

(课程实践汇报封面模版)

合肥工业大学
《机械优化设计》课程实践
研究汇报

班 级： 机设六班

学 号： 20230581

姓 名： 李继鑫

讲课老师： 王卫荣

日 期： 2023年5月7日

一维搜索

$$\min f(x) = \begin{cases} \frac{c * \cos x}{d * [(x - 2)^2 + 3]} & [0, 2\pi] \\ & [0, 10] \end{cases}$$

注：其中 c、d 为待定系数，用于确定选择的函数是哪一种。

- C 语言程序段如下：

```
#include <stdio.h>

#include<math.h>

#define p 3.14

float fun(float x,float c,float d);

void main(void)

{

float a0,a1,a2,r,a,b;

float y1=0.0000,y2=0.0000,u;

float c,d;

u=0.618;

printf("input[a,b]and r:a= b= r= ");

scanf("%f%f%f",&a,&b,&r);

printf("choose only ONE function number c=1 0 or d=0 1\n");

scanf("%f%f",&c,&d);

if(c==1) d=0;

    else c=0,d=1;
```

```
a1=b-u*(b-a),y1=fun(a1,c,d);
a2=a+u*(b-a),y2=fun(a2,c,d);
do
{
  if(y1>=y2)
  {
    a=a1;
    a1=a2,y1=y2;
    a2=a+u*(b-a),y2=fun(a2,c,d);
  }
  else
  {
    b=a2;
    a2=a1,y2=y1;
    a1=b-u*(b-a),y1=fun(a1,c,d);
  }
}while(fabs((b-a)/b)>r && fabs((y2-y1)/y2)>r);
a0=0.5*(a+b);
printf("The best result a0=%f\n",a0);
}
/*****function editing*****/
```

```

float fun(float x,float c,float d)
{
    float Y;

    Y=c*cos(x)+d*((x-2)*(x-2)+3);

    return(Y);
}

```

- 选择第二的函数 (0, 1) 极小值 $a_0=2.000970$, 理论值为 2, 对的。

```

"F:\教程\Debug\0.618.exe"
input[a,b]and r:a= b= r= 0 10 0.00001
choose only ONE function number c=1 0 or d=0 1
0 1
The best result a0=2.000970
Press any key to continue

```

- 选择第一的函数 (1, 0) 极小值 $a_0=3.140134$, 理论值为 π , 对的。

```

"F:\教程\Debug\0.618.exe"
input[a,b]and r:a= b= r= 0 6.28 0.00001
choose only ONE function number c=1 0 or d=0 1
1 0
The best result a0=3.140134
Press any key to continue

```

(一) 单位矩阵

- C 语言程序实现 n 阶单位矩阵如下：

```
#include<stdio.h>
```

```
void main(void)
```

```
{
```

```
    int n,i,j;
```

```
    int d=1,c=0;
```

```
    printf("Input the dimension of array n=\n");
```

```
        scanf("%d",&n);
```

```
        printf("the array of n*n is:\n");
```

```
        for(i=0;i<n-1;i++)
```

```
        {
```

```
            for(j=0;j<n-1;j++)
```

```
            {
```

```
                if(i==j) printf("%3d",d);
```

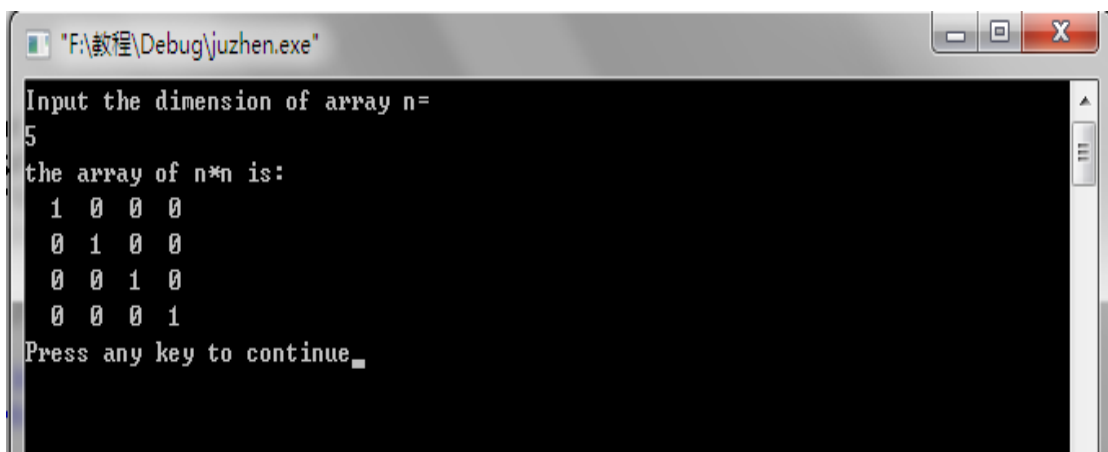
```
                else printf("%3d",c);
```

```
            }
```

```
        printf("\n");
```

