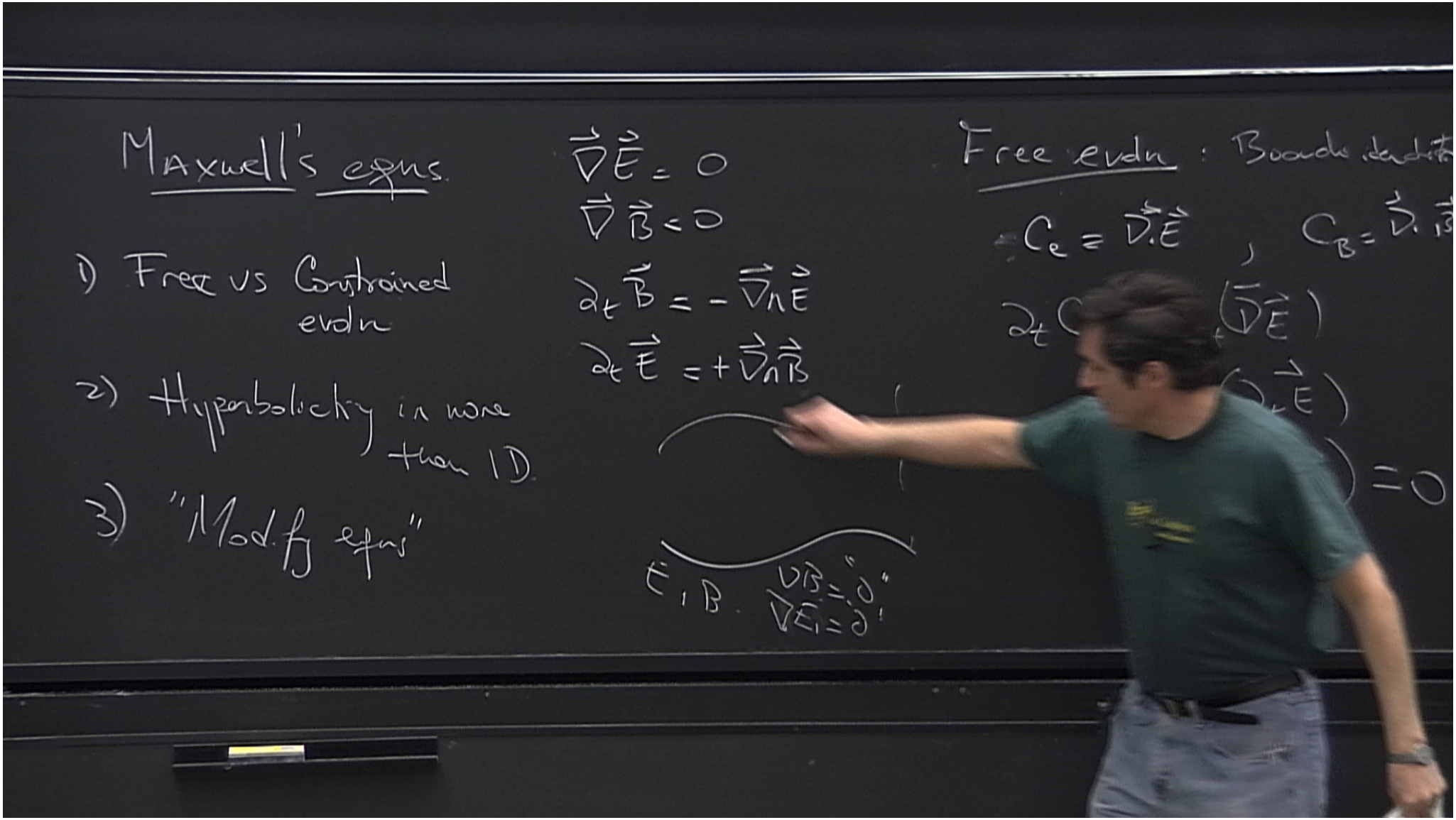
Title: Explorations in Numerical Relativity - Lecture 9

Date: Apr 13, 2012 11:30 AM

URL: http://pirsa.org/12040049

Abstract:


Maxwell's eqns.
1) Free vs Constrained evoln
2) Hyperbolicity in more than 1D
3) "Modified eqns"
$\vec{\nabla}\vec{E} = 0$
$\vec{\nabla}\vec{B} = 0$
$\partial_t \vec{B} = -\vec{\nabla}\times\vec{E}$
$\partial_t \vec{E} = +\vec{\nabla}\times\vec{B}$
E, B
$\vec{\nabla}B = 0$"
$\vec{\nabla}E = 0$!
Free evoln : Boundary
$C_E = \vec{\nabla}\cdot\vec{E}$
$C_B = \vec{\nabla}\cdot\vec{B}$
∂_t
$(\vec{\nabla}\vec{E})$
$\vec{E}$
$= 0$

$\vec{\nabla}\vec{E} = 0$

$\vec{\nabla}\vec{B} = 0$

$\partial_t \vec{B} = -\vec{\nabla} \wedge \vec{E}$

$\partial_t \vec{E} = +\vec{\nabla} \wedge \vec{B}$

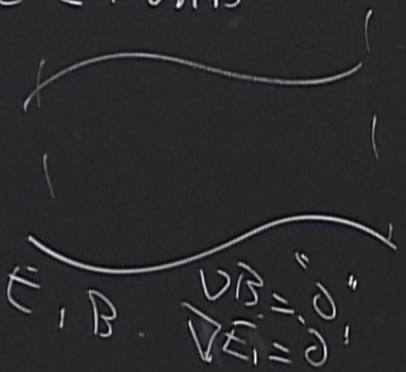
Free evdn : Boundaries $\nabla_{[a} F_{bc]} = 0$

