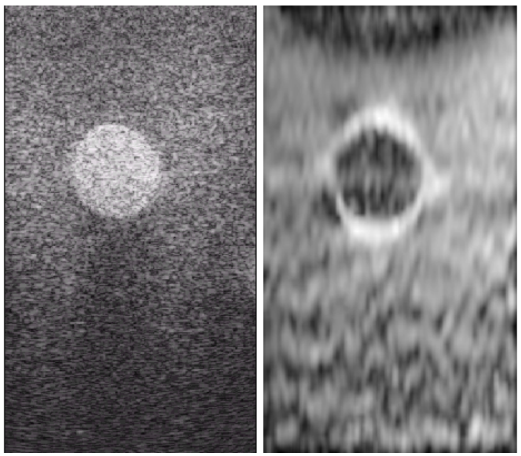
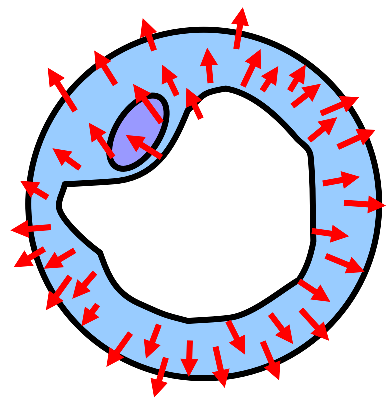
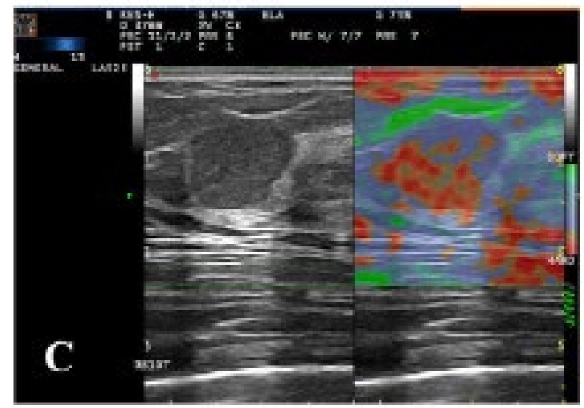
Derivative fun …

Motivation

- Strain Imaging (Differentiation)







Y. Shi, R. S. Witte, M. O'Donnell, 'Identification of vulnerable atherosclerotic plaque using IVUS-based thermal strain imaging', *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 52, no. 5, pp. 844-850, May 2005

W. R. Jia, T. Luo, Y. J. Dong, X. X. Zhang, W. W. Zhan, J. Q. Zhou, 'Breast elasticity imaging techniques: Comparison of strain elastography and shear-wave elastography in the same population', Ultrasound in Med. & Biol., Vol. 47, No. 1, pp. 104-113, 2021.

AHA Atherosclerotic Lesion Characterization

- Types I, II, III, IV, V, VI
  - Types I, II, and III lesions
    - Initial, fatty streak, and intermediate lesions
    - Grow through lipid accumulation
  - Type IV atheroma lesion
    - Lipids pools coalesce to form a core.
  - Type V (fibroatheroma) and VI (complicated)
    - Growth through accelerated smooth muscle growth and collagen deposition by fibroblasts.
    - Type VI (complicated) involves surface defect (thrombosis or hematoma).
    - Type Vb - layer of calcium in fibrous cap.
    - Type VI lesion has three subdivisions:
      - VIa, surface disruption
      - VIb, hematoma or hemorrhage
      - VIc, thrombosis.
- Morbidity and mortality due to IV and V

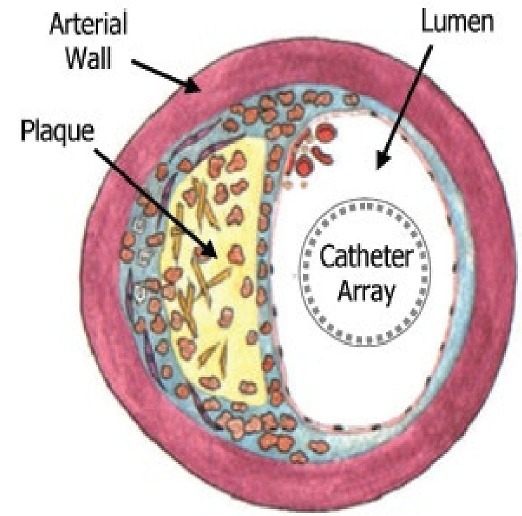
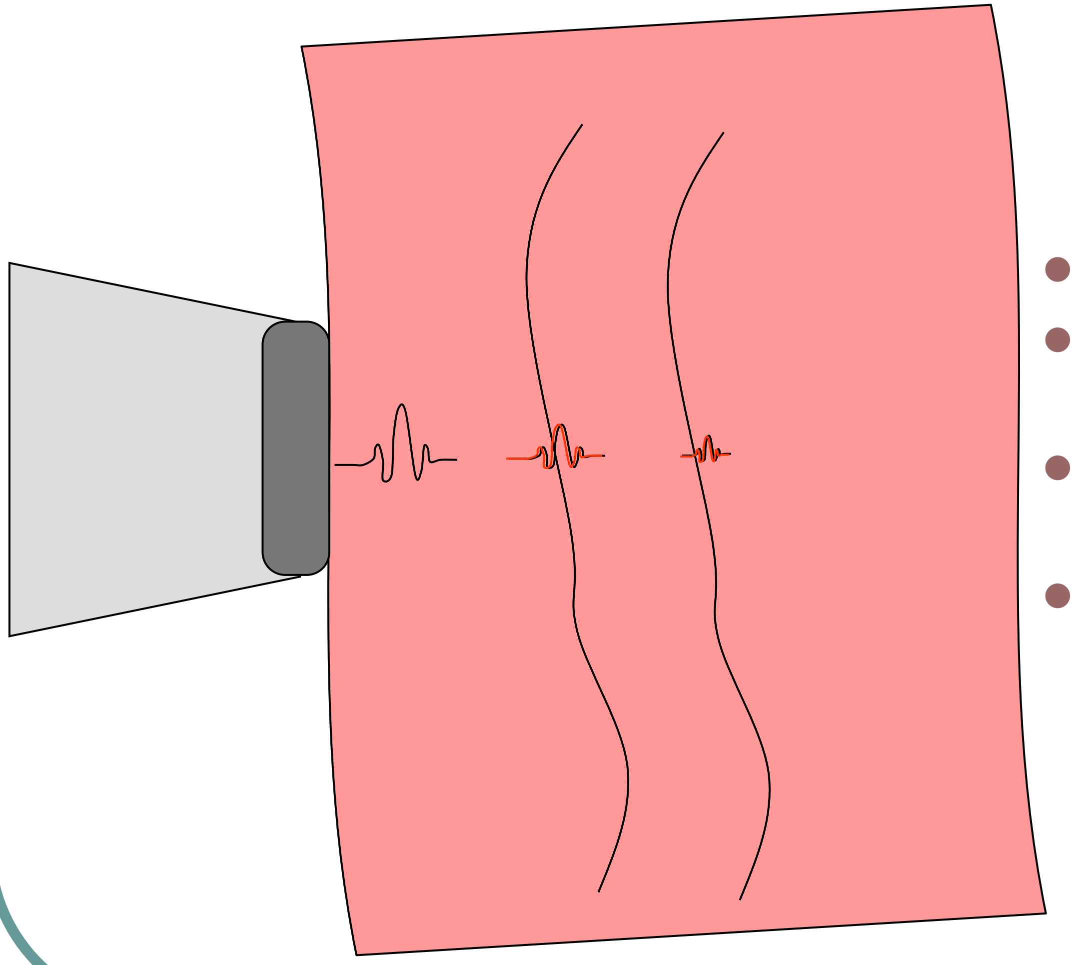


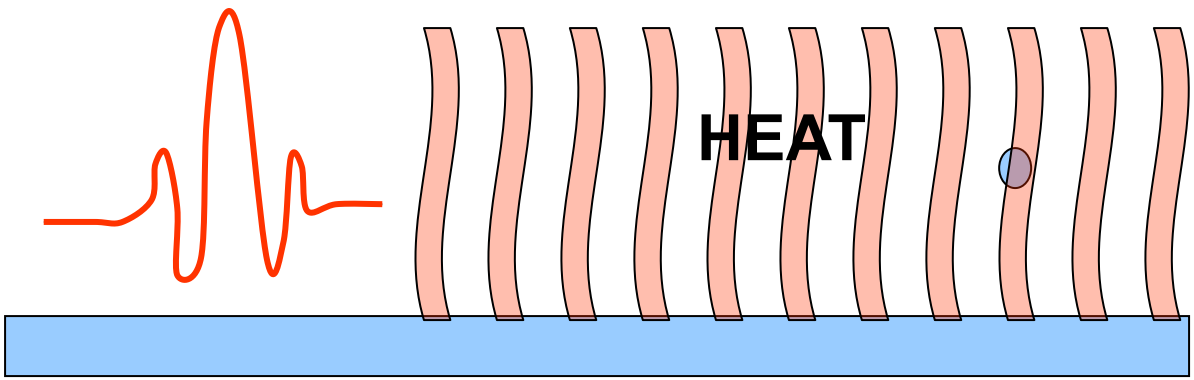
Fig. 1. IVUS imaging of rupture-prone inflamed plaque (adapted from [3]).

Ultrasound basics



- Piezoelectric transducer
- Pulse frequency typically 5- 40 MHz
- Handheld (sector, linear, annular)
- Echo-time and intensity recorded
  - Assuming constant speed of sound, range can be determined

Temporal Strain



Change in speed of sound due to temperature

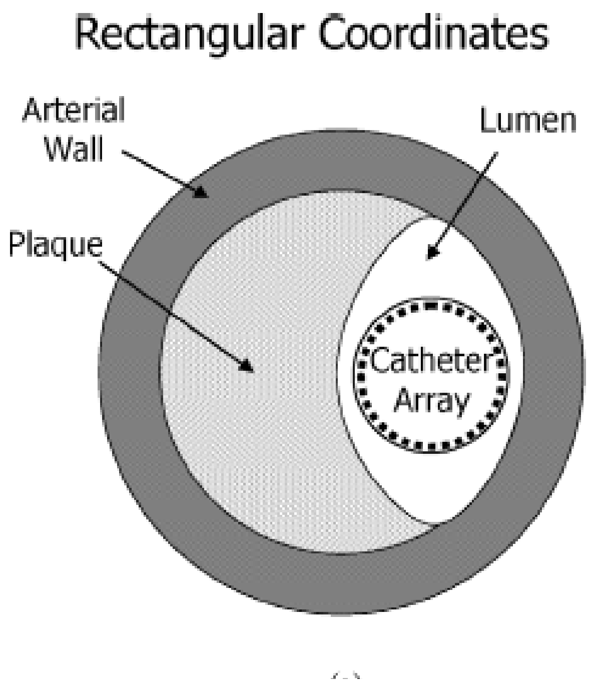
$$\frac{\partial}{\partial t}[\delta t(z)] = (\beta(z) - \lambda(z))\delta\theta(z), \qquad (1)$$

