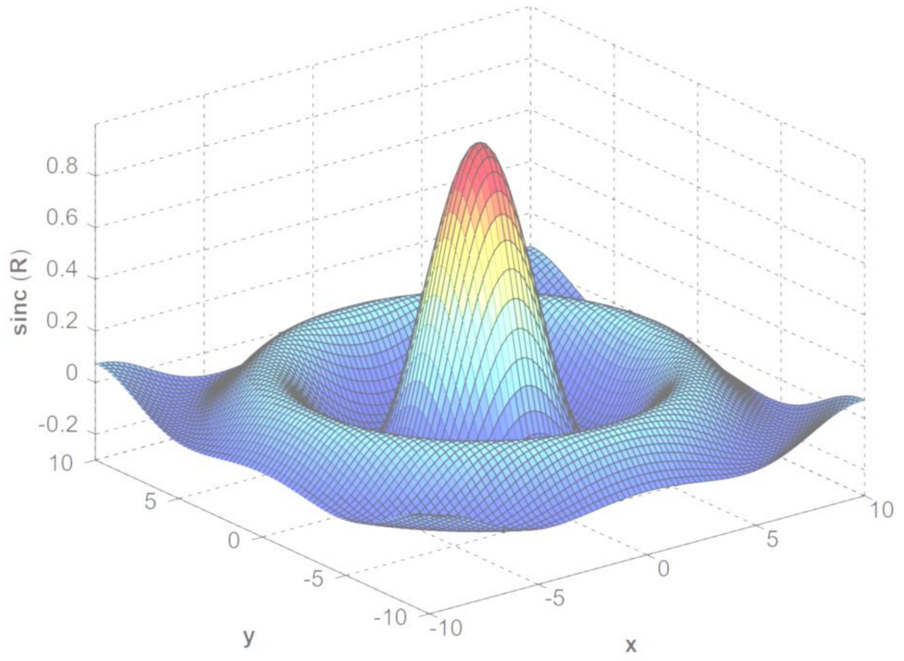
2021년도 2학기

응용전산및실습 II (02) #5



교과목명 : 응용전산 및 실습 II (02)

담당교수 : 이 수 형

E-mail : soohyong@uu.ac.kr

교재명 : 유인물

Matlab 그래프

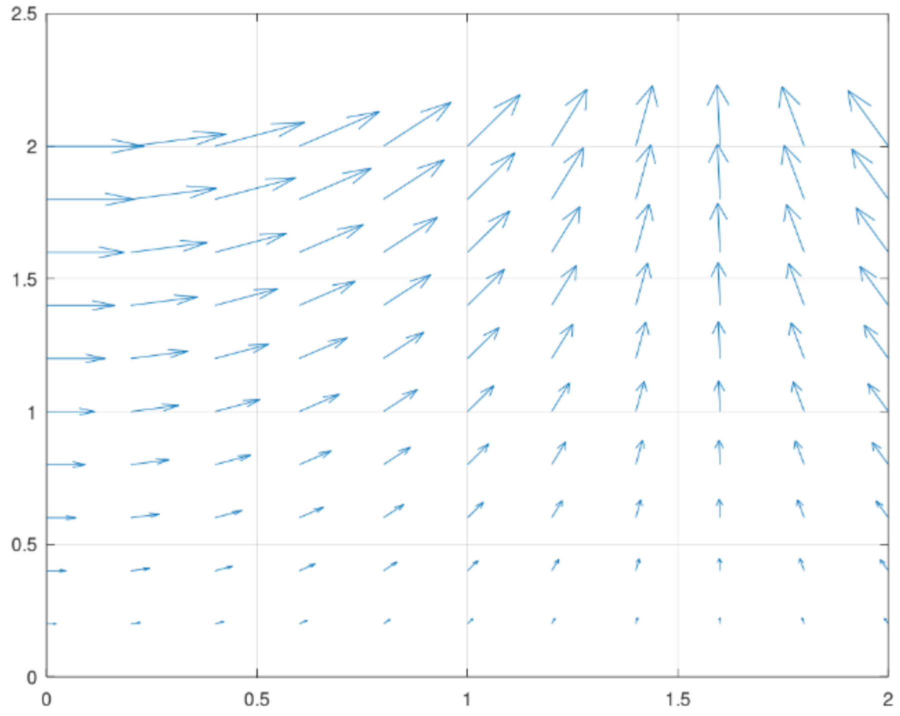
벡터 표시 #1

- 벡터(화살표)의 표시
  - quiver, quiver3 함수를 사용
  - 예] $\mathbf{F} = y \cdot \cos x \, \mathbf{a}_x + y \cdot \sin x \, \mathbf{a}_y$

```
[x,y] = meshgrid(0:0.2:2,0:0.2:2);
u = cos(x).*y;
v = sin(x).*y;

figure
quiver(x,y,u,v)
grid
```

