Nous allons ici tracer las solutions approchées d'une équation différentielle du premier ordre du type :
$y' = F(x, y)$, avec la condition initiale : $y_0 = y(x_0)$.
On ne se pose aucun problème quant à l'existence et l'unicité des solutions.

1 Un peu de Maple

1.1 Des listes

Maple peut manipuler des listes comme : `[[X,Y],[X',Y'], ...,[X",Y"]]`.
qu'on peut mettre dans une variable comme `Points`
L'opérateur `op(...)` ote les crochets extérieurs d'une liste,
ainsi : `Points :=[op(Points),[X,Y]]` ajoute un point à la liste de points `Points`...

1.2 Des graphes

`plot(Points)` engendre le graphe formé des segments de droites reliant les points successifs de `Points`.
Celui ci peut au besoin se mettre dans une variable comme `A` ou `B`...
On regardera aussi l'option `color=...` de `plot`.
Les couleurs usuelles sont `black, yellow, red, blue, green...`
`display([A,B])` trace le graphe simultané des 2 (ou 3...) graphes indiqués. Avoir choisi des couleurs permet de différencier les graphes.

2 Méthode d'Euler

2.1 La méthode avec un pas h

On part d'un point M de coordonnées (X_M, Y_M) appartenant à la solution approchée.
Le point suivant est le point E de coordonnées (X_E, Y_E), tel que $X_E = X_M + h$ et tel que le point E est sur la droite passant par M de pente $F(X_M, Y_M)$.

Voir la figure ci-dessous.

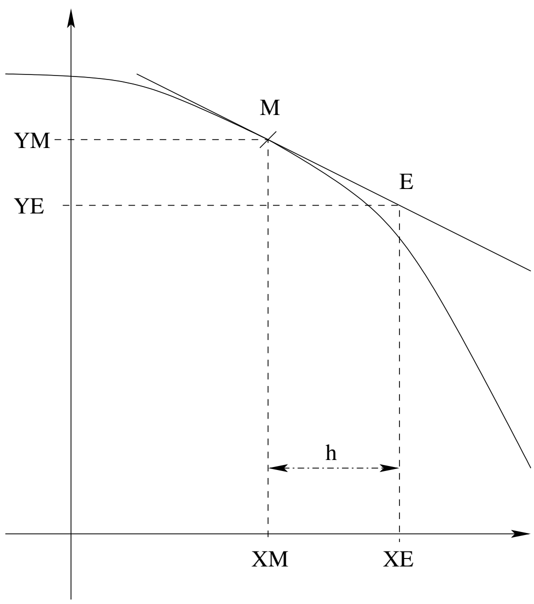


Fig. 1 – Méthode d'Euler

On itère le procédé en partant du point (X_0, Y_0) et en reprenant à chaque étape le point E comme nouveau point M, autant de fois qu'il le faut pour que X décrive l'intervalle demandé.

2.2 Questions

1) Calculer les coordonnées de E en fonction de celles de M, de F et de h.

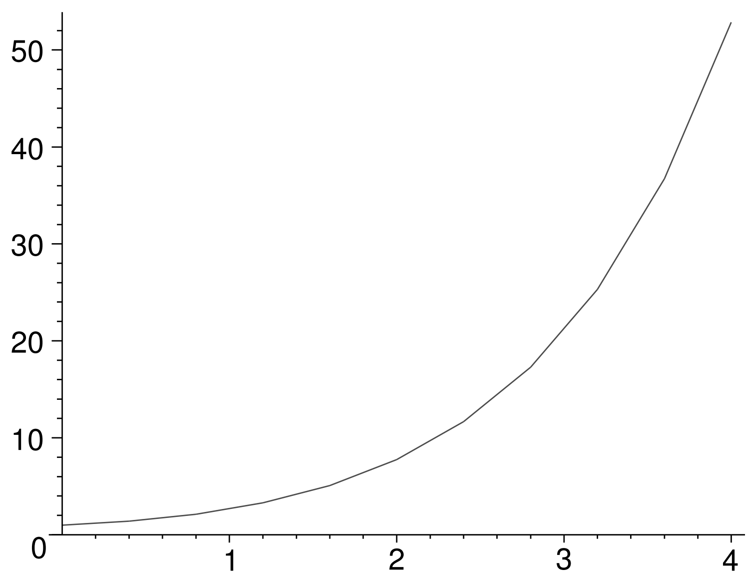
Solution : On a dans l'énoncé : $X_E = X_M + h$.
Par ailleurs : $Y_E = Y_M + h\,F(X_M, Y_M)$ puisque $F(X_M, Y_M)$ est la pente de la droite (ME).

