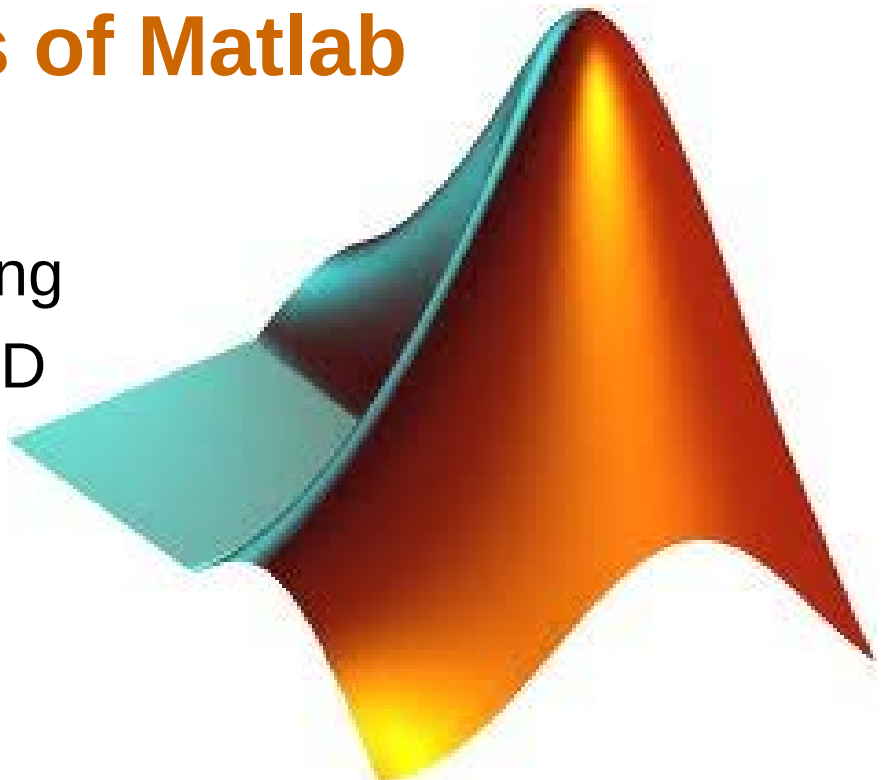
Matlab

Jetzt geht‘s erst richtig los:

Kap. I – Basics of Matlab

Variablentypen und Nutzung

Visualisierung in 2D und 3D



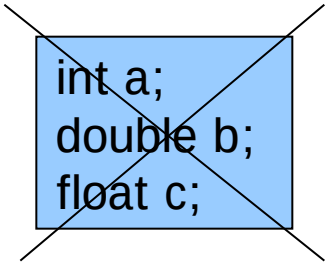
Matlab

– Effiziente Programme durch **Vektorisierung**

➢ **In Matlab gibt es nur Matrizen:**

- ein Skalar ist eine 1x1 Matrix
- ein Zeilenvektor ist eine 1xn Matrix
- ein Spaltenvektor ist eine mx1 Matrix

➢ **In Matlab braucht man keine Deklarierung der Variablen:**



✓ Alle Variablen sind vorab als “double precision” deklariert:

```
>> x = 5;
>> whos x
```

Name	Size	Bytes	Class	Attributes
x	1x1	8	double	

Matlab: Variable Name

- Variable naming rules
  - must be unique in the first 63 characters
  - must begin with a letter
  - may not contain blank spaces or other types of punctuation
  - may contain any combination of letters, digits, and underscores
  - are case-sensitive
  - should not use Matlab keyword
- Pre-defined variable names
  - Pi and some more (see later)

Matlab: Datentypen (Class) und Zuweisung

```
>> x = 5;  % class(x)  double ist default
```

Es gibt aber noch:

Integer	intn,uintn, n=8,16,32,64	(1 bis 8 bytes)
Real	single, double	(4, 8 bytes)
Complex	complex re + img i	(16 bytes)
Logical	true (1) or false (0)	(1 byte)
Character	char	(1 byte)
String	„many“ characters	(„many“ bytes)

```
>> a=int8(2);
b=int8(5);
>>a/b
     0
>> exp(i*pi)
-1.0000 + 0.0000i
>> log(-2)
0.6931 + 3.1416i
```

```
>> x = char(65)
x =
A
>>uint8(‘A‘)
65
>> x = ‘A‘+2
x =
67
```

```
>> char(x)
x=
C
char(2*'A'-9)
y
```