x, y 위치에 u, v 성분의 벡터로 표시

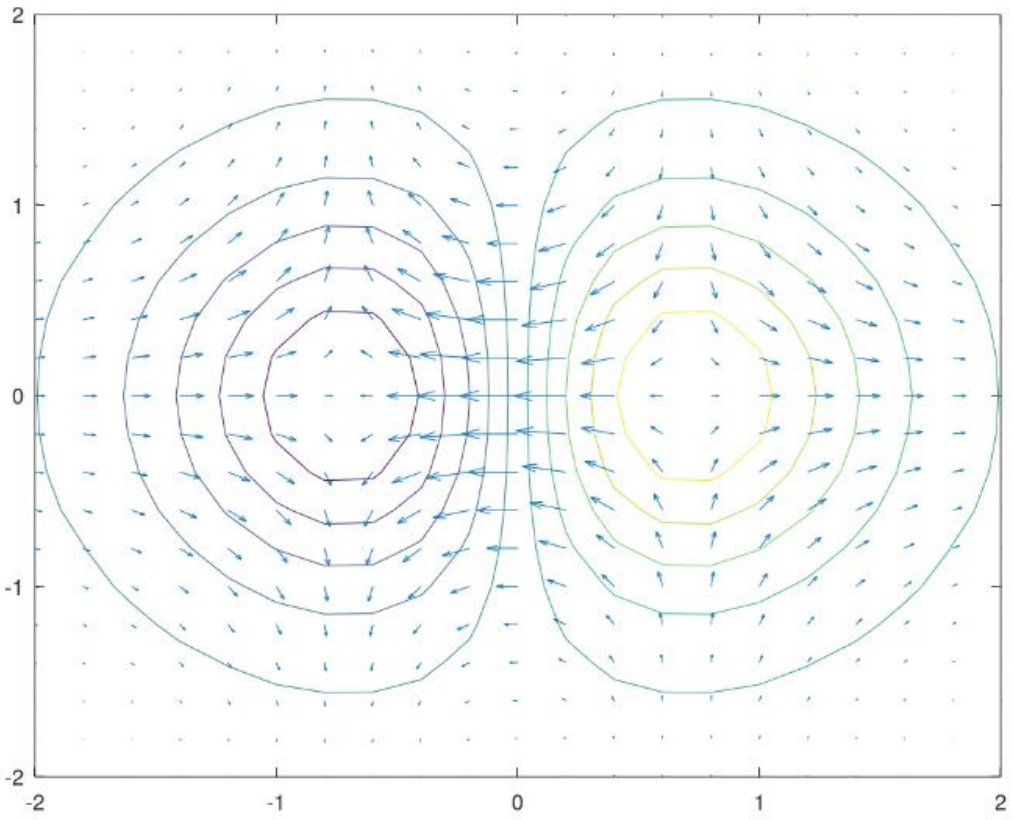


벡터 표시 #2

- 전위의 분포가 $V = xe^{-x^2-y^2}$으로 분포할 때 전계 $\mathbf{E}$의 분포를 나타내어라.
  - 전계의 세기는 $\mathbf{E} = -\nabla V$이므로

```
[X,Y] = meshgrid(-2:.2:2);
Z = X.*exp(-X.^2 - Y.^2);
[DX,DY] = gradient(Z,.2,.2);

figure
contour(X,Y,Z)
hold on
quiver(X,Y,-DX,-DY)
hold off
```



벡터 표시 #3

- 3차원 벡터(화살표)의 표시 : quiver3
  - `quiver3(x,y,z,u,v,w)` : x,y,z 위치에 u,v,w 벡터

- 3차원 그림의 보는 각도 설정하기
  - `view(az, el)` : azimuth(방위), elevation(고도)
  - `view([az el])`

