Chapter 09.00D

Physical Problem for Optimization Computer Engineering

Problem Statement

An image is a collection of gray level values at a set of predetermined sites known as pixels, which are arranged in an array. These gray level values are also known as image intensities.

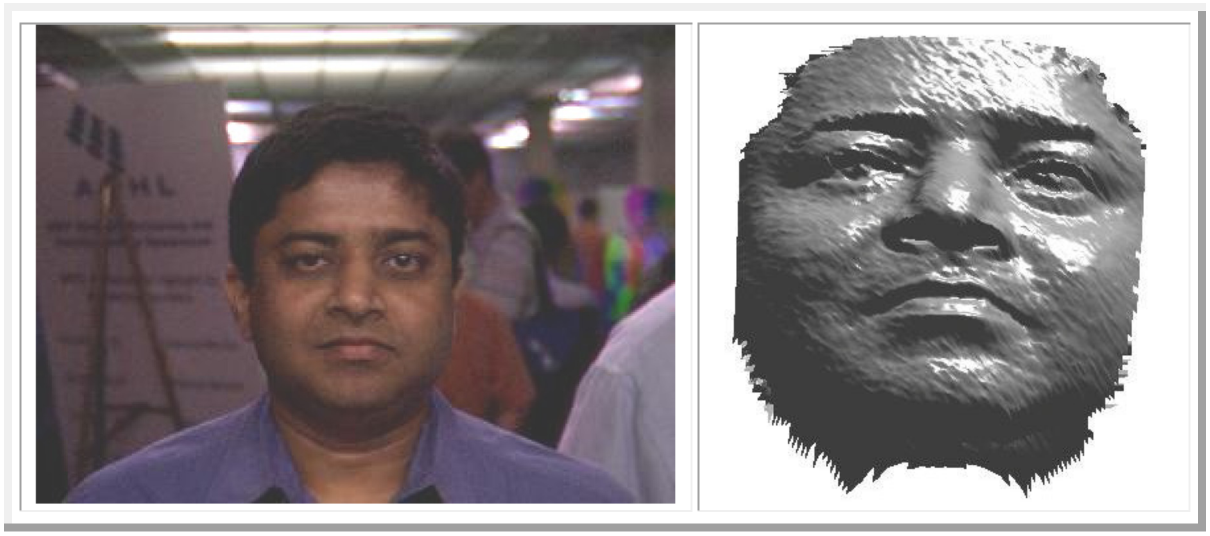


Figure 1: On the left is a typical image collected using a camera. On the right is a three dimensional shape data of the face.

On the left of Figure 1 is an example of an image that we are familiar with. Each point in that image can be indexed using two coordinates, the row and column number, or u and v. It is two dimensional or 2D for short. For each (u,v), we have an intensity of color value. Given just an image, it is not possible to create how an object will appear viewed from another direction. For instance, given just the frontal view of the face, it is not possible to generate a side or titled view of the face. However, if in addition to the 2D image view, we have three dimensional (3D) shape data, then it is possible to do this. This is what we will explore in this problem. An example of the 3D shape is show in Figure 1 on the right. Each point in that data represents an actual 3D point in space, indexed by, (X,Y,Z) coordinates. Taken together, they represent the shape of the face. What is missing is the texture or the color information. The 2D image gives us that texture information. Note that for display purposes, we have shaded the surface with artificial lighting (using Matlab's *patch* function.). The actual data is basically a set of 3D points as shown in the 3d plot (Figure 2). There are many different 3D cameras on the market. We used the Konica-Minolta camera for collection of the face data.