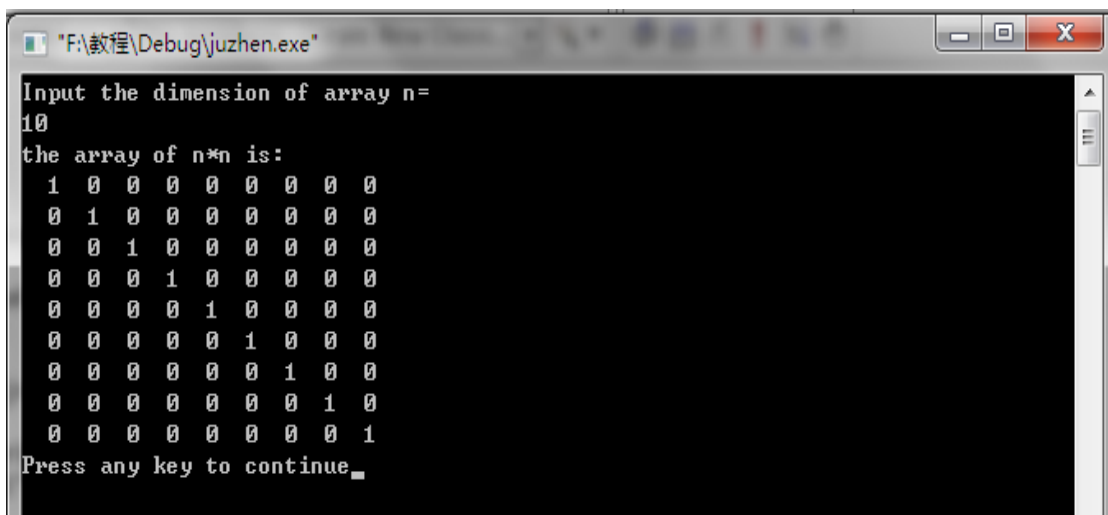
```
}  
  
}
```

- 试验数据验证如下，取 $n=5$ 得到 5 阶单位矩阵：



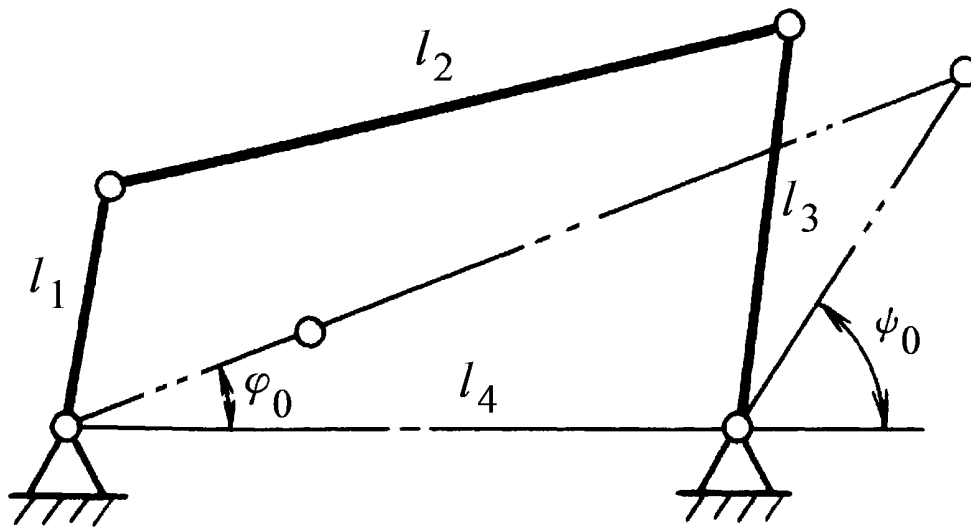
```
"F:\教程\Debug\juzhen.exe"  
Input the dimension of array n=  
5  
the array of n*n is:  
1 0 0 0  
0 1 0 0  
0 0 1 0  
0 0 0 1  
Press any key to continue_
```

- 取 $n=10$ 得到 10 阶单位矩阵：



```
"F:\教程\Debug\juzhen.exe"  
Input the dimension of array n=  
10  
the array of n*n is:  
1 0 0 0 0 0 0 0 0 0  
0 1 0 0 0 0 0 0 0 0  
0 0 1 0 0 0 0 0 0 0  
0 0 0 1 0 0 0 0 0 0  
0 0 0 0 1 0 0 0 0 0  
0 0 0 0 0 1 0 0 0 0  
0 0 0 0 0 0 1 0 0 0  
0 0 0 0 0 0 0 1 0 0  
0 0 0 0 0 0 0 0 1 0  
0 0 0 0 0 0 0 0 0 1  
Press any key to continue_
```

(二) 连杆机构问题优化设计



- 修改程序实现过程如下:

```
unit sumt_fgh;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, Math;
```

```
type
```

```
arr3 = array[1..55] of real;
```

```
type
```

```
TForm3 = class(TForm)
```

```
private
```

```

    { Private declarations }

public

    { Public declarations }

end;

var

    Form3: TForm3;

    Procedure ffx;

    Procedure ggx;

    Procedure hhx;

implementation

uses sumt_1, sumt_2;

{$R *.DFM}

//      应用惩罚函数法优化措施计算优化问题，顾客应首先根据详细问题

// 建立其优化的数学模型(①目的函数；②不等式约束函数；③等式约束函数。)；

//      min  F[x]

//      s.t. GX[j]≤0  (j=1,2,...kg)

//      HX[j]=0  (j=1,2,...kh)

// 再将目的函数、约束函数按 DELPHI 语言改写成体现式，

//      fx := ...;

```



```

//    gx[1] := ...;

//    gx[2] := ...;

//    .

//    .

//    gx[kg] := ...;

//    hx[1] := ...;

//    hx[2] := ...;

//    .

//    .

//    hx[kh] := ...;

// 替代下面的 procedure ffx 段中的目的函数的计算体现式 fx;

// 替代下面的 procedure ggx 段中的目的函数的计算体现式 gx[1]、gx[2]...gx[kg]

// 替代下面的 procedure hhx 段中的目的函数的计算体现式 hx[1]、hx[2]...hx[kh]}

procedure ffx; //目的函数

var y0,y1,z,t,b,a,y2 : double;

var i :integer;

begin

    with form1.sumt do    begin

        fx:=0;

        for i:= 1 to 30 do

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/865340241022011214>