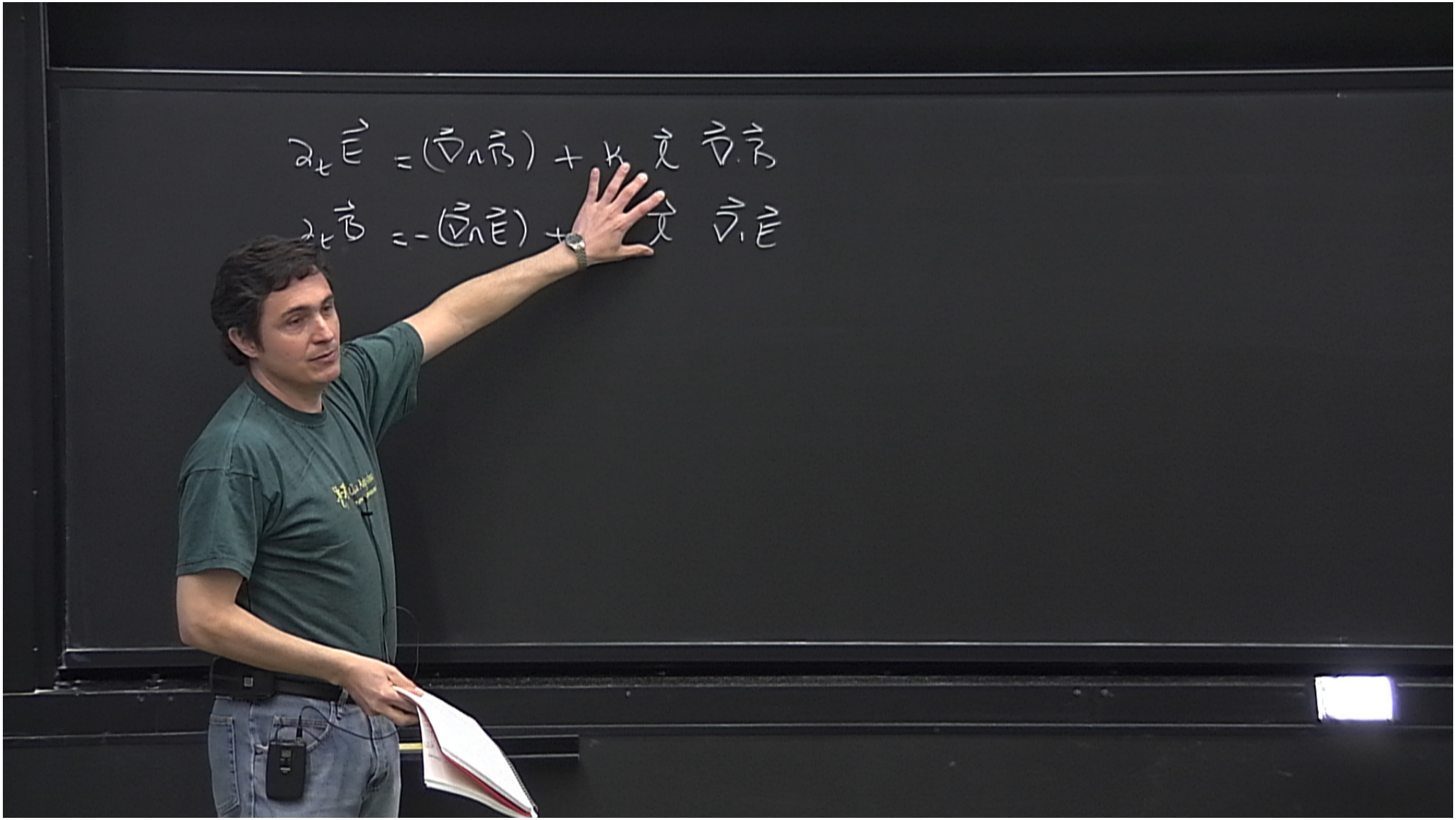
$C_E = \vec{\nabla}.\vec{E}$, $C_B = \vec{\nabla}.\vec{B}$

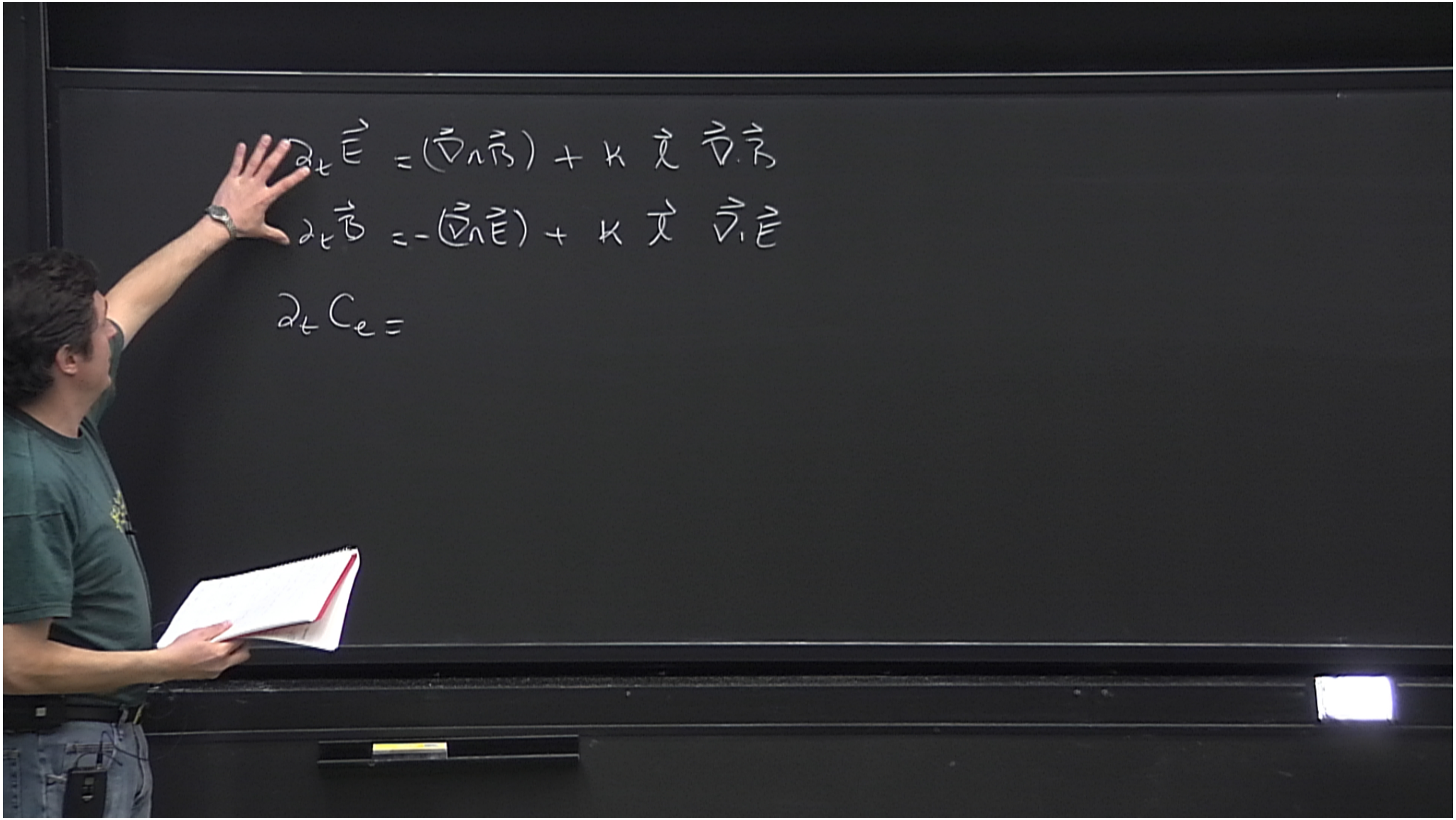
$\partial_t C_E = \partial_t (\vec{\nabla}\vec{E})$

