**二维可压缩磁流体模拟（用MATLAB语言复原工作）**

陈宗华(Zong-hua Chen, 814484233@qq.com,15307756701)

玉林师范学院（东校区）物理与电信工程学院

广西玉林市教育东路1303号，537000

2019-08

**摘 要**

本工作是围绕谢华生博士的著作《计算等离子体物理导论》[1]第5章5.42节撕裂模及磁重联的内容展开讨论的。此书是等离子体物理数值计算与模拟的入门教程，通过具体的算例来帮助初学者理解相关的物理概念和物理图像，既有新意又有实用性，对于初学入门者来说，是一本不可多得的书。书中的算例均提供了相关的代码，基本上是以MATLAB语言为主，有些算例并没有完全用MATLAB语言编写，例如撕裂模及磁重联的那一节[2]的算例提供的是Fortran代码，对于多种代码语言不能一一精通的初学者来说这是一个挑战。此外，作者提供的参考著作(傅竹风，1995《空间等离子体数值模拟》)[3]网上缺货，电子版在全国磁约束核聚变专业群也未能求得。为此，我按照书籍提供的线索及其提供的Fortran代码注释说明，用MATLAB语言对撕裂模及磁重联这一节的内容进行复原工作。

# 1 二维可压缩磁流体方程

平板模型，所有的变量都在(x,z)平面，各变量y方向是均匀的。

## 1.1 原始方程

原始方程没有霍尔项， 为常数。



其中，使用条件



考虑到，我们使用 来表示，例如



把上述方程写成显示的二维方程组（已经归一化）



## 1.2 归一化

对于方程组，归一化的密度，压强，长度，速度和时间的归一化分别是，，。

参数估计



磁Lundquist数。

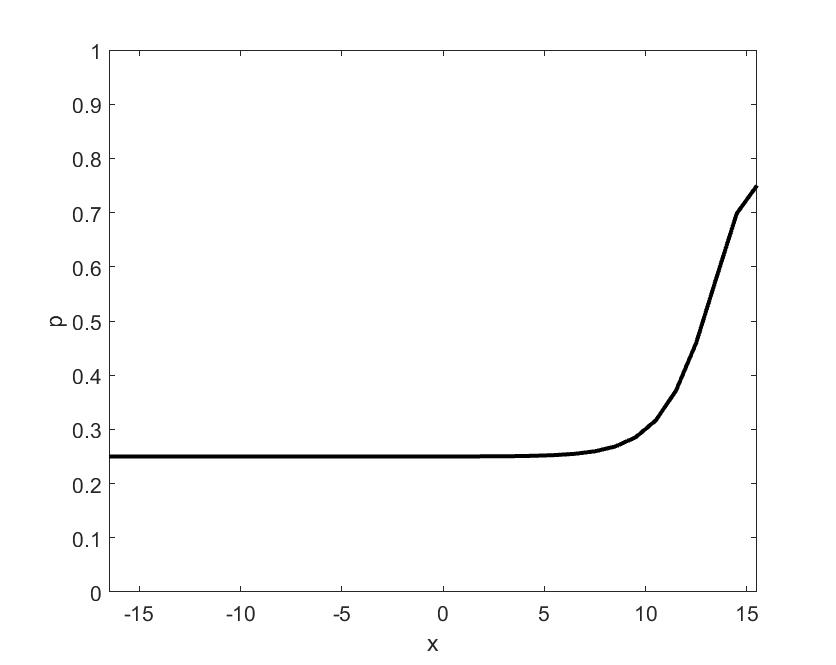
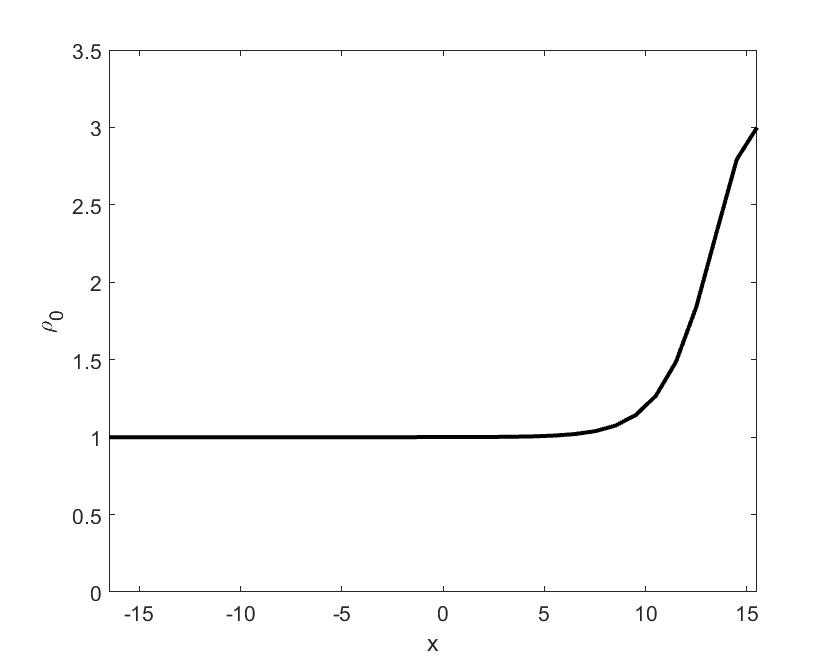
## 1.3 离散化

