Evaluation des fonctions

Les trois méthodes:

Méthode 1(M1) : q(a/(2^k)) = c0*(a/(2^k))^{0} + c1*(a/(2^k))^{1} +...+ cn*(a/(2^k))^n, où n est le degré du polynôme.

Méthode 2(M2) : q(a/(2^k)) = (c0*a^{0}*2^(k*n) + c1*a^{1}*2^(k*(n-1)) +...+ cn*a^n*2^(k*(n-n)))/2^(n*k)

Horner (M3)

La complexité

Dans la suite, les fonctions de la bibliothèque GMP comme mpz_mul_ui() , mpz_mul_2exp() etc.. seront comptés comme une opération.

Soit n le degré des polynômes, et C(n) le nombre d'opération pour un degré n.

Pour M1 : C(n+1)=C(n)+3 . On a 13+3*n opérations.

Pour M2 : C(n+1)=C(n)+4. On a 20+4*n opérations.

Pour M3 : C(n+1)=C(n)+3. On a 14+3*n opérations.

La complexité pour les trois méthodes semble être linéaire, O(n).

Evaluation des fonctions

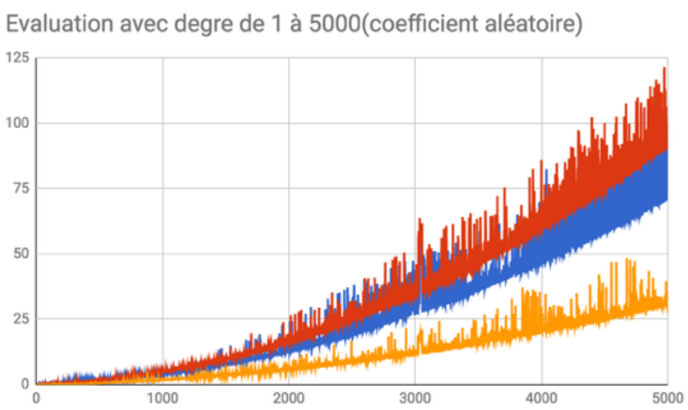
On a effectué l'évaluation avec des coefficients aléatoire qui sont crées par la fonction **mpz_urandomb** (mpz_t rop, gmp_randstate_t state, mp_bitcnt_t n), qui génère les coefficients de 0 à 2^(n-1), et avec un if(rand()%==0) pour obtenir des valeurs au négatives.

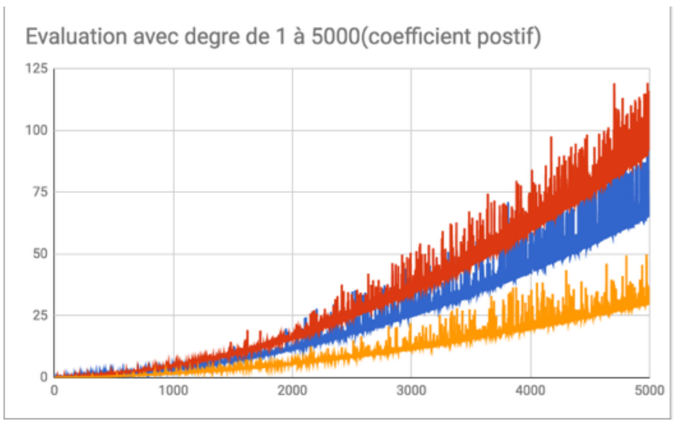
On a choisi: degré k=5000, a=12345 et k=5.

On a évalué avec des coefficients aléatoires(positif et négatif) et avec des coefficients positifs.

Les trois méthodes sont représentées par couleur : M1 M2 M3. x=degré et y= temps

(Le fichier E1_temps contient le temps d'exécution pour chaque incrémentation de n, pour pouvoir insérer les résultats dans l'excel, ils sont en seconde et ils sont multiplié par 10000.)





Les dénominateurs et les numérateurs obtenu sont identiques pour les trois méthodes.

Le fait qu'on a des coefficients négatifs n'a pas d'impact sur le temps d'exécution.

A partir de ces résultats on peut en déduire qu'en termes d'efficacité on a : M3>M1>M2.

Les courbes ci-dessus ne sont pas exactement linéaire mais plutôt polynomiale.