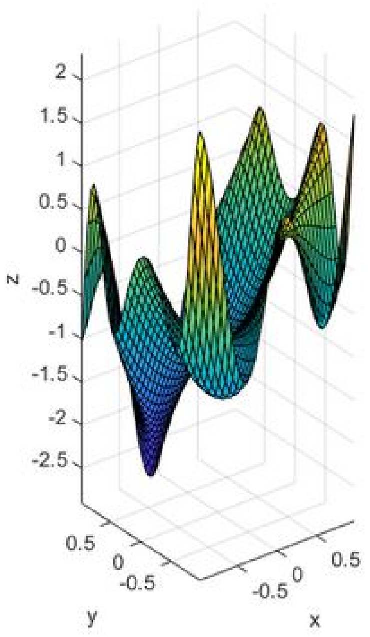
SVD Decomposition of a Surface and Dimension Reduction

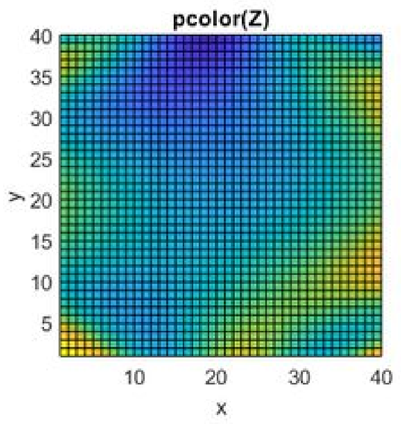
Low Rank Approximation of a Surface

The code for this example is here.

Consider a surface lying over the square patch in the x, y-plane given by $[-1,1] \times [-1,1]$. The z-coordinates are produced by a function $z = f(x,y)$. We can almost view this as an infinite matrix, for example, at row $y = .25$, and column $x = -.15$, there is the entry $z = f(-.15, .25)$. Discretize $[-1,1]$ by splitting it into 40 sub-intervals all of width $2/40 = 0.05$ and get a 40×40 matrix of $z_{i,j}$ values corresponding to $f(x_i, y_j)$ where (x_i, y_j) is the center of the $(i,j)^{\text{th}}$ square area, $A_{i,j}$ with area $dA_{i,j} = (0.05)^2$. Here $x_i = (-1 + 0.025) + i \cdot 0.05 = -0.975 + i \cdot 0.05$. Similarly $y_j = -0.975 + j \cdot 0.05$. The function taken for this example is $f(x,y) = \sin(2\pi xy^2 - y) - \cos(4\pi x^2 y + x) + x^2 - y^3$. The 40×40 matrix of values will be called Z, and MATLAB can provide a surface plot together with a coloring of the matrix that represents the z values.

This is all very easy to achieve and plot in MATLAB. Here is code taken from the provided MATLAB live script SVDProject.mlx:





Here is what Z looks like as a matrix:

$$Z = \begin{bmatrix} 2.3070 & 2.1675 & 1.9471 & 1.6540 & \cdots \\ 2.1036 & 1.8324 & 1.5112 & 1.1539 & \cdots \\ 1.6833 & 1.3435 & 0.9860 & 0.6241 & \cdots \\ 1.1526 & 0.8045 & 0.4634 & 0.1390 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

It is important to understand how the indices in the Z-matrix correspond to the x, y-plane so here is X and Y are

$$X = \begin{bmatrix} -0.9750 & -0.9250 & -0.8750 & -0.8250 & \cdots \\ -0.9750 & -0.9250 & -0.8750 & -0.8250 & \cdots \\ -0.9750 & -0.9250 & -0.8750 & -0.8250 & \cdots \\ -0.9750 & -0.9250 & -0.8750 & -0.8250 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

and

$$Y = \begin{bmatrix} -0.9750 - 0.9750 - 0.9750 - 0.9750 & \cdots & & & \\ -0.9250 - 0.9250 - 0.9250 - 0.9250 & \cdots & & & \\ -0.8750 - 0.8750 - 0.8750 - 0.8750 & \cdots & & & \\ -0.8250 - 0.8250 - 0.8250 - 0.8250 & \cdots & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

So $X(i,j) = x_j$ and $Y(i,j) = y_i$ so $z(x_j, y_i) = z(X(i,j), Y(i,j)) = Z(i,j)$.