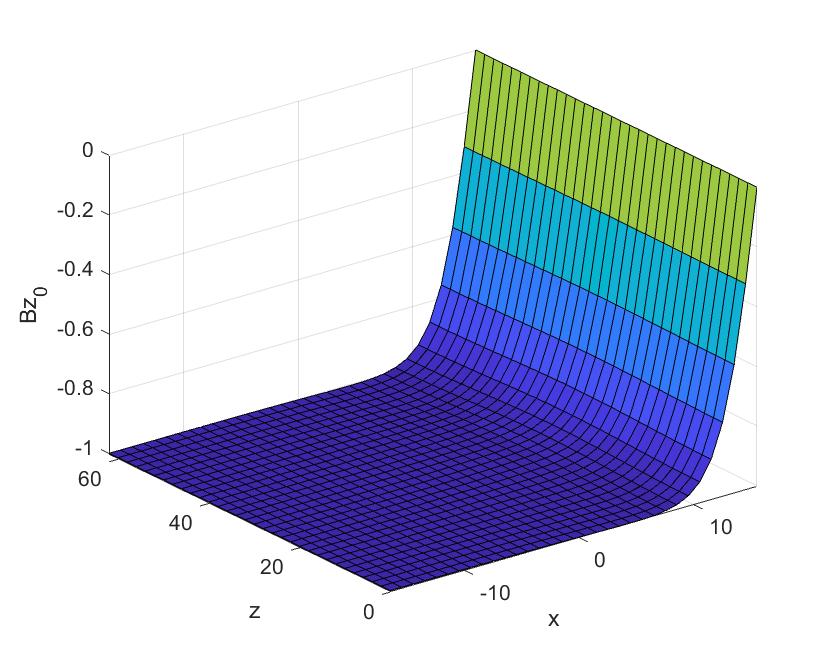
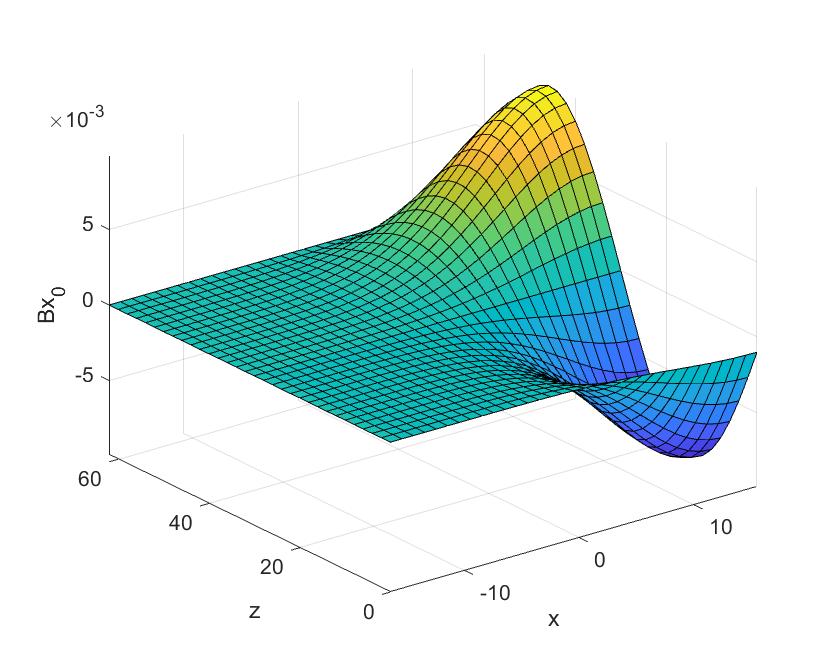
对于方程组右手边的空间微分，采用中心差分格式如下



