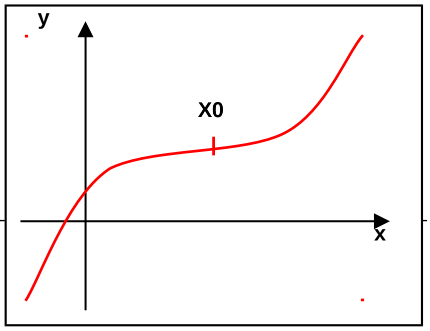
Développement limité

Au voisinage d'un point x0 :

$$f(x)=f(x0)+\frac{x-x0}{1!}f'(x0)+\frac{(x-x0)^2}{2!}f''(x0)+\frac{(x-x0)^3}{3!}f'''(x0)+\ldots+\frac{(x-x0)^n}{n!}f^n(x0)+0(x^{n+1})$$

Si la fonction est convergente au point x0, le terme suivant est toujours plus petit que le terme précédent. Le terme suivant ajoute donc de la précision par rapport au calcul déjà effectué, jusqu'à que cette précision soit considérée comme négligeable. En effet, en informatique, le nombre réel de type double possède 15 chiffres significatifs. Il n'est donc pas indispensable d'aller au delà d'une telle précision.



En prenant x0 = 0: si la fonction est définie pour ce point là.

$$f(x)=f(0)+\frac{x}{1!}f'(0)+\frac{x^2}{2!}f''(0)+\frac{x^3}{3!}f'''(0)+\ldots+\frac{x^n}{n!}f^n(0)+0(x^{n+1})$$

Recherche de la fonction exponentielle : f(x) = e^x

La fonction existe pour le point x0 = 0, donc avec f(x) = e^x , nous avons :

$$\begin{cases} f(x)=e^x \\ f'(x)=e^x \\ f''(x)=e^x \end{cases} \Rightarrow \begin{cases} f(0)=1 \\ f'(0)=1 \\ f''(0)=1 \end{cases}$$ donc :

Formule explicite :

$$\boxed{e^x=1+\frac{x}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}+\ldots+\frac{x^n}{n!}+0(x^{n+1})}$$

recherchons → e^1 = 1 + 1 + 0.5 + 0.16667 + 0.041 + …..

Au fur et à mesure de la progression, le terme tend en effet vers 0, et entre temps, la précision s'affine.

soit aussi $$\boxed{e^x=1+\sum_{i=1}^{n}\frac{x^i}{i!}}$$

Formule de récurrence :

$$\begin{cases} t_1=? \\ t_n=q.t_{n-1} \end{cases}$$ avec $$t_i=\frac{x^i}{i!}$$ → $$\begin{cases} i=1 \rightarrow t_i=x \\ i=n \rightarrow t_n=\frac{x^n}{n!} \\ i=n-1 \rightarrow t_{n-1}=\frac{x^{n-1}}{(n-1)!} \end{cases}$$ → $$q=\frac{t_n}{t_{n-1}}=\frac{x^n.(n-1)!}{x^{n-1}.n!}=\frac{x}{n}$$ soit

$$\boxed{\begin{cases} t_1=x \\ t_n=\frac{x}{n}.t_{n-1} \end{cases}}$$

Algorithme de la fonction exponentielle

Fonction exp(**d** x : réel) : réel
Déclaration :
  variable résultat, terme, n, résultatAvant : réel
Début
  terme ← x
  résultat ← terme+1
  résultatAvant ← résultat-1
  n ← 2
  Tant que résultat ≠ résultatAvant **Faire**
    résultatAvant ← résultat
    terme ← terme . x / n
    résultat ← résultat + terme
    n ← n+1
  Fin faire
  exp() ← résultat
Fin

Implémentation de la fonction exponentielle

```
//--------------------------------------------------------
double exp(double x)
{
  double terme = x, resultat = terme+1.0;
  double resultatAvant = resultat-1.0, n = 2.0;
  while (resultat!=resultatAvant)
  {
    resultatAvant = resultat;
    terme *= x/n;
    resultat += terme;
    n++;
  }
  return resultat;
}
//--------------------------------------------------------
double exp(double x)
{
  double terme = x, resultat = terme+1.0;
  double resultatAvant = resultat-1.0, n = 2.0;
  while (resultat!=resultatAvant)
  {
    resultatAvant = resultat;
    resultat += terme *= x/n++;
  }
  return resultat;
}
//--------------------------------------------------------
```

Recherche de la fonction sin : f(x) = sin x

La fonction existe pour le point x0 = 0, donc avec f(x) = sin x , nous avons :

$$\begin{cases} f(x)=\sin x \\ f'(x)=\cos x \\ f''(x)=-\sin x \\ f'''(x)=-\cos x \end{cases} \Rightarrow \begin{cases} f(0)=0 \\ f'(0)=1 \\ f''(0)=0 \\ f'''(0)=-1 \end{cases}$$

donc :

Formule explicite :

$$\boxed{\sin x = x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\ldots\ldots+(-1)^p.\frac{x^{2p+1}}{(2p+1)!}+0(x^{2p+2})}$$

soit aussi

$$\boxed{\sin x=\sum_{i=0}^{p}(-1)^i\frac{x^{2i+1}}{(2i+1)!}}$$

Formule de récurrence :

$$\begin{cases} t_0=? \\ t_n=q.t_{n-1} \end{cases} \text{ avec } t_i=(-1)^i\frac{x^{2i+1}}{(2i+1)!} \rightarrow \begin{cases} i=0\rightarrow t_0=x \\ i=n\rightarrow t_n=(-1)^n\frac{x^{2n+1}}{(2n+1)!} \\ i=n-1\rightarrow t_{n-1}=(-1)^{n-1}\frac{x^{2n-1}}{(2n-1)!} \end{cases}$$

$$q=\frac{t_n}{t_{n-1}}=\frac{(-1)^n x^{2n+1}.(2n-1)!}{(-1)^{n-1}.x^{2n-1}.(2n+1)!}=\frac{(-1)x^2}{2n(2n+1)}$$

soit

$$\boxed{\begin{cases} t_0=x \\ t_n=\frac{-x^2}{2n(2n+1)}.t_{n-1} \end{cases}}$$

Algorithme de la fonction sinus

Fonction sin(**d** x : réel) : réel
Déclaration :
  variable résultat, terme, n, résultatAvant : réel
Début
  terme ← x
  résultat ← terme
  résultatAvant ← résultat-1
  n ← 1
  Tant que résultat ≠ résultatAvant **Faire**
    résultatAvant ← résultat
    terme ← terme . –x² / 2n / (2n+1)
    résultat ← résultat + terme
    n ← n+1
  Fin faire
  sin() ← résultat
Fin

Implémentation de la fonction sinus

```
//-----------------------------------------------------
double sin(double x)
{
  double terme = x, resultat = terme;
  double resultatAvant = resultat-1.0, n = 1.0;
  while (resultat!=resultatAvant)
  {
    resultatAvant = resultat;
    resultat += terme *= -x*x / (2.0*n+1.0) / (2.0*i);
    n++;
  }
  return resultat;
}
//-----------------------------------------------------
```