Matlab: Umwandlung Text->Zahl->Text

```
>> s = '12345.6';                    % String mit 7 Characters
>> s+1                               % Wahnsinn, da ASCII-Werte
   51   52   53   54   47   55       % um 1 erhöht werden.

>> z = str2num(s)                    % Umwandlung String in Zahl
z =
   1.2346e+04                        % Zahlendarstellung „short“
   1.23456000000000e+04              % nach  >> format long

>> num2str(z)
   12345.6

>> num2str(z,3)                      % nur 3 signifikante Stellen
   1.23e+04
```

Matlab: Some special variables/values

beep

pi **(π)**

eps **(2.2204e-16 genau)**

inf **(e.g. 1/0)**

i, j **($\sqrt{-1}$)**

NaN **(Not a Number)**

x = [ ] (empty)

```
>> 1/0
Warning: Divide by zero.
ans =
   Inf
>> pi
ans =
   3.1416
>> i
ans =
   0+ 1.0000i
>> 0/0
ans =
   NaN
```

Matlab: Special Characters

Special Characters [ ] ( ) {} = ' . ... , ; : % ! @

Matlab

- **Vergleichsoperatoren**
  - kleiner/größer < >
  - gleich/ungleich ==, ~=
  - größergleich >=
  - kleinergleich <=

- **Logische Operatoren**
  - und && (& bei vektoriellem Vergleich)
  - oder || (| bei vektoriellem Vergleich)
  - nicht ~

Matlab: Vectors and Matrices

- How do we assign values to vectors?

```
>> A = [1 2 3 4 5]
A =
   1   2   3   4   5
```

A row vector –
values are separated by
commas (,) or spaces.

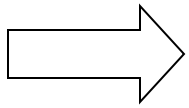
```
>> B =
[10;12;14;16;18]
B =
      10
      12
      14
      16
      18
```

A column vector –
values are separated by
semi–colon (;).

Matlab: Vectors and Matrices

- How do we assign values to matrices ?

```
>> A = [1 2 3;4 5 6;7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
```



$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Columns separated by space or a comma

Rows separated by semi-colon

Matlab

- **Skalar** 12 `>> x=12`
- **Vektor** (1,2) oder $\begin{pmatrix}1\\2\end{pmatrix}$ `>> x=[1,2]` oder `x=[1,2]'`
- **Matrizen**
  - Beliebige Matrizen $\begin{pmatrix}1&3&5&7\\2&4&6&8\end{pmatrix}$

    ```
    >> [1 3 5 7;2 4 6 8]
    ```
  - Spezielle Matrizen $\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$

    ```
    >> eye(3)
    >> ones(2,4)
    >> zeros(1,3)
    ```

    $\begin{pmatrix}1&1&1&1\\1&1&1&1\end{pmatrix}$

    $\begin{pmatrix}0&0&0\end{pmatrix}$
  - Zufallszahlen

    ```
    >> rand(3)
    >> rand(100,100)
    ```

```
>> rand(3)

ans =

    0.8381    0.3795    0.7095
    0.0196    0.8318    0.4289
    0.6813    0.5028    0.3046
```

Matlab

- **Matrizen indizieren**
  - Dimension

```
>> M=rand(3,4)   % m-by-n Matrix
>> size(M,1)     % m = Zeilen
>> size(M,2)     % n = Spalten
```

  - Alle Elemente als Liste

```
>> M(:)
```

Matlab

Werte aus Matrix extrahieren

```
>> A = [1 2 3;3 2 1]
A =
     1     2     3
     3     2     1
```

```
>> B = A'
B =
     1     3
     2     2
     3     1
```

```
>> b = A(1,:)
b =
     1     2     3
>>b = A(:,1)
b =
     1
     2
     3
```

```
>> A(2,2:end)
     2     1
```

```
>> B(:,3)=[4 5 6]'
B =
     1     3     4
     2     2     5
     3     1     6
```

Matlab

Concatenation of Matrices

```
>> x = [1 2];  y = [4 5];  z=[ 0 0];
>>A = [x y]
   1   2   4   5

>> B = [x; y]
     1   2
     4   5
```

```
>> C = [x y ;z]
Error:
??? Error using ==> vertcat CAT arguments dimensions are not consistent.
```