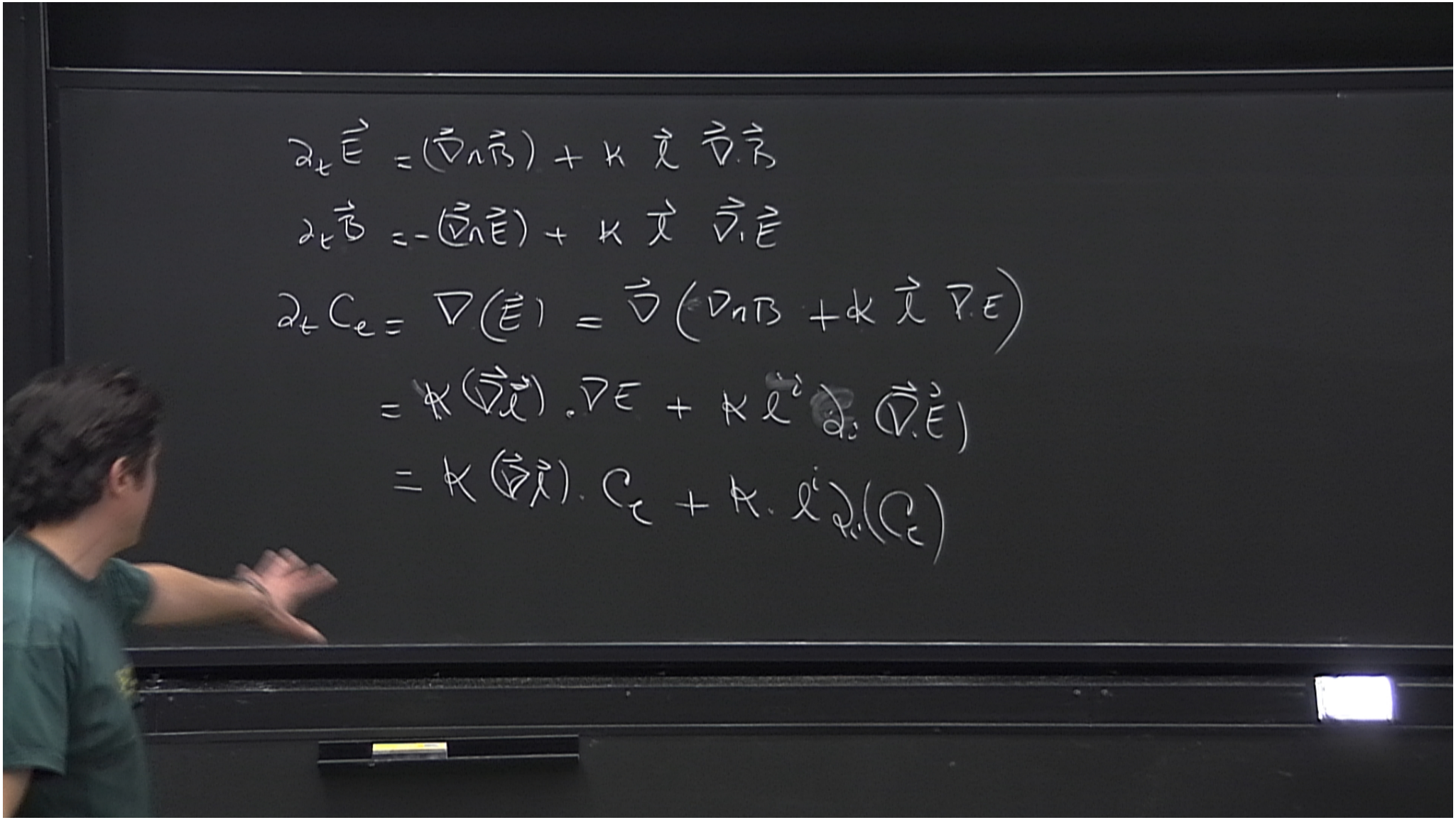
$= \vec{\nabla}(\partial_t \vec{E})$

$= \vec{\nabla}(\vec{\nabla} \wedge \vec{B}) = 0$




$\partial_t \vec{E} = (\vec{\nabla} \wedge \vec{B}) + k \chi \vec{\nabla} \cdot \vec{B}$
$\partial_t \vec{B} = -(\vec{\nabla} \wedge \vec{E}) + \chi \vec{\nabla} \cdot \vec{E}$


$\partial_t \vec{E} = (\vec{\nabla} \wedge \vec{B}) + k \chi \vec{\nabla} \cdot \vec{B}$
$\partial_t \vec{B} = -(\vec{\nabla} \wedge \vec{E}) + k \chi \vec{\nabla} \cdot \vec{E}$
$\partial_t C_e =$


$\partial_t \vec{E} = (\vec{\nabla} \wedge \vec{B}) + k \vec{\chi} \vec{\nabla} \cdot \vec{B}$
$\partial_t \vec{B} = -(\vec{\nabla} \wedge \vec{E}) + k \vec{\chi} \vec{\nabla} \cdot \vec{E}$
$\partial_t C_e = \nabla(\vec{E}) = \vec{\nabla}(\nabla \wedge B + k \vec{\chi} \nabla \cdot E)$
$= k(\vec{\nabla}\vec{\chi}) \cdot \nabla E + k \vec{\chi} \partial_t (\vec{\nabla} \cdot \vec{E})$
$= k(\vec{\nabla}\vec{\chi}) \cdot C_e + k \cdot \vec{\chi} \partial_t (C_e)$

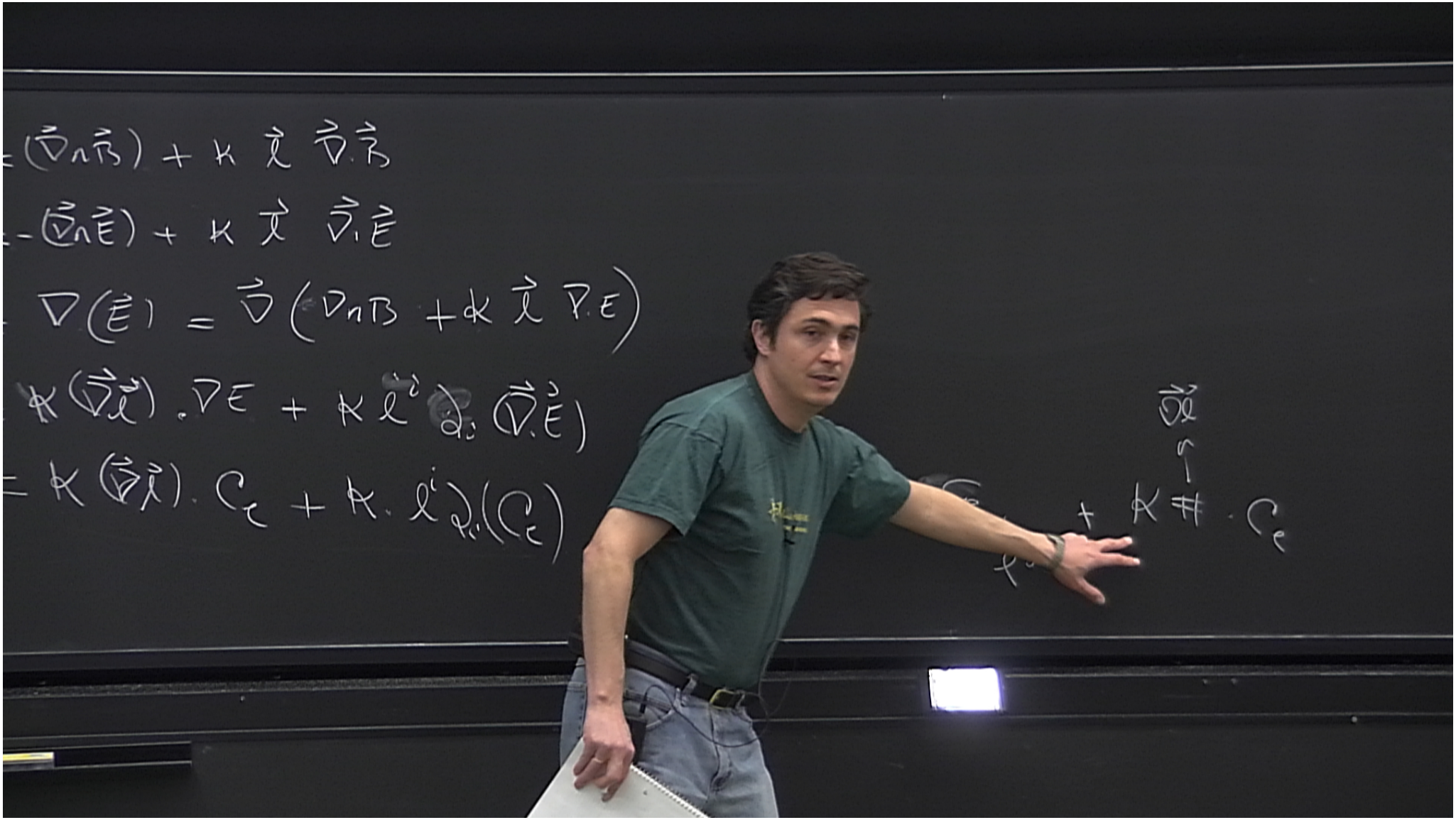
$$\partial_t \vec{E} = (\vec{\nabla} \wedge \vec{B}) + k\, \vec{\xi}\, \vec{\nabla}.\vec{B}$$