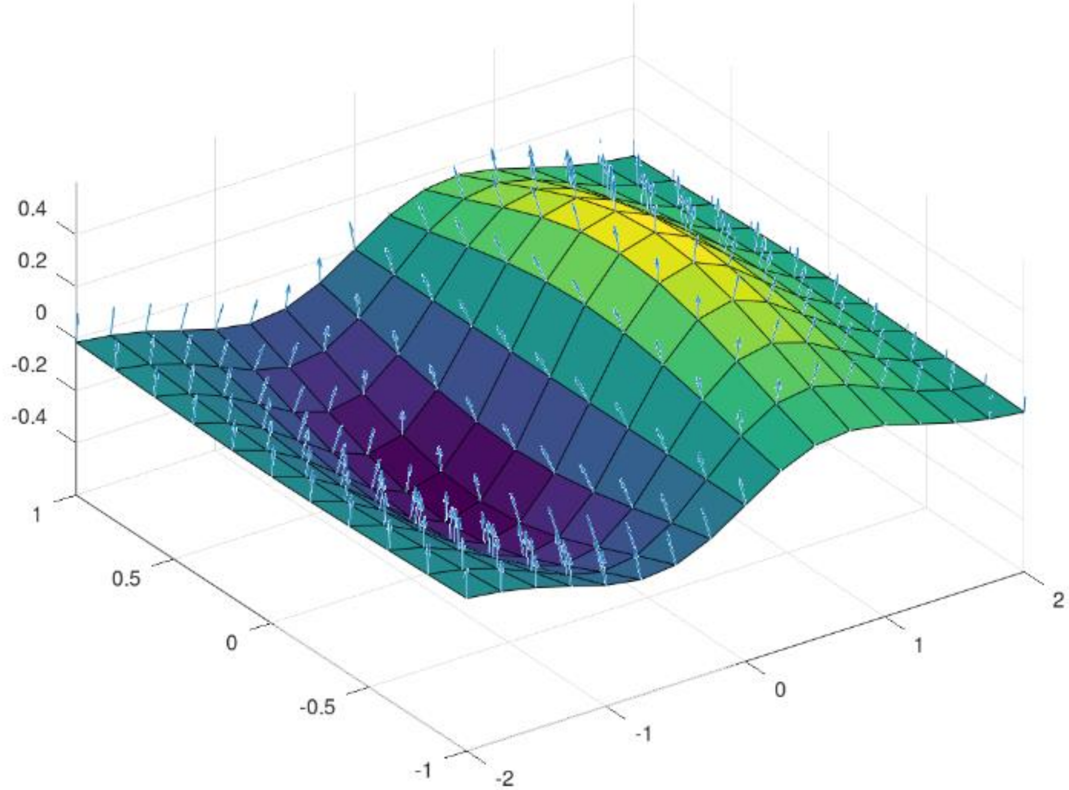
벡터 표시 #4

- 예] $z = xe^{-x^2-y^2}$인 곡면의 법선 그리기
  - 곡면의 법선 구하는 함수 : surfnorm()

```
[X,Y] = meshgrid(-2:0.25:2,-
1:0.2:1);
Z = X.* exp(-X.^2 - Y.^2);
[U,V,W] = surfnorm(X,Y,Z);

figure
quiver3(X,Y,Z,U,V,W,0.5)

hold on
surf(X,Y,Z)
view(-35,45)
axis([-2 2 -1 1 -.6 .6])
hold off
```



응용 예제

- 예제
  - $P_1(-1,0,0)$에 $Q_1 = 1$[C], $P_2(1,0,0)$에 $Q_2 = -1$[C]의 전하가 존재하는 경우 $-2 \leq x, y, z \leq 2$의 영역에서의 전계의 세기 $\mathbf{E}$를 표현하라.
  - 범위 $-2 \leq x, y, z \leq 2$ 내의 점 $P(x, y, z)$에서 주위 전하에 의해 생성되는 전계의 세기

$$\boxed{\mathbf{E} = \frac{Q_i}{4\pi\epsilon_0 R}\mathbf{a}_R, \qquad i = 1, 2}$$

  - x, y, z가 모두 변하는 경우의 수식의 값을 계산할 필요가 있음 ⇒ 3차원 격자
  - 범위내부에서의 E의 방향 : quiver3()
  - 3차원 공간 내부에서의 전계의 세기 E의 크기 : 표현이 어려움 ⇒ slice()

응용 예제 : meshgrid

- meshgrid
  - meshgrid() 함수
    - ➢ x, y 벡터들을 이용하여 x-y평면의 격자 생성
  - [X, Y] = meshgrid(x, y) : 2차원 격자
    - ➢ 인수가 1개인 경우 : x, y축의 값이 동일하게 적용됨

```
>> [X,Y]=meshgrid(1:3, 1:3)
X =

   1   2   3
   1   2   3
   1   2   3

Y =

   1   1   1
   2   2   2
   3   3   3
```

