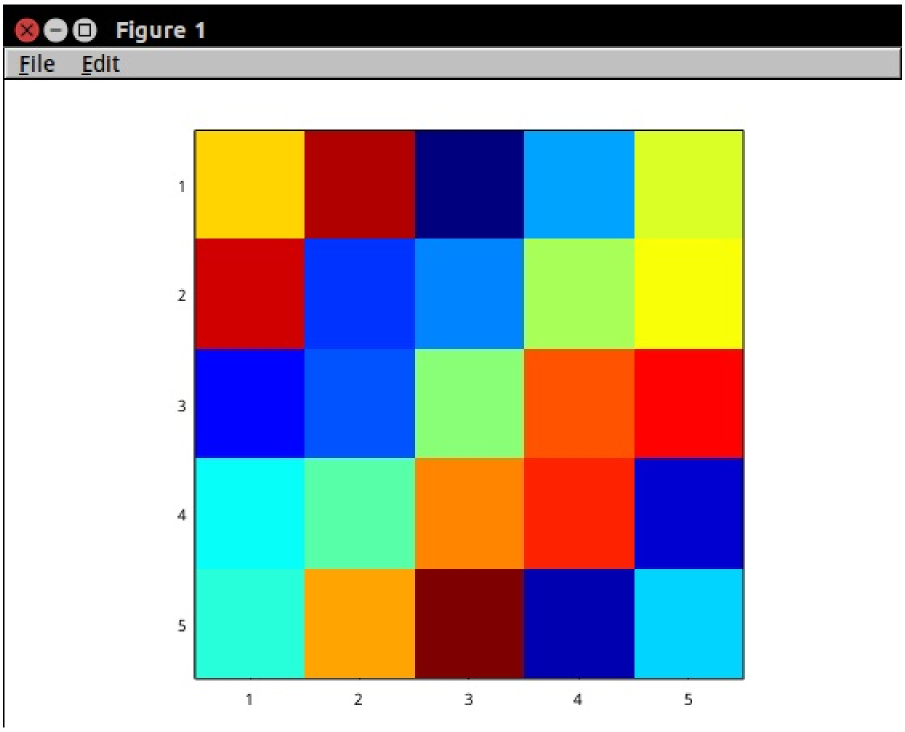
1.

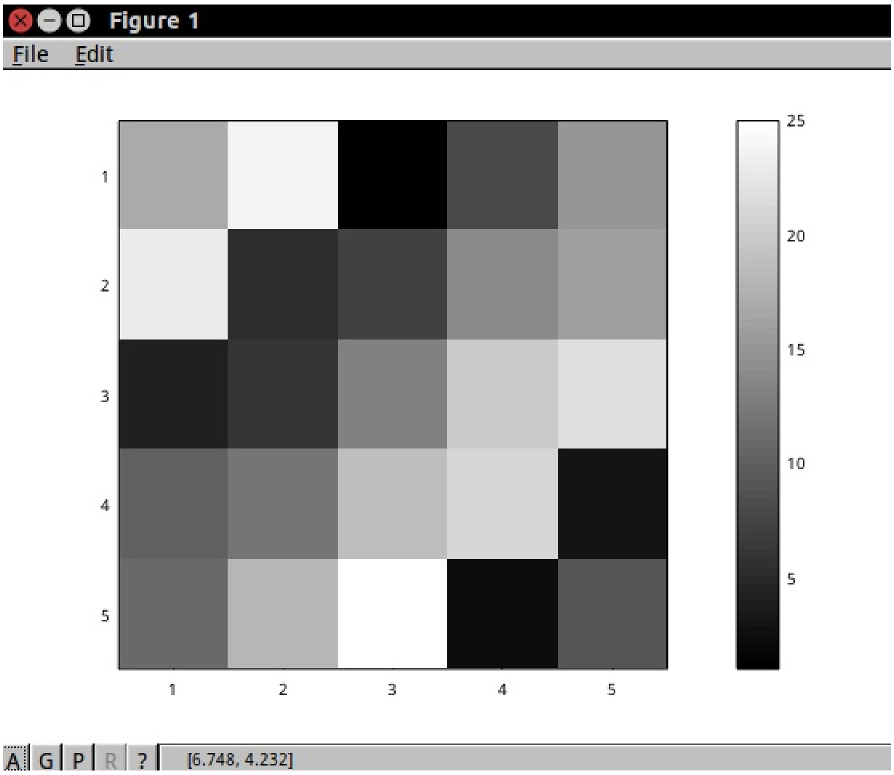
```
octave:109> A = magic(5)
A =

   17   24    1    8   15
   23    5    7   14   16
    4    6   13   20   22
   10   12   19   21    3
   11   18   25    2    9

octave:110> imagesc(A)
```



```
octave:111> imagesc(A), colorbar, colormap gray;
```