Let $Z = U\Sigma V^T$ be the SVD decomposition of Z so that $Z = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$, where $k = \text{rank}(Z)$.

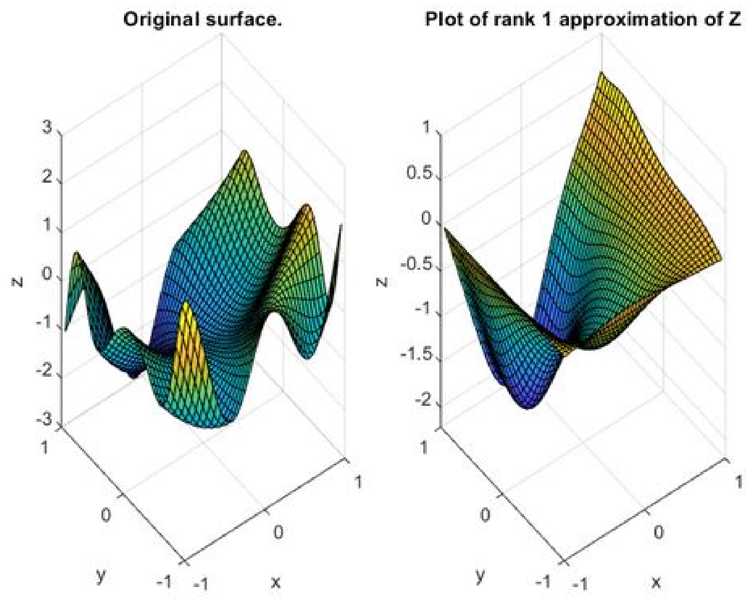
Let $Z(r)$ be the "best" rank-r approximation of Z, this is given by

$$Z(r) = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$$

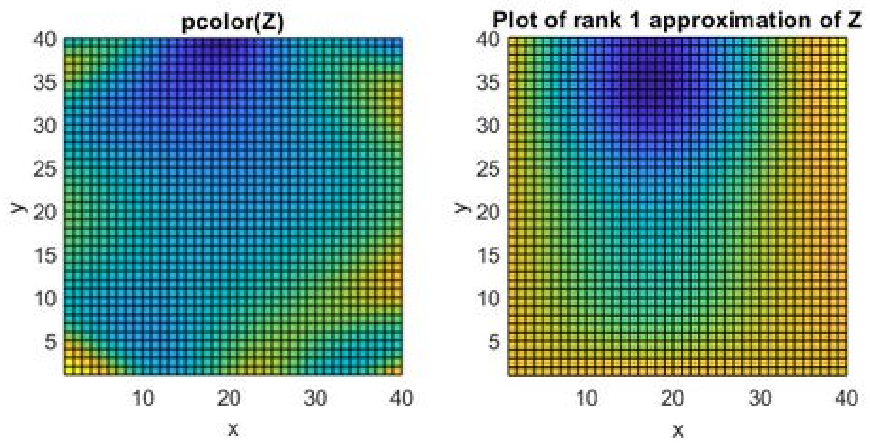
Think about the interpretation of $\boldsymbol{u}_i \boldsymbol{v}_i^T$ here. These are rank-1 matrices, each of which can be viewed as a surface/image, and the original image surface/image Z is the weighted sum of these rank-1 surfaces/images. The rank-1 surface $\boldsymbol{u}_i \boldsymbol{v}_i^T$ is very simple to interpret. All horizontal slices look like v_i, itself viewed as a function of "x" and u_i viewed as a function of "y." So it is as if $f_i(x,y) = u_i(y)v_i(x)$ is a separable function of two variables, and $f(x,y) = \sum_i \sigma_i f_i(x,y) = \sum_i \sigma_i u_i(y) v_i(x)$. Then the *rank-r* approximation to f is $f_r(x,y) = \sum_{i=1}^{r} \sigma_i u_i(y) v_i(x)$ for $r \leq k = \text{rank}(Z)$.