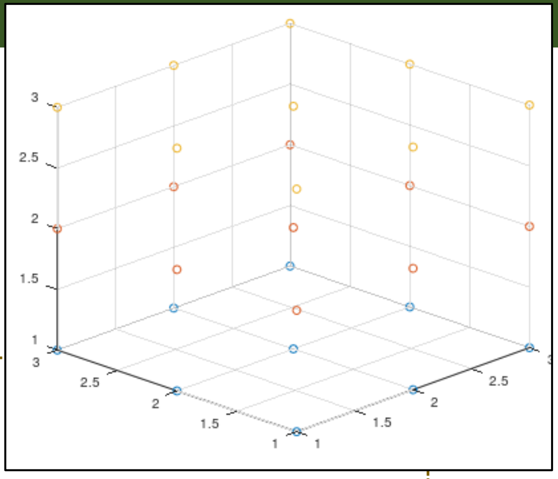
```
>> [X,Y]=meshgrid(1:3)
X =

   1   2   3
   1   2   3
   1   2   3

Y =

   1   1   1
   2   2   2
   3   3   3
```

응용 예제 : meshgrid



- meshgrid
  - [X,Y,Z] = meshgrid(x, y, z) : 3차원 격자
    - ➢ 인수가 1개인 경우 : 출력변수가 3개인 경우에는 3차원 격자가 생성됨

```
>> [X,Y]=meshgrid(1:3)
X =

   1    2    3
   1    2    3
   1    2    3

Y =

   1    1    1
   2    2    2
   3    3    3
```

```
>> [X,Y,Z]=meshgrid(1:3)
X =

ans(:,:,1) =

   1    2    3
   1    2    3
   1    2    3

ans(:,:,2) =

   1    2    3
   1    2    3
   1    2    3

ans(:,:,3) =

   1    2    3
   1    2    3
   1    2    3
```

```
Y =

ans(:,:,1) =

   1    1    1
   2    2    2
   3    3    3

ans(:,:,2) =

   1    1    1
   2    2    2
   3    3    3

ans(:,:,3) =

   1    1    1
   2    2    2
   3    3    3
```

```
Z =

ans(:,:,1) =

   1    1    1
   1    1    1
   1    1    1

ans(:,:,2) =

   2    2    2
   2    2    2
   2    2    2

ans(:,:,3) =

   3    3    3
   3    3    3
   3    3    3
```

응용 예제

- 전계의 세기 **E** 계산

```
qs = [-1 0 0 1; 1 0 0 -1]; % [x, y, z, Q]

eps0 = 8.854e-12;

[Px Py Pz] = meshgrid(-2:0.5:2);
Ex = zeros(size(Px));
Ey = Ex;
Ez = Ex;
Ep = Ex;
```

각 행 : [x, y, z 좌표, 전하량 Q]

3차원 격자 생성

전계의 세기를 저장할 변수 초기화

```
for k=1:length(qs(:,1))
  Rx = Px - qs(k,1);
  Ry = Py - qs(k,2);
  Rz = Pz - qs(k,3);
  R = sqrt(Rx.^2 + Ry.^2 + Rz.^2);
  E = qs(k,4) ./ (4 * pi * eps0 * R);
  Ep = Ep + E;
  ax = Rx ./ R;
  ay = Ry ./ R;
  az = Rz ./ R;
  Ex = Ex + E .* ax;
  Ey = Ey + E .* ay;
  Ez = Ez + E .* az;
end
```

- $\mathbf{R}$: 점 $P(x,y,z)$와 전하의 위치 $P_k(P_{k,x},P_{k,y},P_{k,z})$의 위치벡터
- $R_x = P_{i,x} - P_x,\ \cdots,\ R = |\mathbf{R}| = \sqrt{R_x^2 + R_y^2 + R_z^2}$