$$\partial_t \vec{B} = -(\vec{\nabla} \wedge \vec{E}) + k\, \vec{\xi}\, \vec{\nabla}.\vec{E}$$

$$\partial_t C_e = \nabla(\vec{E}) = \vec{\nabla}\left(\nabla \wedge B + k\, \vec{\xi}\, \nabla.E\right)$$

$$= k(\vec{\nabla}\vec{\xi}).\nabla E + k\,\xi^i \partial_i(\vec{\nabla}.\vec{E})$$

$$\partial_t C_e = k(\vec{\nabla}\vec{\xi}).C_e + k.\xi^i \partial_i(C_e) \rightarrow C_{e,t} = k\, C_{e,\text{drag}}$$



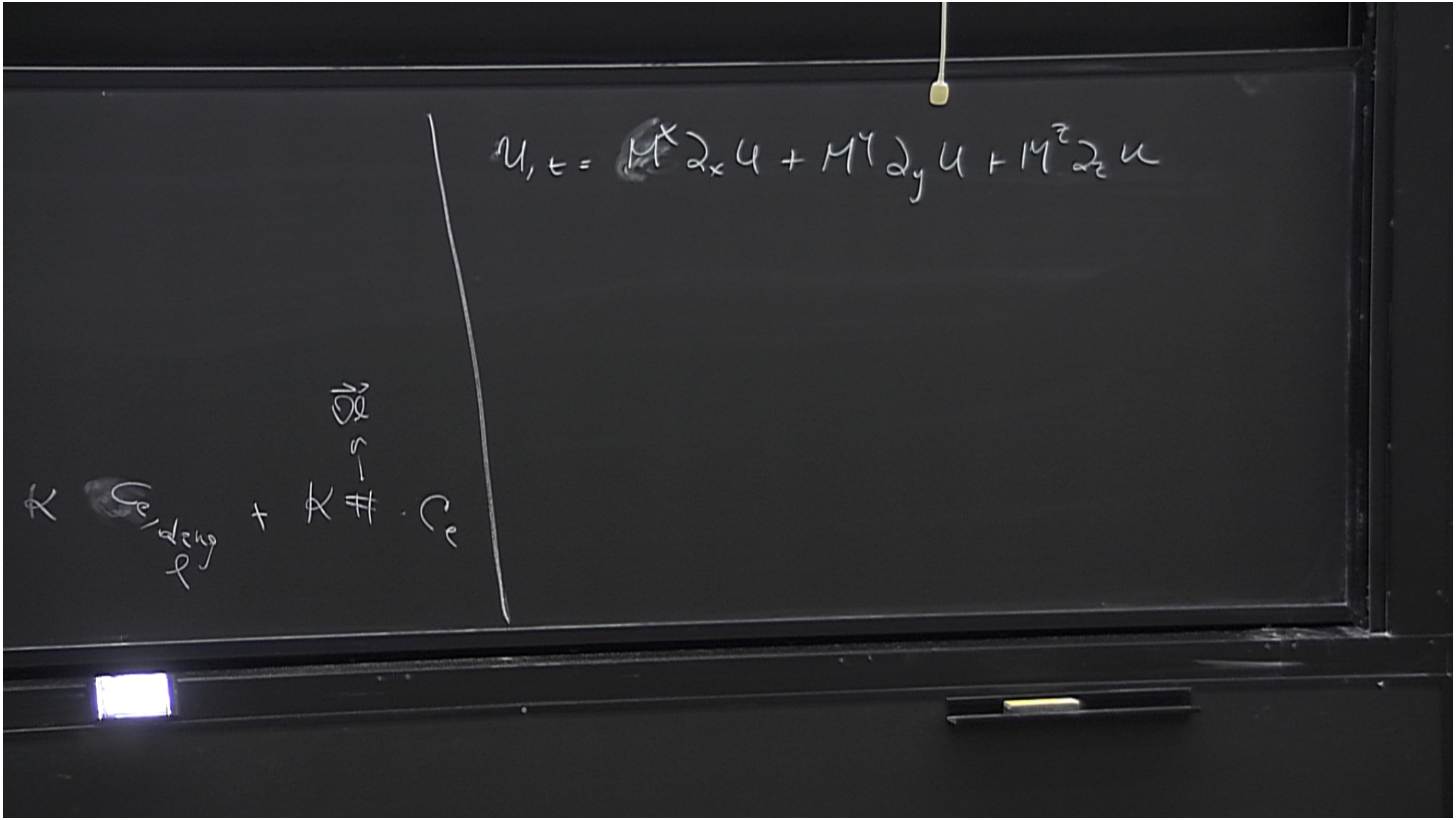
$$\partial_t \vec{E} = (\vec{\nabla} \times \vec{B}) + \kappa\, \vec{\chi}\, \vec{\nabla} \cdot \vec{B}$$

