	0.75	3.67	1.86
Hybrid trilinear interpolation					
($\Delta\rho_0 = 20$)	1.55	1.01	0.59	3.49	1.49
($\Delta\rho_0 = 30$)	1.54	1.01	0.59	2.82	1.49
($\Delta\rho_0 = 40$)	1.54	1.01	0.61	2.82	1.50
Hybrid polynomial interpolation					
($\Delta\rho_0 = 20$)	1.66	0.99	0.71	3.14	1.62
($\Delta\rho_0 = 30$)	1.63	0.99	0.61	2.86	1.52
($\Delta\rho_0 = 40$)	1.61	0.99	0.53	2.56	1.45

Globules

On average, the mean method most accurately generated oblique planes, with both the trilinear and polynomial interpolations providing cross sections of similar accuracy. For this data set, the hybrid algorithms did worse than the purely continuous methods (**Figure 5**); the proximity method did poorest of all, probably because there are no edges in the globule data sets and continuity is never broken. The trilinear and polynomial interpolations may also have had trouble with the peaks in the center of each sphere.

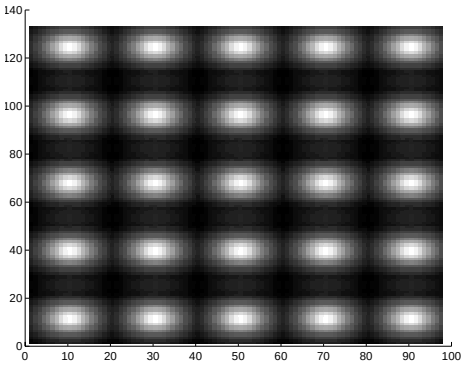

Figure 5. Globules: Polynomial hybrid method, with $\theta = 45°$, $\phi = 0°$, $(45, 45, 50)$.

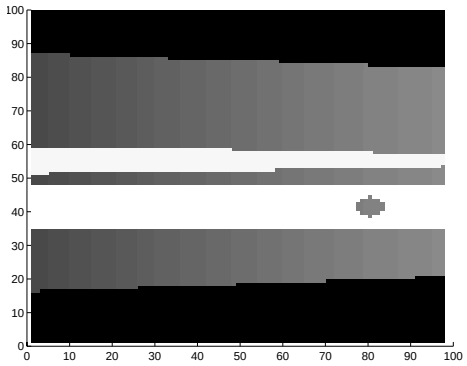

Figure 6. Arm: Polynomial hybrid method, with $\theta = 10°$, $\phi = 0°$, $(40, 80, 50)$.

Arm

Here we found the situation almost totally reversed. The proximity method and the hybrid algorithms (**Figure 6**) all performed significantly better than the purely continuous methods, with the mean method doing particularly badly.

This is quite reasonable, because our arm has many very sharp edges as we go from bone to muscle. The mean method should fail on any discontinuities, and this is what we see. The discontinuity detection seems to be working, because the hybrid algorithms perform noticeably better than the trilinear and polynomial methods.

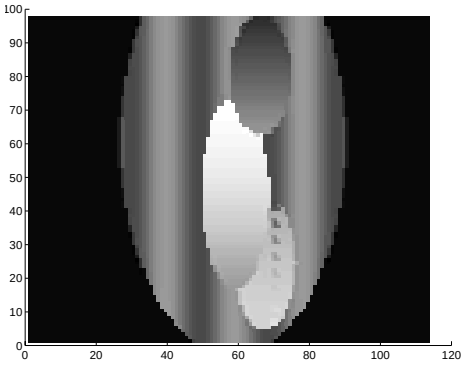

Figure 7. Generic organ: Polynomial hybrid method, with $\theta = 0°$, $\phi = 30°$, $(50, 50, 50)$.

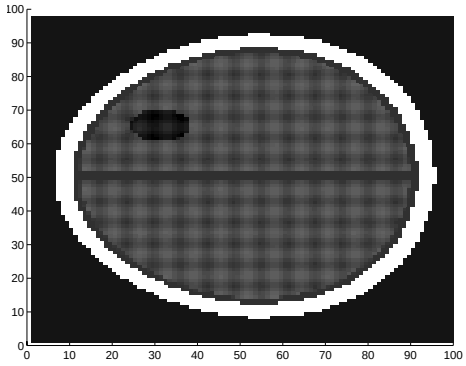

Figure 8. Brain, polynomial hybrid method, $\theta = 5°$, $\phi = 0°$, $(50, 50, 50)$.

