

Ce qu'il faut retenir de la dérivation numérique

✿ définir une fonction à partir d'un vecteur d'abscisse

```
x=-1:0.1:1;  
y=x.^3-5;
```

✿ représenter cette fonction à l'aide de plot

```
plot(x,y,'ko-')  
xlabel('axe des x')  
ylabel('axe des y')  
title('y=f(x)')
```

✿ utiliser des schémas simples de dérivation numérique

↳ dérivée première par différence non-centrée

méthode scalaire (non-recommandée !)

```
n=length(x);  
for i=1:n-1  
dync(i)=(y(i+1)-y(i))/(x(i+1)-x(i));  
end
```

méthode vectorielle (recommandée !)

```
dync=(y(2:end)-y(1:end-1))./(x(2:end)-x(1:end-1));
```

↳ dérivée première par différence centrée

méthode scalaire (non-recommandée !)

```
n=length(x);  
for i=2:n-1  
dyc(i)=(y(i+1)-y(i-1))/(x(i+1)-x(i-1));  
end
```

méthode vectorielle (recommandée !)

```
dyc=(y(3:end)-y(1:end-2))./(x(3:end)-x(1:end-2));
```

↳ dérivée seconde par différence centrée

méthode scalaire (non-recommandée !)

```
n=length(x);  
for i=2:n-1  
d2yc(i)=(y(i+1)-2*y(i)+y(i-1))/(x(i+1)-x(i))^2;  
end
```

méthode vectorielle (recommandée !)

```
d2yc=(y(3:end)-2*y(2:end-1)+y(1:end-2))./(x(3:end)-x(2:end-1)).^2;
```

* représenter les dérivées numériques

```
plot(x(2:end),dync) % pour la différence non-centrée à droite  
plot(x(1:end-1),dync) % pour la différence non-centrée à gauche  
plot(x(2:end-1),dyc) % pour la dérivée première par différence centrée  
plot(x(2:end-1),d2yc) % pour la dérivée seconde par différence centrée
```



Conseils pour éviter les erreurs classiques

➔ penser à vider la mémoire (`clear all`) et fermer les fenêtres ouvertes (`close all`) lorsque le comportement de Matlab devient douteux ;

➔ ne pas confondre les opérations vectorielles (*, /, ^) avec les opérations composante par composante (.*, ./, .^):

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \quad A * B = \begin{pmatrix} 3 & 6 & 9 \\ 6 & 12 & 18 \\ 9 & 18 & 27 \end{pmatrix} \quad A .* B = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

➔ dans un plot, s'assurer que les vecteurs d'abscisse et d'ordonnée ont la même longueur : ne pas hésiter utiliser les commandes `length` (longueur d'un vecteur) et `size` (dimensions d'une matrice).

➔ pour éviter ce problème dans le cas présent, on peut également faire artificiellement en sorte que le vecteur contenant la dérivée et le vecteur d'abscisse aient la même longueur :

% exemple pour la dérivée première par différence centrée

```
dyc=(y(3:end)-y(1:end-2))./(x(3:end)-x(1:end-2));
```

```
dyc=[NaN dyc NaN];
```

```
plot(x,dyc)
```