2) Ecrire une procédure `Euler` à 5 paramètres : `F, X0, Y0, Xfin et n`,
qui calcule la liste des $n+1$ points de la solution approchée pour les paramètres donnés.

```
Euler := proc(F, X0, Y0, Xfin, n)
local XM, YM, XE, YE, h, Points;
  XM := evalf(X0);
  YM := evalf(Y0);
  h := evalf((Xfin - X0)/n);
  Points := [[XM, YM]];
  to n do
    YE := YM + h * F(XM, YM);
    XE := XM + h;
    Points := [op(Points), [XE, YE]];
    XM := XE;
    YM := YE
  end do;
  plot(Points)
end proc
```

3) Tracer la solution approchée d'Euler pour l'équation différentielle $y' = x + y$ avec $(x_0, y_0) = (0, 1)$ sur $[0, 4]$ avec $n = 10$.

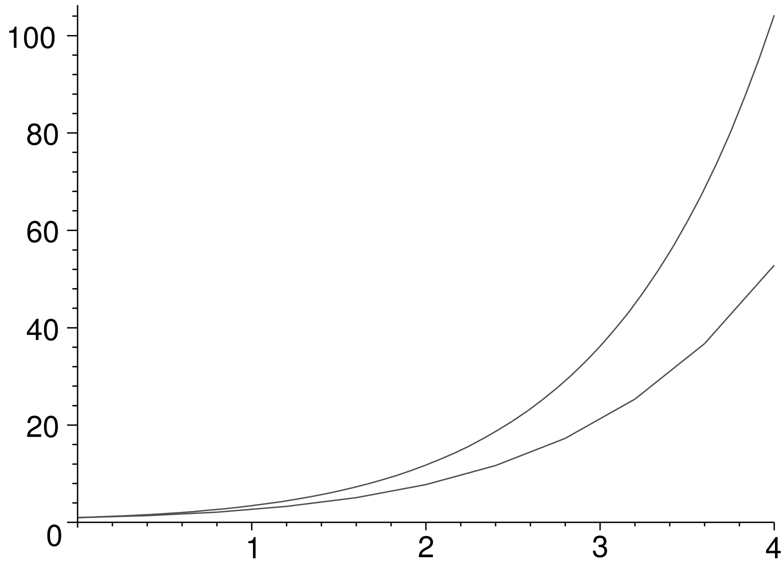
```
> Euler((u,v)->u+v,0,1,4,10);
```



4) Tracer sur un même graphe la solution exacte et la solution approchée d'Euler.

Solution : Notons d'abord qu'on obtient facilement la solution exacte de cette équation différentielle linéaire du premier ordre : $y = 2\,e^x - x - 1$.

```
> Sol:=plot(2*exp(x)-x-1,x=0..4):
> E:=Euler((u,v)->u+v,0,1,4,10):
> display([Sol,E]);
```



3 Méthode de Runge-Kutta d'ordre 2

3.1 La méthode avec un pas de h

En partant du point M appartenant à la solution approchée de la méthode de Runge-Kutta d'ordre 2, on construit d'abord le point d'Euler E.
On considère le point RK_1, milieu du segment (ME), de coordonnées (X_{RK_1}, Y_{RK_1}).
Le point suivant de la solution approchée est le point RK_2, de coordonnées (X_{RK_2}, Y_{RK_2}),
avec : $X_{RK_2} = X_E = X_M + h$, et qui, de plus, appartient à la droite passant par M et de pente $F(X_{RK_1}, Y_{RK_1})$

Voir la figure ci-dessous.