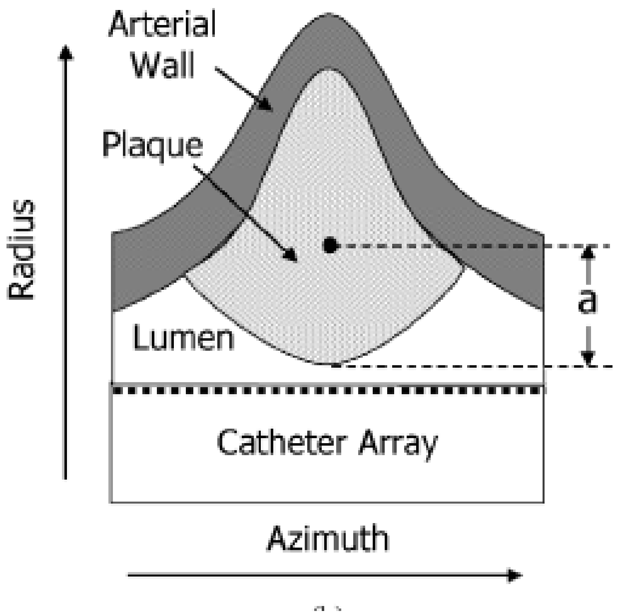
Change in echo time with respect to time

Coefficient of thermal expansion

Temperature change

Simulated Lesion Configuration





Phantom experiment setup

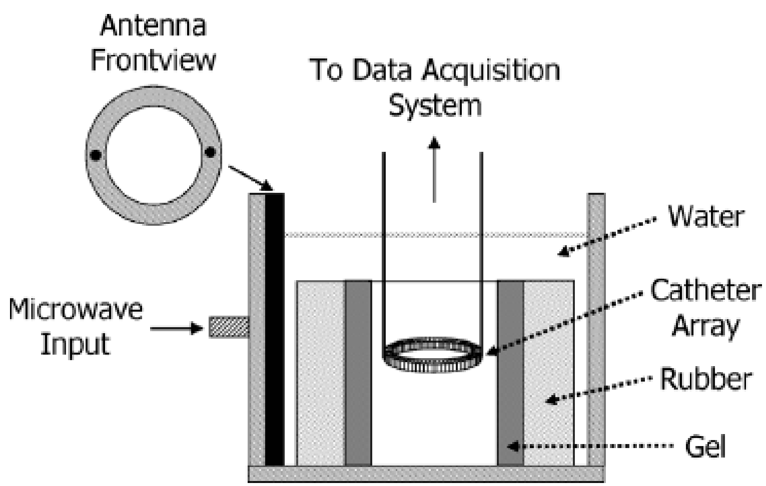


Fig. 3. Illustration of experimental setup.

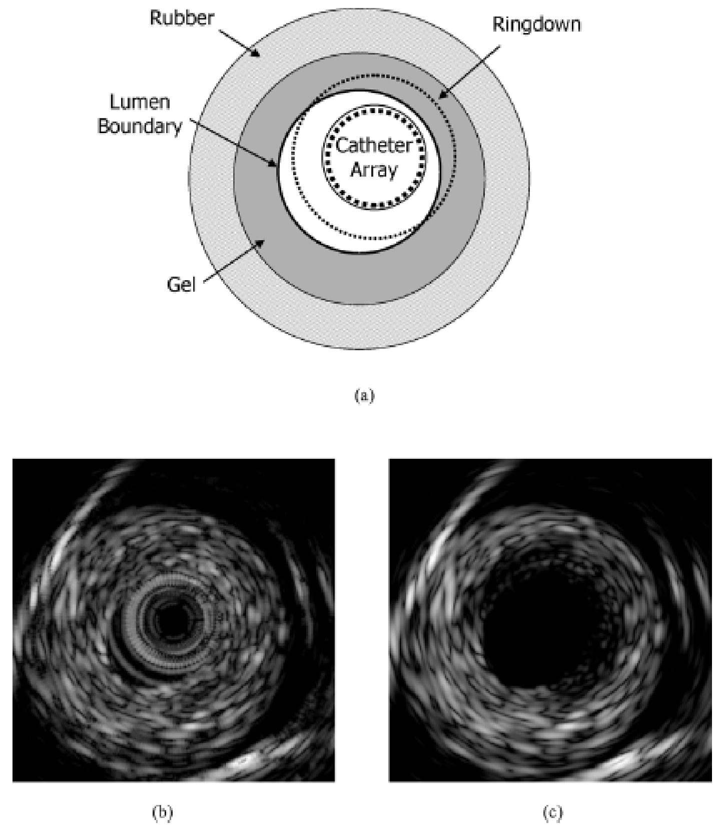


Fig. 4. Illustration of ringdown effects and lumen detection. (a) Phantom geometry. (b) B-scan image with ringdown. (c) Processed B-scan image.

Simulation Results

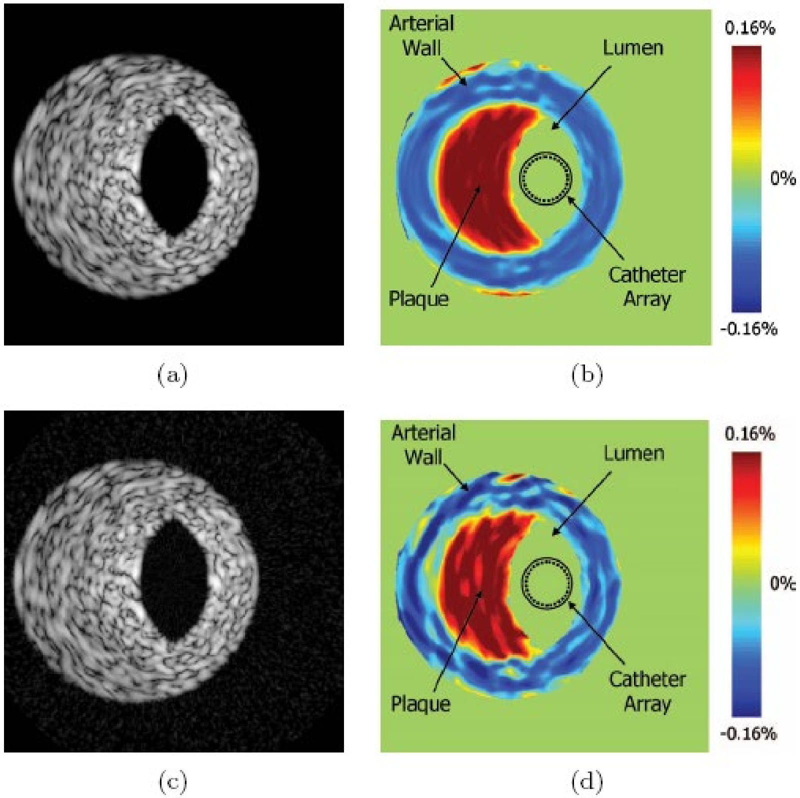


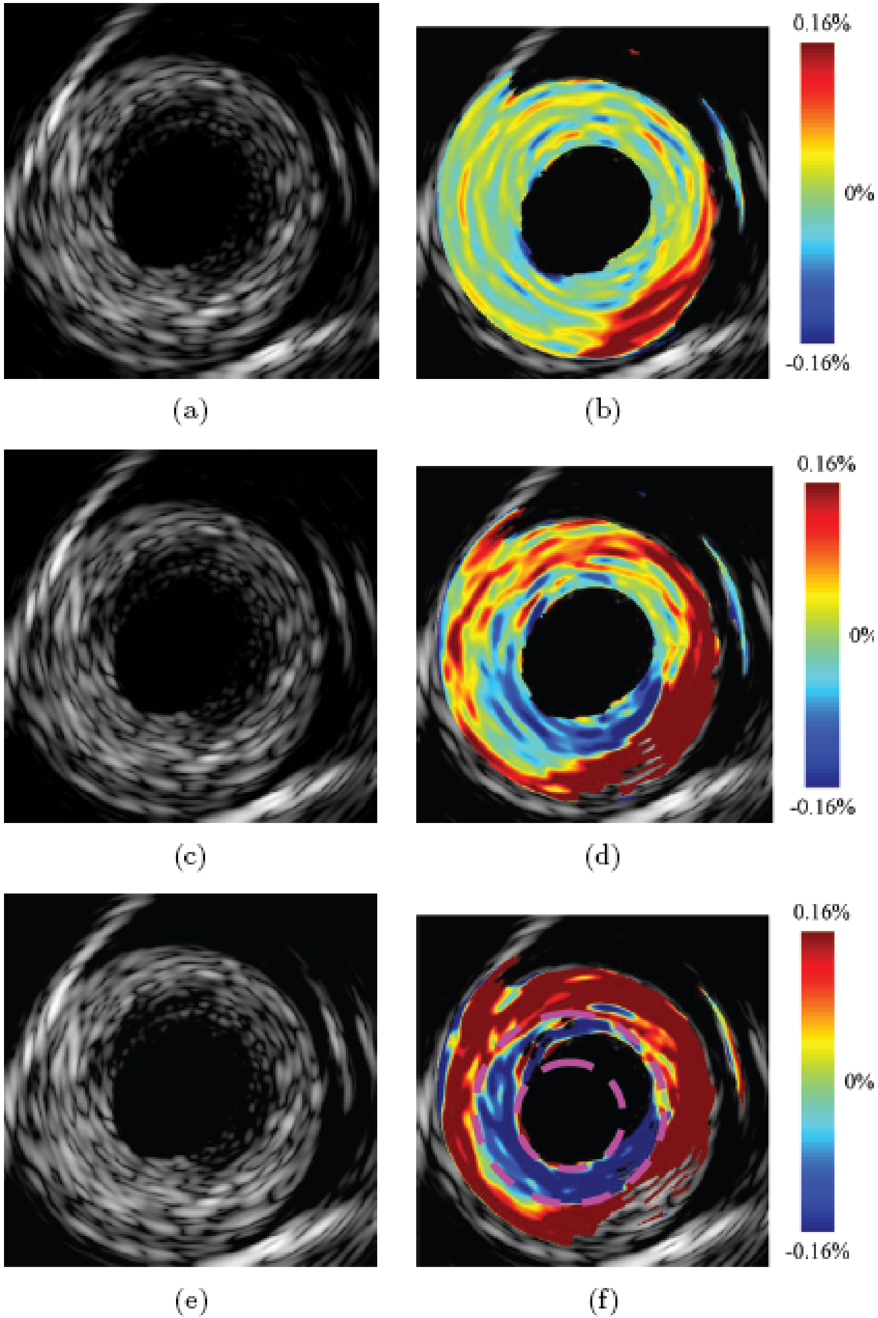
Fig. 5. Simulated B-scan and radial temporal strain images with a 1°C temperature rise. (a) B-scan image (SNR = 40 dB). (b) Temporal strain image (SNR = 40 dB). (c) B-scan image (SNR = 20 dB). (d) Temporal strain image (SNR = 20 dB). The dimensions of all images are 8 mm × 8 mm.

$$\frac{\partial}{\partial t}[\delta t(z)] = (\beta(z) - \lambda(z))\delta\theta(z), \qquad (1)$$

where $t(z)$ is the round trip time delay of an echo from a scatterer at z, $\beta(z)$ is the linear coefficient of thermal expansion, and $\lambda(z)$ is a linear coefficient that determines sound speed variation versus temperature. In general, β is much smaller than λ in soft tissues, which reduces the temporal strain to a function of $\lambda(z)$ and $\delta\theta(z)$. The $\lambda(^\circ\mathrm{C}^{-1})$ for water-bearing tissue ranges from 0.7×10^{-3} to 1.3×10^{-3} and for lipid-bearing tissue from -1.3×10^{-3} to -2×10^{-3} [25]. The difference in λ implies diverging temporal strains, differentiating water-bearing and lipid-bearing tissue with high contrast. A two-dimensional speckle tracking algo-