The rank-1 approximation

Here are the images of the rank-1 approximations to Z:

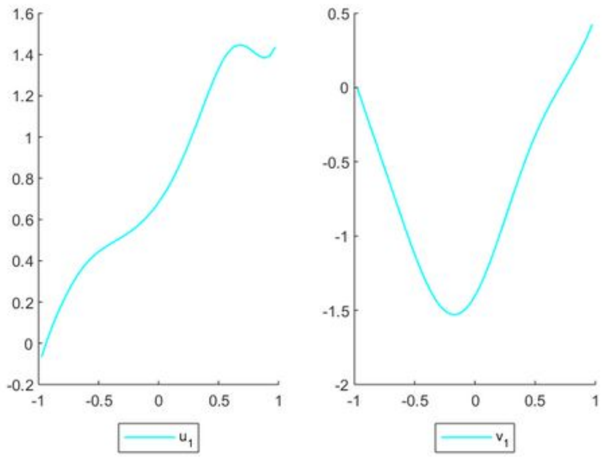


Original surface compared to the rank-1 approximation $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$

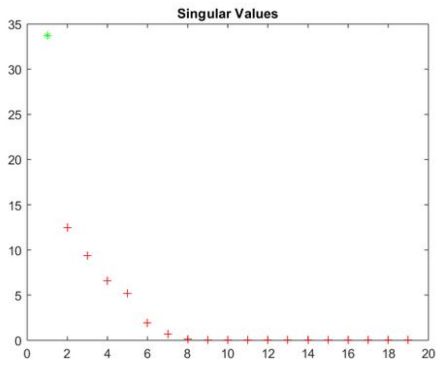


Coloring of the two matrices Z and $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$ (the "best" rank-1 approximation to Z).

Here are the images of the two functions generating these, so think of this as $f_1(x, y) = \sigma_1 u_1(x) v_1(y)$.



The scree plot gives some idea about how well the rank-1 surface approximates the original.



We can measure the percentage of the *information* or *variance* in Z captured by our rank-r approximation $Z(r)$ as

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{k} \sigma_i^2}$$

For this rank-1 approximation we have

$$\frac{\sigma_1^2}{\sum_{i=1}^{k} \sigma_i^2} = 78.07\%$$

so about 78.07% of the "information" in the original surface is captured by this rank-1 approximation.

Another familiar measure for how well the rank-1 approximation $Z(1) = \sigma_1 u_1 v_1^T$ would be to take the L^2 distance between the surfaces. This would be

$$\|f(x,y) - f_1(x_y)\|^2 = \iint_A (f(x,y) - f_1(x,y))^2 \, dx \, dy$$

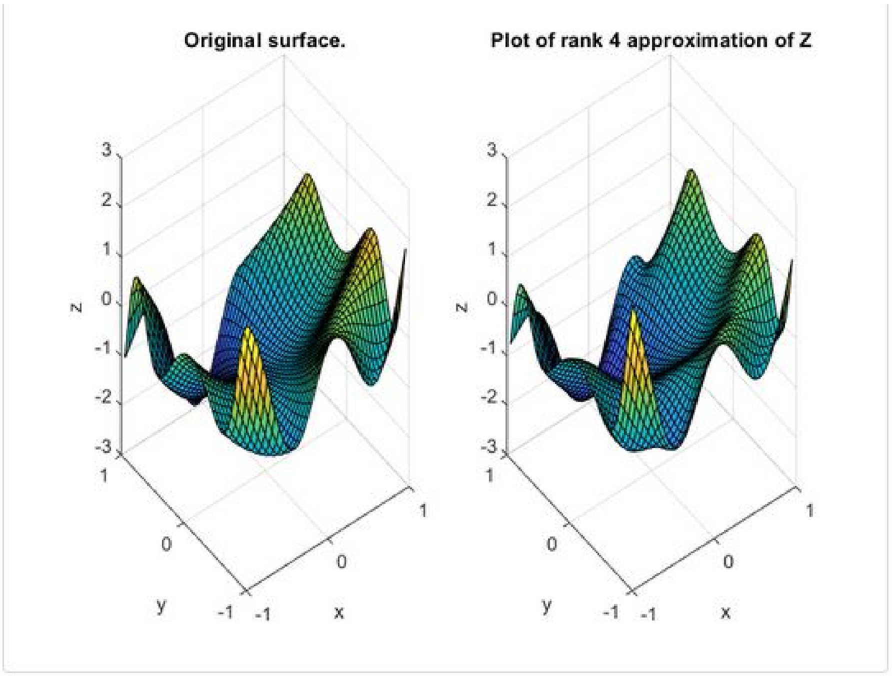
which is approximated here by:

$$L^2\text{-error: } \sum_i \sum_j (Z_{i,j} - Z(1)_{i,j})\, dA_{i,j} = \sum_i \sum_j (Z_{i,j} - \sigma_i u_1(i) v_1^T(j))^2\, dA_{i,j}$$

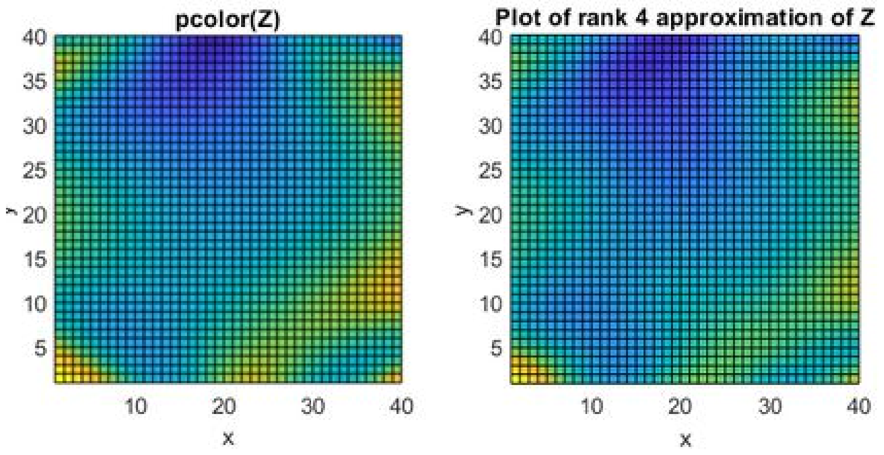
For the rank-1 approximation, we get the L^2-error to be 0.7999.

Rank-4 Approximation.

Above, the rank-1 approximation was investigated. Here, we look at the ran-4 approximation. The rank-4 approximation $Z(4) = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \sigma_3 \boldsymbol{u}_3 \boldsymbol{v}_3^T + \sigma_4 \boldsymbol{u}_4 \boldsymbol{v}_4^T$ captures about 97.736% of the information! Here are plots of the rank-4 approximation of Z vs Z itself.

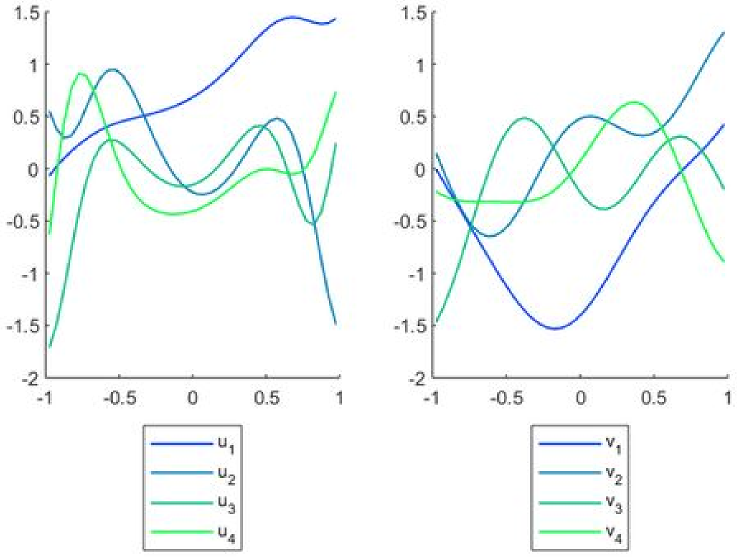


Original surface compared to $Z(4)$ (the rank-4 approximation)