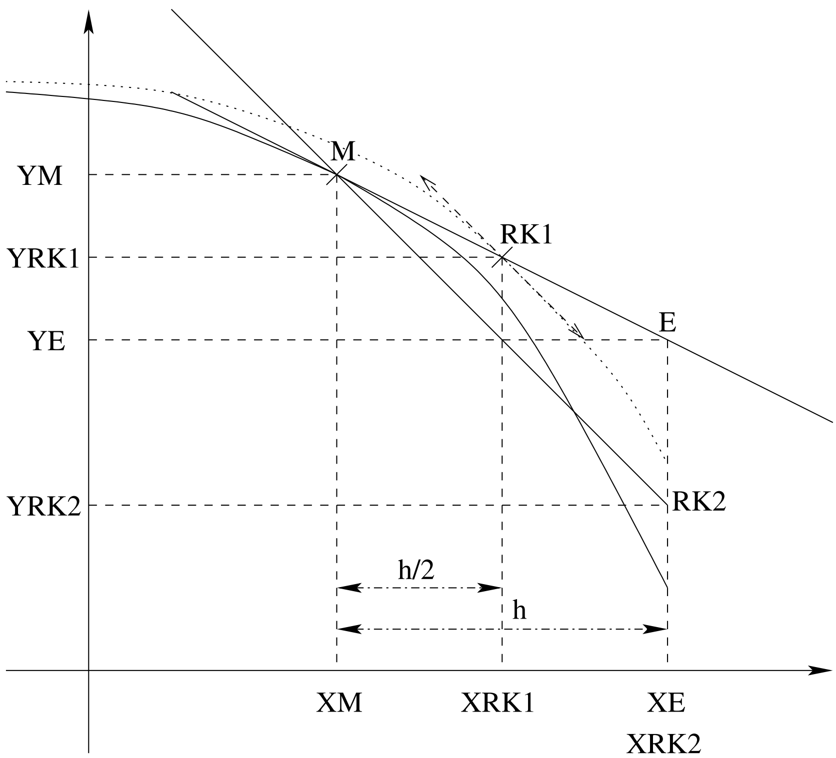


FIG. 2 – Méthode de Runge-Kutta d'ordre 2

On itère le procédé en partant du point (X_0, Y_0) et en reprenant à chaque étape le point E comme nouveau point M, autant de fois qu'il le faut pour que X décrive l'intervalle demandé.

3.2 Questions

1) Calculer les coordonnées de RK_2 en fonction de celles de M, de F et de h.

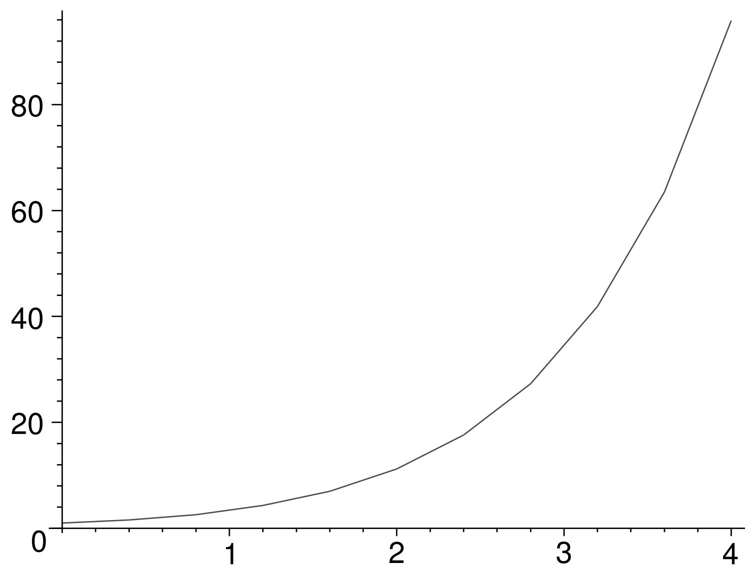
Solution : On a dans facilement : $X_{RK_1} = X_M + \dfrac{h}{2}$.
Par ailleurs : $Y_{RK_1} = Y_M + \dfrac{h}{2} F(X_M, Y_M)$ puisque $F(X_M, Y_M)$ est la pente de la droite (ME).
Bien sûr, $X_{RK_2} = X_M + h$.
Enfin : $Y_{RK_2} = Y_M + h\,F(X_{RK_1}, Y_{RK_1})$ puisque la pente de la droite $(M\,RK_2)$ est $F(X_{RK_1}, Y_{RK_1})$.

