第一次大作业

陈文宇　　徐维震　　刘小端　　高鹏智　　阴雅萱

2022 年 10 月 25 日

摘要

学习 latex 使我快乐

目录

表格

插图

1 教学测试 LaTeX

1.1 不换行的公式编辑

a,b,c 的关系为 $a+b=c$，（$a+b=c$）[1] $a+b=c$ 还是我瞎编的

1.2 换行居中的公式编辑

a,b,c 的关系如下：

$$a+b=c$$

依旧是我瞎编的

且 a,b,c 满足：

$$a*b=c$$

1.3 希腊字母

ω,π,σ,α,γ,β

1.4 上下标

$x^2-2x+1=0$,$y_2-y_1=0$

1.5 数学公式

$$y=\sin^2 x+\cos^2 x$$

$$y=\log_3 x$$

$$y=\ln x$$

$$y=\sqrt[2]{x^2+1}$$

参见 1.5

[1] 我瞎编的

1.6 分式

1/3

$$\frac{1}{3}$$

$\frac{x^2+1}{x^3-1}$ $\frac{x^3+1}{x^4+2}$

1.7 行间公式

$$\frac{x^2+1}{x^3-1}$$

$$\frac{x^2+1}{x^3-1}$$

$$\frac{x^2+1}{x^3-1}$$

1.8 自动编号的 equation 环境（可引用）

交换律详见公式 (1), 结合律详见公式 (2)

$$a+b=b+a \tag{1}$$

$$(a*b)*c=a*(b*c) \tag{2}$$

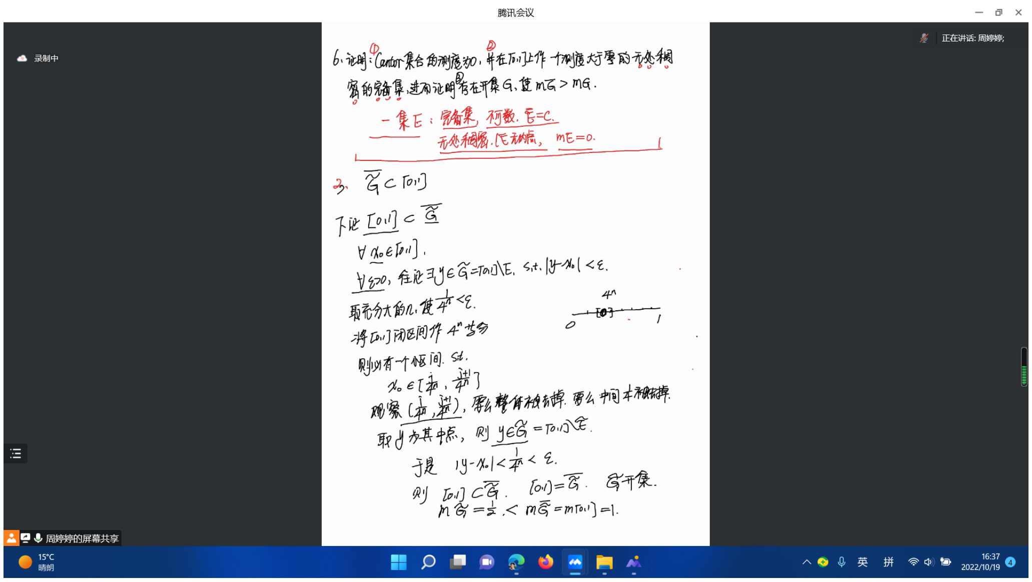
$$(a*b)*c=a*(b*c)$$

2 特殊环境

2.1 列表

:	1	:	1	one
:	11	:	3	eleven

2.2 插图



2.3 盒子

|mbox 命令可以生成一个最基本的盒子|

| makebox 命令可以生成一个最基本的盒子 |

| makebox[][l] 命令可生成一个最基本的盒子 |

| makebox[][r] 命令能生成一个最基本的盒子|

| makebox[][s] 命令会生成一个最基本的盒子|

|mbox 命令可以生成一个最基本的盒子|

| framebox 命令可以生成一个最基本的盒子 |

| framebox[][l] 命令可生成一个最基本的盒子 |

| framebox[][r] 命令能生成一个最基本的盒子|

| framebox[][s] 命令会生成一个最基本的盒子|

三字经：人之初
性本善
性相近
习相远

千字文：天地玄黄
宇宙洪荒

这是一个垂直盒子的领域。[a]