对于方程组的左边的时间偏微分，采用的是四阶龙格库塔法格式求解。

## 1.4 初始条件





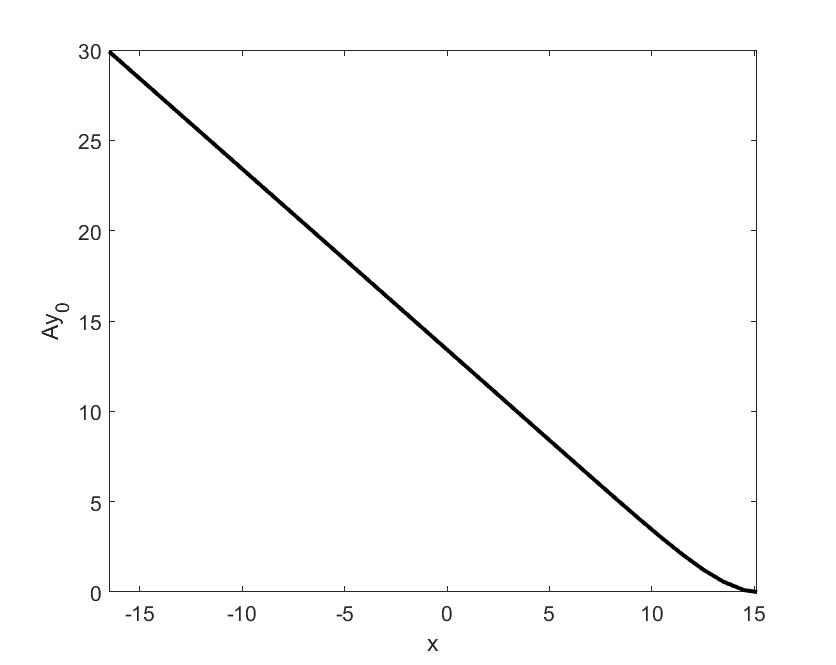


图1 初始参数空间分布

## 1.5 边界条件

在z方向上采用的是周期性边界条件，在x方向上用的是混合边界条件。在x轴的下边界采用条件为



而在x轴的下边界采用的是自由边界条件。在x轴的上边界采用条件为



# 2 MATLAB求解代码结构

变量U(ni,nj,5)存放了5个二维变量，分别代表。本工作是基于MATLAB语言重新对著作[2, 4]提供的Fortran代码进行改写，结合MATLAB的矩阵运算优势，把原来单点循环赋值运算，全部改写成矩阵赋值运算，使代码得到了很大程度上的简化。为擅于MATLAB语言而不会用Fortran的初学者提供参考。另外，在Fortran代码的子程序“Subroutine right(xo,xi)”中，求解的右边pt项本来是对x进行求微分，但实际上输入的字母有误，具体是子程序中“-rdz\*(pt(i+1,j)-pt(i-1,j)))”正确应该是乘以rdx。由于rdx和rdz是个比较相接近的常数，没有对整体的趋势产生显著影响。