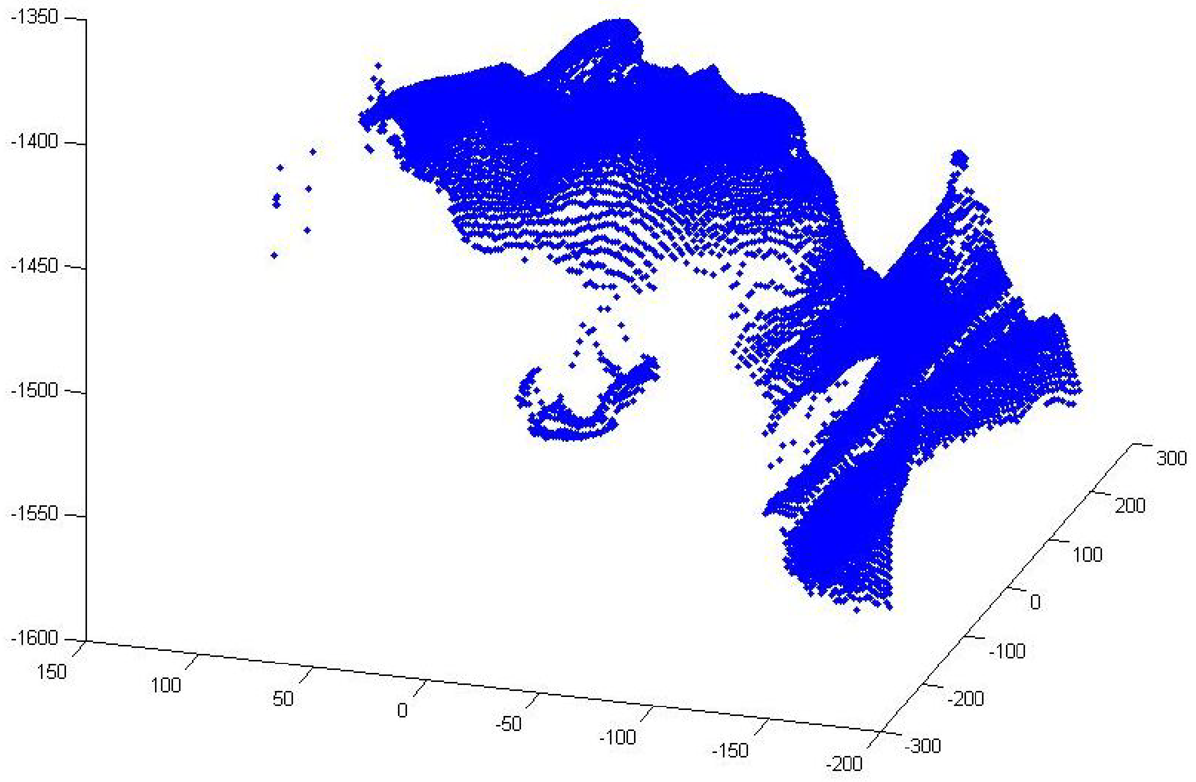


Figure 2: Three-dimensional points on the face.

Solution

Physics of the Problem

We start by considering the geometry of image formation. How are the 3D coordinates in the world related to the 2D pixel locations in the image? Light from a 3D point, P, passes through the camera lens and registers on a point P in the image, as depicted on the left in Figure 3. We have abstracted the camera to be a pin-hole camera. The equations get complicated when considering the actual lens system, but the essence of the geometrical relationship is capture by the pin-hole model of the camera. The underlying geometry can be depicted as in Figure 3. A point P in the world is projected onto a point p in the image. The line pP passes through the lens center.