Generic Organ

We found that both of the hybrid formulas, trilinear and polynomial, produced equally favorable results (**Figure 7**). Once again, the arithmetic mean method performed poorly, followed by the proximity, polynomial, and trilinear methods, whose results were comparable. We suspect that this is the case because for this data set we used smooth functions to generate each of the different tissue types.

Brain

Due to the general smoothness of each lobe and the sharp contrast of the skull, the hybrid polynomial method with a high value of $\Delta\rho_0$ produced the most accurate results. The proximity method produced very accurate results, surpassing the trilinear and polynomial methods but falling short of the hybrid methods (**Figures 8–10**). The most inaccurate results were produced by the arithmetic mean method, and **Figure 11** clearly shows how it fails by trying to smooth the edges of the skull.

Residual Plots

Another way to examine the algorithms is to plot the residuals, which allows us to see exactly where the a method breaks down. In **Figures 12–14**, large positive or negative residuals stand out in white or gray.

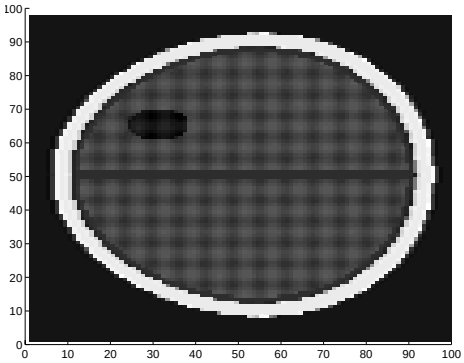

Figure 9. Brain, polynomial method, $\theta = 0°$, $\phi = 30°$, $(50, 50, 50)$.

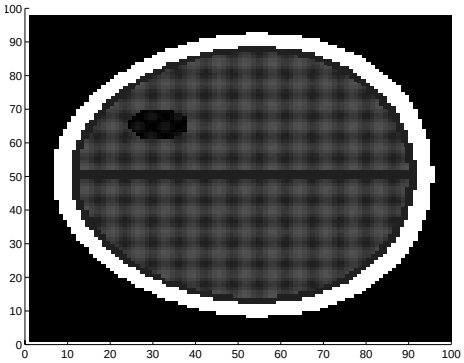

Figure 10. Brain, proximity method, $\theta = 5°$, $\phi = 0°$, $(50, 50, 50)$.

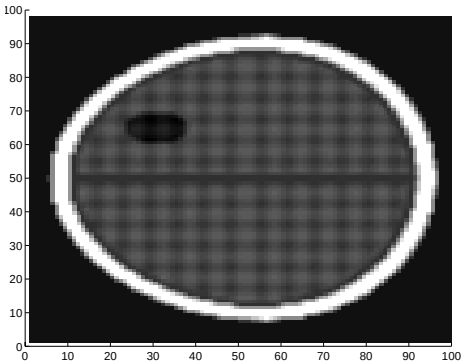

Figure 11. Brain, density mean, $\theta = 0°$, $\phi = 30°$, $(50, 50, 50)$.

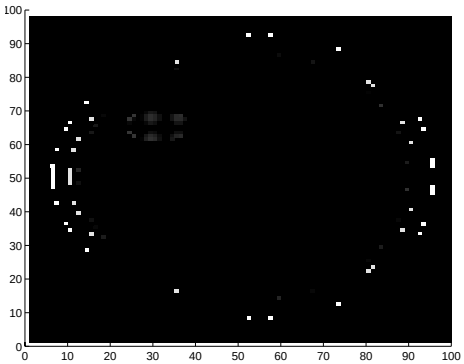

Figure 12. Brain, proximity residuals, $\theta = 5°$, $\phi = 0°$, $(50, 50, 50)$.