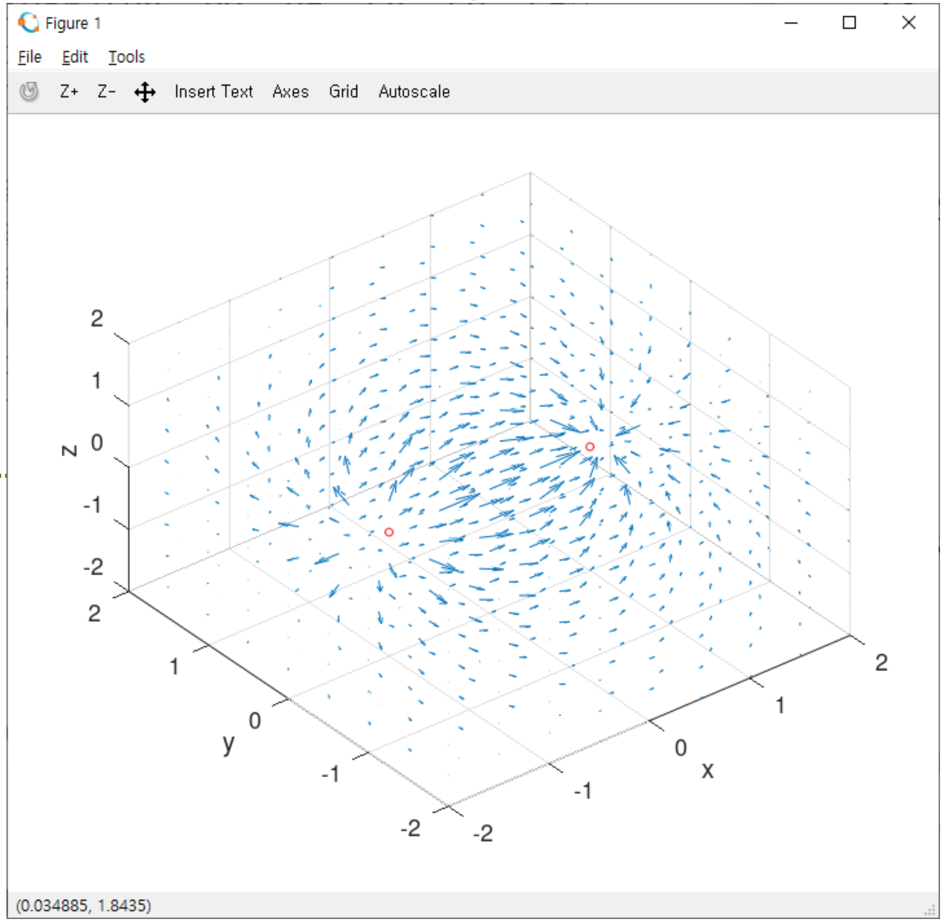
- $\mathbf{E} = \frac{Q_k}{4\pi\epsilon_0 R}\mathbf{a}_R$, $\mathbf{E} = E_x\mathbf{a}_x + E_y\mathbf{a}_y + E_z\mathbf{a}_z$
- $\mathbf{a}_R = \frac{\mathbf{R}}{R},\ a_{R,x} = \frac{R_x}{R}$

응용 예제

- 전계의 세기 E 벡터의 표현

```
plot3(qs(:,1), qs(:,2), qs(:,3), 'ro');
hold on;
quiver3(Px, Py, Pz, Ex, Ey, Ez,'maxheadsize', 0.3);
grid on;
hold off;

xlabel('x');
ylabel('y');
zlabel('z');
set(gca, 'FontSize', 18);
axis([-2 2 -2 2 -2 2]);
```



응용 예제

- 전계의 세기 **E** 계산

```
qs = [-1 0 0 1; 1 0 0 -1]; % [x, y, z, Q]

eps0 = 8.854e-12;

[Px Py Pz] = meshgrid(-2.0001:0.1:2);
Ex = zeros(size(Px));
Ey = Ex;
Ez = Ex;
Ep = Ex;
```

각 행 : [x, y, z 좌표, 전하량 Q]

3차원 격자 생성 (범위 수정)

전계의 세기를 저장할 변수 초기화

```
for k=1:length(qs(:,1))
   Rx = Px - qs(k,1);
   Ry = Py - qs(k,2);
   Rz = Pz - qs(k,3);
   R = sqrt(Rx.^2 + Ry.^2 + Rz.^2);
   E = qs(k,4) ./ (4 * pi * eps0 * R);
   Ep = Ep + E;
   ax = Rx ./ R;
   ay = Ry ./ R;
   az = Rz ./ R;
   Ex = Ex + E .* ax;
   Ey = Ey + E .* ay;
   Ez = Ez + E .* az;
end
```

- $\mathbf{R}$: 점 $P(x,y,z)$와 전하의 위치 $P_k(P_{k,x},P_{k,y},P_{k,z})$의 위치벡터
- $R_x = P_{i,x} - P_x, \cdots, R = |\mathbf{R}| = \sqrt{R_x^2 + R_y^2 + R_z^2}$

- $\mathbf{E} = \frac{Q_k}{4\pi\epsilon_0 R}\mathbf{a}_R$, $\mathbf{E} = E_x\mathbf{a}_x + E_y\mathbf{a}_y + E_z\mathbf{a}_z$
- $\mathbf{a}_R = \frac{\mathbf{R}}{R}, \ a_{R,x} = \frac{R_x}{R}$