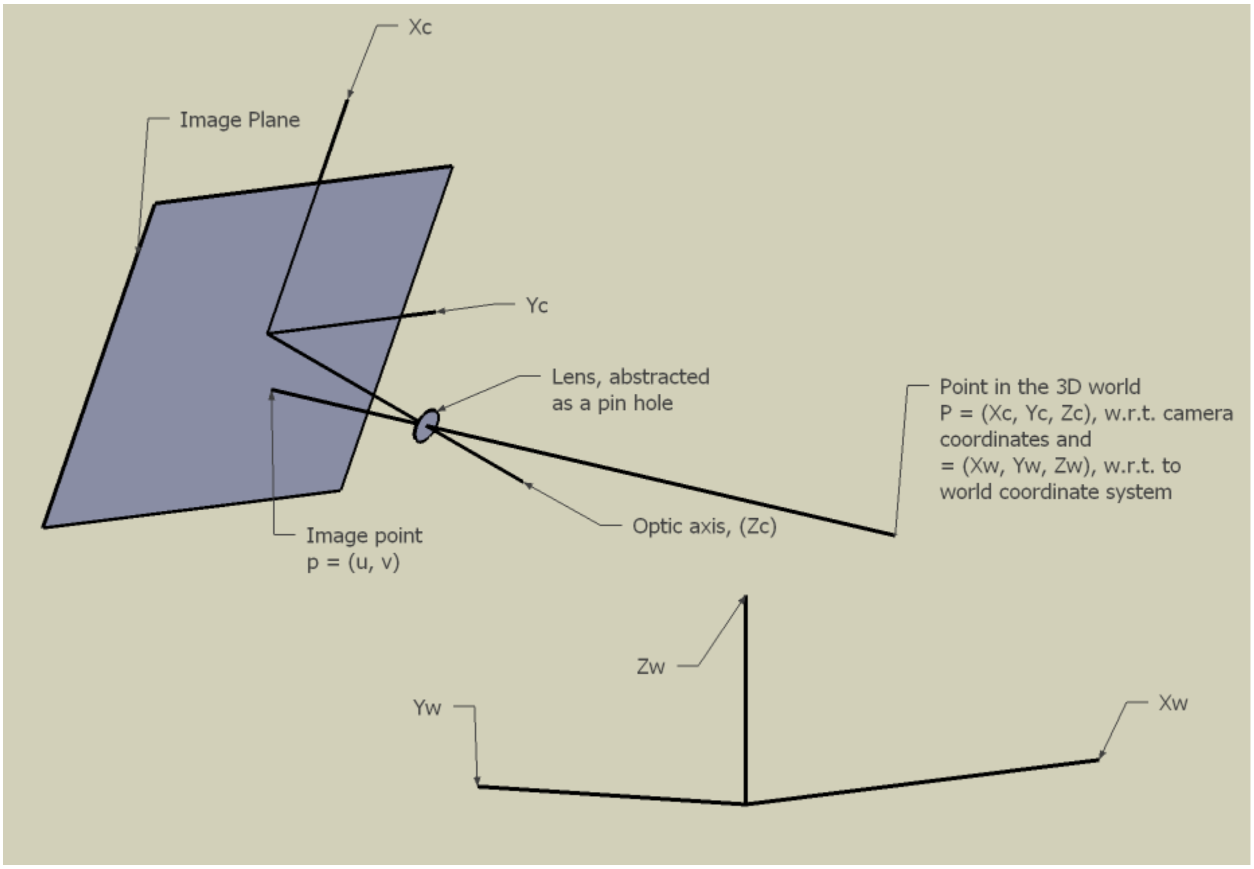

Xc
Image Plane
Yc
Lens, abstracted as a pin hole
Point in the 3D world P = (Xc, Yc, Zc), w.r.t. camera coordinates and = (Xw, Yw, Zw), w.r.t. to world coordinate system
Optic axis, (Zc)
Image point p = (u, v)
Zw
Yw
Xw

Figure 3: Projective geometry. Coordinates of a plane with respect to the image are related to the world point coordinates.

We have two coordinate systems, one rooted in the 3D world, which we will refer to as the world coordinate system and the other rooted at the center of the image called the camera coordinate system. The camera-based and world-based coordinates of any given 3D points, P, are $P_C = [X_C, Y_C, Z_C]^T$ and $P_w = [X_w, Y_w, Z_w]^T$, respectively. These two coordinate values are related via a rigid rotation, R, and translation, T:

$$P_c = RP_w + T$$

or expanding in matrix notation we have

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

The rotation matrix can be expressed as a product of three rotation matrices, each capturing rotation along one of the three coordinate axes.

$$R(\alpha,\beta,\gamma) = R_x(\alpha)R_y(\beta)R_z(\gamma)$$

or in matrix form

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The image coordinate of the projected point, $p = (u,v)$, will be related to the camera-based coordinates of the 3D point, $P_C = [X_C, Y_C, Z_C]^T$, using what is known as the perspective projection equations.