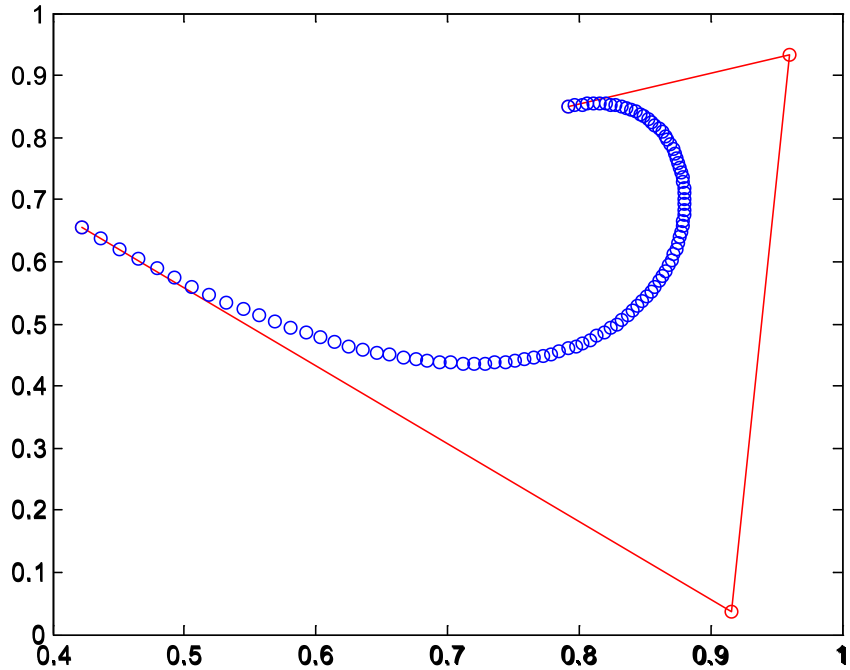
Phantom Results

- Potential problems
  - Speckle tracking
  - Ringdown artifacts
  - Thermal heterogeneity
  - Motion
  - Antenna design



Fun with splines…

Example



monotonic in x

x= 0.4218	y = 0.6557
0.7922	0.8491
0.9157	0.0357
0.9595	0.9340

non-monotonic in x

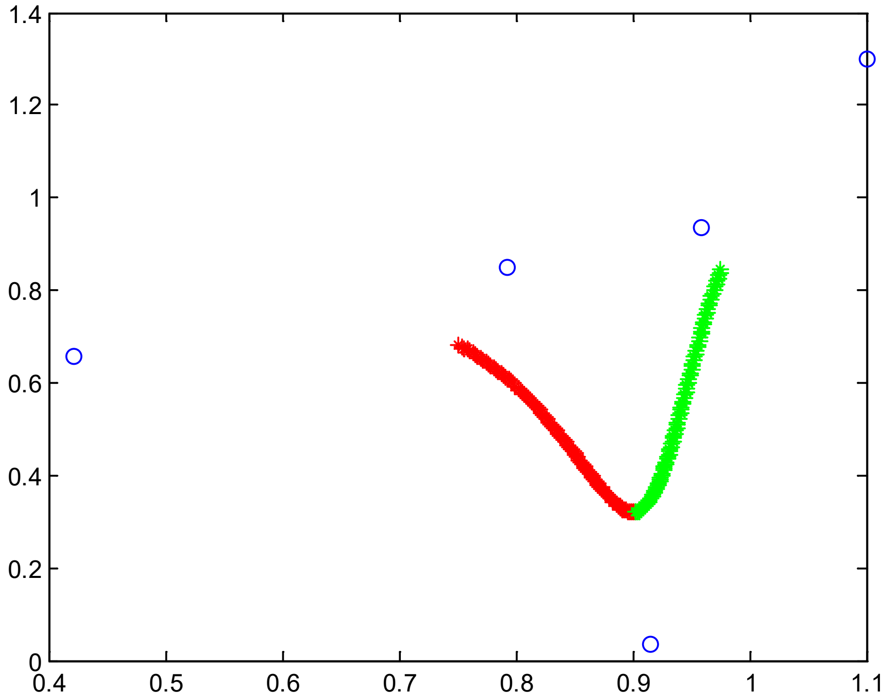
x= 0.4218	y = 0.6557
0.9157	0.0357
0.7922	0.8491
0.9595	0.9340

non-monotonic in x

x= 0.4218	y = 0.6557
0.9157	0.0357
0.9595	0.9340
0.7922	0.8491

```
>> u=[0:.01:1]';
>> xi=(1-u).^3*x(1)+3*(1-u).^2.*u*x(2)+3*(1-u).*u.^2*x(3)+u.^3*x(4);
>> yi=(1-u).^3*y(1)+3*(1-u).^2.*u*y(2)+3*(1-u).*u.^2*y(3)+u.^3*y(4);
```

B-spline example ...



x=	0.4218	y=	0.6557
	0.7922		0.8491
	0.9157		0.0357
	0.9595		0.9340
	1.1000		1.3000

```
>>xi=1/6*(1-u).^3*x(1)+1/6*(3*u.^3-6*u.*u+4)*x(2)+1/6*(-3*u.^3+3*u.*u+3*u+1)*x(3)+1/6*u.^3*x(4);
>>yi=1/6*(1-u).^3*y(1)+1/6*(3*u.^3-6*u.*u+4)*y(2)+1/6*(-3*u.^3+3*u.*u+3*u+1)*y(3)+1/6*u.^3*y(4);
>>xi2=1/6*(1-u).^3*x(2)+1/6*(3*u.^3-6*u.*u+4)*x(3)+1/6*(-3*u.^3+3*u.*u+3*u+1)*x(4)+1/6*u.^3*x(5);
>>yi2=1/6*(1-u).^3*y(2)+1/6*(3*u.^3-6*u.*u+4)*y(3)+1/6*(-3*u.^3+3*u.*u+3*u+1)*y(4)+1/6*u.^3*y(5);
```

D. Rueckert et al., ‘Nonrigid registration using free-form deformations: Application to breast MR images, *IEEE Transactions on Medical Imaging*, Vol. 18, No. 8, pp. 712-721, August, 1999

3781 citations WOS!
6388 on Google Scholar!

1293(WOS), 2206(GS) - JCG
384(WOS), 676(GS) - MIM

Scientific Problem

women. Typically, the detection of breast cancer in MRI requires the injection of a contrast agent such as Gadolinium DTPA. It is known that the contrast agent uptake curves of malignant disease differ from benign disease and this property can be used to identify cancerous lesions [3]. To quantify the rate of uptake, a 3-D MRI scan is acquired prior to the injection of contrast media, followed by a dynamic sequence of 3-D MRI scans. The rate of uptake can be estimated from the difference between pre- and postcontrast images. Any motion of the patient between scans, or even normal respiratory and cardiac motion, complicates the estimation of the rate of uptake of contrast agent by the breast tissue.

Basic Background

The goal of image registration in contrast-enhanced breast MRI is to relate any point in the postcontrast enhanced sequence to the precontrast enhanced reference image, i.e., to find the optimal transformation $\mathbf{T}$: $(x, y, z) \mapsto (x', y', z')$ which maps any point in the dynamic image sequence $I(x, y, z, t)$ at time t into its corresponding point in the reference image $I(x', y', z', t_0)$, taken at time t_0. In general, the motion of the breast is nonrigid so that rigid or affine transformations alone are not sufficient for the motion correction of breast MRI. Therefore, we develop a combined

Method Initialization

The global motion model describes the overall motion of the breast. The simplest choice is a rigid transformation which is parameterized by 6 degrees of freedom, describing the rotations and translations of the breast. A more general class of transformations are affine transformations, which have six additional degrees of freedom, describing scaling and shearing. In 3-D, an affine transformation can be written as

$$\mathbf{T}_{\text{global}}(x, y, z) = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \theta_{14} \\ \theta_{24} \\ \theta_{34} \end{pmatrix} \quad (2)$$

Modeling Nonrigid Deformations

$$\mathbf{T}_{\text{local}}(x, y, z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(u) B_m(v) B_n(w) \phi_{i+l,\, j+m,\, k+n} \quad (3)$$

where $i = \lfloor x/n_x \rfloor - 1$, $j = \lfloor y/n_y \rfloor - 1$, $k = \lfloor z/n_z \rfloor - 1$, $u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$, $w = z/n_z - \lfloor z/n_z \rfloor$ and where B_l represents the lth basis function of the B-spline [22], [23]

$$B_0(u) = (1-u)^3/6$$
$$B_1(u) = (3u^3 - 6u^2 + 4)/6$$
$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6$$
$$B_3(u) = u^3/6.$$

An Objective Function to Smooth

$$\mathbf{T}_{\text{local}}(x, y, z) = \sum_{l=1}^{L} \mathbf{T}_{\text{local}}^{l}(x, y, z).$$

$$\mathcal{C}_{\text{smooth}} = \frac{1}{V} \int_0^X \int_0^Y \int_0^Z \left[\left(\frac{\partial^2 \mathbf{T}}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \mathbf{T}}{\partial y^2} \right)^2 + \left(\frac{\partial^2 \mathbf{T}}{\partial z^2} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{T}}{\partial xy} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{T}}{\partial xz} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{T}}{\partial yz} \right)^2 \right] dx\, dy\, dz \tag{5}$$

An Objective Function to Smooth

An additional objective function term designed to maximize overlap in structures

$$\mathcal{C}_{\text{similarity}}(A, B) = H(A) + H(B) - H(A, B) \qquad (6)$$

where $H(A)$, $H(B)$ denote the marginal entropies of A, B and $H(A, B)$ denotes their joint entropy, which is calculated from the joint histogram of A and B. If both images are aligned, the mutual information is maximized. It has been shown by Studholme [14] that mutual information itself is not independent of the overlap between two images. To avoid any dependency on the amount of image overlap, Studholme suggested the use of normalized mutual information (NMI) as a measure of image alignment

$$\mathcal{C}_{\text{similarity}}(A, B) = \frac{H(A) + H(B)}{H(A, B)}. \qquad (7)$$

Optimization

calculate the optimal affine transformation parameters Θ by maximising eq. (7)

initialize the control points Φ.

repeat

calculate the gradient vector of the cost function in eq. (8) with respect to the non-rigid transformation parameters Φ:

$$\nabla \mathcal{C} = \frac{\partial \mathcal{C}(\Theta, \Phi^l)}{\partial \Phi^l}$$

while $||\nabla \mathcal{C}|| > \epsilon$ **do**

recalculate the control points $\Phi = \Phi + \mu \frac{\nabla \mathcal{C}}{||\nabla \mathcal{C}||}$

recalculate the gradient vector $\nabla \mathcal{C}$

increase the control point resolution by calculating new control points Φ^{l+1} from Φ^l.

increase the image resolution.

until finest level of resolution is reached.