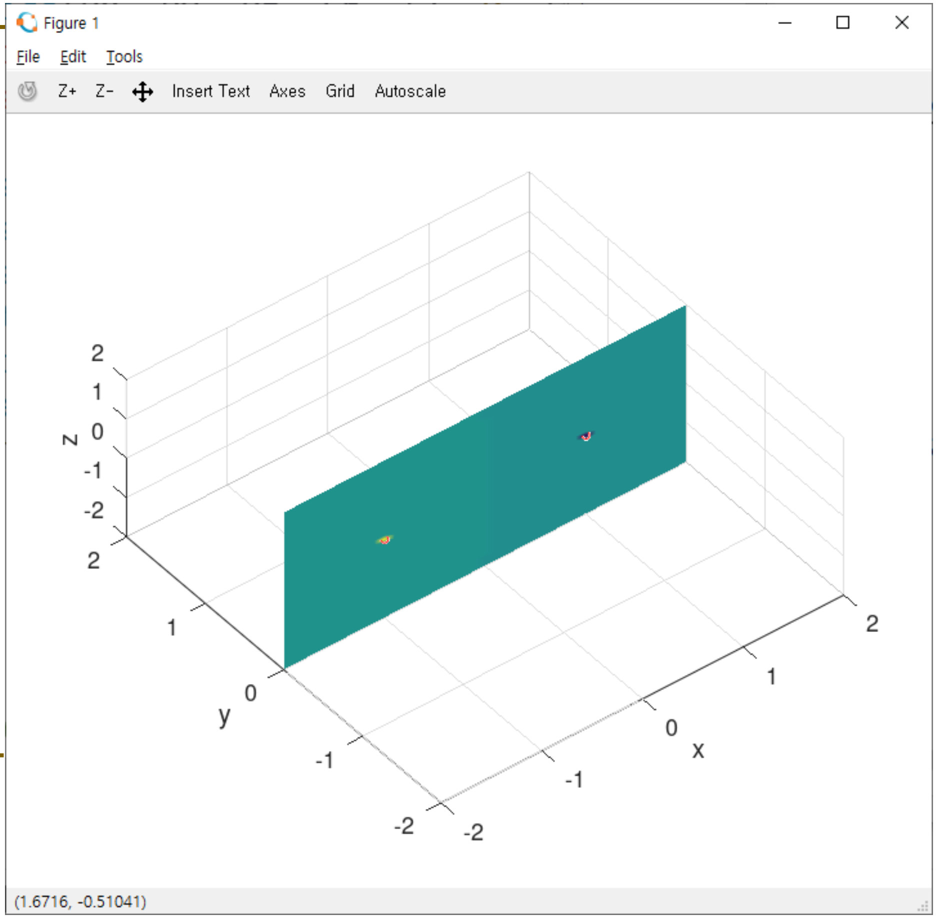
응용 예제

- slice() 함수 : 공간의 단면의 값을 표현
  - `slice(x,y,z,v,sx,sy,sz)` : (x,y,z) – 3차원 격자, v – 3차원 공간에서의 값
    sx, sy, sz – 자를 단면

```
plot3(qs(:,1), qs(:,2), qs(:,3), 'ro');
hold on;
slice(Px, Py, Pz, Ep, [], 0, []);
grid on;
shading interp;

xlabel('x');
ylabel('y');
zlabel('z');
set(gca, 'FontSize', 18);
axis([-2 2 -2 2 -2 2]);
view([-38 65]);
```

$y = 0$에 대한 단면만 계산



  - `slice(…, 0, [-1 0 1], []);`

다항식 다루기

Matlab 다항식 다루기

- 다항식 : 이공학에서 문제를 풀고 모델링하기 위해서 자주 사용되는 수학적인 표현
  - $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$

- Matlab에서의 다항식의 표현
  - $8x + 5$ `p=[8 5]`
  - $2x^2 - 4x + 10$ `d=[2 -4 10]`
  - $6x^2 - 150 = 6x^2 - 0x - 150$ `h=[6 0 -150]`
  - $5x^5 + 6x^2 - 7x = 5x^5 + 0x^4 + 0x^3 + 6x^2 - 7x + 0$

    `c=[5 0 0 6 -7 0]`

  $\Rightarrow$ 각 계수들로 이루어진 벡터를 사용하여 표현

* 참고 : 매트랩의 정석, Amos Gilat저, 인피니티 북스

다항식의 값 계산

- 주어진 x에 대해서 다항식 함수의 값을 계산하는 함수
  - `polyval(p, x)`
    p : 다항식 계수 벡터, x : 주어진 변수
- 다항식의 계산
  - $f(x) = x^5 - 12.1x^4 + 40.59x^3 - 17x^2 - 71.95x + 35.88$
  - $f(9)$의 값을 계산하고, 구간 $-1.5 \le x \le 6.7$에 대한 그래프를 그려라

```
>> p=[1 -12.1 40.59 -17.015 -71.95 35.88];
>> polyval(p, 9)

ans =
   7.2611e+03

>> x=-1.5:0.5:6.7;
>> y=polyval(p,x);
>> plot(x,y);
```