# 3 复原模拟结果

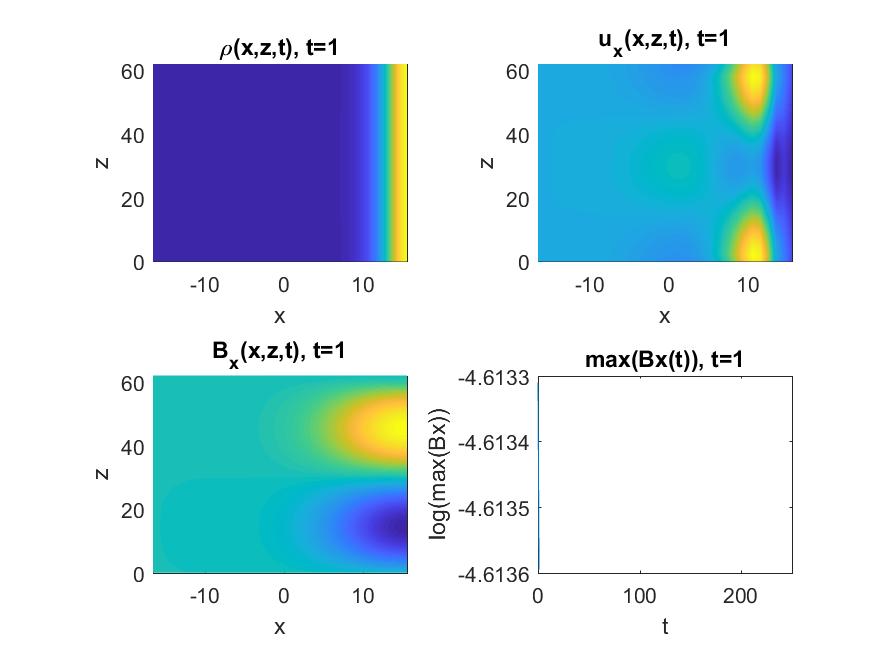
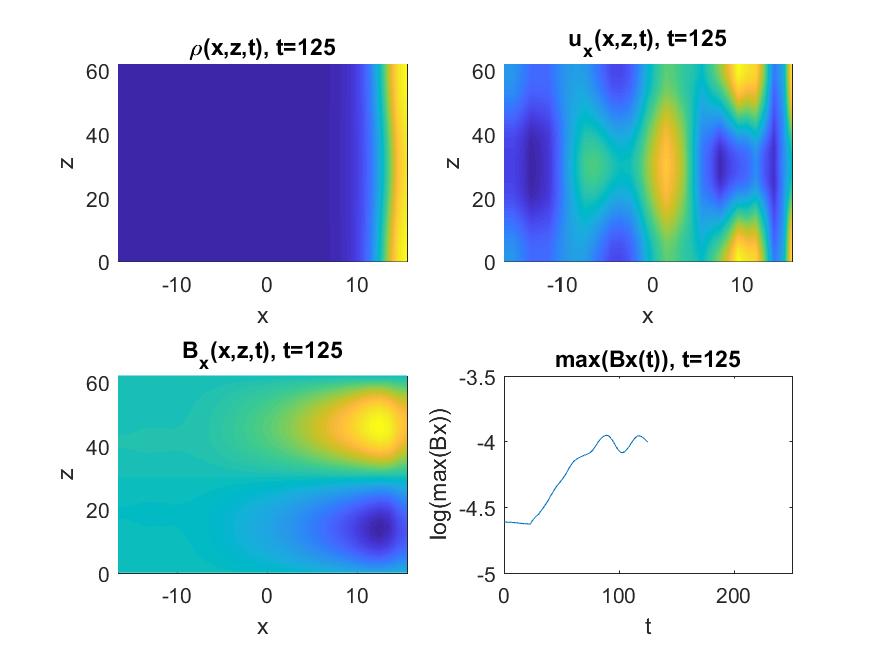
 