Matlab

- **Matrizen indizieren**
  - Mit Logik

```
>> x=2:7
   2 3 4 5 6 7
>> x>4
   0 0 0 1 1 1
```

  - Werte ausgeben

```
>> x(x>4)
   5 6 7
```

Matlab

- **Matrizen sortieren/umformen**
  - sortieren

```
>> x=6:-1:1
   6 5 4 3 2 1
>> sort(x)
   1 2 3 4 5 6
```

  - umformen

```
>> reshape(x,2,3)
   1 3 5
   2 4 6
```

Matlab: Matrices Operations

Given A and B:

```
>> A = [1 2 3;4 5 6;7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9
```

```
>> B = [3 5 2; 5 2 8; 3 6 9]

B =

     3     5     2
     5     2     8
     3     6     9
```

Addition

```
>> X = A + B

X =

     4     7     5
     9     7    14
    10    14    18
```

Subtraction

```
>> Y = A - B

Y =

    -2    -3     1
    -1     3    -2
     4     2     0
```

Product

```
>> Z = A * B

Z =

    22    27    45
    55    66   102
    88   105   159
```

Transpose

```
>> T = A'

T =

     1     4     7
     2     5     8
     3     6     9
```

Matlab

- **Matrixoperationen**
  - inneres Produkt

```
>> x=[1 -1]     (x‘ ist Spaltenvektor!)
>> x*x‘
2
```

  - äußeres Produkt

```
>> x‘*x
    1 -1
   -1  1
```

Matlab

- **Vektormanipulationen**
  - elementweise Operationen mit „.“

```
>> [1 2 3].*[1 10 100]
     1 20 300
>> [10 20 30]./[5 20 60]
    2.0000 1.0000 0.5000
>> [2 4 8].^2
    4 16 64
```

Matlab: Matrix Manipulation

- Determinante einer Matrix: `d = det(A)`
- Eigenvalues and eigenvectors: `[V,D] = eig(A)`
- Singular value decomposition: `[U,S,V] = svd(A)`
- Orthogonal-triangular decomposition: `[Q,R] = qr(A)`
- LU factorization: `[L,U] = lu(A)`
- Matrix rank: `a = rank(A)`
- Condition number: `a = cond(A)`

Matlab Task:

plot the function sin(x) between $0 \leq x \leq 4\pi$

1. Create an x-array of 100 samples between 0 and 4π.

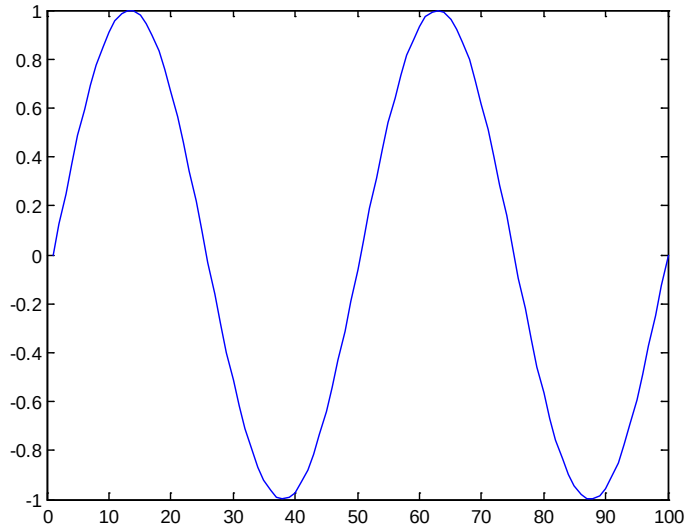
```
>> x = [0:0.1:4*pi];
```

2. Calculate sin(…) of the x-array

```
>> y = sin(x);
```

3. Plot the y-array

```
>> plot(y)
```



Matlab Task: Plot $e^{-x/3}\sin(x)$ between $0 \le x \le 4\pi$

- Create an x-array of 100 samples between 0 and 4π.

```
>> x = [0:0.1:4*pi];
```

```
>>x = linspace(0,4*pi,100);
```