Algorithm overview

$$\mathbf{T}_{\text{global}}(x, y, z) = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \theta_{14} \\ \theta_{24} \\ \theta_{34} \end{pmatrix} \quad (2)$$

Global Alignment

$$\mathbf{T}_{\text{local}}(x, y, z) = \sum_{l=1}^{L} \mathbf{T}_{\text{local}}^{l}(x, y, z).$$

Nonrigid Local Adjustments

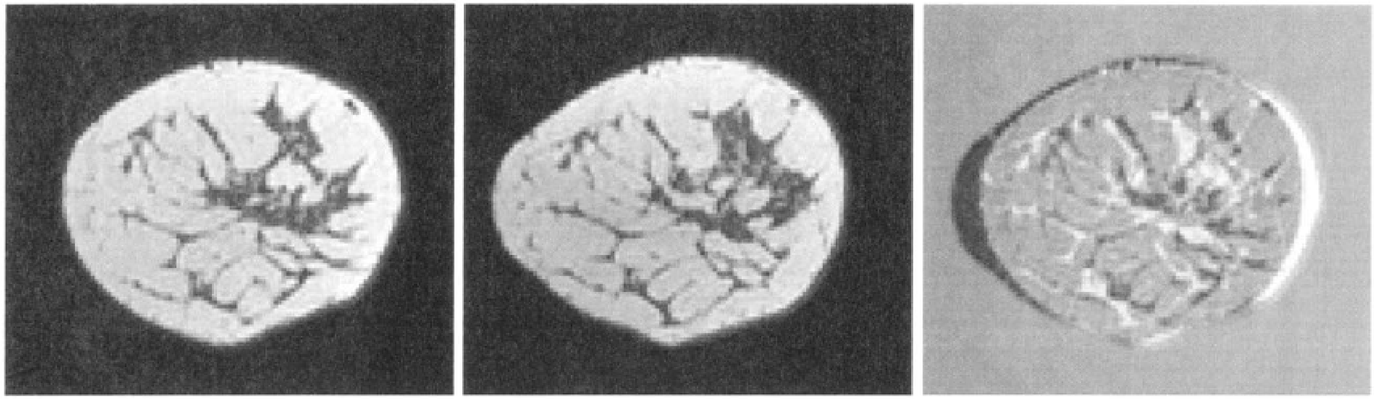
$$\mathcal{C}_{\text{similarity}}(A, B) = \frac{H(A) + H(B)}{H(A, B)}.$$

Transformation Characterization

$$\mathcal{C}_{\text{smooth}} = \frac{1}{V} \int_0^X \int_0^Y \int_0^Z \left[\left(\frac{\partial^2 \mathbf{T}}{\partial x^2}\right)^2 + \left(\frac{\partial^2 \mathbf{T}}{\partial y^2}\right)^2 + \left(\frac{\partial^2 \mathbf{T}}{\partial z^2}\right)^2 + 2\left(\frac{\partial^2 \mathbf{T}}{\partial xy}\right)^2 + 2\left(\frac{\partial^2 \mathbf{T}}{\partial xz}\right)^2 + 2\left(\frac{\partial^2 \mathbf{T}}{\partial yz}\right)^2 \right] dx\, dy\, dz \quad (5)$$

Adjust and Optimize

Example of misregistration



(a) (b) (c)

Fig. 2. Example of misregistration caused by motion of a volunteer. (a) Before motion. (b) After motion. (c) After subtracting (b) from (a) without registration.

Example result

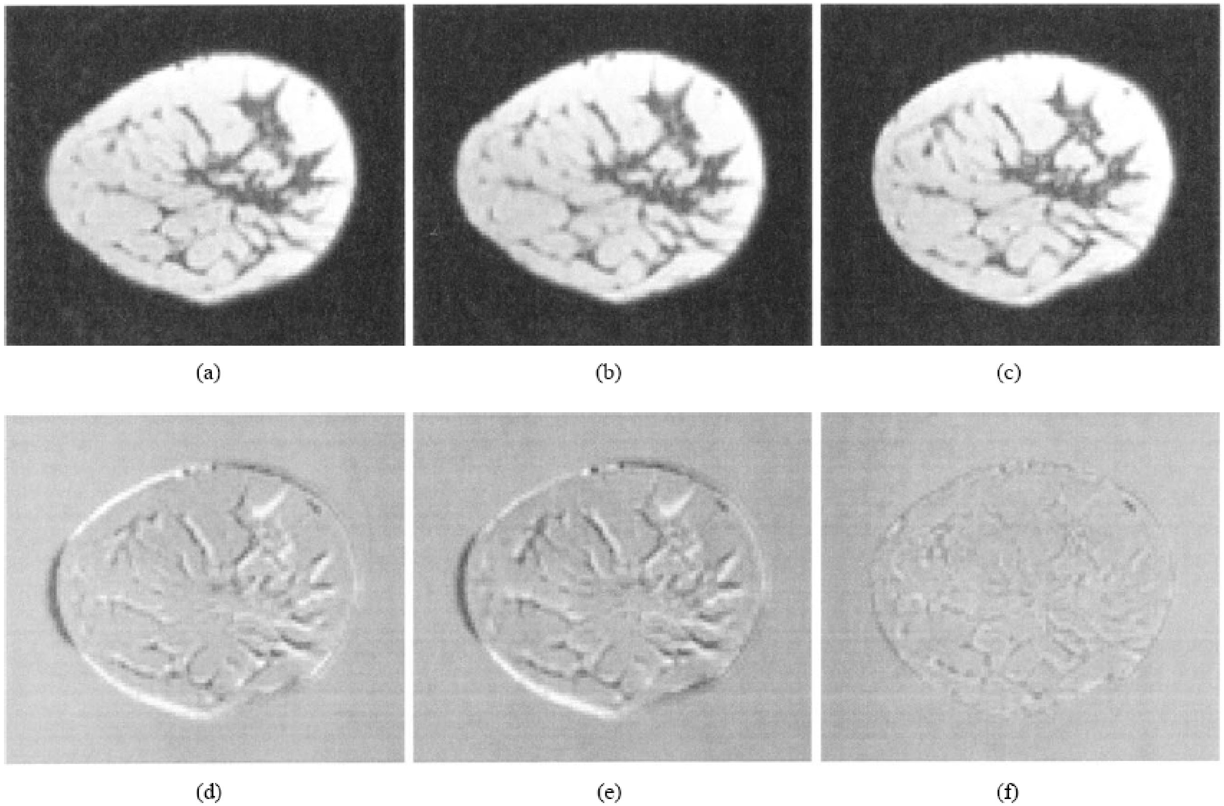


Fig. 3. Example of different transformations on the registration for the volunteer study in Fig. 2: after (a) rigid, (b) affine, and (c) nonrigid registration. The corresponding difference images are shown in (d)–(f).

Example result

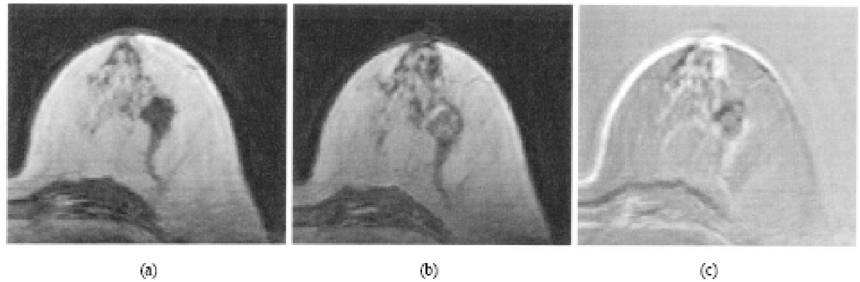


Fig. 6. Example of misregistration in a contrast-enhanced patient study. (a) Before injection of the contrast medium. (b) After injection of the contrast medium. (c) After subtraction of (a) and (b) without registration.

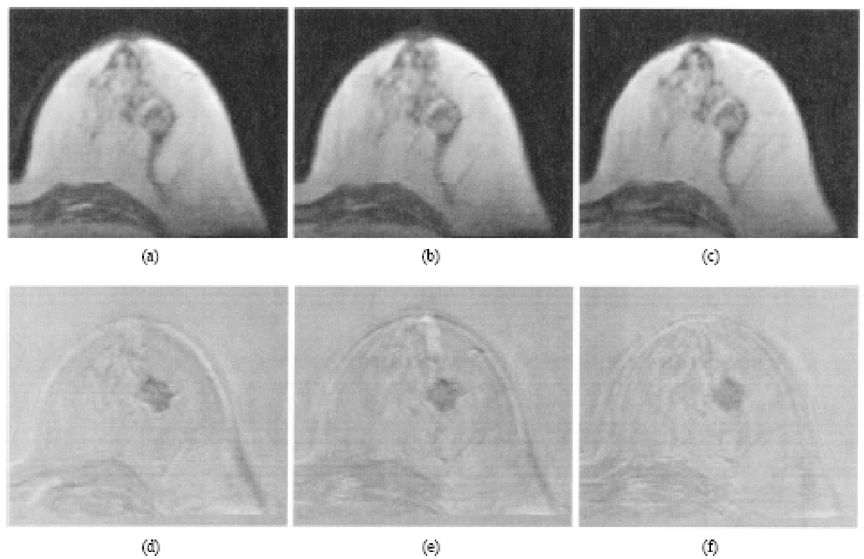


Fig. 7. Example of different transformations on the registration for the patient study in Fig. 6. (a) After rigid. (b) After affine. (c) After nonrigid registration. The corresponding difference images are shown in (d)–(f).

Example result

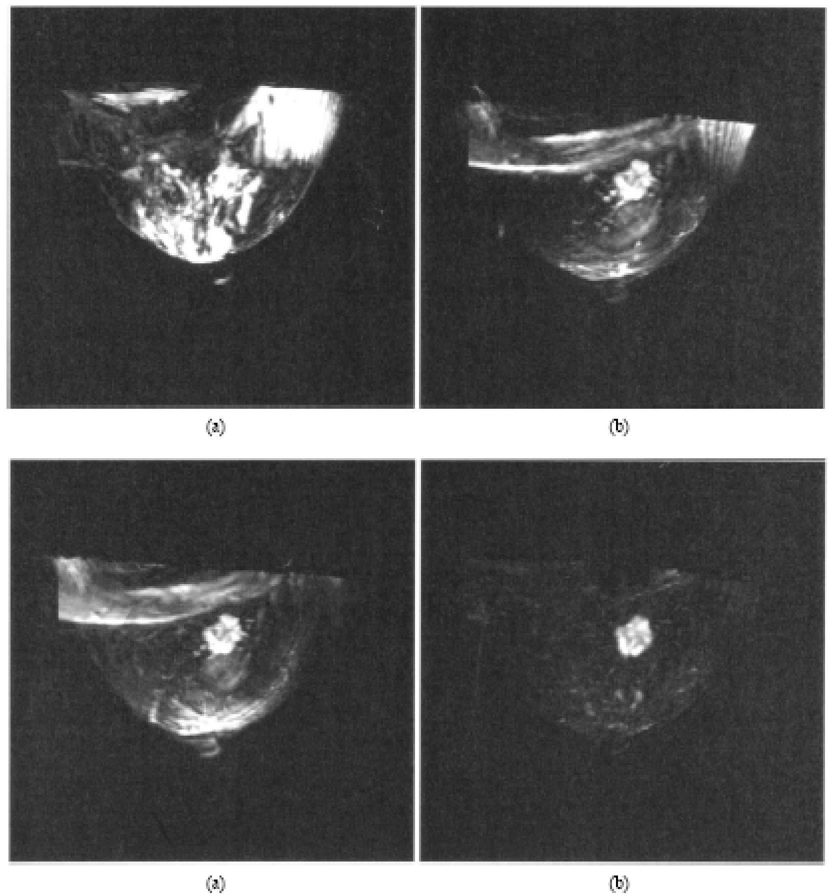


Fig. 8. A MIP of the difference images of the patient study in Fig. 6. (a) Without registration. (b) With rigid. (c) With affine. (d) With nonrigid registration. The tumor can be recognized after registration with all three techniques, but is most clearly visible in (d).

