Homework 1

杨中义 物理学系 20307110314

2023.9.16

1. Problem 1 Finding solution

1.1. Problem Description

采用Fortran90编写程序，在0到200内找方程的整数解:$a^5+b^5+c^5+d^5=e^5$。

1.2. Code Description

题目要求求解一个整数的五元方程，由于方程是高次的，并没有什么简单直接的解法，只能对 $[0,200]$ 内的整数进行遍历求解。由于该方程在形式上具有(a,b,c,d)的轮换对称性，因此对于一组解，交换他们的顺序依然构成一组解。我们对这种因为交换而带来的解的简并性并不感兴趣，因此在代码中我们假定了$a\leqslant b\leqslant c\leqslant d$。这样也可以提高搜索的效率。

本程序使用 Fortran90 编写，源代码为/finding_solution/finding_solution.f90

1.3. Pseudo Code

Algorithm 1 Finding integer solutions of the given equation

Input: /

Output: All solutions that satisfy the equation

1: integer a, b, c, d, e

2: do $a\in[0,200], b\in[a,200], c\in[b,200], d\in[c,200], e\in[1,200]$: $\Rightarrow$ traversing integers in [0,200]

3: if $a^5+b^5+c^5+d^5=e^5$:

4: print(a,b,c,d,e) $\Rightarrow$ print out one of the the expected solution

5: end if

6: end do

1.4. Testing Case

经过遍历，我们只找到一组符合要求的非平凡解如下图所示

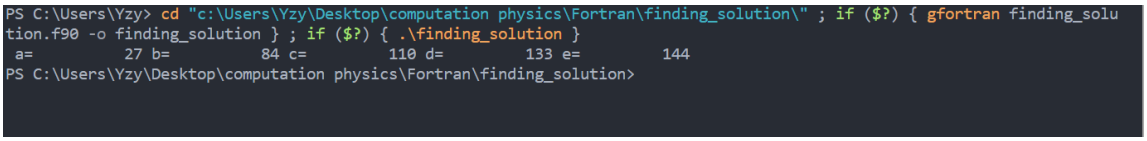

Figure 1: The only solution found from the given equation

2. Problem 2 Game:24point

2.1. Problem Description

24点游戏是儿时玩的主要益智类游戏之一，玩法为：从一副扑克中抽取4张牌，对4张牌使用加减乘除中的任何方法，使计算结果为24。例如，1,5,5,5，通过 (5 - (1 / 5)) * 5 = 24，最快算出24者胜。请采用 Fortran90 编程求解24点游戏的解。

2.2. Code Description

本程序要求输入四个数字，并判断是否存在一种解法能够使用+, —, *, / 四个运算符号，以及一定的运算顺序计算出24点。若有，则输出其中一个可能的解。

由于括号可以改变运算的顺序，算符自身在 Fortran90 中有优先级差异，本程序的任意两个数之间的运算都用括号括住，从而完全控制表达式的运算顺序。此时只需要对于三个运算符的顺序进行排列。一共有 6 种情况，不重复的有 5 种，分别是：

$$\begin{array}{l}(((a\cdot b)\cdot c)\cdot d)\\((a\cdot(b\cdot c))\cdot d)\\(a\cdot((b\cdot c)\cdot d))\\(a\cdot(b\cdot(c\cdot d)))\\((a\cdot b)\cdot(c\cdot d))\end{array}\tag{1}$$

其中字母表示的是整数，"·"表示运算符号。

本程序用$situationA()$, $situationB()$, $situationC()$, $situationD()$, $situationE()$分别表示这5种情况对应的函数，此外定义 calculate() 函数进行两个数之间的运算；在给定运算顺序的情况下定义find_solution()函数进行 4 个输入数字的全排列、3个运算符的遍历。

本程序使用 Fortran90 编写，源代码为:/24_point/24_point.f90。

2.3. Pseudo Code

Algorithm 2 Game:24 point

Input: 4 integers

Output: one(or none) of the solution for the game

```
found = find_solution(4 numbers)
if found ← 1 then
    at least one solution
else if found ← 0 then
    print: no solution
end

function find_solution (a, b, c, d)                    ⇒ sort 4 numbers and choose 3 operators
for every sort of 4 numbers do
        if situationA = 24:
            print,(((a · b) · c) · d)
        else if situationB = 24:
            print,((a · (b · c)) · d)
        else if situationC()=24:
            print,(a · ((b · c) · d))
        else if situationD()=24:
            print,(a · (b · (c · d)))
        else if situationE()=24:
            print,((a · b) · (c · d))
        end if
end do

function calculate(a, b,operator)                      ⇒ calculate the expression"a(operator)b"
return (a(operator)b)

function situationA()
calculate (((a · b) · c) · d)
function situationB()
calculate ((a · (b · c)) · d)
function situationC()
calculate (a · ((b · c) · d))                          ⇒ 5 different types of orders of calculations
function situationD()
calculate (a · (b · (c · d)))
function situationE()
calculate ((a · b) · (c · d))
```

2.4. Testing Case

随机选择了几个数组用本程序进行计算，得到了正确的结果，如下所示。

```
PS C:\Users\Yzy\Desktop\computation physics\Fortran\24_point> cd "c:\Users\Yzy\Desktop\computation physics\Fortran\24_poin
t\" ; if ($?) { gfortran 24_point.f90 -o 24_point } ; if ($?) { .\24_point }
 Input 4 numbers ranging from 1 to 10:
1,2,4,8
 (((          1 -          2 )+          4 )*          8 )
```

Figure 2: Solution to the given numbers :1,2,4,8

```
PS C:\Users\Yzy\Desktop\computation physics\Fortran\24_point> cd "c:\Users\Yzy\Desktop\computation physics\Fortran\24_poin
t\" ; if ($?) { gfortran 24_point.f90 -o 24_point } ; if ($?) { .\24_point }
 Input 4 numbers ranging from 1 to 10:
1,3,6,7
 (((          1 *          7 )-          3 )*          6 )
```

Figure 3: Solution to the given numbers : 1,3,6,7

```
PS C:\Users\Yzy\Desktop\computation physics\Fortran\24_point> cd "c:\Users\Yzy\Desktop\computation physics\Fortran\24_poin
t\" ; if ($?) { gfortran 24_point.f90 -o 24_point } ; if ($?) { .\24_point }
 Input 4 numbers ranging from 1 to 10:
1,1,1,1
 No solution for your input.
 Please try again.
```

Figure 4: No solution for the given numbers : 1,1,1,1