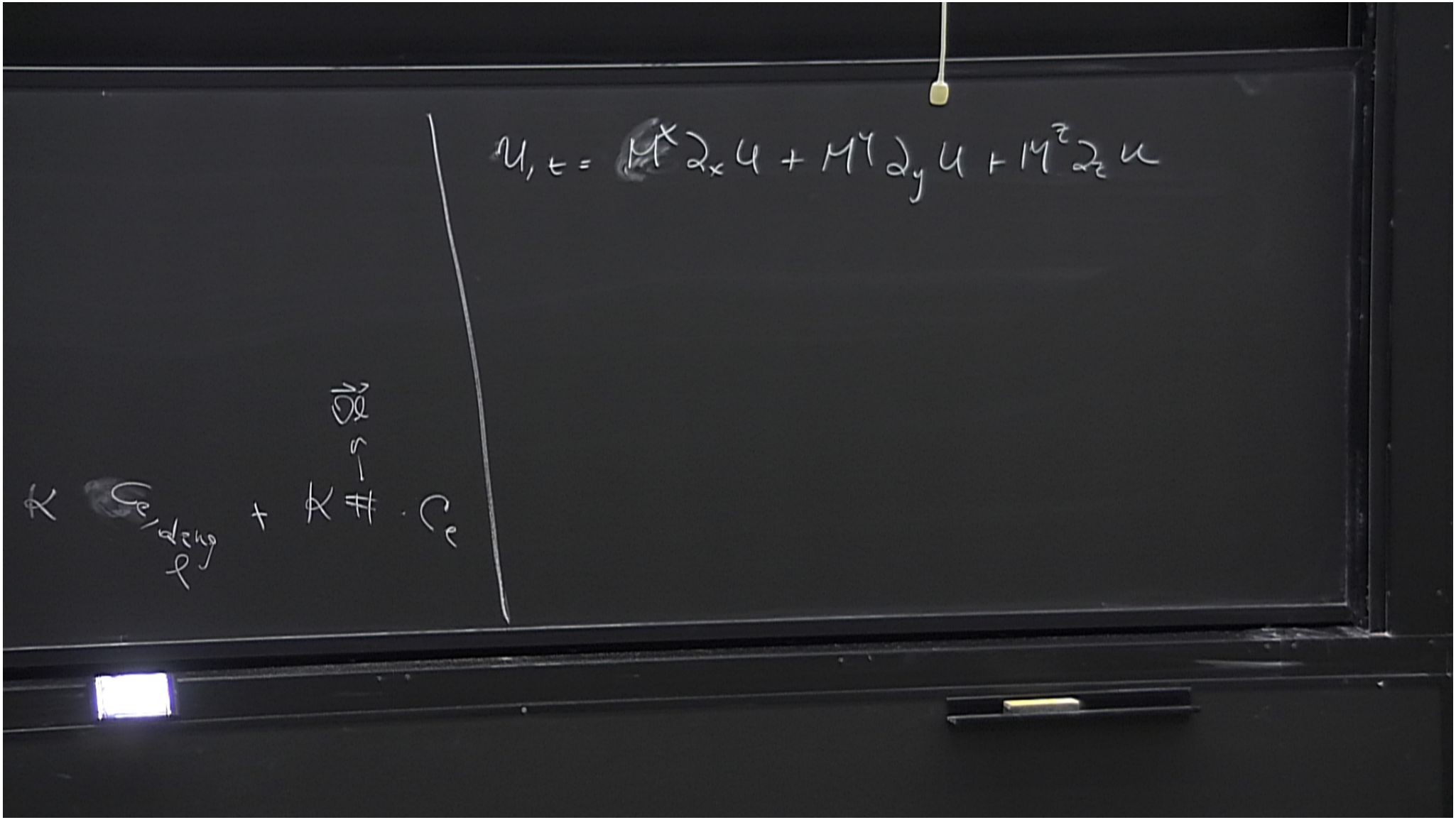
$$\partial_t \vec{B} = -(\vec{\nabla} \times \vec{E}) + \kappa\, \vec{\chi}\, \vec{\nabla} \cdot \vec{E}$$

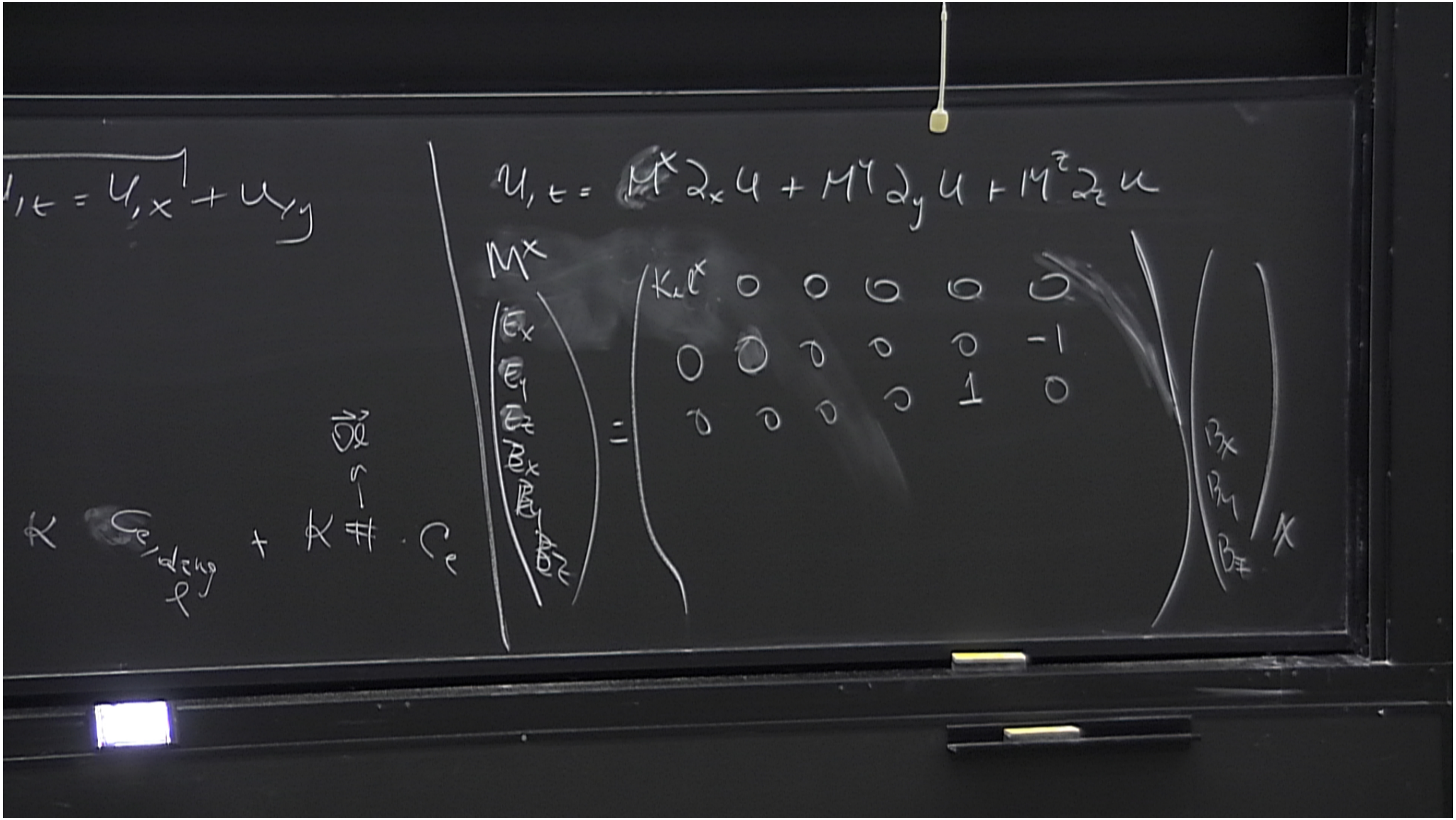
$$\partial_t C_e = \nabla(\vec{E}) = \vec{\nabla}(\nabla \times B + \kappa\, \vec{\chi}\, \nabla \cdot E)$$

