Fundamentals

Evaluation of Polynomial

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1$$

when you see this polynomial , how do you evaluate its arithmetic?

for example: suppose x=1/2 , we may manipulate the polynomial arithmetic in a normal way:

$$P(\frac{1}{2}) = 2 * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} + 3 * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} - 3 * \frac{1}{2} * \frac{1}{2} + 5 * \frac{1}{2} - 1$$

but how can we reduce the multiplication and additions?

another way is to transform the polynomial into a nested one:

$$P(\frac{1}{2}) = -1 + \frac{1}{2} * (5 - \frac{1}{2} * (-3 + \frac{1}{2} * (3 + 2 * \frac{1}{2})))$$

as you can see , we have reduced the polynomial to 4 multiplication with the same 4 additions.

so there is the more general form:

$$\begin{aligned} P(x) &= c_1 + c_2x + c_3x^2 + c_4x^3 + c_5x^5 \\ &= c_1 + x(c_2 + x(c_3 + x(c_4 + x(c_5)))) \end{aligned}$$

also in the interpolation calculations , the formula has a more general form:

$$P(x) = c_1 + (x - r_1)(c_2 + (x - r_2)(c_3 + (x - r_3)(c_4 + (x - r_4)(c_5))))$$

the r1 r2 r3 r4 represent the base point

from the formula above we can see that the count of multiplication is only determined by degree of polynomial.

Here is the c++ code implementing the evaluation of nested polynomial

```
//the code of monomial
double NA::monomial(int degree, double coeff,double x, double b) {
    std::cout<<degree<<" "<<coeff<<" "<<x<<" "<<b<<std::endl;
    double begin=x;
    for (int i = 1; i <degree ; ++i) {
        x*=begin;
    }
    return coeff*x+b;
}

//the code of nest polynomial
double NA::NestPolynomial(int degree, double coeff[], double x, double
basePoint[]) {
    double result=coeff[degree];
    for(int i=degree; i>0;--i){
        std::cout<<result<<" ";
        result=monomial(1,result,(x-basePoint[i-1]),coeff[i-1]);
```

```
    }
    return result;
}
```