Example Result

TABLE I

COMPARISON OF THE AVERAGE REGISTRATION ERROR OF THE VOLUNTEER STUDIES IN TERMS OF SQUARED SUM OF INTENSITY DIFFERENCES (SSD) AND CORRELATION COEFFICIENT (CC) FOR DIFFERENT TYPES OF TRANSFORMATION. THE SPLINE-BASED FFD HAS BEEN EVALUATED AT A CONTROL POINT SPACING OF 20, 15, AND 10 mm

Registration	SSD (mean)	SSD (variance)	CC
No registration	38.52	53.90	0.8978
Rigid	23.63	33.38	0.9604
Affine	21.38	29.84	0.9689
Affine + FFD ($20mm$)	14.35	23.43	0.9877
Affine + FFD ($15mm$)	13.28	20.91	0.9895
Affine + FFD ($10mm$)	12.53	19.25	0.9905

Last time … Numerical Differentiation

Last time … Numerical Differentiation

- Let's say I want to know $\frac{\partial u}{\partial x}$ where u is the displacement of small piece of tissue …
- Hey! A Taylor Series has derivatives… what can I do?

This is what I want!!!

$$u_{i+1} = u_i + h u_i' + \frac{h^2}{2!} u_i'' + \frac{h^3}{3!} u_i''' + \dots$$

Last time … Numerical Differentiation

Let's solve…

First Forward Difference

$$u_{i+1} - u_i = hu_i' + \frac{h^2}{2!}u_i'' + \frac{h^3}{3!}u_i''' \ldots.$$

$$\varepsilon_x = u_i' = \frac{du_i}{dx} = \frac{u_{i+1} - u_i}{h} - \frac{h}{2!}u_i'' + \ldots.$$

Or how about a 1st order backward difference

First Backward Difference

$$u_{i-1} = u_i - hu_i' + \frac{h^2}{2!}u_i'' - \frac{h^3}{3!}u_i''' + \ldots.$$

$$\varepsilon_x = \frac{\partial u_i}{\partial x} = \frac{u_i - u_{i-1}}{h} + \frac{h}{2!}u_i'' + \ldots.$$

Last time … Numerical Differentiation

$$u_{i+1} = u_i + hu_i' + \frac{h^2}{2!}u_i'' + \frac{h^3}{3!}u_i''' + \ldots.$$

$$u_{i-1} = u_i - hu_i' + \frac{h^2}{2!}u_i'' - \frac{h^3}{3!}u_i''' + \ldots.$$

$$u_{i+1} - u_{i-1} = 2hu_i' + \frac{2h^3}{3!}u_i''' + \ldots.$$

Truncation error

$$\varepsilon_x = u_i' = \frac{du}{dx} = \frac{u_{i+1} - u_{i-1}}{2h} - \frac{h^2}{3!}u_i''' + \ldots.$$

Subtract and solve for u_i'

Last time … Numerical Differentiation

- How to do get higher order?

$$(-2)\left[u_{i+1} = u_i + hu_i' + \frac{h^2}{2!}u_i'' + \frac{h^3}{3!}u_i''' \ldots.\right]$$

$$+ \quad u_{i+2} = u_i + (2h)u_i' + \frac{(2h)^2}{2!}u_i'' + \frac{(2h)^3}{3!}u_i''' \ldots.$$

$$u_{i+2} - 2u_{i+1} = -u_i + h^2u_i'' + h^3u_i''' + \ldots$$

2nd Order Forward Difference

$$u_i'' = \frac{u_{i+2} - 2u_{i+1} + u_i}{h^2} - hu_i''' + \ldots$$

Last time … Numerical Differentiation

$$u_{i+1} = u_i + hu_i' + \frac{h^2}{2!}u_i'' + \frac{h^3}{3!}u_i''' + \frac{h^4}{4!}u''''_i$$

$$+ \quad u_{i-1} = u_i - hu_i' + \frac{h^2}{2!}u_i'' - \frac{h^3}{3!}u_i''' + \frac{h^4}{4!}u''''$$

$$u_{i+1} + u_{i-1} = 2u_i + h^2u_i'' + \frac{h^4}{12}u''''_i$$

2nd Order Center Difference

$$u_i'' = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{h^2}{12}u''''_i$$

Last time … Numerical Differentiation

- How to do get same order, but more accuracy?

This is what I want to higher accuracy!!!

$$u_{i+1} = u_i + hu_i' + \frac{h^2}{2!}u_i'' + \frac{h^3}{3!}u_i''' + \ldots$$

- Need two Taylor Expansions

$$u_{i+1} = u_i + hu_i' + \frac{h^2}{2!}u_i'' + \frac{h^3}{3!}u_i''' + \ldots$$

$$u_{i+2} = u_i + 2hu_i' + \frac{4h^2}{2!}u_i'' + \frac{8h^3}{3!}u_i''' + \ldots$$

Numerical Differentiation

$$(-4)\left[u_{i+1} = u_i + hu_i' + \frac{h^2}{2!}u_i'' + \frac{h^3}{3!}u_i''' \ldots.\right]$$

$$+\ \frac{u_{i+2} = u_i + (2h)u_i' + \dfrac{(2h)^2}{2!}u_i'' + \dfrac{(2h)^3}{3!}u_i''' \ldots.}{u_{i+2} - 4u_{i+1} = -3u_i - 2hu_i' + \dfrac{8h^3}{3!}u_i''' + \ldots}$$

$$u' = \frac{-u_{i+2} + 4u_{i+1} - 3u_i}{2h} + \frac{4h^2}{3!}u_i''' + \ldots$$

Last time … Numerical Differentiation

- Forward/Backward difference… for every increase in order or accuracy, costs 1 extra function evaluation
  - e.g. f' to 2nd order accuracy .. 2 points gets f' to $O(h)$ so add 1 points and you'll get f' to $O(h^2)$
- Center difference… have to sample *symmetrically*, get higher order accuracy for the same # of evaluations as the FWD/BWD diff. (i.e. one order higher)

Last time …Error analysis

•However, must also factor in rounding error …

Function evaluation on computer:

$$\hat{f}_i \quad = \quad f_i \quad + \quad \varepsilon_i$$

Computer value | Exact value | Round-off value

$$\hat{f}_i' = \frac{\hat{f}_{i+1} - \hat{f}_{i-1}}{2h} = \frac{f_{i+1} - f_{i-1}}{2h} + E_{trunc} + \frac{\varepsilon_{i+1} - \varepsilon_{i-1}}{2h}$$

$$f_i' + E_{trunc}$$

Exact Taylor expansion

Round-off error

Last time …Error analysis

$$\therefore \quad \left|\hat{f}_i' - f_i'\right| = \left|E_{trunc} + E_{round-off}\right|$$
$$\leq \left|E_{trunc}\right| + \left|E_{round-off}\right| = \left|\frac{h^2 f_i'''\ (\varsigma)}{6}\right| + \left|\frac{\varepsilon_{i+1} - \varepsilon_{i-1}}{2h}\right|$$

How does discretization affect error?

Error contributions have opposite trends as h $\downarrow$
….. Round-off increases, truncation decreases

Makes sense … we have already seen how truncation error decreases when we decrease step size … but decreasing step size has implications … more evaluations … more round-off

Last time … Numerical Integration

Last time … Numerical Integration

revisit the Taylor Series

Define: $I(x) \equiv \int_a^x f(x)\,dx$ … now expand $I(x)$ in Taylor Series about a point

$$I(x) = I(a) + I'(a)(x-a) + \frac{I''(a)(x-a)^2}{2!} + \dots$$

$$= I(a) + f(a)(x-a) + \frac{f'(a)(x-a)^2}{2!} + \dots$$

Now evaluate at $x=b$, then $h = b-a$

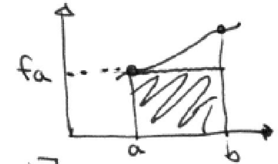
$$I(b) = I(a) + h f_a + \frac{h^2 f_a'}{2} + \frac{h^3 f_a''}{3!} + \dots$$

Last time … Numerical Integration

Generate various expressions by truncating Taylor Series

- simplest:

$$I(b) = \cancel{I(a)}^{\,0} + h f_a + \frac{h^2 f'(\xi)}{2} \qquad \xi \in [a,b]$$



- Next "better" approximation:

$$I(b) = \cancel{I(a)}^{\,0} + h f_a + \frac{h^2 f_a'}{2} + \frac{h^3 f''(\xi)}{3!} \qquad \xi \in [a,b]$$

use $O(h)$ difference expression for f_a'

$$= h f_a + \frac{h^2}{2}\left[\frac{f_b - f_a}{h} - \frac{h}{2} f_a'' + O(h^2)\right] + \frac{f_a'' h^3}{3!} + \dots$$

$$= \frac{h}{2}\left(f_b + f_a\right) - \frac{h^3}{12} f_a'' + HOT$$

$\vdots$

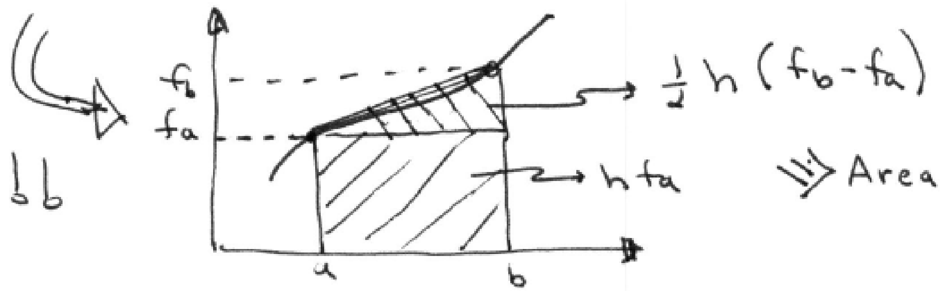
Last time … Numerical Integration

$$\vdots$$

$$= \frac{h}{2}(f_b + f_a) - \frac{h^3 f''(\xi)}{12} \quad \xi \in [a,b]$$

- once again

Trapezoidal Rule !!



$$\Rightarrow \text{Area} = \left(\frac{f_b + f_a}{2}\right) h$$

"exact for polynomial of degree 1 or less"

Leading error $\sim f''$

Last time … Numerical Integration

$\vdots$

$$= \sum_{j=1}^{N} \left[\frac{h}{2} \left(f_j + f_{j-1} \right) - \frac{h^3}{12} f''(\xi_j) \right] \qquad \xi_j \in [x_{j-1}, x_j]$$

$$= \frac{h}{2} \left[f_0 + f_N + 2 \sum_{j=1}^{N-1} f_j \right] - \underbrace{\frac{h^3}{12} \sum_{j=1}^{N} f''(\xi_j)}_{N \overline{f''(\xi_j)}}$$

average value

So

$Nh = b - a$

$$I(b) = \underbrace{\frac{h}{2} \left[f_a + f_b + 2 \sum_{j=1}^{N-1} f_j \right]}_{\text{Composite Trap Rule}} - \underbrace{\left(\frac{b-a}{12} \right) h^2 \overline{f''(\xi_j)}}_{\text{Error term}}$$

Composite Trap Rule b

Error term one order lower in h relative to single application

Last time … Numerical Integration

what happens in composite application?