图2 场随时间演化t=1s及125s。

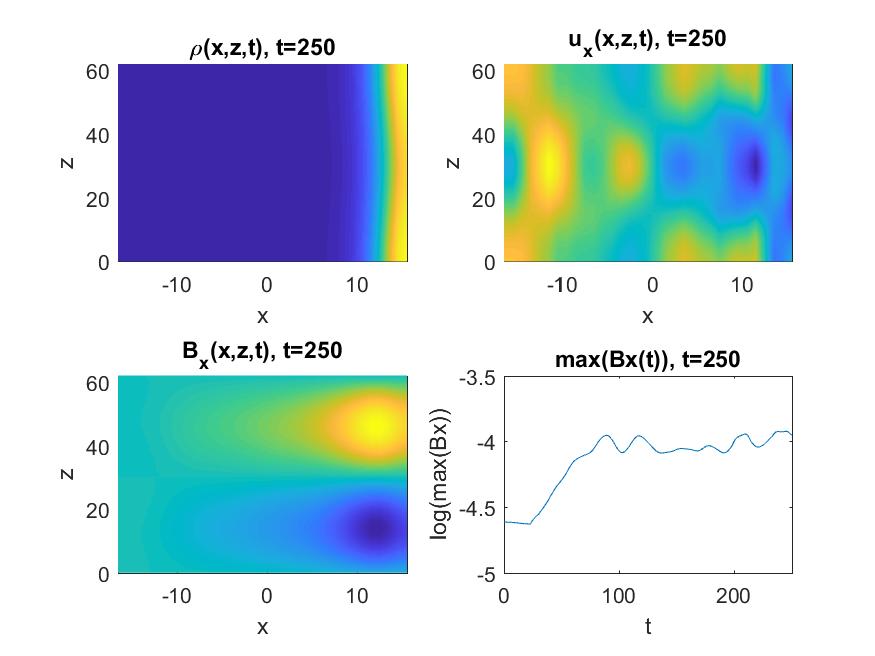


图3 场随时间演化t=250s.

图2-3为一组典型的非线性模拟结果，其中我们可以看到初始扰动调整后，开始进入一段指数增长的线性阶段，然后达到非线性饱和，饱和后基本稳态演化。

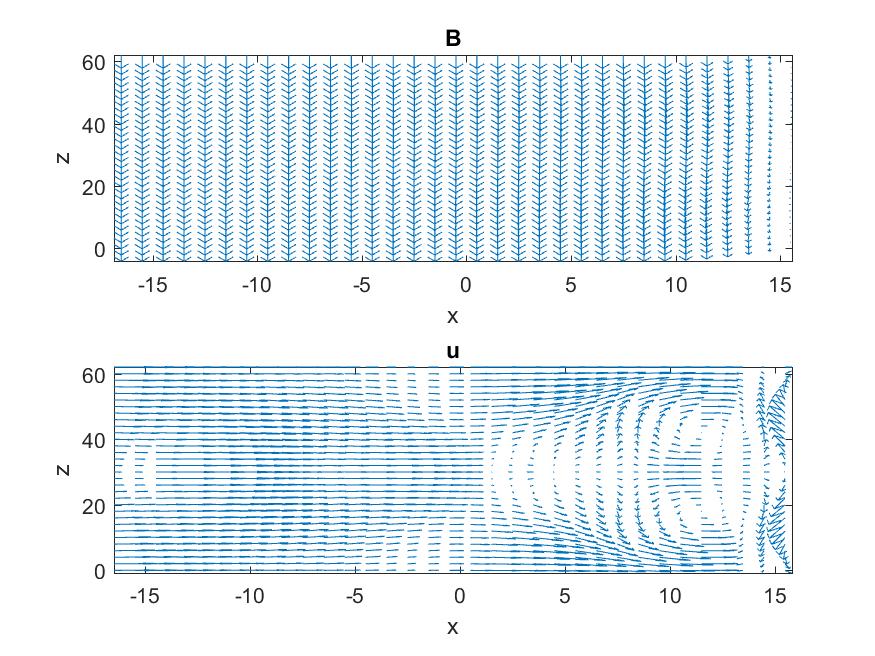
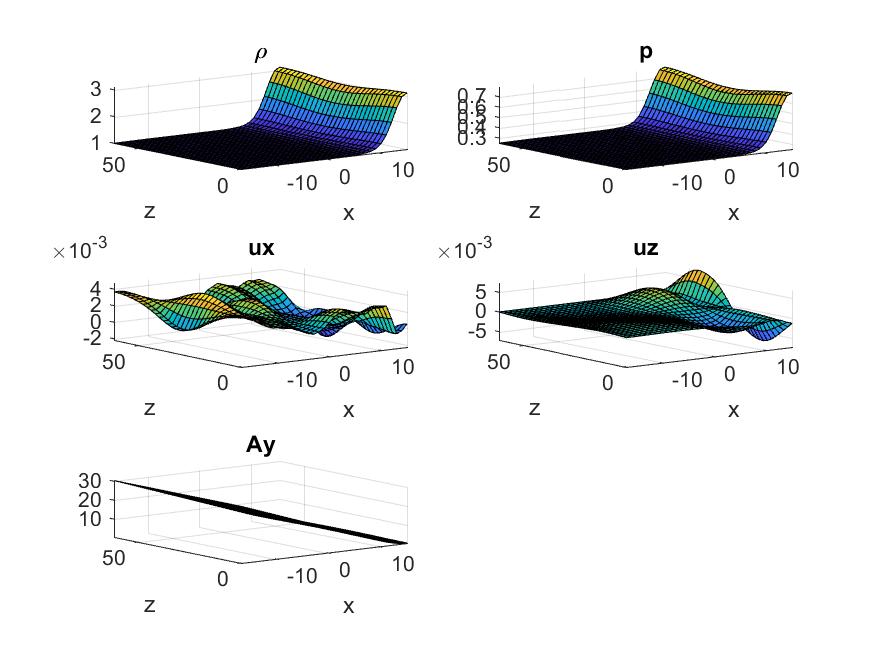