$$u = -f\frac{X_c}{Z_c}$$

$$v = -f\frac{Y_c}{Z_c}$$

where f is the focal length of the lens, captured by the distance between the pin-hole (optic center) and the image plane. These equations can be derived using ratio-based relationships in similar triangle geometry. Given these geometrical relationships, one can directly relate the world-based coordinates of a point with its image coordinates. It is a nonlinear relationship, involving a ratio of sin and cosine functions, with the focal length, 3 rotation angles and the 3 translation values as the parameters.

$$u = U(X_w, Z_w; \alpha, \beta, \gamma, T_x, T_y, T_z, f)$$

$$v = V(Y_w, Z_w; \alpha, \beta, \gamma, T_x, T_y, T_z, f)$$

Our task is to register the given 2D texture map with the 3D shape data. First, we will have to mark corresponding points between the two data. For instance, note the coordinates of the nose tip in the image and the 3D data. Other facial features that can be easily corresponded are the inside and outside corners of the eyes, points on the nostril boundary, marking on the forehead and so on. These correspondences will give us a list of N paired point sets: $\{(u_i, v_i), (X_i, Y_i, Z_i) \mid i = 1,2,.....,N\}$.

Given this, we have to find the best rotation and translation values that will register the given points. After we have estimated the rotation and translation values, then we can easily register rest of the points in the images using the perspective projection equations that we have outlined earlier. Thus, our optimization problem at these points is given by

$$\arg\min_{\alpha,\beta,\gamma,T_xT_yT_z,f} \sum_{i=1}^{N} (u_i - U(X_i, Z_i; \alpha, \beta, \gamma, T_x, T_y, T_z, f))^2 + (v_i - V(Y_i, Z_i; \alpha, \beta, \gamma, T_x, T_y, T_z, f))^2$$

Note that we are optimizing the distance between the image location of the 3D points based on the estimate rotation, translations, and the actual observed location in the image.

Worked Out Example

We select the coordinates of the facial features as shown in Figure 4. The corresponding points are given in the following Table 1.

Table 1: Values of projected point at coordinates.

$u_i =$	$v_i =$	$X_i =$	$Y_i =$	$Z_i =$
1	-47	0.5	70.1	1405.4
20	-43	28.4	63.8	1426.8
-22	-43	27.4	62.3	1425.0
1	-31	3.4	42.7	1408.1
16	-24	23.9	35.8	1422.2
2	-22	3.3	30.7	1390.6
-14	-23	20.0	34.2	1418.4
37	0	54.1	0.9	1446.2
12	0	24.1	2.3	1436.1
-13	1	14.2	0.8	1428.2
-35	1	48.4	2.3	1437.6
1	10	4.9	12.5	1419.1