Recall with Truncation Error:

$$E_{TRUNC} = N \frac{h^3}{12} \overline{f''(\xi)} \xrightarrow{N=\frac{b-a}{h}} \frac{h^2}{12}(b-a)\overline{f''(\xi)}$$

Similarly with Round off:

$$E_{ROUND} = N \frac{h}{2}(\epsilon_a+\epsilon_b) \Longrightarrow \underline{\underline{\frac{(b-a)}{2}}}(\epsilon_a+\epsilon_b)$$

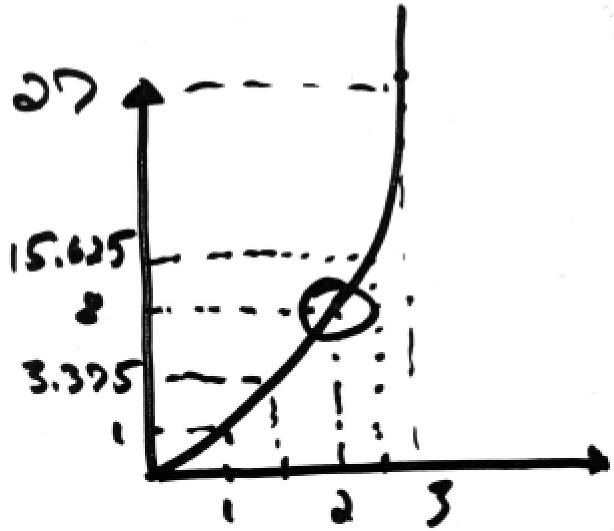
Independent of h!!

$$|E_{ROUND}| \le (b-a)\bar{\epsilon} \quad \text{where} \quad \bar{\epsilon} = \max\{\epsilon_{a_i}, \epsilon_{b_i}\} \; i=1,2,\ldots N$$

Quick example…

How does grid size effect things?

$f(x) = x^3$, $f'(x) = 3x^2$



$f'(x=2) = 12$

Look @ FWD Diff

$h = 1$

$$\frac{f_{i+1} - f_i}{h} = \frac{27-8}{1} = 19$$

→ cut h/2

$$\frac{f_{i+1} - f_i}{h} = \frac{15.625-8}{1/2} = 15.25$$

How does grid size effect things?

How about BKD DIFF.

$h=1$

$$\frac{f_i - f_{i-1}}{h} = \frac{8-1}{1} = 7$$

$h = 1/2$

$$\frac{f_i - f_{i-1}}{h} = \frac{8-3.375}{.5} = 9.25$$

How about center diff.

$h=1$

$$\frac{f_{i+1} - f_{i-1}}{2h} = \frac{27-1}{2} = 13$$

$h = 1/2$

$$\frac{f_{i+1} - f_{i-1}}{2h} = \frac{15.625-3.375}{1} = 12.25$$

How does grid size effect things?

TABLE

ANALYTIC => $f'(x) = 12$

	h=1	Error	h=1/2	Error
FWD DIFF	19	7	15.25	3.25
BWD DIFF	7	5	9.25	2.75
Center Diff	13	1	12.25	0.25

How does grid size effect things?

Recall: 1^{st} FWD DIFF is $O(h)$
1^{st} BWD DIFF is $O(h)$
1^{st} Center Diff is $O(h^2)$

$\Rightarrow$ For FWD, BWD DIFF halfing step roughly ~~cuts~~ error in half

$\Rightarrow$ For CENTER DIFF halfing step cuts by factor of 4, i.e.

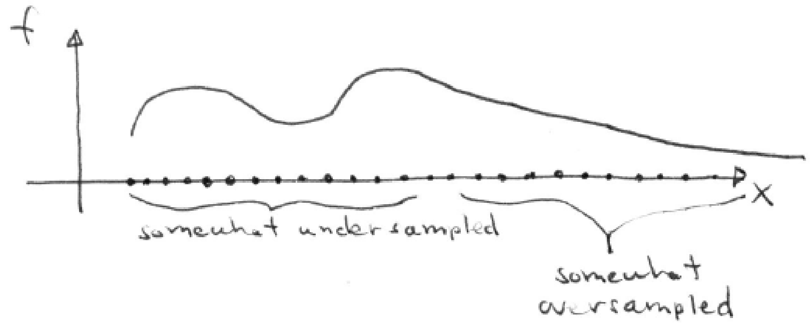
$$\varepsilon_{NEW} = O\left(\left(\tfrac{h}{2}\right)^2\right) \Rightarrow O\left(\tfrac{1}{4}h^2\right)$$

Adaptive Quadrature

Numerical Integration

Adaptive Quadrature

- Try to place sample pts where needed most.
  i.e. where function varies most rapidly

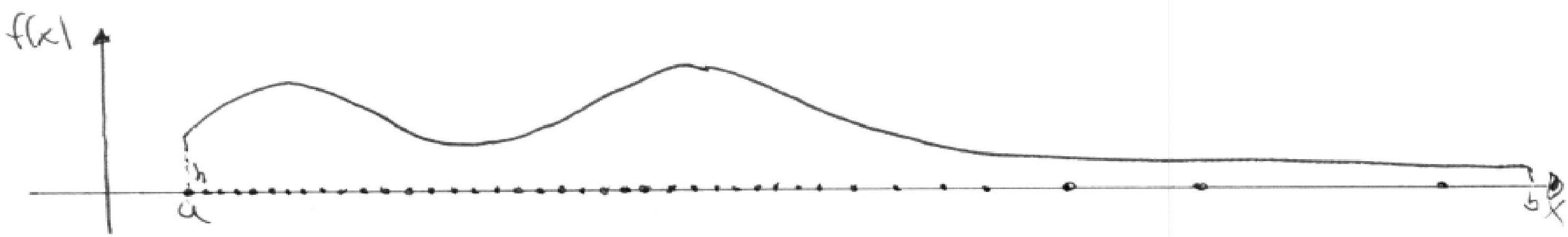


- i.e. really want error uniformly distributed ... to do this would need to know exact answer (don't have it) or estimate error bound and systematically subdivide until below a prescribed tolerance

Numerical Integration

- Strategy: Try to estimate error bound through successive refinement of a quadrature rule

  - want an algorithm that would produce the following set of sample points



Sprinkle nodes such that $h_i = \frac{b-a}{2^{m_i}}$ m_i = integer

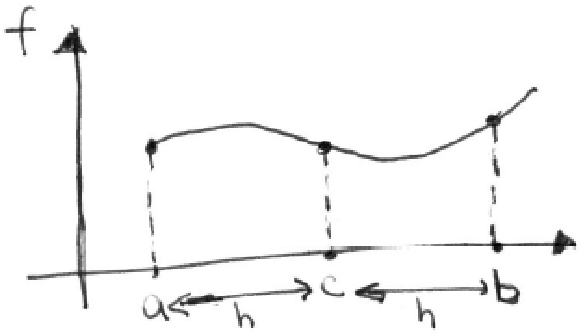
How to find the desired m_i's?

Numerical Integration

Example: Develop an Adaptive Quadrature Rule for Simpson Integration

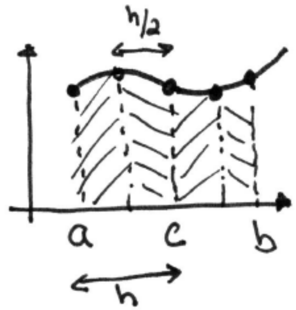
- Define $S(a,b) \equiv$ Simpson on $[a,b]$

  then $$\int_a^b f(x)\,dx = S(a,b) - \frac{h^5 f^{IV}(\xi)}{90}$$



Numerical Integration

subdivide



Applying Simpson's Rule in composite fashion for a refinement yields:

Average btw two panels

$$\int_a^b f(x)\,dx = S(a,c) + S(c,b) - \frac{2\,\overline{f^{iv}}(\xi)\left(\frac{h}{2}\right)^5}{90}$$

i.e. above represents two applications of the rule with spacing h/2

Numerical Integration

$\Rightarrow$ rewriting

$$\int_a^b f(x)\,dx = S(a,c) + S(c,b) - \frac{1}{16}\left(h^5 \frac{\overline{f^{iv}(\xi)}}{90}\right)$$

Now recall for single panel application

$$\int_a^b f(x)\,dx = S(a,b) - \frac{h^5}{90} f^{iv}(\xi)$$

$\Rightarrow$ key assumption $\Rightarrow$ $\underset{\text{single app.}}{f^{iv}(\xi)} \approx \underset{\text{2 panel app.}}{\overline{f^{iv}}(\xi)} = \tilde{f}^{iv}(\xi)$

Numerical Integration

using this assumption, we can set equal to each other since both are approximations to $\int_a^b f(x)\,dx$

$$S(a,b) - \frac{h^5}{90} f^{iv}(\tilde{\xi}) = S(a,c) + S(c,b) - \frac{h^5}{16} f^{iv}(\tilde{\xi})$$

rewriting

$$\frac{15}{16}\left(\frac{h^5 f^{iv}(\tilde{\xi})}{90}\right) = S(a,b) - S(a,c) - S(c,b)$$

or

$$\underbrace{\frac{1}{16}\left(\frac{h^5 f^{iv}(\tilde{\xi})}{90}\right)}_{\text{Recall this is error associated with}} = \frac{1}{15}\left(S(a,b) - S(a,c) - S(c,b)\right)$$

$$\int_a^b f(x)\,dx = S(a,c) + S(c,b) - \frac{1}{16}\left(\frac{h^5 f^{iv}(\tilde{\xi})}{90}\right)$$

Numerical Integration

so I could say

$$\left| \int_a^b f(x)\,dx - S(a,c) - S(c,b) \right| = \frac{1}{15} \left| S(a,b) - S(a,c) - S(c,b) \right|$$

what does this say?

It says that if I choose an error tolerance ϵ to be the difference between analytic and approximation (ϵ represents LHS value of above exp.), the procedure of splitting my interval in half introduces a multiplier of 15.

Numerical Integration

⇒ In other words, to meet my tolerance of ϵ, I need to make sure that

$$\underline{S(a,b) - S(a,c) - S(c,b) < 15\epsilon}$$

So…. Our algorithm looks like….

Numerical Integration

How to find the desired m_i's?