- Calculate sin(…) of the x-array

```
>> y = sin(x);
```

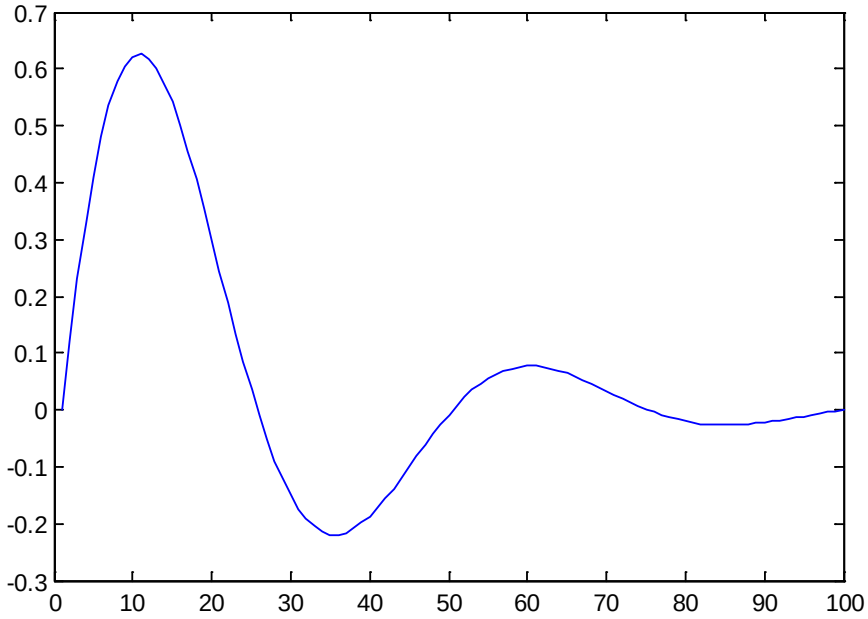
- Calculate $e^{-x/3}$ of the x-array

```
>> y1 = exp(-x/3);
```

- Multiply the arrays y and y1 correctly (y.*y1 NICHT y*y1)

```
>> y2 = y.*y1;
```

- Plot the y2-array



Matlabs Display Facilities (2D)

- plot(…)

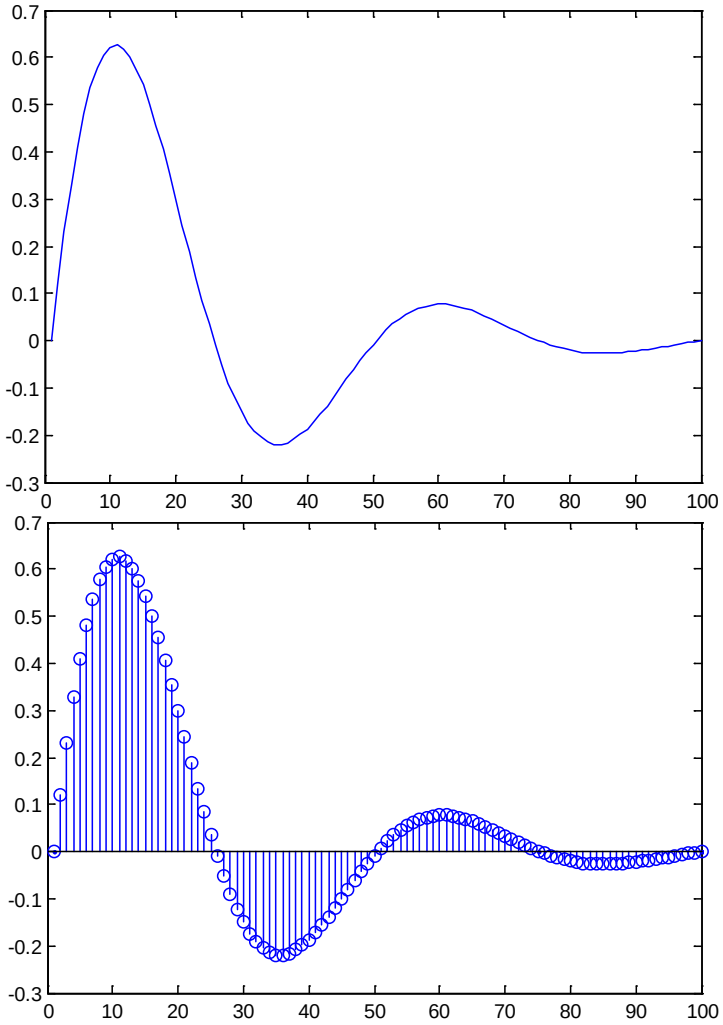
```
Beispiele:
>> x=linspace(0,4*pi,100);
>> y=sin(x);
>> plot(y)
>> plot(x,y)
```