[a] 脚注来自迷你业

2.4 浮动体

浮动体位置参数的设定参见表 1

表 1: 浮动体的位置参数

参数	含义
h	当前位置（代码所处的上下文）
t	顶部
b	底部
p	单独成页
!	在决定位置是忽视限制

注 1：这是浮动体的位置参数

2.5 gather 环境和 gather* 环境

$$a \times b = c \tag{3}$$

$$b \times a = c \tag{4}$$

$$a \times b = c$$

$$b \times a = c$$

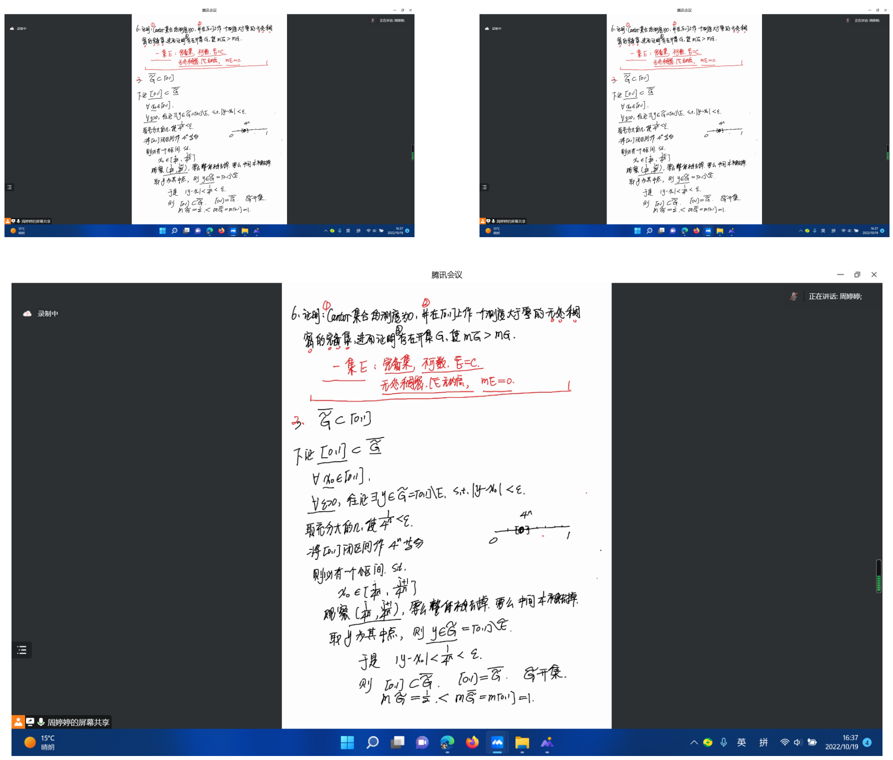


图 1: 随便的插图

3 数学矩阵的基本形式

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \qquad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{Bmatrix} 1 & 0 \\ 0 & 1 \end{Bmatrix} \qquad \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \qquad \begin{Vmatrix} 1 & 0 \\ 0 & 1 \end{Vmatrix}$$

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

3.1 数字矩阵的常用省略符

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ & \ddots & \vdots \\ 0 & & a_{nn} \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & \dots & a_{1n} \\ \vdots & \iddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}$$

3.2 分块矩阵

$$\begin{bmatrix} \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} & \text{0} \\ \text{0} & \begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix} \end{bmatrix}$$

3.3 三角矩阵

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ & a_{22} & \dots & a_{2n} \\ & & \ddots & \vdots \\ \multicolumn{2}{c}{\raisebox{1.3ex}[0pt]{\Huge 0}} & & a_{nn} \end{bmatrix}$$

3.4 跨列的省略号

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ \hdotsfor{4} \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