Start:

Section refinement

a $\qquad S(a,b) = S_{10}$ $\qquad b$

$S_{11}(a, \frac{a+b}{2}) \qquad S_{21} = S(\frac{a+b}{2}, b)$

If $|S_{10} - S_{11} - S_{21}| < 15\epsilon$ STOP $\Rightarrow \left|\int_{1,2} f\,dx - S_{11} - S_{21}\right| < \epsilon$

else

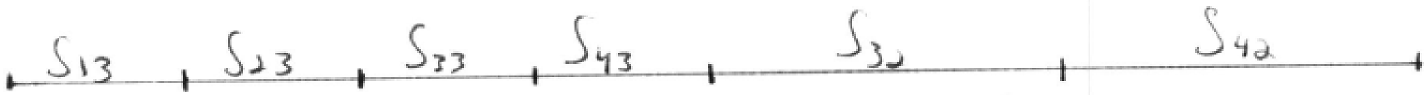
$S_{12} = (a, \frac{a+b}{4}) \qquad S_{22} \qquad S_{32} \qquad S_{42}$

If $|S_{11} - S_{12} - S_{22}| < 15(\frac{\epsilon}{2})$ $\qquad$ If $|S_{21} - S_{32} - S_{42}| < 15(\frac{\epsilon}{2})$

stop $\Rightarrow \left|\int_{1,2} f\,dx - S_{12} - S_{22}\right| < \frac{\epsilon}{2}$ $\qquad$ stop $\Rightarrow \left|\int_{3,4} f\,dx - S_{32} - S_{42}\right| < \frac{\epsilon}{2}$

else

Numerical Integration



If $|S_{12} - S_{13} - S_{23}| < 15\left(\frac{\epsilon}{4}\right)$ If $|S_{22} - S_{33} - S_{43}| < 15\left(\frac{\epsilon}{4}\right)$

stop else

⋮

Numerical Integration

Gauss Quadrature

- so far methods have equally-spaced sampling pts or at least integer multiple of smallest h
- Question: Is there an optimal set of sample pts such that we can integrate exactly polynomial higher than N+1 for N even?
- seems as though if we write

$$I = \sum_{i=1}^{N} a_i \, f(x_i)$$

has 2N degrees of freedom

$\uparrow$ N $\qquad$ $\uparrow$ N

Find a_i' s

Numerical Integration

If choose a_i, x_i carefully; expect that we should be able to integrate exactly a polynomial of degree $2N-1$

- Example: Consider case of 2 sample points

$$I = a_1 f(x_1) + a_2 f(x_2)$$

4 parameters; expect to be able to integrate a cubic or lower exactly.

Recall: Trapezoidal rule only exact for linear

i.e. let
$$\left.\begin{aligned} f(x) &= 1 \\ &= x \\ &= x^2 \\ &= x^3 \end{aligned}\right\}$$
Integrate and solve 4 equations in 4 unknowns

Numerical Integration

$$\int_{-1}^{1} dx = 2 = a_1 + a_2$$

$$\int_{-1}^{1} x\,dx = \frac{x^2}{2}\Big|_{-1}^{1} = 0 = a_1 x_1 + a_2 x_2$$

$$\int_{-1}^{1} x^2\,dx = \frac{x^3}{3}\Big|_{-1}^{1} = 2/3 = a_1 x_1^2 + a_2 x_2^2$$

$$\int_{-1}^{1} x^3\,dx = \frac{x^4}{4}\Big|_{-1}^{1} = 0 = a_1 x_1^3 + a_2 x_2^3$$

4 nonlinear equations in 4 unknowns ... solve w/ Newton for systems (or by hand in this case)

Numerical Integration

solution: $a_1 = a_2 = 1$; $-x_1 = x_2 = \sqrt{1/3}$

x_i's are called Gauss points

a_i's are called the weights

- can generalize to an N-pt formula : Exact for polynomials degree 2N-1
- turns out that x_i's (Gauss Pts) are roots on Legendre Polynomials on [-1, 1] with weights : $\int_{-1}^{1} L_{N,i}(x)\,dx = \omega_i$

  N^{th} order Lagrange centered at each x_i
- Legendre polynomial is orthogonal set on [-1, 1] ⇒ key is these have a weight of unity

Numerical Integration

- Fortunately; Gauss pts and weights are tabulated

- However, Gauss pts assume $\int_{-1}^{1} f(x)\,dx$ $\therefore$ Need to transform general integral $\int_a^b f(y)\,dy$

Transform f(y) to f(x) where x Gauss points

let $y = \frac{a+b}{2} + \frac{b-a}{2} x$; $dy = \frac{b-a}{2} dx$

actual position on $[a,b]$

Gauss pt

If $x = -1 \Rightarrow y = a$

If $x = 1 \Rightarrow y = b$

so $$\int_a^b f(y)\,dy = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2} x\right) dx$$

Numerical Integration

so $$\int_a^b f(y)\,dy = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2}x\right)dx$$

$$= \frac{b-a}{2}\sum_{i=1}^{N} w_i f(y_i)$$

where

$$y_i = \frac{a+b}{2} + \frac{b-a}{2}x_i$$

- See text for elegant proof of Gaussian Quadrature

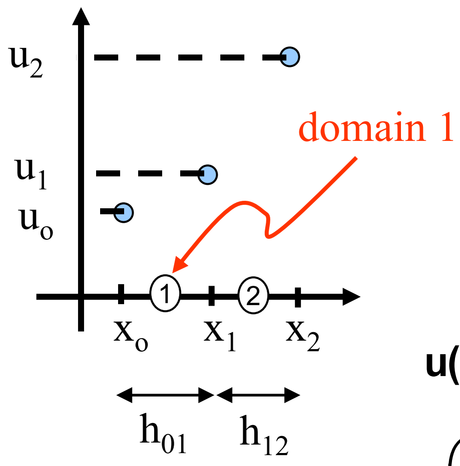
→ Numerical Analysis by Burden and Faires

→ This form is often referred to as

Gauss-Legendre Integration

Let’s step back a bit and view again…

Let's reinterpret integration!



Recall…

… so now I can express $\mathbf{u_i}$ anywhere in the domain of $\mathbf{0 \leq x_i \leq 1}$ using our Lagrange basis…

$$u(x) = \sum_{i=0}^{1} u_i L_{1,i}(x) = u_o L_{1,0}(x) + u_1 L_{1,1}(x)$$

$$u\left(x = x_o + \frac{h_{01}}{3}\right) = u_o L_{1,0}\left(x_o + \frac{h_{01}}{3}\right) + u_1 L_{1,1}\left(x_o + \frac{h_{01}}{3}\right)$$

$$u\left(x = x_o + \frac{h_{01}}{3}\right) = u_o \frac{\left(x_1 - \left(x_o + \frac{h_{01}}{3}\right)\right)}{h_{01}} + u_1 \frac{\left(x_o + \left(\frac{h_{01}}{3} - x_o\right)\right)}{h_{01}}$$

$$u\left(x = x_o + \frac{h_{01}}{3}\right) = \frac{2}{3} u_o + \frac{1}{3} u_1$$

Gives interpolated value in domain

FEM Spine - Numerical Integration

Strategy:
$$\int_a^b f(x)dx \approx \sum_{i=0}^{N} a_i f(x_i)$$

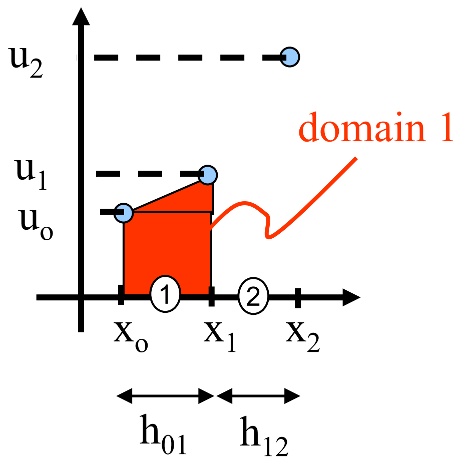
i.e. a weighted sum of function evaluations

Like Calc: $$\int_a^b f(x)dx = \mathrm{Lim}_{\Delta x \to 0} \sum_{i=0}^{N} f(x_i)\Delta x_i \quad \text{but...}$$

Don't take limit: $$\int_a^b f(x)dx = \sum_{i=0}^{N} f(x_i)\Delta x_i \quad + \text{Error}$$

Could expand f(x) in a Lagrange basis:
$$f(x) = \sum_{i=0}^{N} f_i L_{N,i}(x) + \text{Error}$$

FEM Spine - Numerical Integration



$$\int_a^b f(x)dx = \int_a^b \sum_{i=0}^{N} f_i L_{N,i}(x) + \int_a^b E_{trunc}$$

$$\int_a^b f(x)dx = \sum_{i=0}^{N} f_i \underbrace{\int_a^b L_{N,i}(x)}_{a_i} + \text{error}$$

$$\int_a^b f(x)dx = f_o \int_a^b L_{1,0}(x)dx + f_1 \int_a^b L_{1,1}(x)dx$$

$$\int_a^b f(x)dx = f_o \int_{x_o}^{x_1} \frac{(x - x_1)}{(x_o - x_1)} dx + f_1 \int_{x_o}^{x_1} \frac{(x - x_o)}{(x_1 - x_o)} dx$$

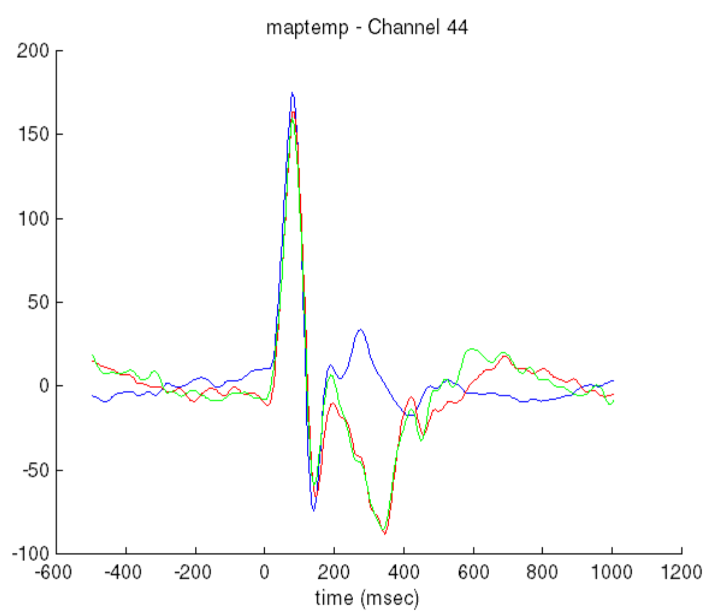
$$\int_a^b f(x)dx = \frac{x_1 - x_o}{2}(f_o + f_1) + \text{error}$$ Trapezoidal rule!

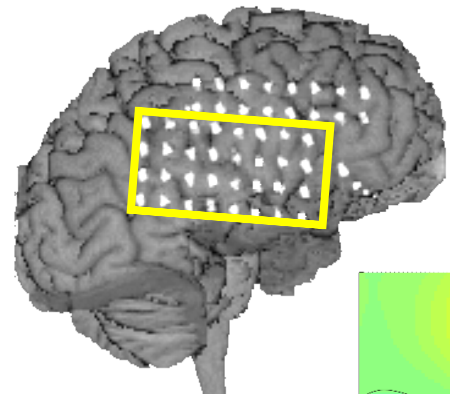
Polynomial Basis/Weighting Functions

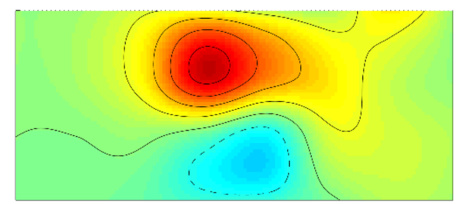
- Key point: When an integral has functions in the integrand represented by a basis expansion, the constant coefficients can slip outside the integral and integration is applied to the user-prescribed basis function, in this case the Lagrange polynomial.
- Result: Integration becomes simple – it is easy to integrate the inter-nodal behavior function if it's just a polynomial!