- stem(…)

```
Beispiel:
>> stem(y)
>> stem(x,y)
```

- bar(…)

```
Beispiel:
>> bar(y)
>> bar(y(1:4:end))
```



Matlabs Display Facilities (2D)

- title(…)

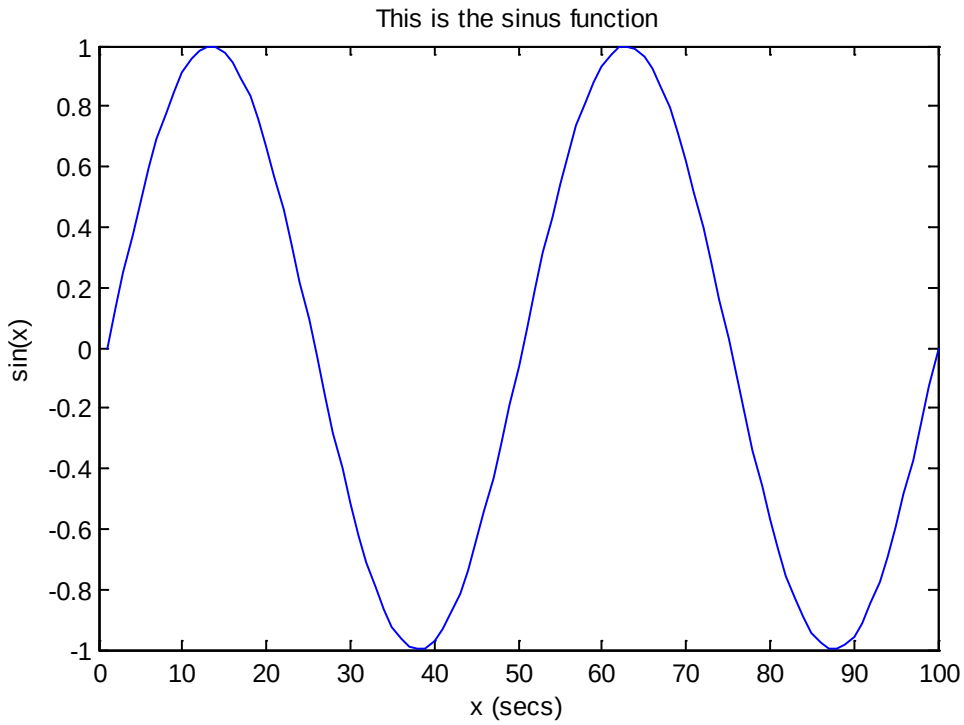
```
>> title('This is the sinus function')
```