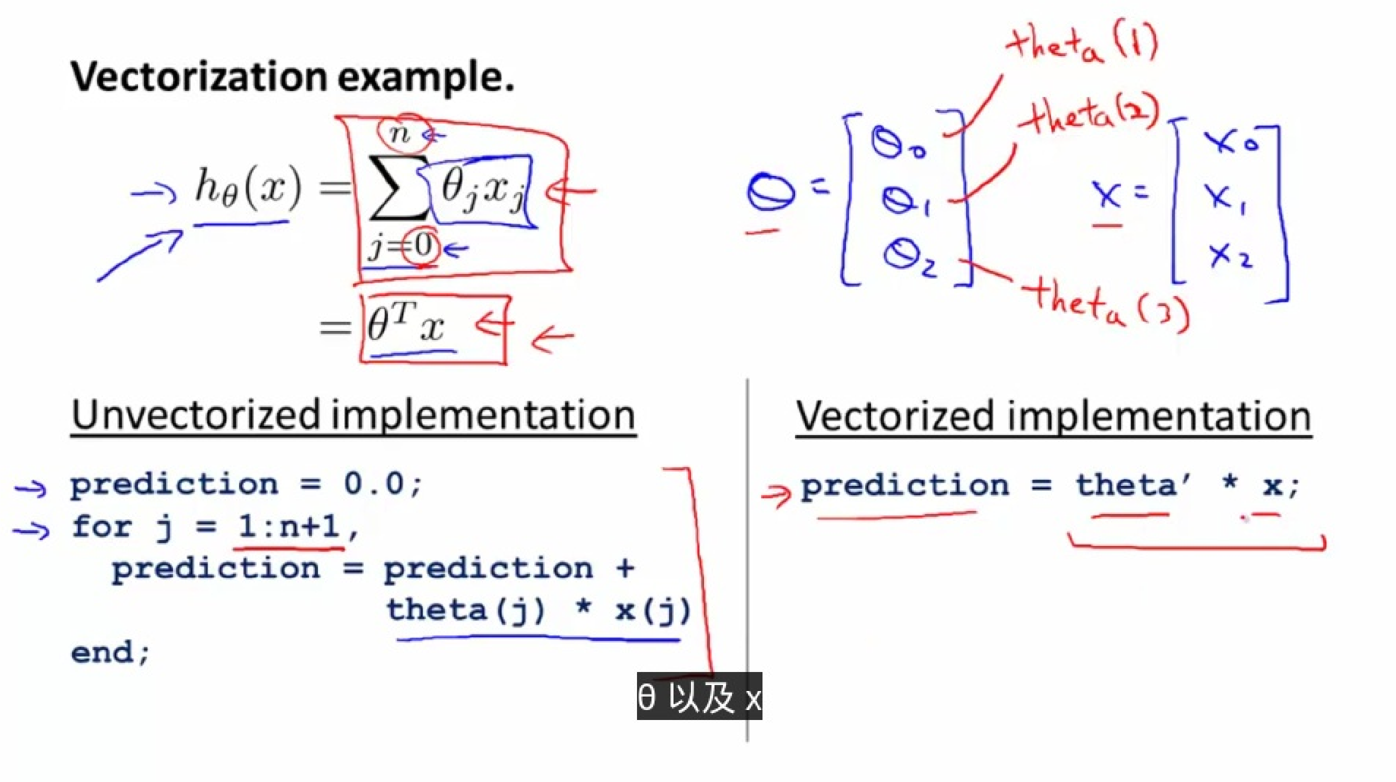
2. for, while, if.

```
octave:114> v = rand(5, 1)
v =

   0.462886
   0.242023
   0.042883
   0.639265
   0.725329

octave:115> for i = v
> disp(i)
> end
   0.462886
   0.242023
   0.042883
   0.639265
   0.725329
octave:116>
```

3. 向量化：



Vectorization example.