다항식의 근 계산

- 다항식의 근을 계산하는 함수
  - `r = roots(p)`
    p : 다항식 계수 벡터, r : 계산된 근을 포함하는 벡터
- 다항식의 근 구하기
  - $f(x) = 4x^2 + 10x - 8$의 근 구하기
  - $f(x) = x^5 - 12.1x^4 + 40.59x^3 - 17x^2 - 71.95x + 35.88$의 근 구하기

```
>> roots([4 10 -8])
ans =
   -3.1375
    0.6375
>> p=[1 -12.1 40.59 -17.015 -71.95 35.88];
>> roots(p)'
ans =
    6.5000    4.0000    2.3000   -1.2000    0.5000
```

다항식의 근 계산

- 다항식의 근으로부터 다항식의 계수를 구하는 함수
  - `p = poly(r)`
    p : 다항식 계수 벡터, r : 근을 포함하는 벡터
- 예제]
  - $f(x) = x^5 - 12.1x^4 + 40.59x^3 - 17x^2 - 71.95x + 35.88$의 근을 구하고, 그 근들로부터 다항식의 계수를 구하기

```
>> p=[1 -12.1 40.59 -17.015 -71.95 35.88];
>> r=roots(p);
>> pp=poly(r)

pp =

    1.0000  -12.1000   40.5900  -17.0150  -71.9500
35.8800
```

다항식의 연산

- 두 다항식의 덧셈 : 계수 벡터를 더함 (길이를 맞춰줘야 함)
  - $f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$
  - $f_2(x) = 3x^3 - 2x - 6$

```
>> p1=[3 15 0 -10 -3 15 40];
>> p2=[3 0 -2 -6];
>> p1 + p2
행렬의 차원이 일치해야 합니다.

>> p1 + [0 0 0 p2]
ans =
     3    15     0    -7    -3    13    34
```

다항식의 연산

- 곱셈 : conv() 함수 사용 (convolution)
  - $f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$
  - $f_2(x) = 3x^3 - 2x - 6$

```
>> p1=[3 15 0 -10 -3 15 40];
>> p2=[3 0 -2 -6];
>> pm=conv(p1, p2)

pm =

     9     45     -6    -78    -99     65    186    -12   -170   -
240
```

➢ $f_m(x) = f_1(x) + f_2(x)$
$= 9x^9 + 45x^8 - 6x^7 - 78x^6 - 99x^5 + 65x^4 + 186x^3 - 12x^2 - 170x - 240$

다항식의 연산

- 나눗셈 : deconv() 함수 사용 (deconvolution)
  - `[q, r] = deconv(u, v)`
    다항식 u를 v로 나누는 경우 몫은 q에, 나머지는 r에 저장됨
  - $f_1(x) = 2x^3 + 9x^2 + 7x - 6$를 $f_2(x) = x + 3$로 나누는 경우

```
>> u=[2 9 7 -6];
>> v=[1 3];
>> [q, r]=deconv(u, v)
q =
     2     3    -2
r =
     0     0     0     0
```

  - $2x^6 - 13x^5 + 75x^3 + 2x^2 - 60$을 $x^2 - 5$로 나누는 경우는?

연습

- 다음 다항식의 그래프를 polyval() 함수를 사용하여 그려라.
  - $f(x) = 0.1x^5 - 0.2x^4 - x^3 + 5x^2 + 235, -6 \leq x \leq 6$
  - $f(x) = 0.008x^4 - 1.8x^3 - 5.4x + 54, -14 \leq x \leq 16$
- 다음의 두 다항식의 곱셈을 수행하라.
  - $(-x^3 + 5x - 1)(x^4 + 2x^3 - 16x + 5)$
  - $x(x - 1.7)(x + 0.5)(x - 0.7)(x + 1.5)$, 이 결과를 $-1.6 \leq x \leq 1.8$에 대해서 그려라.
- 연속적인 두 수의 곱이 6,972인 경우, 다항식 연산을 사용하여 두 정수를 구하라.
- 다음의 다항식 나눗셈을 수행하라.
  - $-10x^6 - 20x^5 + 9x^4 + 10x^3 + 8x^2 + 11x - 3$ 나누기 $2x^2 + 4x - 1$