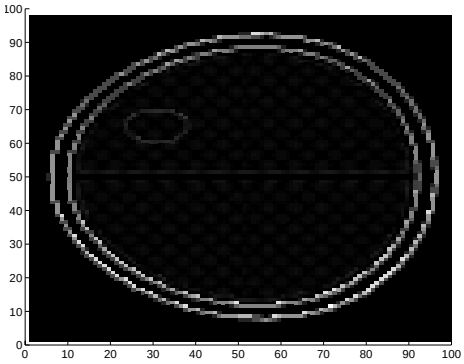

Figure 13. Brain, density mean, residuals, $\theta = 0°$, $\phi = 30°$, $(50, 50, 50)$.

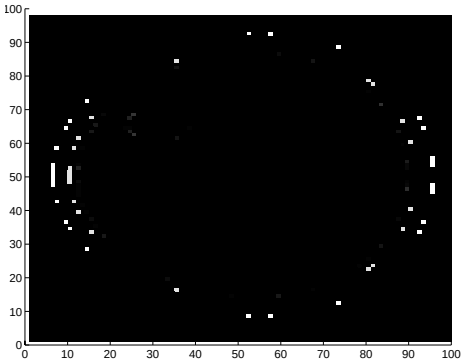

Figure 14. Brain, polynomial hybrid method, residuals, $\theta = 5°$, $\phi = 0°$, $(50, 50, 50)$.

The proximity method was generally accurate over most of the brain, except for the dark area in one of the lobes and a few areas where the sharp discontinuity of the skull also produced errors (**Figure 12**).

The arithmetic mean algorithm clearly shows errors around all the edges (**Figure 13**).

Finally, the hybrid polynomial algorithm has inaccuracies near the edge of the skull, but it handles the dark area very well (**Figure 14**).

Overall Results

- Averaging over all four data sets (see the last column of **Table 1**), the most accurate algorithm for generating oblique planes through arrays of three-dimensional data is the hybrid polynomial algorithm (with $\Delta\rho_0 = 40$), which produced an average error 16% less than the proximity method.
- The hybrid trilinear algorithm (with $\Delta\rho_0 = 30$) did almost as well, generating an average error 14% less than the proximity method.
- The proximity algorithm produced reasonably accurate planes, with average error 7% less than the continuous polynomial algorithm and about 1% better than the continuous trilinear algorithm.
- The trilinear algorithm performed better than the polynomial method, but neither handled discontinuities well enough to produce results as good as the hybrid methods.
- The arithmetic mean algorithm produced particularly poor results, because of the large errors that it makes around a discontinuity of any kind.

Strengths and Weaknesses

The hybrid polynomial algorithm is flexible and could easily be expanded to handle data arrays of any size. It can take a slice through any point at any angle. Moreover, it is effective at interpolating through smooth regions but still preserves the sharpness of edges in the original data. Even if the closest point is on the wrong side of the discontinuity, the image is still qualitatively correct: It shows a sharp edge. The threshold constant $\Delta\rho_0$ allows the user to choose how much smoothing is done.

The hybrid algorithm will miss small discontinuities in the data, and it does not look for nondifferentiable points. If there are cusps, the algorithm will probably not notice them and attempt to smooth through these points, even though to do so is inappropriate.

Future Work

The next step is to implement better methods for interpolating in the continuous regions, using larger sets of points. Cubic and quartic splines might be effective, as might other types of polynomials, or perhaps a method of rational function interpolation.

The discontinuity detection algorithm could be improved by expanding it to look for cusps and nondifferentiability. The discontinuities could also be used to perform automatic tissue typing; the algorithm might then be able automatically to output images showing just brain gray matter, or showing just tumor tissue. Even with the current software, a more detailed investigation of the dynamics of $\Delta\rho_0$ would be very useful.

Most important, the algorithm needs thorough testing against actual MRI data.

References

Acton, Forman S. 1990. *Numerical Methods That Work*. Washington DC: Mathematical Association of America.

Garcia, Alejandro L. 1994. *Numerical Methods for Physics*. Englewood Cliffs, NJ: Prentice-Hall.

Hornak, Joseph P. 1997. The Basics of MRI. `http://www.cis.rit.edu/htbooks/mri/` . (7 February 1998).

Lancaster, Peter, and Kestutis Salkauskas. 1986. *Curve and Surface Fitting*. London: Academic Press.

Press, William H., et al. 1988. *Numerical Recipes in C*. New York: Cambridge University Press.

Summit, Steve. *C Programming FAQs*. 1996. New York: Addison-Wesley.