图4 场随时间演化t=250s.

**参考文献**

[1] 谢华生．计算等离子体物理导论[M]．北京：科学出版社，2018．

[2] 谢华生．撕裂模及磁重联[M]//谢华生．计算等离子体物理导论．北京：科学出版社，2018：107-111．

[3] 傅竹风，胡友秋．空间等离子体数值模拟[M]．安徽科学技术出版社，1995．

[4] 谢华生．Hua-sheng XIE\* （谢华生）[EB/OL]．[2019-08-27]．http://hsxie.me/．

# 附录：MATLAB程序

# MHD2D\_main

二维可压缩磁流体模拟（用MATLAB语言复原工作） 陈宗华(Zong-hua Chen, [814484233@qq.com](mailto:814484233@qq.com),15307756701) 玉林师范学院（东校区）物理与电信工程学院 广西玉林市教育东路1303号，537000 2019-08

close;clear;clc;

global dx dz gamma nu eta dt Bx Bz

nx=33;

nz=32;

nt=5000;

dt=0.05; % 要满足 dt < min(dx,dz)/[sqrt(1.0+0.5\*gamma\*beta)\*va]

dx=1.0; dz=2.0;

xl=dx\*nx/2; zl=dz\*(nz-1);

xx=-xl:dx:xl-dx;

zz=0:dz:zl;

[x,z]=meshgrid(xx,zz);

x=x';z=z';

gamma=1.66667; %

eta=0.01;

nu=0.05;

beta=0.5; % beta in x=Lx

rho0=1.0; % rho0 -- mass density rho0 in x=Lx

B0=1.0; % b0 -- B0

bl=3.0; % bl -- width of current sheet

va=sqrt(B0\*B0/rho0);% Alfven velocity

p0=0.5\*beta\*B0\*B0; % p0 -- pressure p0 in x=Lx

t0=0.5\*beta\*va\*va; % t0 -- temperature T0 in x=Lx

cfl=1.5; % Courant–Friedrichs–Lewy condition parameter

U=zeros(nx,nz,5);

% % 初始场分布

s=((1:nx)-nx)\*dx/bl;

b=B0\*tanh(s);

p=p0+0.5\*(B0.^2-b.^2);

rho=p/t0;

Ay=B0.\*bl.\*log(cosh(s));

U(:,:,1)=repmat(rho',1,nz);

U(:,:,2)=repmat(p',1,nz);

U(:,:,5)=repmat(Ay',1,nz);

% % 初始场扰动

nw=1;% nw -- number of initial perturbation

abd=10; % abd -- perturbation width in x

am=[0.01,zeros(1,9)];% am -- prtb amplitude

s=((2:nx)-nx)\*dx/abd;

for m=1:nw

sii=exp(-s.\*s).\*am(m)\*B0;

sij=sin((2\*m\*((1:nz)-0.5\*nz)/(nz-1)+0.5)\*pi)\*dz\*(nz-1)/(2.0\*m\*pi);

end

U(:,:,5)=U(:,:,5)+[zeros(1,nz);sii'\*sij];

t=0;

rho0=U(:,:,1);p0=U(:,:,2);ux0=U(:,:,3);uz0=U(:,:,4);Ay0=U(:,:,5);[Bx0,Bz0]=calcBxz(t,U);

for it=1:nt

t=t+dt;