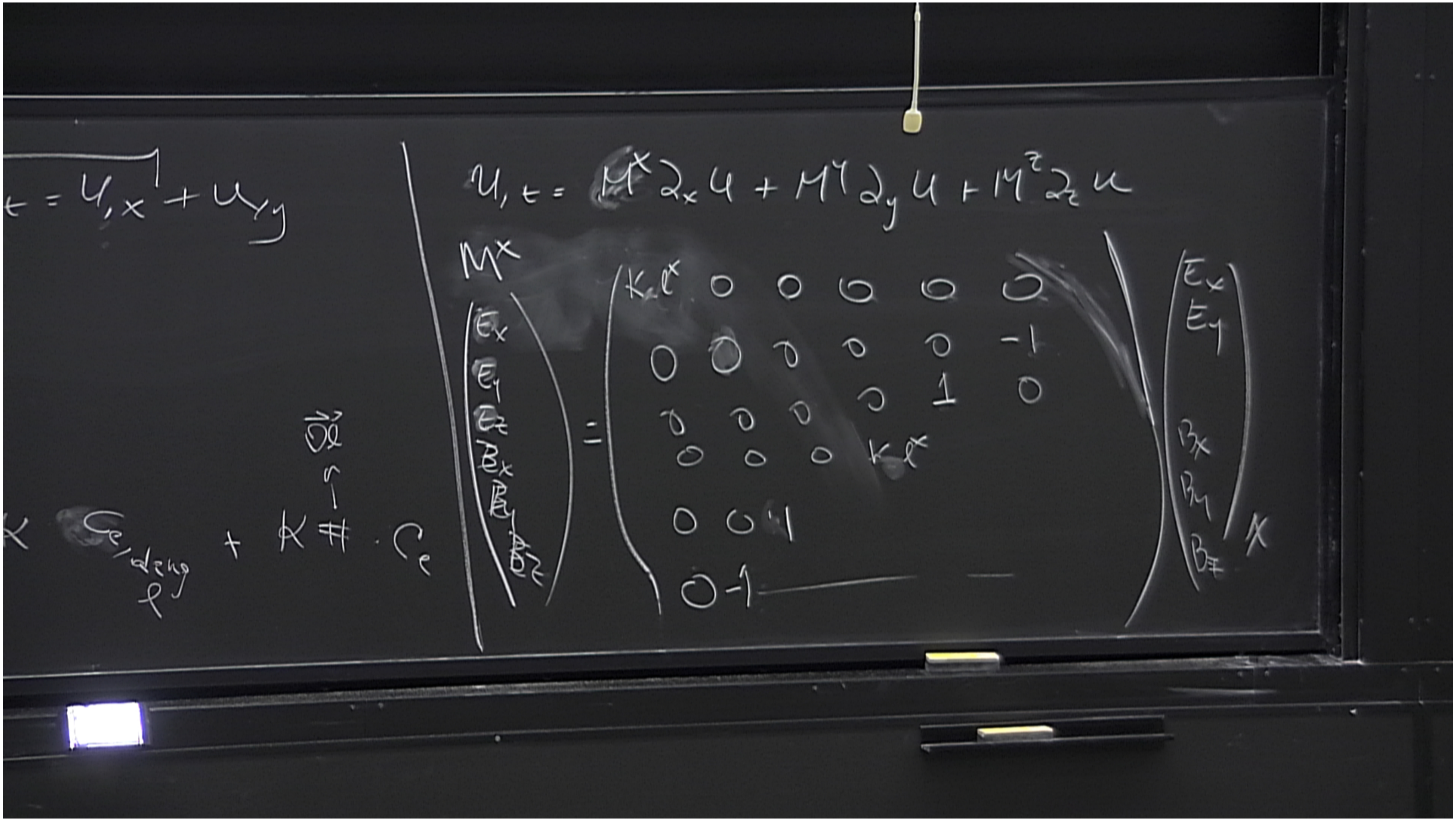
$$= \kappa(\vec{\nabla}\vec{\chi}) \cdot \nabla E + \kappa \ell^2 \partial_i (\vec{\nabla} \cdot \vec{E})$$

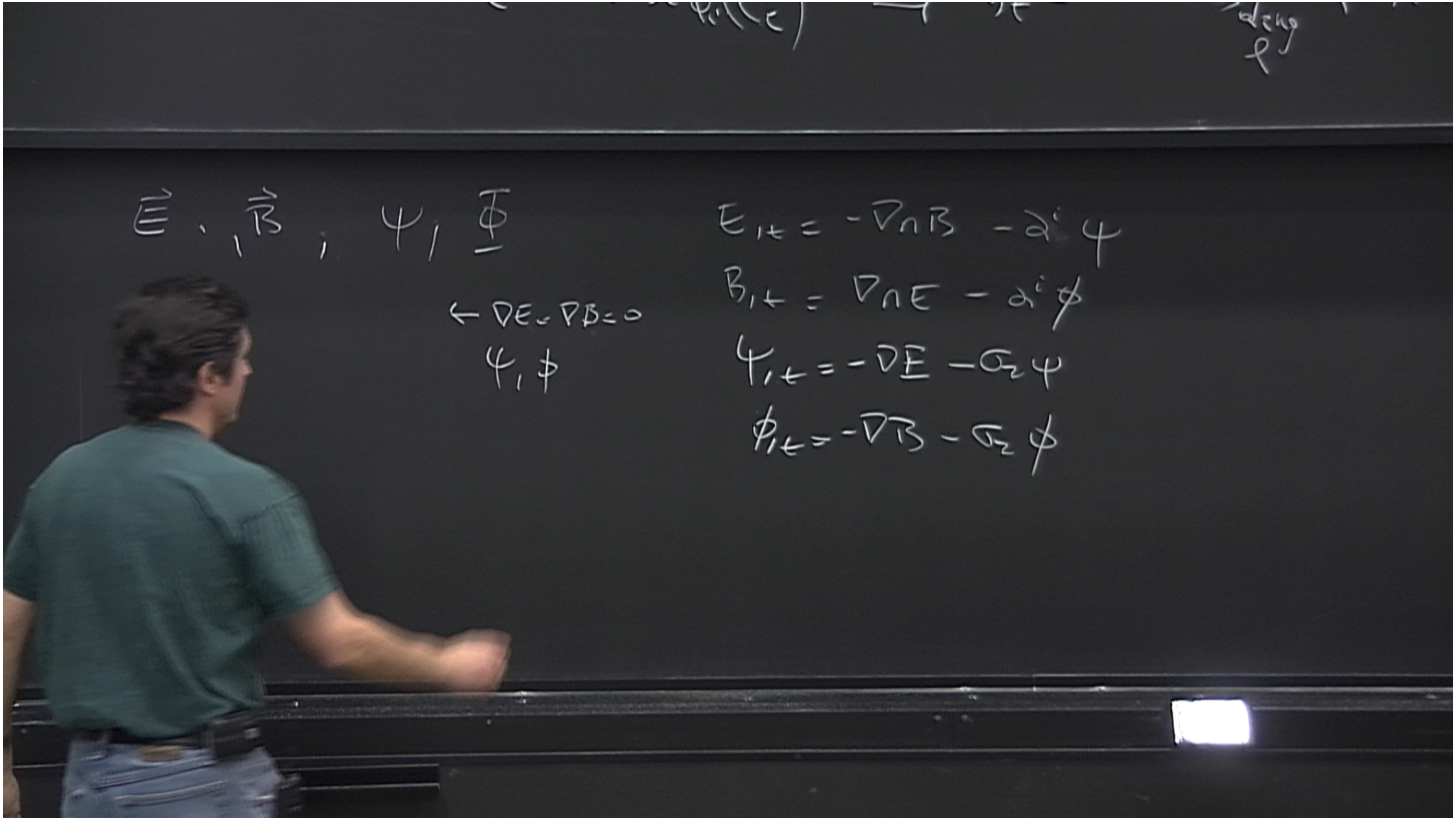
$$\partial_t C_e = \kappa(\vec{\nabla}\vec{\chi}) \cdot C_e + \kappa \cdot \ell^2 \partial_i(C_e) \rightarrow C_{e,t} = \kappa\, C_{e,\text{diag}} + \kappa \#$$