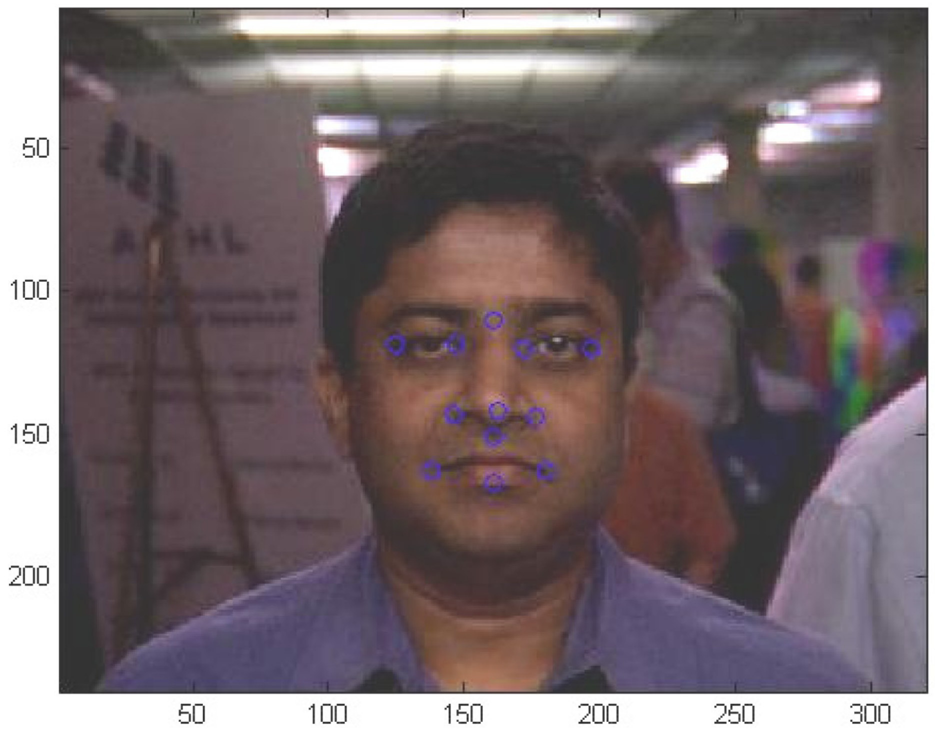
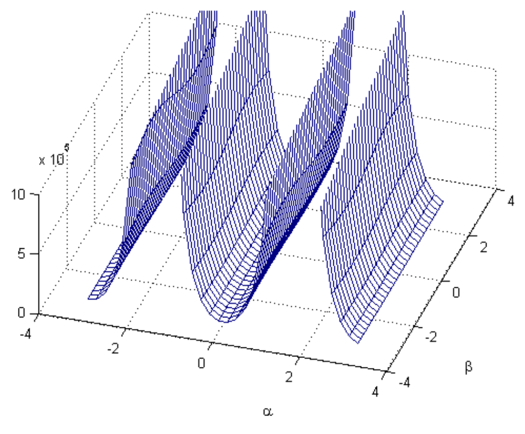
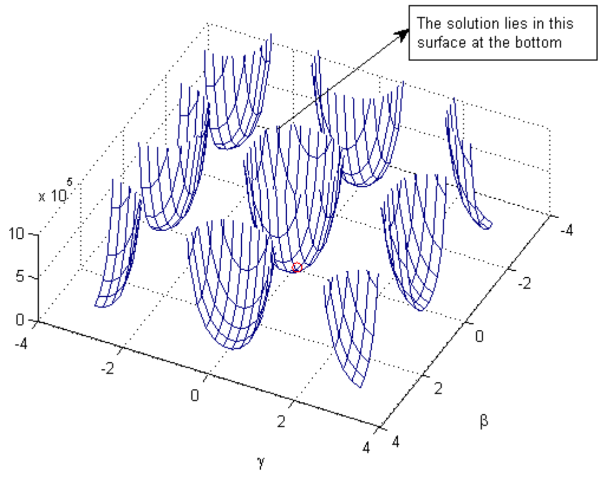


Figure 4: Facial features that were selected to estimate the transformation between and the image and 3D face data are marked with blue circles. The corresponding points are also selected from the 3D face data.

Since the optimization form is a sum of squares, we used the MATLAB function lsqnonline to find the optimum rotation and translation values. For this optimization, we kept the focal length, f, fixed at 1000. Our initial estimates for the rotation and translation values were all zero. Figure 5 shows the change in the optimized value with iteration. We see that the error stabilizes in 25 iterations for this example.

The final residual was 36.534050.
The estimated rotation angles were $(\alpha = -1.019942, \beta = -1.964836, \gamma = 2.506232)$ degrees, and the translations were $(T_x = 46.337847, T_y = 62.571441, T_z = -13.514498)$





(a) (b)

Figure 5: Two-dimensional slices through the six dimensional error function. (a) Angles α and β are varied, while the rest are held at the found solution values. (b) Angles β and γ are varied while the rest are held constant at the found solution values.

Some examples of the typical objective function are visualized in Figure 5. The objective function is six-dimensional, so it is hard to visualize. We have shown here some two-dimensional slices through the function. In other words, two of the values are held constant while the others are held fixed at their found (optimal) values. Note that the variables plotted here are angles, hence the axes wrap around. We see that the surface itself is quite smooth; however, there are multiple solutions. Therefore, it is important that the initial condition be chosen to fall in the “bucket” corresponding to the solution. The fortunate characteristic is that the buckets are far apart.

Using these estimates, we can map the rest of the image texture onto the 3D face data. Figure 6 shows some example views of the final mapped data. We can image the views of the face as it would appear from any angle. We are done!