[Bx,Bz]=calcBxz(t,U);

Bxm(it)=max(max(Bx));

U=RK4(t,U);

if mod(it,20)==0

figure(2)

subplot(3,2,1);surf(x,z,U(:,:,1)); xlabel('x');ylabel('z');title('\rho');axis tight

subplot(3,2,2);surf(x,z,U(:,:,2)); xlabel('x');ylabel('z');title('p');axis tight

subplot(3,2,3);surf(x,z,U(:,:,3)); xlabel('x');ylabel('z');title('ux');axis tight

subplot(3,2,4);surf(x,z,U(:,:,4)); xlabel('x');ylabel('z');title('uz');axis tight

subplot(3,2,5);surf(x,z,U(:,:,5)); xlabel('x');ylabel('z');title('Ay');axis tight

figure(3)

subplot(2,1,1);quiver(x,z,Bx,Bz,2); xlabel('x');ylabel('z');title('B');axis tight

subplot(2,1,2);quiver(x,z,U(:,:,3),U(:,:,4),2); xlabel('x');ylabel('z');title('u');axis tight

figure(4)

subplot(2,2,1);pcolor(x,z,U(:,:,1)); shading('interp');

xlabel('x');ylabel('z');title(['\rho(x,z,t), t=',num2str(t)])

subplot(2,2,2);pcolor(x,z,U(:,:,3)); shading('interp');

xlabel('x');ylabel('z');title(['u\_x(x,z,t), t=',num2str(t)])

subplot(2,2,3);pcolor(x,z,Bx); shading('interp');

xlabel('x');ylabel('z');title(['B\_x(x,z,t), t=',num2str(t)])

subplot(2,2,4);

plot([1:it]\*dt,log(Bxm));xlabel('t');ylabel('max(Bx(t))');title(['max(Bx(t)), t=',num2str(t)])

xlabel('t');ylabel('log(max(Bx))');title(['max(Bx(t)), t=',num2str(t)])

xlim([0,nt\*dt])

drawnow

end

end

function Uo=RK4(t,Ui)

% U(:,:,i),其中 i= 1 2 3 4 5 --> rho p ux uz ay

global dt

k1=rightEq(t,Ui);

U1=Ui+0.5\*dt\*k1;

k2=rightEq(t,U1);

U2=Ui+0.5\*dt\*k2;

k3=rightEq(t,U2);

U3=Ui+dt\*k3;

k4=rightEq(t,U3);

Uo=Ui+dt\*(k1+2\*k2+2\*k3+k4)/6;

end

function k=rightEq(t,U)

% U(:,:,i),其中 i= 1 2 3 4 5 --> rho p ux uz ay

global gamma nu eta dx dz Bx Bz

rho=U(:,:,1);

p=U(:,:,2);

ux=U(:,:,3);

uz=U(:,:,4);

Ay=U(:,:,5);

pt=p+0.5\*(Bx.^2+Bz.^2);% pt=p+b^2/2

[nx,nz,~]=size(U);

ic=1:nx;

jc=1:nz;

ip=ic+1;ip(nx)=1;

im=ic-1;im(1)=nx;

jp=jc+1;jp(nz)=1;

jm=jc-1;jm(1)=nz;

k(:,:,1)=-ux.\*(rho(ip,:)-rho(im,:))/(2\*dx) ...

-uz.\*(rho(:,jp)-rho(:,jm))/(2\*dz) ...

-rho.\*((ux(ip,:)-ux(im,:))/(2\*dx) ...

+(uz(:,jp)-uz(:,jm))/(2\*dz));

k(:,:,2)=-ux.\*(p(ip,:)-p(im,:))/(2\*dx) ...

-uz.\*(p(:,jp)-p(:,jm))/(2\*dz) ...

-gamma.\*p.\*((ux(ip,:)-ux(im,:))/(2\*dx) ...

+(uz(:,jp)-uz(:,jm))/(2\*dz));

k(:,:,3)=-ux.\*(ux(ip,:)-ux(im,:))/(2\*dx) ...

-uz.\*(ux(:,jp)-ux(:,jm))/(2\*dz) ...