3.5 行内小矩阵

复数 $z=(x,y)$ 还可以用矩阵 $\left(\begin{smallmatrix} x & -y \\ y & x \end{smallmatrix}\right)$ 来表示

3.6 array 环境

$$\begin{array}{c|c} \frac{1}{2} & 0 \\ \hline 0 & \frac{a}{b} \end{array}$$

$$\begin{array}{c} \left(\begin{array}{ccc|ccc} a & \cdots & a & b & \cdots & b \\ & \ddots & \vdots & \vdots & \adots & \\ & & a & b & & \\ \hline & & & c & \cdots & c \\ & \multicolumn{2}{c|}{\Huge 0} & \vdots & & \vdots \\ & & & c & \cdots & c \end{array}\right) \begin{array}{l} \left.\rule{0mm}{10mm}\right\}p \\ \left.\rule{0mm}{10mm}\right\}q \end{array} \\ \underbrace{\rule{25mm}{0mm}}_{m} \underbrace{\rule{25mm}{0mm}} \end{array}$$

日本人口老龄化影响 [1]
“数值分析”课程教学探讨 [2]
基于 LaTeX 的 Web 数学 [3]

4 参考文献的编辑

参考文献

[1] 张翠玉. 日本人口老龄化对储蓄率的影响研究. 硕士, 吉林大学, 2020.

[2] 陈小美 娄朋林彭叶辉, 谭敏. 地方高校面向数学师范专业认证的“数值分析”课程教学探讨. 教师, (35):79–80, 2021.

[3] 陈立辉，苏伟，蔡川，陈晓薇. 基于 latex 的 web 数学2022-06.

5 代码展示

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

//1.编写 h1 函数
//2.编写 h2 函数
//3.带入结点,每个函数都会返回一个函数值,将其相乘,
//  4个这样的乘积加权后得到每部分的函数值
//4.三个部分函数值相加为插值多项式的值 p
//主函数: 定义参数, 初始化, 调用 p 即可

//编写 h1 h2 利用其 封装在 p 函数里
double h1(int T,int j,double x[],double xs);
double h2(int T,int j,double x[],double xs);
double px1(double xs,double ys,int N,int m
    ,double x[],int n,double y[],double f[],double fx[],double fy[],double fxy[]);
double px2_ex(double xs,double ys,int N,int m,
    ,double x[],int n,double y[],double f[],double fx[],double fy[]);
double px2(double xs,double ys,int N,int m,double x[],int n,double y[],
    ,double f[],double fx[],double fy[]);
double l_2(int T,int j,double x[],double xs);

int main(){
//给出 结点指标 N*N
//待求点(xs,ys)所在分片位置(x[m],y[n]),(x[m+1],y[n]),(x[m],y[n+1]),(x[m+1],y[n+1])
//步长h, 结点(x[],y[])
//录入 f fx fy fxy 的值
// 返回 待求点的插值多项式的值 p
int N=6,m,n,i,j;
```

```
double h;
double x[N],y[N];
double f[N*N],fx[N*N],fy[N*N],fxy[N*N];
double xs=1.0/3.0,ys=2.0/3.0;
double p;

//初始化
h=1.0/(N-1);
for(i=0;i<N;i++){
x[i]=h*i;
y[i]=h*i;
}
for(i=0;i<N;i++){
for(j=0;j<N;j++){
f[j*N+i]=sin(pow(x[i],2)*y[j]+1);
fx[j*N+i]=2*x[i]*y[j]*cos(pow(x[i],2)*y[j]+1);
fy[j*N+i]=pow(x[i],2)*cos(pow(x[i],2)*y[j]+1);
fxy[j*N+i]=2*x[i]*cos(pow(x[i],2)*y[j]+1)
-2*pow(x[i],3)*y[j]*sin(pow(x[i],2)*y[j]+1);
}
}
/*
int M=10;
h=0.2/M;
for(i=1;i<M;i++){

xs=0.2+h*i;
m=floor((N-1)*xs);

ys=0.4+1e-6;
n=floor((N-1)*ys);
p=px2(xs,ys,N,m,x,n,y,f,fx,fy);
printf("question2:待求点插值多项式的为: %.12lf\n",p);
```