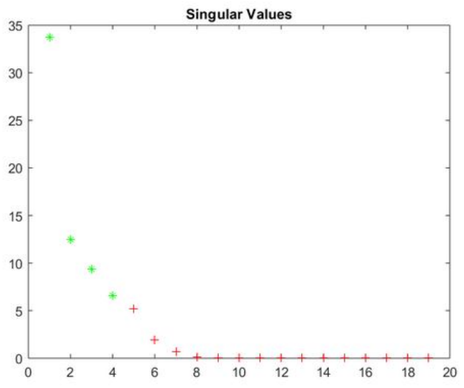
Here are the coloring of the two matrices Z and $Z(4)$ (the "best" rank-4 approximation to Z).

It is clear that more details of the original surface are captured here.



Plot of $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4$ (the horizontal slices) and $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4$ (the vertical slices).

The following is again the scree plot of the singular values showing the relative weights of the 4 singular values used.



$$\frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2}{\sum_{i=1}^k \sigma_i^2} = 0.9785 = 97.85\%$$

So about 97.85% of the original information from the surface is contained in the rank-4 approximation.

The L^2-error is

$$\sum_{\substack{1 \le i \le 41 \\ 1 \le j \le 41}} (Z(i,j) - Z(4)(i,j))^2 \, dA = 0.07828$$

Which is significantly smaller than the rank-1 L^2-error.

Image Reduction Project

A surface can simply be interpreted as a matrix of numbers; similarly, a gray image can be interpreted as a matrix of "gray values," which, if one was so inclined, could itself be interpreted as a surface. Now that we have investigated low-rank approximations of a surface, let's consider low-rank approximations of an image. This immediately leads to the idea of dimension reduction and one method of *compressing* images.

The file "SVDImRed.mlx" has code that reads in an image from the web or your computer, converts the image to a gray image which is just a matrix like Z above, in this case the matrix is called "GrayImg." An SVD decomposition is created GrayImg $= USV^T$. You can view the image reconstructed exactly as described above for the reconstruction of the surface by using the command "show(M,N)" which will reconstruct the image using the M^{th} through N^{th} singular values/singular vectors, i.e.,

$$\text{GrayImg}(M, N) = \sum_{i=M}^{N} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$$

Here, you should complete the following tasks and prepare a report on what you find.

- Look at the first ten rank-1 reconstructions, that is, run show(i,i) for each i for values $i = 1$ through $i = 10$. Comment on what you think the i^{th} rank-1 image (singular value) is capturing from the original image. Next, look at the rank-10 reconstruction with show(1,10). Comment on what you notice. You might also try looking at show(5,20) and other combinations.
- How much data is in the original image? Each gray value is a 1-byte = 8-bit positive integer value between 0 and 255. So, each pixel requires 1 byte of information.
- How much data must be stored for the rank-10 reconstruction of the image? What is the percent reduction in data storage?
- What percentage of the variance is accounted for in the rank-10 reconstruction?
- What rank reconstruction is required to capture 99% of the initial variance (information)? What is the percent reduction in storage for this rank? Discuss how this image looks compared to the original.

Recall that the variance explained is

$$\text{variance explained} = \frac{\sum_{i=M}^{N} \sigma_i^2}{\sum_{i=1}^{k} \sigma_i^2},$$

where k is the rank of the matrix corresponding to the original image.

Here is a sample project. Your work does not need to be exactly like this. This is simply one example of what you might do and notice.