-1./rho.\*(pt(ip,:)-pt(im,:))/(2\*dz) ... % 原程序有错误(pt(ipx,:)-pt(imx,:))/(2\*dz)中应该是除以dx的

+1./rho.\*(Bx.\*(Bx(ip,:)-Bx(im,:))/(2\*dx)+Bz.\*(Bx(:,jp)-Bx(:,jm))/(2\*dz)) ...

+nu./rho.\*((ux(ip,:)+ux(im,:)-2\*ux(ic,:))/(dx\*dx)+(ux(:,jp)+ux(:,jm)-2\*ux(:,jc))/(dz\*dz));

k(:,:,4)=-ux.\*(uz(ip,:)-uz(im,:))/(2\*dx) ...

-uz.\*(uz(:,jp)-uz(:,jm))/(2\*dz) ...

-1./rho.\*(pt(:,jp)-pt(:,jm))/(2\*dz) ...

+1./rho.\*(Bx.\*(Bz(ip,:)-Bz(im,:))/(2\*dx)+Bz.\*(Bz(:,jp)-Bz(:,jm))/(2\*dz)) ...

+nu./rho.\*((uz(ip,:)+uz(im,:)-2\*uz(ic,:))/(dx\*dx)+(uz(:,jp)+uz(:,jm)-2\*uz(:,jc))/(dz\*dz));

k(:,:,5)=-ux.\*(Ay(ip,:)-Ay(im,:))/(2\*dx) ...

-uz.\*(Ay(:,jp)-Ay(:,jm))/(2\*dz) ...

+eta.\*((Ay(ip,:)+Ay(im,:)-2\*Ay(ic,:))/(dx\*dx)+(Ay(:,jp)+Ay(:,jm)-2\*Ay(:,jc))/(dz\*dz));

% x的下边界, i=1

k(1,:,1)=k(2,:,1);

k(1,:,2)=k(2,:,2);

k(1,:,3)=0;

k(1,:,4)=k(2,:,4);

k(1,:,5)=0;

% x的上边界, i=end

k(end,:,1)=0 ...

-uz(end,:).\*(rho(end,jp)-rho(end,jm))/(2\*dz) ...

-rho(end,:).\*((ux(end,:)-ux(end-1,:))/(1\*dx) ...

+(uz(end,jp)-uz(end,jm))/(2\*dz));

k(end,:,2)=0 ...

-uz(end,:).\*(p(end,jp)-p(end,jm))/(1\*dz) ...

-gamma.\*p(end,:).\*((ux(end,:)-ux(end-1,:))/(1\*dx) ...

+(uz(end,jp)-uz(end,jm))/(2\*dz));

k(end,:,3)=0;

k(end,:,4)=0 ...

-uz(end,:).\*(uz(end,jp)-uz(end,jm))/(2\*dz) ...

-1./rho(end,:).\*(pt(end,jp)-pt(end,jm))/(2\*dz) ...

+1./rho(end,:).\*(Bx(end,:).\*(Bz(end,:)-Bz(end-1,:))/(1\*dx)+0) ...

+nu./rho(end,:).\*((2\*uz(end-1,:)-2\*uz(end,:))/(dx\*dx)+(uz(end,jp)+uz(end,jm)-2\*uz(end,jc))/(dz\*dz));

k(end,:,5)=0 ...

-uz(end,:).\*(Ay(end,jp)-Ay(end,jm))/(2\*dz) ...

+eta.\*((2\*Ay(end-1,:)-2\*Ay(end,:))/(dx\*dx)+(Ay(end,jp)+Ay(end,jm)-2\*Ay(end,jc))/(dz\*dz));

end

function [Bx,Bz]=calcBxz(t,U)

global dx dz

[nx,nz,~]=size(U);

ic=1:nx;

jc=1:nz;

ip=ic+1;ip(nx)=1;

im=ic-1;im(1)=nx;

jp=jc+1;jp(nz)=1;

jm=jc-1;jm(1)=nz;

Bx=-(U(:,jp,5)-U(:,jm,5))/(2\*dz);

Bz= (U(ip,:,5)-U(im,:,5))/(2\*dx);

Bx(end,:)=-(U(end,jp,5)-U(end,jm,5))/(2\*dz);

Bz(end,:)=0;

end

[Published with MATLAB® R2017a](http://www.mathworks.com/products/matlab/)