$$h_\theta(x) = \boxed{\sum_{j=0}^{n} \boxed{\theta_j x_j}} = \boxed{\theta^T x}$$

theta(1)
theta(2)
theta(3)
$\Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \Theta_2 \end{bmatrix}$
$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$

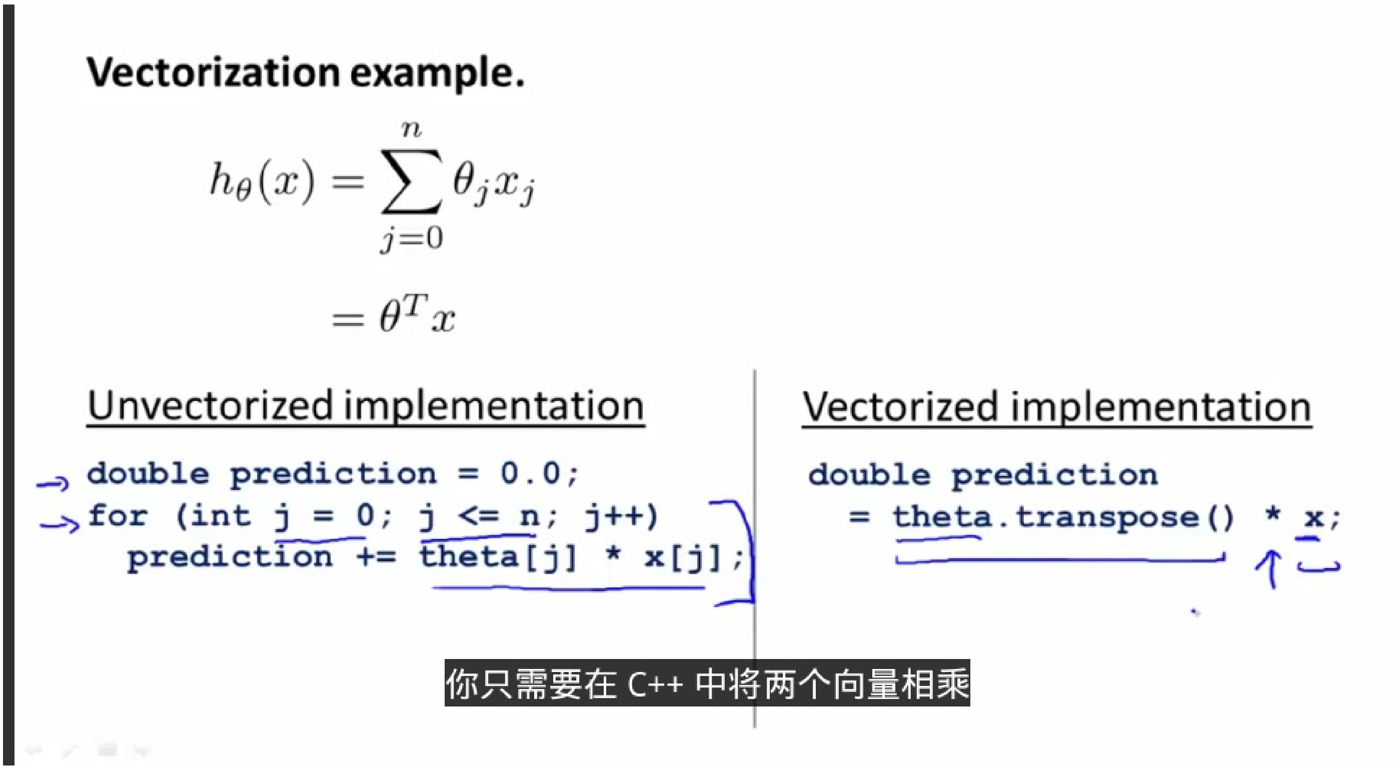
Unvectorized implementation

```
prediction = 0.0;
for j = 1:n+1,
  prediction = prediction +
               theta(j) * x(j)
end;
```

Vectorized implementation

```
prediction = theta' * x;
```

θ 以及 x



Vectorization example.