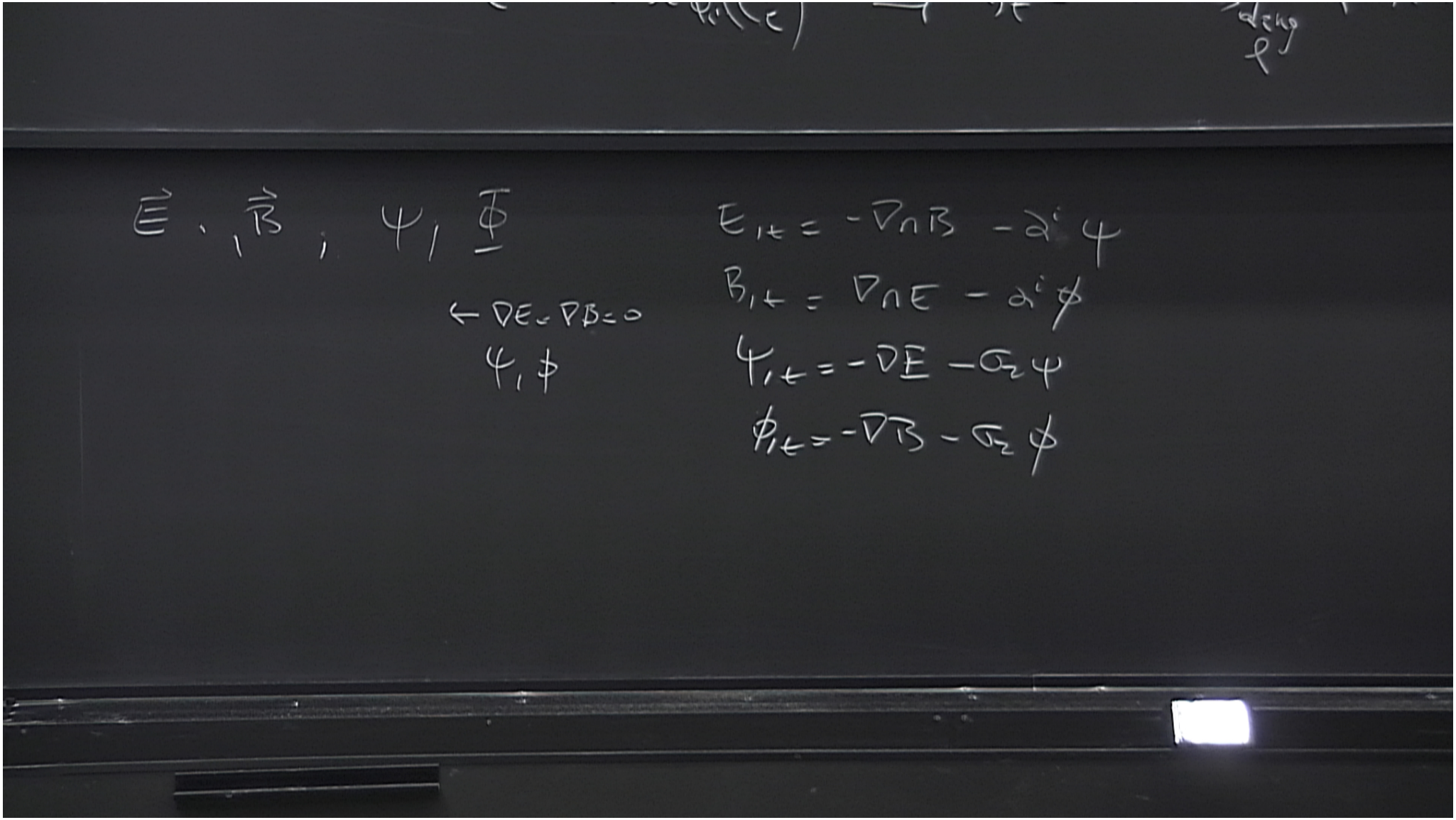

$u_{,t} = M^x \partial_x u + M^y \partial_y u + M^z \partial_z u$
K Gelding $+ K \# \cdot Ce$