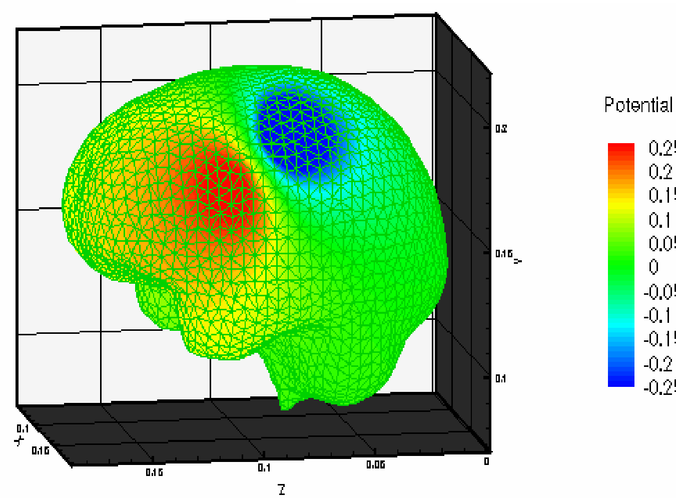
Let’s think spatially…

Epileptogenic sources …









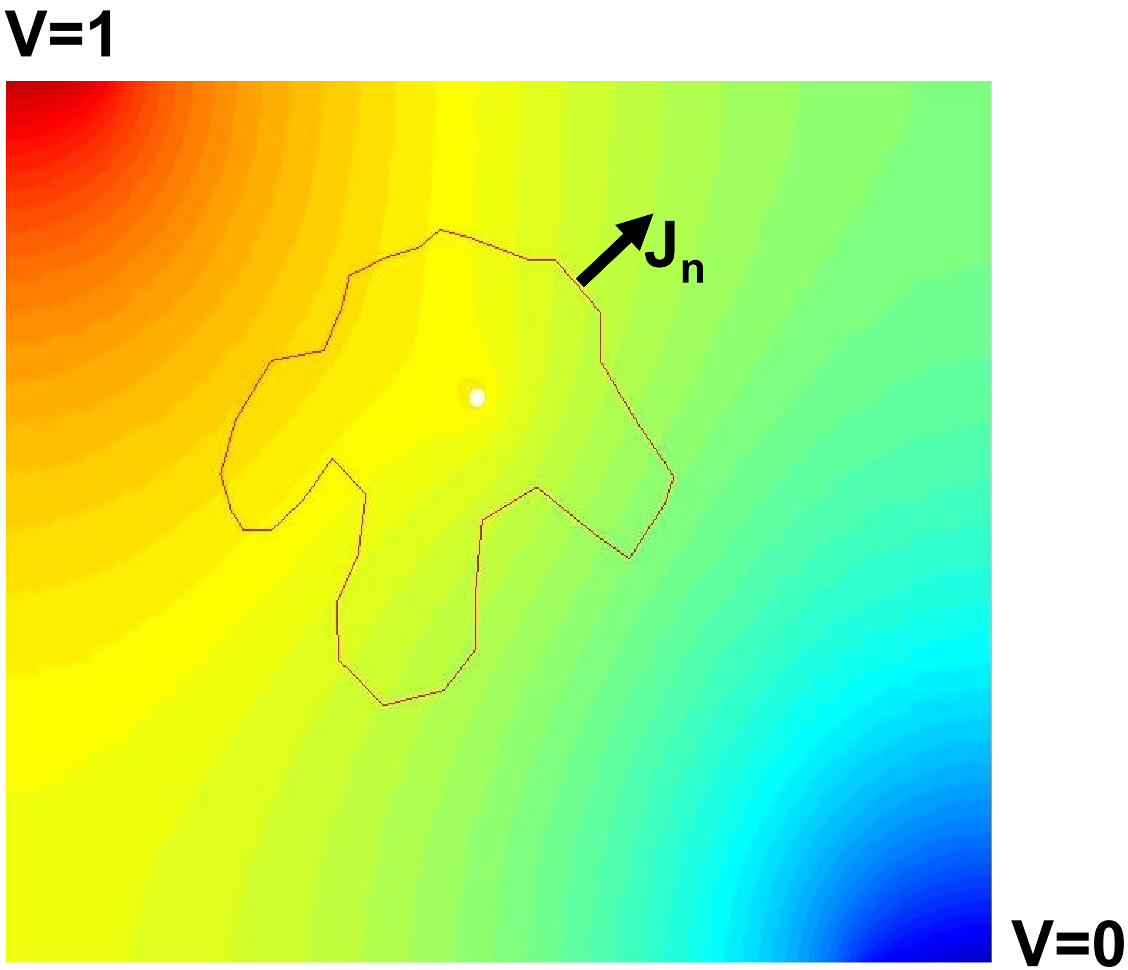
Provocative questions …

PQ 1: So when I say conservation of flow, what does that mean to you?

PQ 2: And can you give me an example?

Numerical Integration

What if?



Numerical Integration

$$\iint_S F \bullet \hat{n} dS = \iiint_V \nabla \bullet F dV$$

Divergence Theorem

- Theorem connects surface integrals and volume integrals.. very powerful
- Although some may think it has its roots in electrostatics, it is independent of physics and is purely a mathematical concept

Numerical Integration

$$\iint_S J \bullet \hat{n}\, dS = \iiint_V \nabla \bullet J$$

So integrating this around a domain ……… tells me something about the content of that domain

Numerical Integration

How would I do this?

$$\iint_S J \bullet \hat{n} dS$$

- Given vectors ([J_x + J_y]) about a contour, how does one calculate in this case a line integral?
- Like everything, we want to break things down to a series of function evaluations that sum to approximate the value of the integral.

Numerical Integration

- Let's expand $\mathbf{J_n}$ as a series of function evaluations

$$J_n = \sum_{j=1}^{n} c_j \varphi(s)_j$$

- Alternatively, we could think of this as expanding the function in a linear basis with a function of a known polynomial form

Numerical Integration

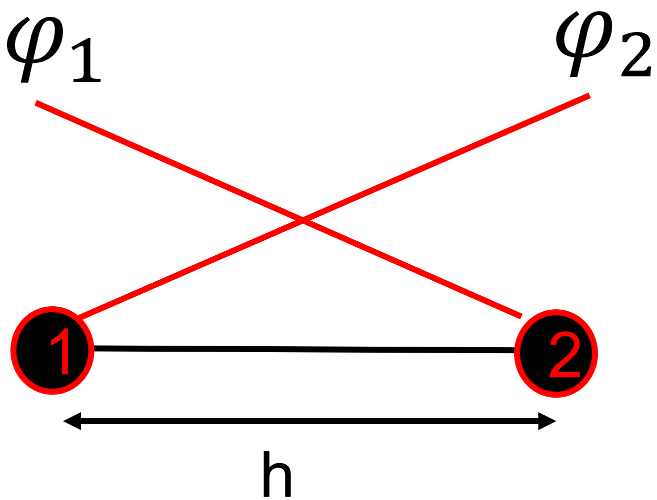
- How to select function?

$$J_n = \sum_{j=1}^{n} c_j \varphi(s)_j$$

- Choose one … some may be more straight-forward than others. If I choose a Lagrange Polynomial, it has an interesting behavior … it transforms the expansion

$$J_n = \sum_{j=1}^{n} J_{n_j} \varphi(s)_j$$

Numerical Integration



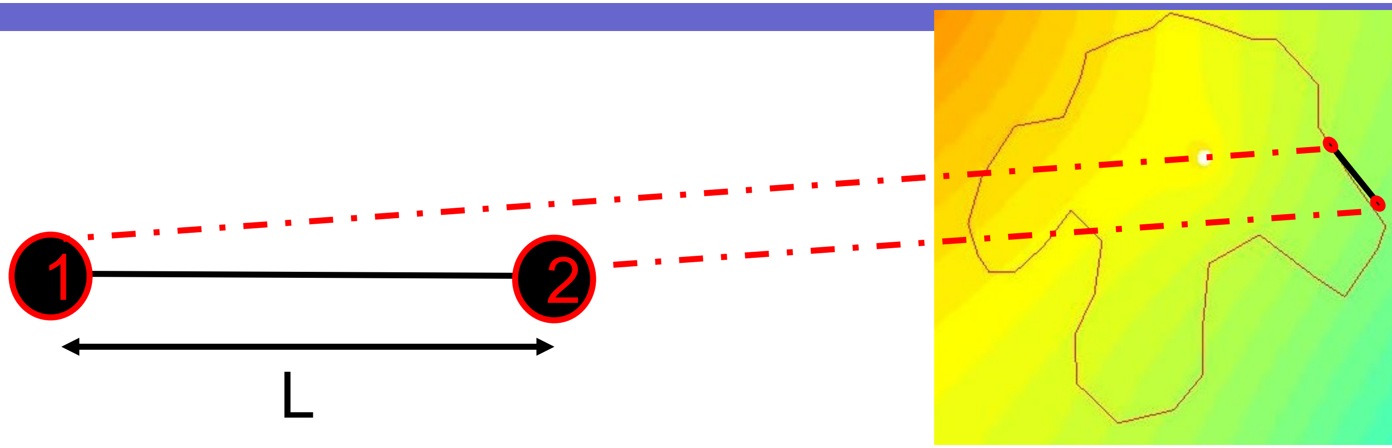
- Rooftop or chapeau function

$$\varphi_1 = \frac{x_2 - x}{h}, \varphi_2 = \frac{x - x_1}{h}$$

$$\int_{x_1}^{x_2} \frac{(x_2 - x)}{h} dx = \frac{x_2^2 - 2x_2x_1 + x_1^2}{2h} = \frac{(x_2 - x_1)^2}{2h}$$

$$\int_{x_1}^{x_2} \frac{(x_2 - x)}{h} dx = \frac{h}{2}$$

Numerical Integration



$$\iint_S J \bullet \hat{n}\, dS \overset{3D-2D\ \mathrm{rep}}{\Rightarrow} \int_\ell J_n\, d\ell = J_{n1}\int_1^2 \varphi_1(\ell)\, d\ell + J_{n2}\int_1^2 \varphi_2(\ell)\, d\ell$$

$$\iint_S J \bullet \hat{n}\, dS \overset{3D-2D\ \mathrm{rep}}{\Rightarrow} \int_\ell J_n\, d\ell = J_{n1}\frac{L}{2} + J_{n2}\frac{L}{2}$$

This represents the integration of $\mathbf{J_n}$ over this length. Just need to go length by length and sum up!

Numerical Integration

- Given [Jx, Jy] – how do you find J_n on a domain contour?

$$\Delta y_i = y_{i+1} - y_{i-1}$$

$$\Delta x_i = x_{i+1} - x_{i-1}$$

$$\hat{\mathbf{n}}_i = \frac{\Delta y_i \hat{\mathbf{x}} - \Delta x_i \hat{\mathbf{y}}}{\Delta S_i}$$