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j$$

$$= \theta^T x$$

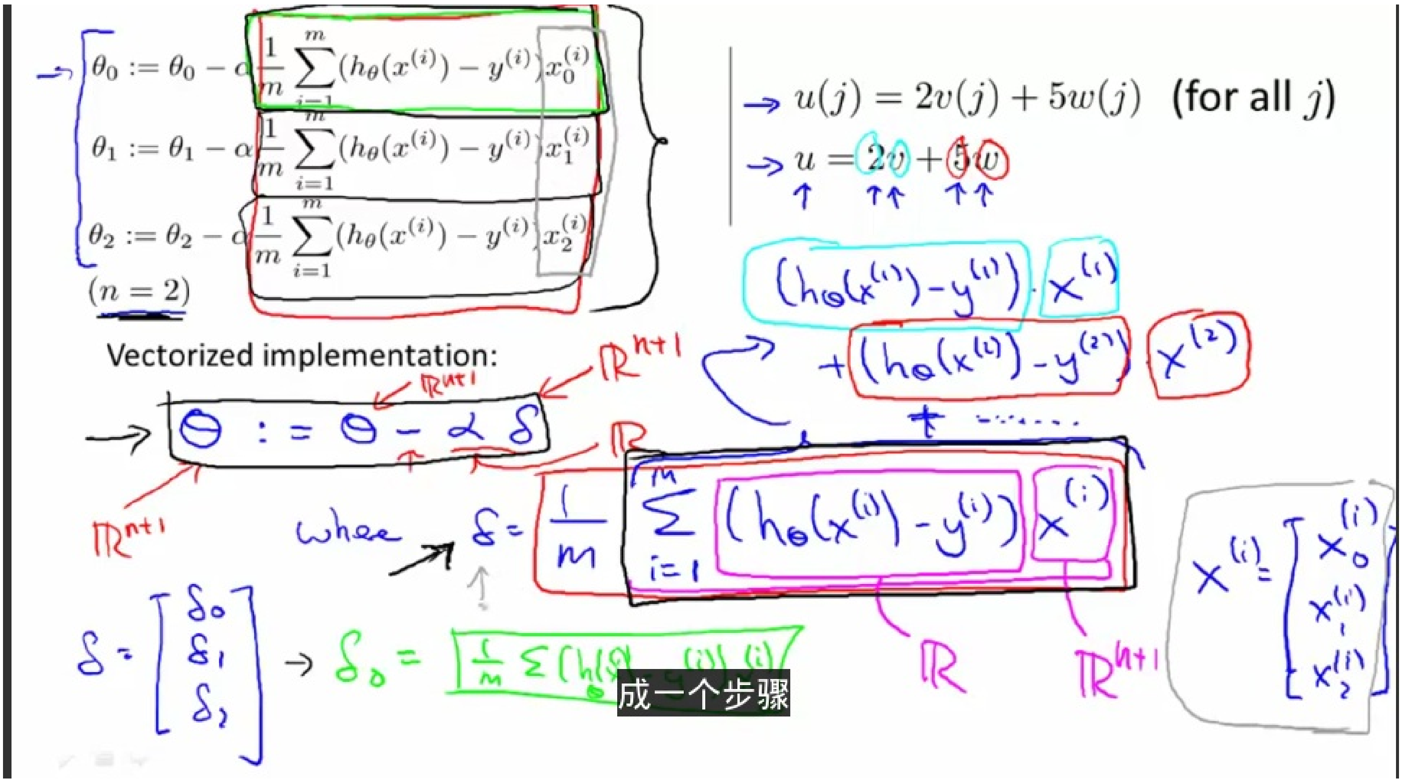
Unvectorized implementation

```
double prediction = 0.0;
for (int j = 0; j <= n; j++)
  prediction += theta[j] * x[j];
```

Vectorized implementation

```
double prediction
  = theta.transpose() * x;
```

你只需要在 C++ 中将两个向量相乘



$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

$(n = 2)$

$u(j) = 2v(j) + 5w(j)$ (for all j)

$u = 2v + 5w$

Vectorized implementation:

$\theta := \theta - \alpha \delta$

where $\delta = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$

成一个步骤

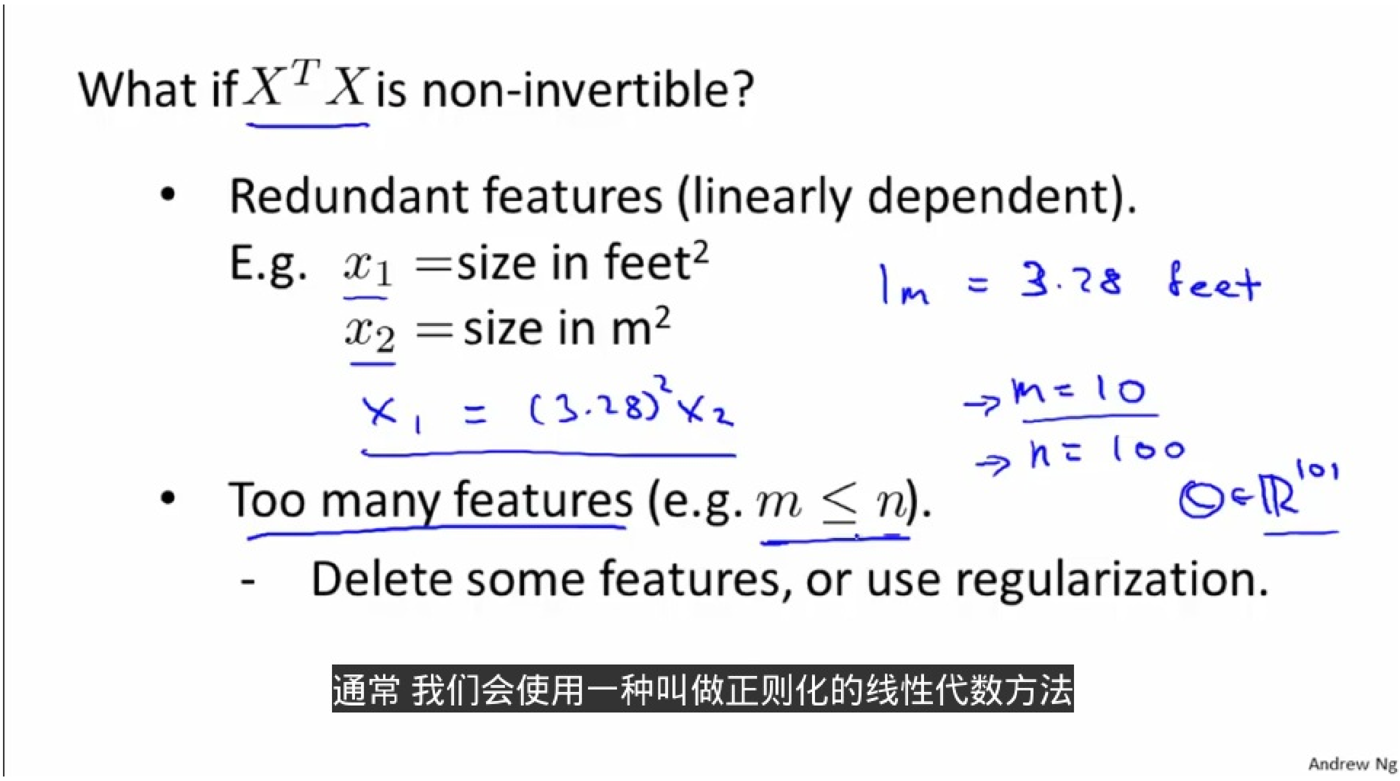
$(h_\theta(x^{(1)}) - y^{(1)}) x^{(1)}$
$+ (h_\theta(x^{(2)}) - y^{(2)}) x^{(2)}$
$+ \cdots$
$\mathbb{R}^{n+1}$
$\mathbb{R}^{n+1}$
$\mathbb{R}$
$\mathbb{R}^{n+1}$
$\mathbb{R}$
$\mathbb{R}^{n+1}$
$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \end{bmatrix}$
$\delta = \begin{bmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \end{bmatrix}$
$\delta_0 = \frac{1}{m} \sum (h(x$

4. 正规方程方法不可逆情况，也即 X'X不可逆：

pinv - 即使矩阵不可逆也可以求出来。 - 伪逆

inv - 矩阵必须是可逆的。

X'X 不可逆是因为： 特征中存在冗余特征，或者特征数大于样本数。例如：



What if X^TX is non-invertible?

- Redundant features (linearly dependent).

  E.g. x_1 = size in feet^2

  x_2 = size in m^2

  $x_1 = (3.28)^2 x_2$

  1m = 3.28 feet

- Too many features (e.g. $m \leq n$).
  - Delete some features, or use regularization.

  $\rightarrow m = 10$

  $\rightarrow n = 100$

  $\theta \in \mathbb{R}^{101}$

通常 我们会使用一种叫做正则化的线性代数方法

Andrew Ng