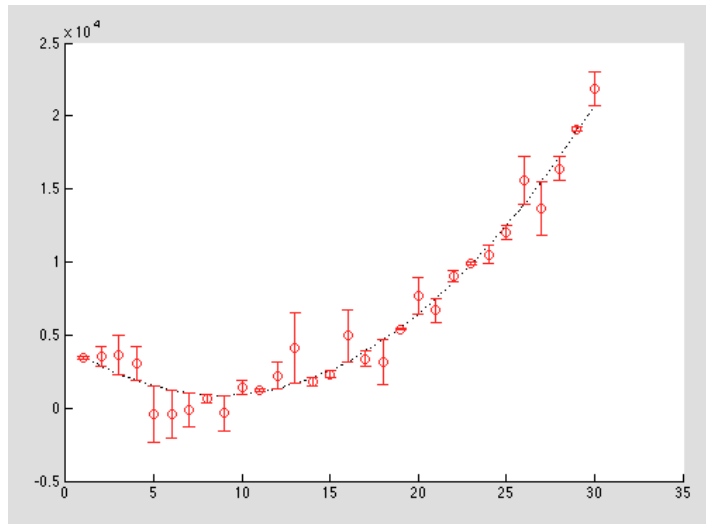


# Exercices programmation – Matlab 3 + Corr

Rappel : Utilisez `doc/help` pour connaître les fonctions manquantes et leurs paramètres !

## Fit et polynôme



Fermez les figures et effacez les variables globales, retenir les affichages.

```
clear all           % efface toutes les variables du workspace
close all          % ferme toutes les fenetres
hold on;           % stack plots
```

Simulez les mesures  $y$  relevées sur un appareil de mesure. Pour cela :

- évaluez le polynôme  $p = x^3 - 6x^2 - 72x - 27$  entre  $x = 1$  et 30,
- ajoutez du bruit d'amplitude égale à la valeur moyenne  $m$  du polynôme  $p$ ,

```
p = [1 -6 -72 -27]; % coef du polynome
x = ...             % abscisse
y1= polyval(p,x); % evalue p en x
m = ...             % calcul de la valeur moyenne de y1
r = ...             % calcul du vecteur bruit
y = y1+r;           % signal simulé y.c. bruit
```

Fittez  $y$  à l'aide d'un polynôme  $p1$  du 2<sup>e</sup> degré

```
p1 = ...
```

Affichez  $p1$  en traitillé

```
plot(...);
```

Calculer le vecteur d'erreur  $e$  entre le polynôme *fitté*  $p1$  et les valeurs simulées  $y$ .

$e = \dots;$

Affichez les mesures avec les erreurs associées :

```
errorbar(x,y,e,'or')

close all
hold on
p = [1 -6 -72 -27];
x =1:30;
y1= polyval(p,x);
r=mean(y1)/2*rand(1,length(x));
y=y1+r;
p1 = polyfit(x,y,2)
plot(x,polyval(p1,x),'-k');
e=polyval(p1,x)-y;
errorbar(x,y,e,'or');
```

## Fonctions anonymes

Écrire la fonction anonyme **Reverse(X)** qui inverse l'ordre les positions des éléments du vecteur X. Le dernier éléments de X devient le premier, l'avant dernier devient le deuxième et le premier élément devient de dernier.

Reverse = @(X) ...

Ex. Reverse ([8, 14, 2, -2, 5]) → [5, -2, 2, 14, 8]

```
Reverse=@(X) X([end:-1:1]);
```

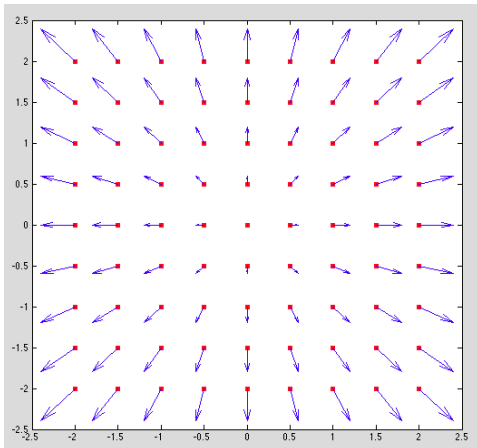
Écrire la fonction anonyme **Pair(X)** qui retourne les valeurs paires du scalaire/vecteur/matrice passé en paramètre, le vecteur de sortie doit être un vecteur ligne avec ses valeurs ordonnées de manière croissantes.

Pair = @(X) ...

Ex. Pair([1:5;6:10]) → 2 4 6 8 10

```
Pair = @(X) sort(X(mod(X,2)==0))'
```

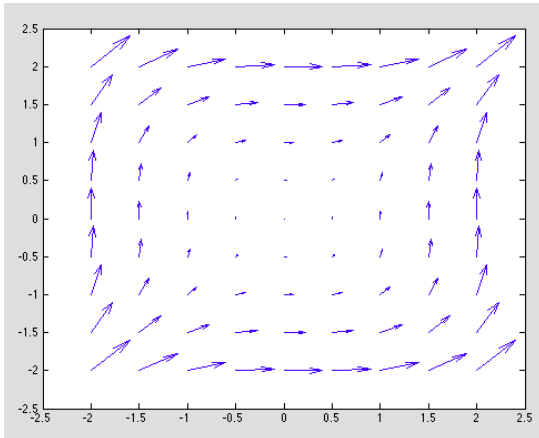
## Vectors field



La fonction `quiver()` permet d'afficher un champ de vecteurs. Ces vecteurs  $(u,v)$  sont définis pour chaque point (carré rouge en  $x,y$ ) d'une grille définie à l'aide de `meshgrid()`.

```
[x,y]=meshgrid(-2:.5:2, -2:.5:2);  
u = x;  
v = y;  
quiver(x,y,u,v)
```

Trouvez  $u$  et  $v$  pour afficher le champ de vecteurs ci-dessous.  
Notez les symétries, ex.  $u(-2) = u(2)$  et  $v(-2) = v(2)$



```
[x,y]=meshgrid(-2:.5:2, -2:.5:2);  
u = ...  
v = ...  
quiver(x,y,u,v)
```

```
[x,y]=meshgrid(-2:.5:2, -2:.5:2);  
u=y.*y;  
v=x.*x;  
plot(x,y,'o')  
quiver(x,y,u,v)
```