다항식의 미분

- 미분 : polyder() 함수 사용 (deconvolution)
  - `k = polyder(p)` : 다항식 p를 미분하여 미분한 다항식의 계수 계산
  - `k = polyder(a, b)` : 다항식 a, b의 곱을 미분하여 미분한 결과 계산
  - `[n d] = polyder(u, v)` : 다항식의 나눗셈을 미분하여 결과를 되돌려줌

```
>> f1=[3 -2 4];
>> f2=[1 0 5];
>> k=polyder(f1)
k =
     6    -2
>> d=polyder(f1,f2)
d =
    12    -6    38   -10
>> [n,d]=polyder(f1,f2)
n =
     2    22   -10
d =
     1     0    10     0    25
```

$f_1(x) = 3x^2 - 2x + 4$ 의 미분은 $6x - 2$

$f_1(x)f_2(x) = (3x^2 - 2x + 4)(x^2 + 5)$의 미분은 $12x^3 - 6x^2 + 38x - 10$

$f_1(x)f_2(x) = \frac{3x^2-2x+4}{x^2+5}$ 의 미분은 $\frac{2x^2+22x-10}{x^4+10x^2+25}$

커브 피팅

- 커브 피팅 (curve fitting)
  - 함수를 데이터 집합에 맞추는 과정으로 회귀분석(regression analysis)이라고 도 함
  - 데이터로부터 수학적 모델을 만드는데 사용함
  - 다양한 함수형태(선형, 다항식, 멱함수, 지수함수 등등)를 사용

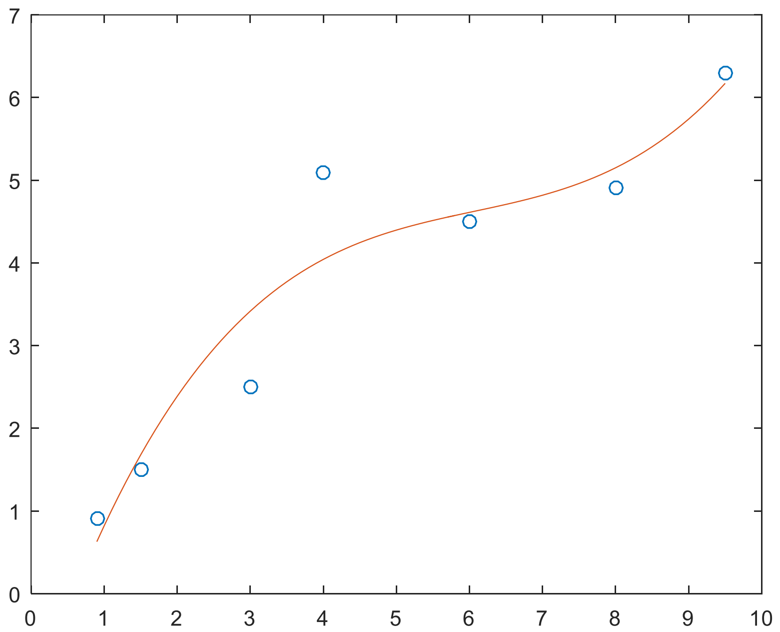
- Matlab에서 지원하는 커브 피팅
  - 다항식 : `polyfit()`
  - 다항식이 아닌 경우 : 멱함수, 지수함수, 로그함수 … → 데이터를 이 함수형태로 가공하여 polyfit() 함수 사용함
  - `polyfit()` 함수 : 데이터 점들에 대한 최상의 함수 계수를 구하는 과정
    - ➢ 최소자승법을 이용하여 오차를 최소화하는 과정으로 계산함

커브 피팅

- 다항식 커브 피팅
  - `p = polyfit(x, y, n);`

    p : 다항식 계수, x & y : 데이터의 좌표, n : 다항식의 차수

```
>> x=[0.9 1.5 3 4 6 8 9.5];
>> y=[0.9 1.5 2.5 5.1 4.5 4.9 6.3];
>> p=polyfit(x,y,3)
p =
    0.0220   -0.4005    2.6138   -
1.4158
>> xp=0.9:0.1:9.5;
>> yp=polyval(p,xp);
>> plot(x,y,'o',xp,yp);
```



- ➢ 데이터들과 가장 근접한 3차 함수의 계수를 계산함
- ➢ $f(x) = 0.022x^3 - 0.4005x^2 + 2.6138x - 1.4158$

커브 피팅

- 차수에 따른 피팅 결과

```
x=[0.9 1.5 3 4 6 8 9.5];
y=[0.9 1.5 2.5 5.1 4.5 4.9 6.3];
for order=1:6
    p=polyfit(x,y,order);
    xp=0.9:0.1:9.5;
    yp=polyval(p,xp);
    figure;
    plot(x,y,'o',xp,yp);
    grid;
    xlabel('x');
    ylabel('y');
    title(sprintf('polyfit : order = %d', order));
end;
```