```
ys=0.4-1e-6;
n=floor((N-1)*ys);
p=px2(xs,ys,N,m,x,n,y,f,fx,fy);
printf("question2:待求点插值多项式的为: %.12lf\n",p);

printf("\n");

}
*/

m=floor((N-1)*xs);

n=floor((N-1)*ys);

p=px1(xs,ys,N,m,x,n,y,f,fx,fy,fxy);

printf("question1:待求点插值多项式的为: %.12lf\n",p);

p=px2_ex(xs,ys,N,m,x,n,y,f,fx,fy);

printf("question2:基于算法一的待求点插值多项式的值为: %.12lf\n",p);


p=px2(xs,ys,N,m,x,n,y,f,fx,fy);

printf("question2:基于算法二的待求点插值多项式的为: %.12lf\n",p);

}

double h1(int T,int j,double x[],double xs){
double h;
if(T==0){
```

```
h=pow((xs-x[j+1])/(x[j]-x[j+1]),2)*(1-2*(xs-x[j])/(x[j]-x[j+1]));
}
if(T==1){
h=pow((xs-x[j])/(x[j+1]-x[j]),2)*(1-2*(xs-x[j+1])/(x[j+1]-x[j]));
}

return h;
}

double h2(int T,int j,double x[],double xs){
double h;
if(T==0){
h=pow(((xs-x[j+1])/(x[j]-x[j+1])),2)*(xs-x[j]);
}
if(T==1){
h=pow(((xs-x[j])/(x[j+1]-x[j])),2)*(xs-x[j+1]);
}

return h;
}

double l_2(int T,int j,double x[],double xs){
double h;
if(T==0){
h=pow(((xs-x[j+1])/(x[j]-x[j+1])),2);
}
if(T==1){
h=pow(((xs-x[j])/(x[j+1]-x[j])),2);
}

return h;
}
```

```
double px1(double xs,double ys,int N,int m,
,double x[],int n,double y[],double f[],double fx[],double fy[],double fxy[]){
double p=0;
int i,j;

for(i=0;i<=1;i++){
for(j=0;j<=1;j++){
p+=h1(i,m,x,xs)*h1(j,n,y,ys)*f[(n+j)*N+m+i] //给出part1的值
+h2(i,m,x,xs)*h1(j,n,y,ys)*fx[(n+j)*N+m+i] //给出part2的值
+h1(i,m,x,xs)*h2(j,n,y,ys)*fy[(n+j)*N+m+i]
+h2(i,m,x,xs)*h2(j,n,y,ys)*fxy[(n+j)*N+m+i];//给出part3的值
}
}

return p;
}

double px2_ex(double xs,double ys,int N,int m
,double x[],int n,double y[],double f[],double fx[],double fy[]){
double p=0;
int i,j;

for(i=0;i<=1;i++){
for(j=0;j<=1;j++){
p+=h1(i,m,x,xs)*h1(j,n,y,ys)*f[(n+j)*N+m+i] //给出part1的值
+h2(i,m,x,xs)*h1(j,n,y,ys)*fx[(n+j)*N+m+i] //给出part2的值
+h1(i,m,x,xs)*h2(j,n,y,ys)*fy[(n+j)*N+m+i];

}
}

return p;
}
```

```
double px2(double xs,double ys,int N,int m,double x[],int n
,double y[],double f[],double fx[],double fy[]){
double p1=0,p2=0,p;
int i,j;

//给出part1的值
for(i=0;i<=1;i++){
for(j=0;j<=1;j++){
p1=p1+(h1(i,m,x,xs)*l_2(j,n,y,ys)+h1(j,n,y,ys)*l_2(i,m,x,xs)
-l_2(i,m,x,xs)*l_2(j,n,y,ys))*f[(n+j)*N+m+i];
}
}
//给出part2的值
for(i=0;i<=1;i++){
for(j=0;j<=1;j++){
p2=p2+h2(i,m,x,xs)*l_2(j,n,y,ys)*fx[(n+j)*N+m+i]
+l_2(i,m,x,xs)*h2(j,n,y,ys)*fy[(n+j)*N+m+i];
}
}

p=p1+p2;

return p;

}
```