Informatique

Polynômes interpolateurs de Lagrange

Méthode des différences divisées

I. Méthode des différences divisées

Les notations suivantes sont celles du problème de l'École de l'Air 93 M - P maths 2.
f est une fonction définie sur [a,b], x_0, ..., x_n sont $n+1$ points de $[a, b]$.
Pour tout $k \in [0, n]$, on note p_k le polynôme d'interpolation de f aux points x_0, ..., x_k et $f[x_0, ..., x_k]$ le coefficient de son terme de degré k.
On montre que pour tout $x \in [a, b]$,

$$p_n(x) = f(x_0) + \sum_{k=1}^{n} f[x_0, ..., x_k](x - x_0)...(x - x_{k-1}), \quad (1)$$

et que si de plus, $p_{k-1}{}^*$ est le polynôme d'interpolation de f aux points x_1, ..., x_k, les p_k se conforment à la relation de récurrence :

$$p_k(x) = \frac{1}{x_k - x_0} ((x - x_0)\, p_{k-1}{}^*(x) - (x - x_k)\, p_{k-1}(x)). \quad (2)$$

Cela donne enfin pour les $f[x_0, ..., x_k]$:

$$f[x_0, ..., x_k] = \frac{1}{x_k - x_0} (f[x_1, ..., x_k] - f[x_0, ..., x_{k-1}]). \quad (3)$$

La relation (2) donne son nom à la méthode.
On remarque enfin que pour tout $i \in [0, n]$, $f[x_i] = f(x_i)$, et on se propose de calculer avec (1) et (2) le polynôme de Lagrange, plus rapidement qu'avec la définition usuelle (polynômes élémentaires).

II. Programmation

Les données sont la fonction f et les points d'interpolation, qui sont contenus dans une liste *points* = $[x_0, ..., x_n]$.
On propose de calculer selon la méthode suivante :

- On stocke dans un premier temps les $f[x_i] = f(x_i)$ dans une liste **`coefs`** ;
- Dans la même liste, on calcule $f[x_0,x_1]$, $f[x_1,x_2]$, ..., $f[x_{n-1},x_n]$ qui remplacent $f[x_1]$, ..., $f[x_n]$ respectivement ($f[x_0]$ est inchangé) ;
- Dans une deuxième passe, on remplace $f[x_1,x_2]$, ..., $f[x_{n-1},x_n]$ par $f[x_0,x_1,x_2]$, ..., $f[x_{n-2},x_{n-1},x_n]$ ($f[x_0,x_1]$ ne bougeant pas) ;

- Ainsi de suite... La dernière étape consiste en le remplacement de $f[x_1, \ldots, x_n]$ par $f[x_0, \ldots, x_n]$.

À la fin de ce processus, la liste **coefs** contient $[f[x_0], f[x_0,x_1], \ldots, f[x_0, \ldots, x_{n-1}], f[x_0, \ldots, x_n]]$.
Dans le même temps, on forme progressivement les produits partiels $(x-x_0)\ldots(x-x_{k-1})$ et le polynôme $p_k(x)$ selon (1).

Écrire une procédure **Lagrange(x,f,points)** retournant le polynôme interpolateur (évalué en *x*) de la fonction f en les points contenus dans la liste donnée comme dernier argument.

```
> Lagrange := proc(x, f, points)
```

III. Tests

On prend pour f la fonction exponentielle :

```
> f := exp :
```

Former le polynôme interpolateur de f en 0, 2, 4.
On pourra utiliser **collect(...,x)** (aide) pour un regroupement des termes en fonction de *x*.

Tracer sur un même schéma les courbes représentatives de f et de son polynôme interpolateur.

IV. Phénomène de Runge

On note ainsi la non-approximation uniforme d'une fonction (même très régulière) par ses polynômes interpolateurs "à pas constant".

Nous disons que ce phénomène est bien observable sur la fonction
$f : x \to \frac{1}{(x^2+1)}$. (On se placera sur l'intervalle $[-5, 5]$.)

```
> f := x ->
```

Écrire une procédure **subdiv** prenant pour argument l'entier **n** et retournant la liste des $n+1$ points de la subdivision de $[-5, 5]$ de pas constant $(\frac{10}{n})$.

En réutilisant la procédure **Lagrange** du **II.**, en déduire une procédure **poly** prenant comme argument l'entier **n** et retournant le polynôme interpolateur de *f* en les points de la subdivision précédente.

À l'aide d'une instruction **plot** appropriée, tracer sur un même schéma les courbes représentatives de *f* et de ses polynômes interpolateurs pour un échantillon de valeurs de *n*. Que penser de la qualité de l'approximation de *f* par ses polynômes interpolateurs ?