- xlabel(…)

```
>>xlabel('x (secs)')
```

- ylabel(…)

```
>>ylabel('sin(x)')
```

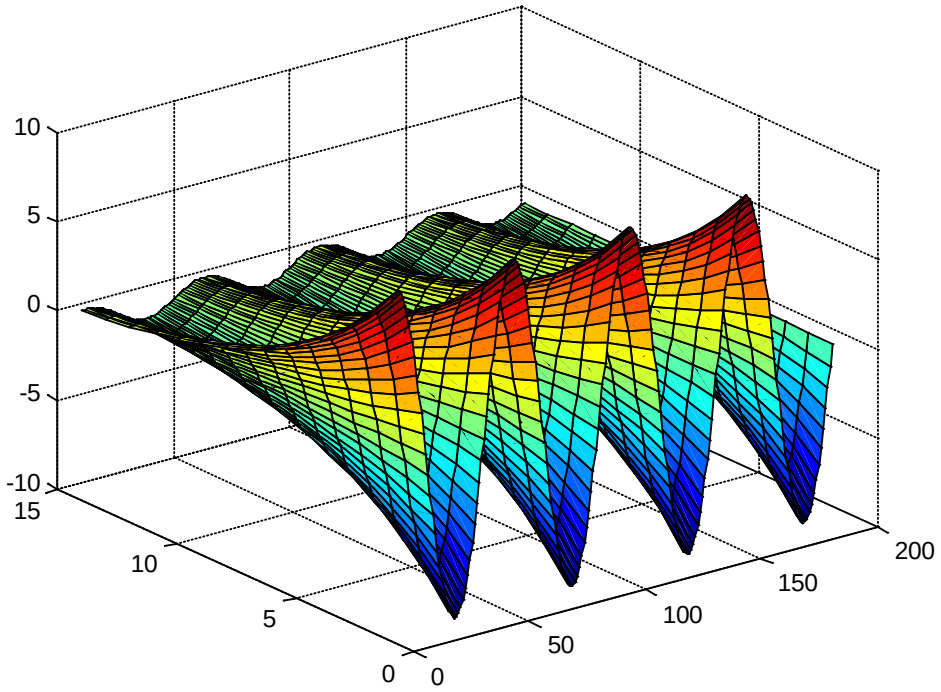


Matlabs Display Facilities (3D)

Example on *mesh* and *surf* – 3 dimensional plot

Supposed we want to visualize a function $\mathbf{Z = 10e^{(-0.4a)} \sin(4\pi t)}$ when *t* is varied from 0.1 to 2 and *a* from 0.1 to 7.

```
>> [t,a] = meshgrid(0.1:.01:2, 0.1:0.5:7);
>> Z = 10.*exp(-a.*0.4).*sin(4*pi.*t);
>> surf(Z);
>> mesh(Z);
```



Matlabs Display Facilities: image

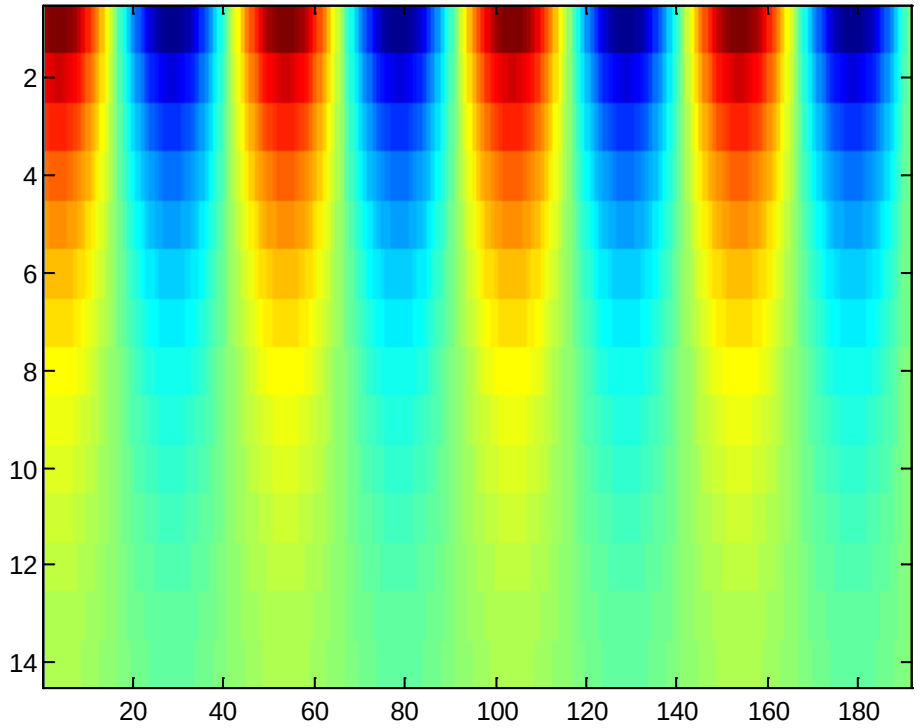
- **Grafische Ausgabe von Daten**
  - neues Ausgabefenster öffnen

```
>> figure
```

  - Daten in einem 2D Bild ausgeben

```
>> imagesc(Z);
>> colorbar
```

sc = scaled image



Matlabs Display Facilities: image

```
>> [x,y] = meshgrid(-10:.1:10,-10:.1:10);
>> z = (x.^2+y.^2)/200;
>> F = exp(-z).*cos(5*z).^2
>> mesh(F)

>> imagesc(F); axis('image')
```