

自适应辛普森积分

Adaptive Simpson Method

Simpson 积分：用二次函数去近似，取积分。

对于二次函数 $F(x) = ax^2 + bx + c$ ，有：

$$\begin{aligned}\int_l^r F(x)dx &= \left(\frac{a}{3}x^3 + \frac{b}{2}x^2 + cx \right) \Big|_l^r \\ &= \frac{a}{3}(r^3 - l^3) + \frac{b}{2}(r^2 - l^2) + c(r - l) \\ &= \frac{r-l}{6} [2a(r^2 + rl + l^2) + 3b(r + l) + 6c] \\ &= \frac{r-l}{6} \left[f(l) + f(r) + 4f\left(\frac{l+r}{2}\right) \right]\end{aligned}$$

所以给出待求函数上三个点 $(l, f(l)), \left(\frac{l+r}{2}, f\left(\frac{l+r}{2}\right)\right), (r, f(r))$ ，就可以用过这三个点的抛物线来近似这一段函数，用上述公式近似 $[l, r]$ 这一段的积分。

自适应：每次把积分区间分成两半，如果左积分加上右积分与总积分相差不大，则认为这一段上积分值近似地不错，不需要再分割；否则继续分割。

Code：

```
1 double simpson(double l, double r){
2     double mid = (l + r) / 2;
3     return (f(l) + 4 * f(mid) + f(r)) * (r - l) / 6;
4 }
5 double solve(double l, double r, double _eps, double I){
6     double mid = (l + r) / 2;
7     double Il = simpson(l, mid), Ir = simpson(mid, r);
8     if(fabs(Il + Ir - I) <= 15 * _eps) return Il + Ir + (Il + Ir - I) / 15;
9     return solve(l, mid, _eps / 2, Il) + solve(mid, r, _eps / 2, Ir);
10 }
11
12 int main(){
13     //...
14     printf("%.6f\n", solve(l, r, 1e-7, simpson(l, r)));
15     return 0;
16 }
```