2) Ecrire une procédure `RK2` à 5 paramètres : `F, X0, Y0, Xfin et n`,
qui calcule la liste des $n+1$ points de la solution approchée pour les paramètres donnés.

```
RK2 := proc(F, X0, Y0, Xfin, n)
local XM, YM, XRK1, YRK1, XRK2, YRK2, h, Points;
  XM := evalf(X0);
  YM := evalf(Y0);
  h := evalf((Xfin - X0)/n);
  Points := [[XM, YM]];
  to n do
    YRK1 := YM + 1/2 * h * F(XM, YM);
    XRK1 := XM + 1/2 * h;
    YRK2 := YM + h * F(XRK1, YRK1);
    XRK2 := XM + h;
    Points := [op(Points), [XRK2, YRK2]];
    XM := XRK2;
    YM := YRK2
  end do;
  plot(Points)
end proc
```

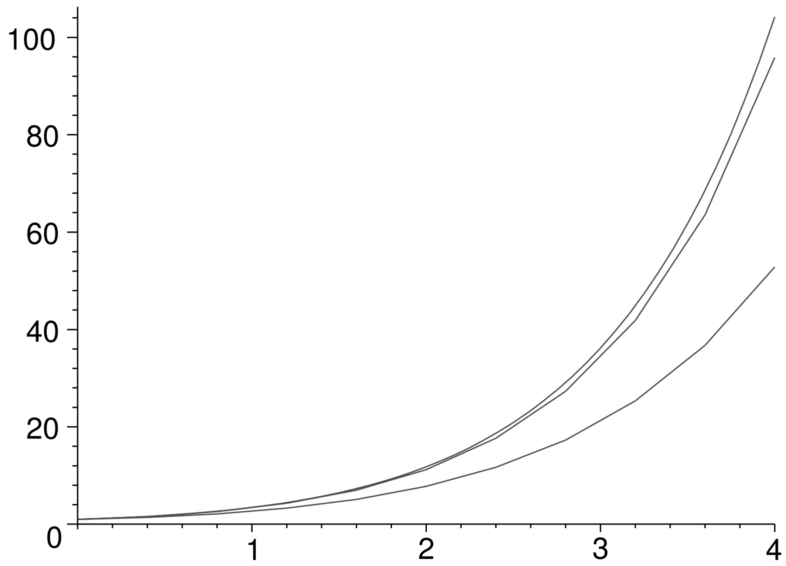
3) Tracer la solution approchée de Runge-Kutta d'ordre 2 pour l'équation différentielle $y' = x + y$ avec $(x_0, y_0) = (0, 1)$ sur $[0, 4]$ avec $n = 10$.

```
> RK2((u,v)->u+v,0,1,4,10);
```



4) Tracer sur un même graphe la solution exacte, la solution approchée d'Euler et la solution approchée de Runge-Kutta d'ordre 2.

```
> Rk2:=RK2((u,v)->u+v,0,1,4,10):
> display([Sol,E,Rk2]);
```



4 Méthode de Runge-Kutta d'ordre 4

4.1 La méthode avec un pas de h

A partir du point M, on construit les point $E,\ RK_1,\ RK_2$.
RK_3 est le milieu du segment (M, RK_2).
Le point suivant de la solution approchée est le point RK_4, de coordonnées (X_{RK_4}, Y_{RK_4}),
avec : $X_{RK_2} = X_E = X_M + h$, et qui, de plus, appartient à la droite passant par M et de pente :

$$\frac{1}{6}\Big(F(X_M, Y_M) + 2\,F\,(X_{RK_1}, Y_{RK_1}) + 2\,F\,(X_{RK_3}, Y_{RK_3}) + F\,(X_{RK_2}, Y_{RK_2})\Big)$$

Voir la figure page suivante.
On itère le procédé en partant du point (X_0, Y_0) et en reprenant à chaque étape le point E comme nouveau point M, autant de fois qu'il le faut pour que X décrive l'intervalle demandé.

4.2 Questions

1) Calculer les coordonnées de RK_4 en fonction de celles de M, de F et de h.