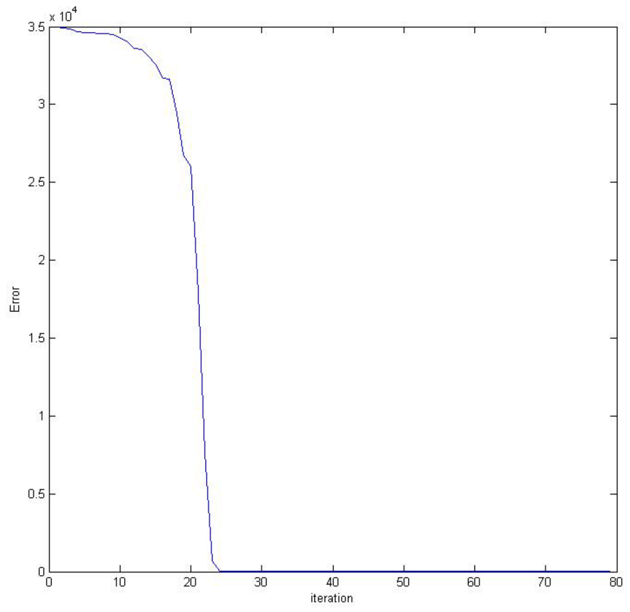


Figure 6: Variation of the optimized error with iteration.

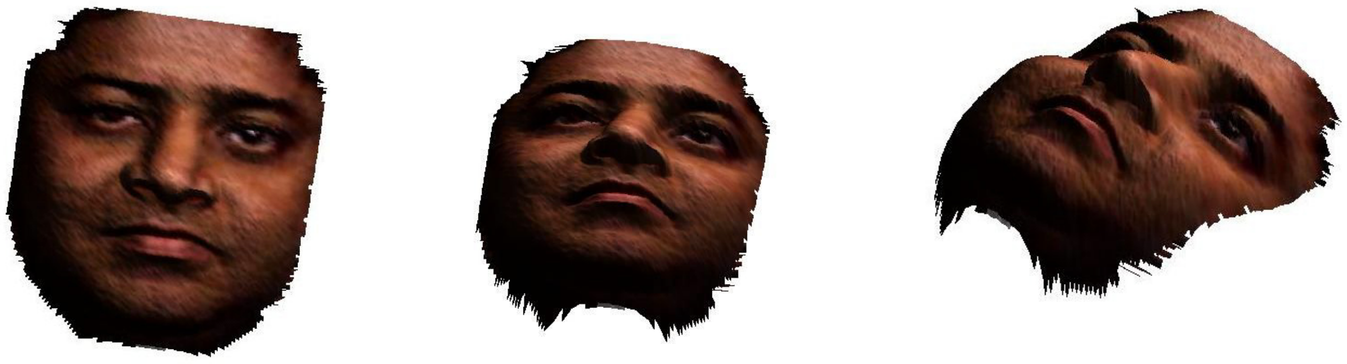


Figure 7: Multiple views of the registered texture and shape data. We can now generate views of the face from any angle.

The following files are available on the web.
http://numericalmethods.eng.usf.edu/mws/com/09opt/0346_frontal.jpeg: face image
http://numericalmethods.eng.usf.edu/mws/com/09opt/0346_frontal.wrl: VRML formatted 3D data of the face.
http://numericalmethods.eng.usf.edu/mws/com/09opt/ReadVRML.m: Matlab file to read the VRML data
http://numericalmethods.eng.usf.edu/mws/com/09opt/Points_2D_3D.m: Code used to generate the example results.

Questions and Assignments

1. In our example we had kept the focal length, f, fixed during the optimization. What happens if one also declares that as an unknown into the optimization?
2. Which parameters are the most sensitive ones? Rotation? Translation? Or focal length?
3. What is the sensitivity of the choice of the initial value?
4. Are we always guaranteed to get the global optimum value? Under what conditions?

OPTIMIZATION	
Topic	Physical optimization problem for computer engineering
Summary	A physical problem of color and texture of 3-D images
Major	Computer Engineering
Authors	Sudeep Sarkar
Date	December 5, 2011
Web Site	http://numericalmethods.eng.usf.edu