Solution : On a dans facilement : $X_{RK_3} = X_M + \dfrac{h}{2}$.
Par ailleurs : $Y_{RK_3} = Y_M + \dfrac{h}{2}\,F(X_{RK_1}, Y_{RK_1})$ puisque $F(X_{RK_1}, Y_{RK_1})$ est la pente de la droite $(M\,RK_2)$.
Bien sûr, $X_{RK_4} = X_M + h$.
Enfin : $Y_{RK_4} = Y_M + \dfrac{1}{6}\Big(F(X_M, Y_M) + 2\,F\,(X_{RK_1}, Y_{RK_1}) + 2\,F\,(X_{RK_3}, Y_{RK_3}) + F\,(X_{RK_2}, Y_{RK_2})\Big)$
en utilisant la pente de la droite $(M\,RK_4)$.

2) Ecrire une procédure RK4 à 5 paramètres : `F, X0, Y0, Xfin et n`,

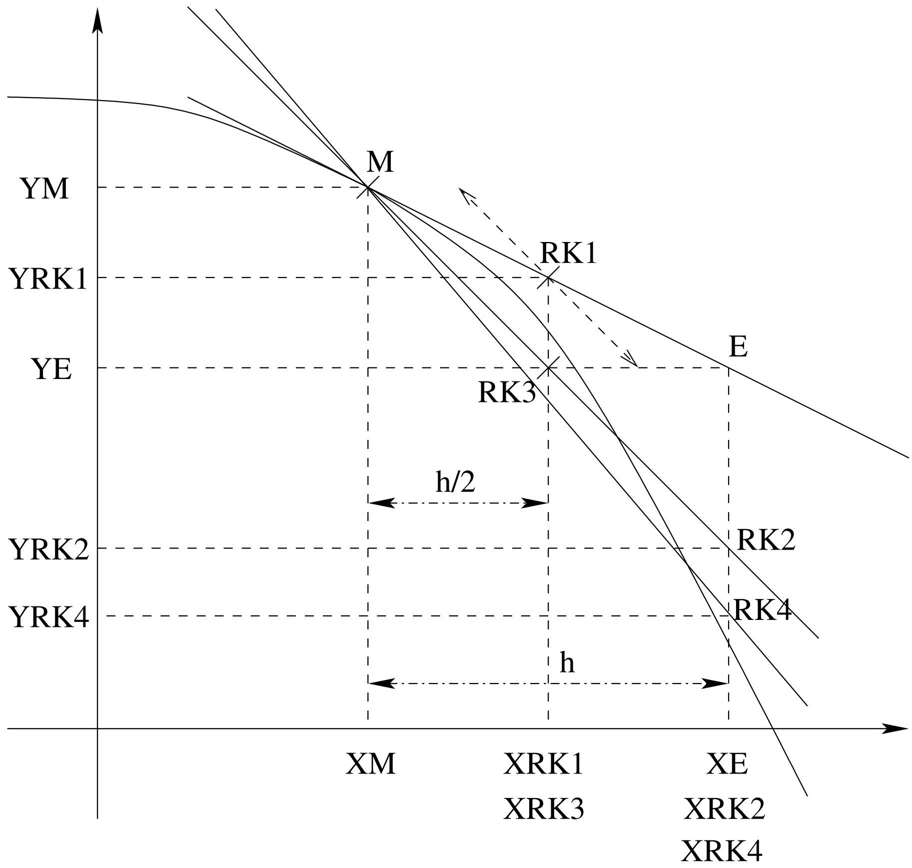


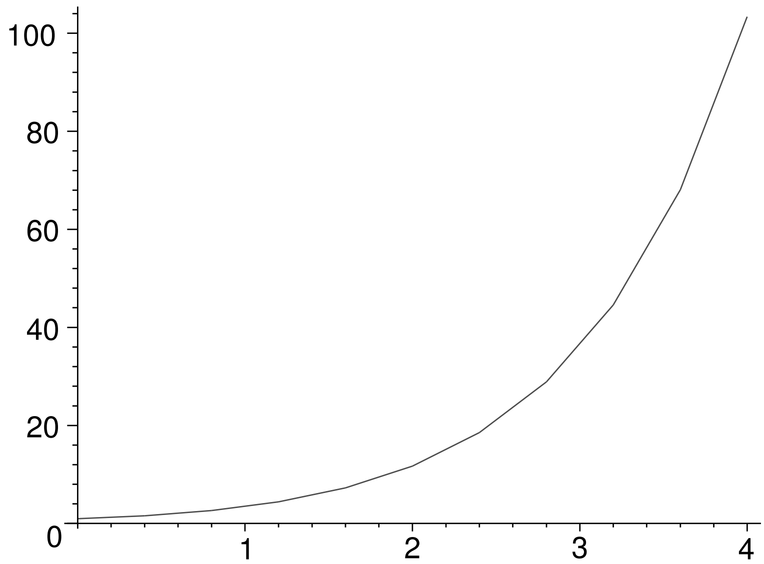
FIG. 3 – Méthode de Runge-Kutta d'ordre 4

qui calcule la liste des $n+1$ points de la solution approchée pour les paramètres donnés.

```
RK4 := proc(F, X0, Y0, Xfin, n)
local XM, YM, XRK1, YRK1, XRK2, YRK2, XRK3, YRK3, XRK4, YRK4, h, Points;
   XM := evalf(X0);
   YM := evalf(Y0);
   h := evalf((Xfin − X0)/n);
   Points := [[XM, YM]];
   to n do
      YRK1 := YM + 1/2 * h * F(XM, YM);
      XRK1 := XM + 1/2 * h;
      YRK2 := YM + h * F(XRK1, YRK1);
      XRK2 := XM + h;
      YRK3 := YM + 1/2 * h * F(XRK1, YRK1);
      XRK3 := XM + 1/2 * h;
      YRK4 := YM+
         1/6 * h * (F(XM, YM) + 2 * F(XRK1, YRK1) + 2 * F(XRK3, YRK3) + F(XRK2, YRK2))
         ;
      XRK4 := XM + h;
      Points := [op(Points), [XRK4, YRK4]];
      XM := XRK4;
      YM := YRK4
   end do;
   plot(Points)
end proc
```

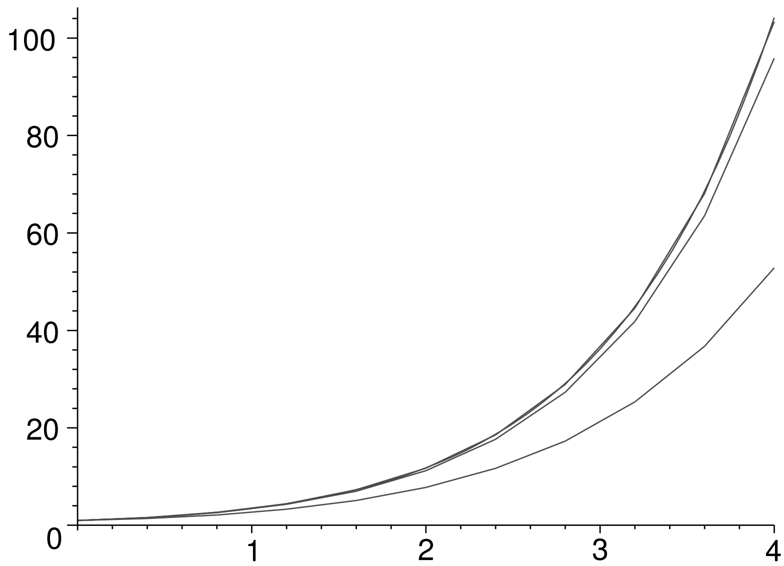
3) Tracer la solution approchée de Runge-Kutta d'ordre 4 pour l'équation différentielle $y' = x + y$ avec $(x_0, y_0) = (0, 1)$ sur $[0, 4]$ avec $n = 10$.

```
> RK4((u,v)->u+v,0,1,4,10);
```



4) Tracer sur un même graphe la solution exacte, la solution approchée d'Euler, la solution approchée de Runge-Kutta d'ordre 2 et la solution approchée de Runge-Kutta d'ordre 4.

```
> Rk4:=RK4((u,v)->u+v,0,1,4,10):
> with(plots):
> display([Sol,E,Rk2,Rk4]);
```



Les courbes sont de la plus basse à la plus haute, la solution d'Euler, de Runge-Kutta d'ordre 2, et, presque confondues, la solution de Runge-Kutta d'ordre 4 et la solution exacte.