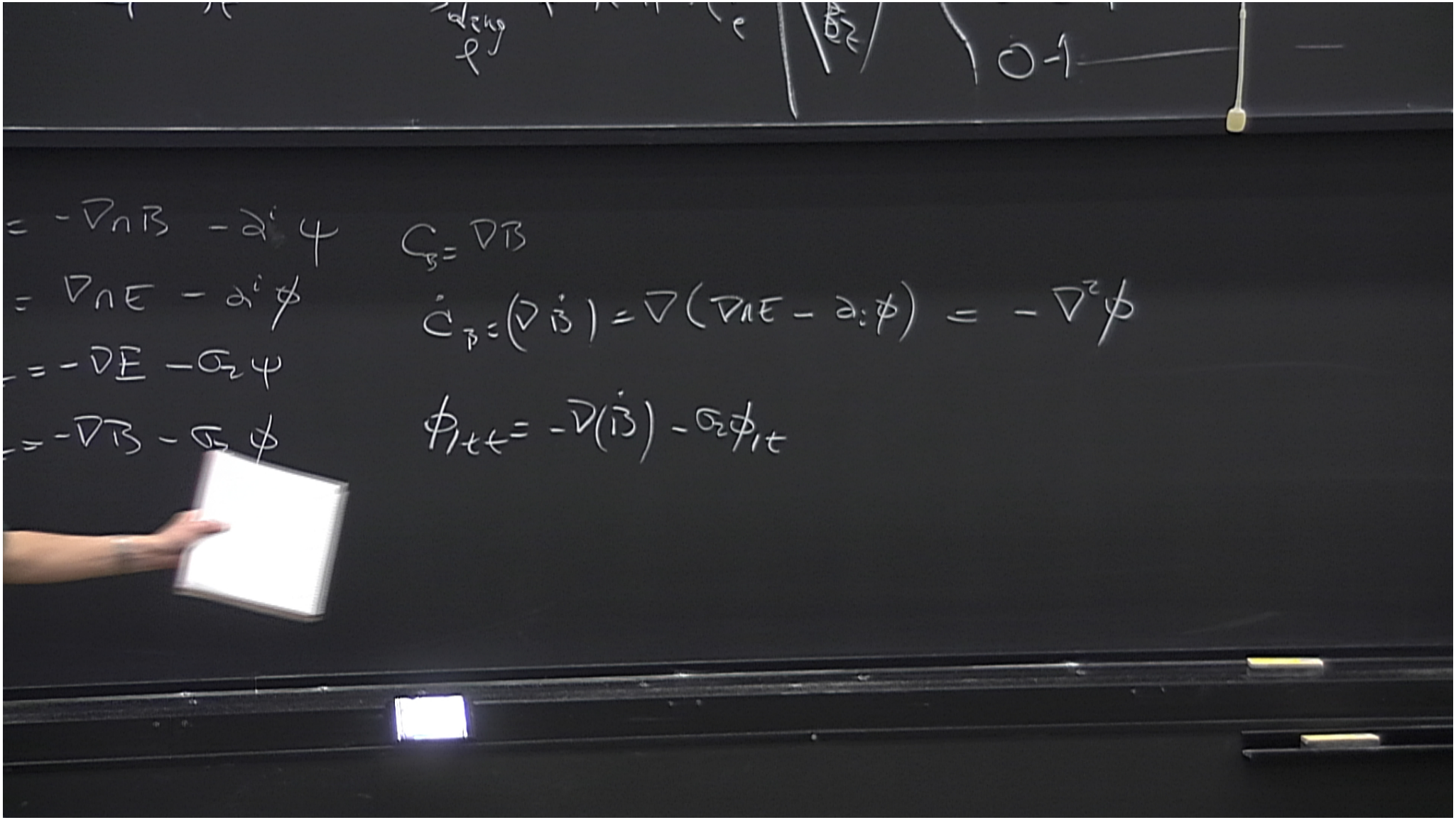

$u_{,t} = M^x \partial_x u + M^y \partial_y u + M^z \partial_z u$
K Gelding $+ K \vec{\partial} \cdot C_e$

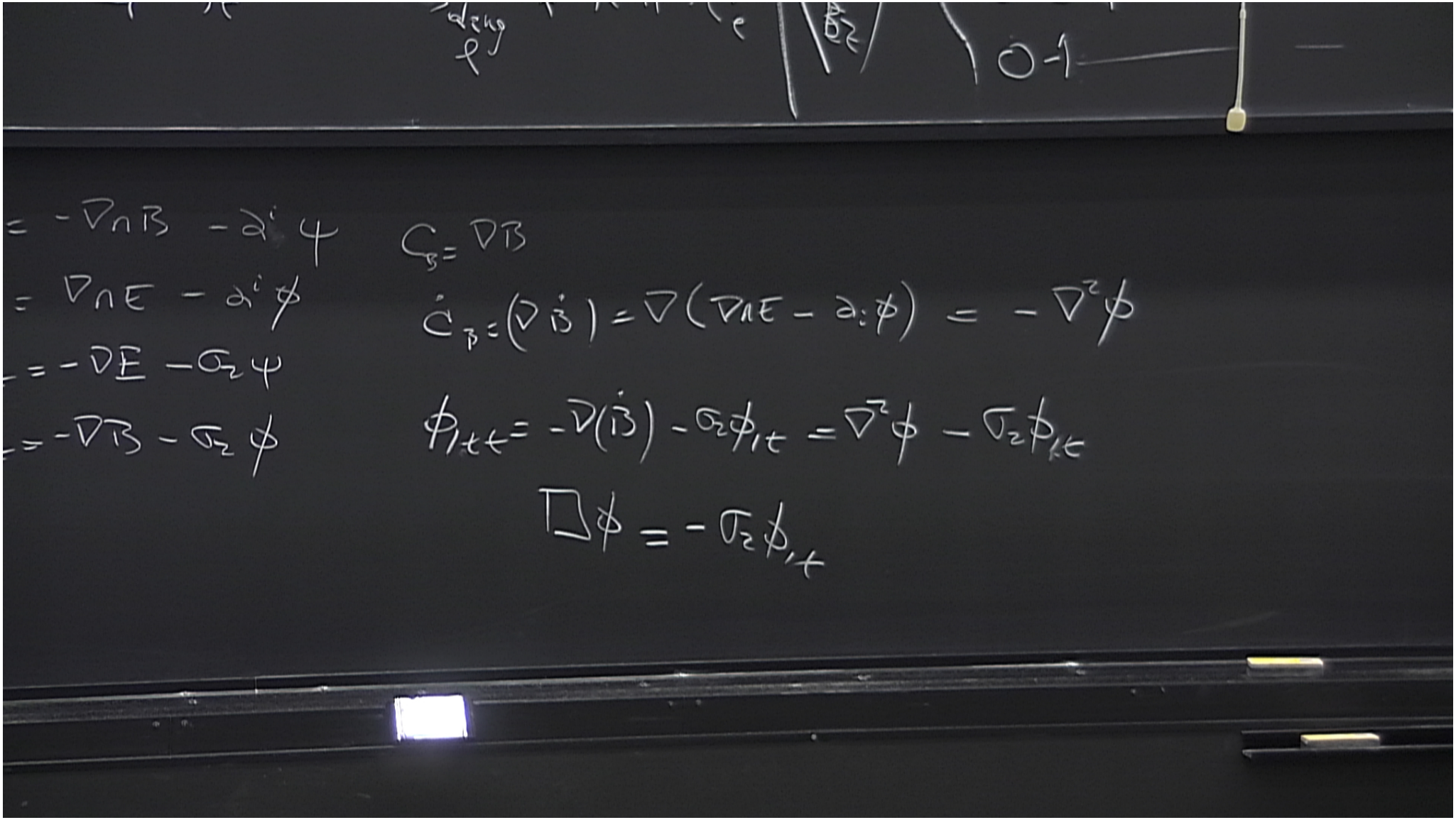

$u_{,t} = u_{,x} + u_{,y}$
$u_{,t} = M^x \partial_x u + M^y \partial_y u + M^z \partial_z u$
M^x
$\begin{pmatrix} E_x \\ E_y \\ E_z \\ B_x \\ B_y \\ B_z \end{pmatrix} = \begin{pmatrix} K_x c^x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} K$
$K \; \frac{c_{\text{leading}}}{\rho} + K \# \cdot c_e$
$\vec{\partial}$

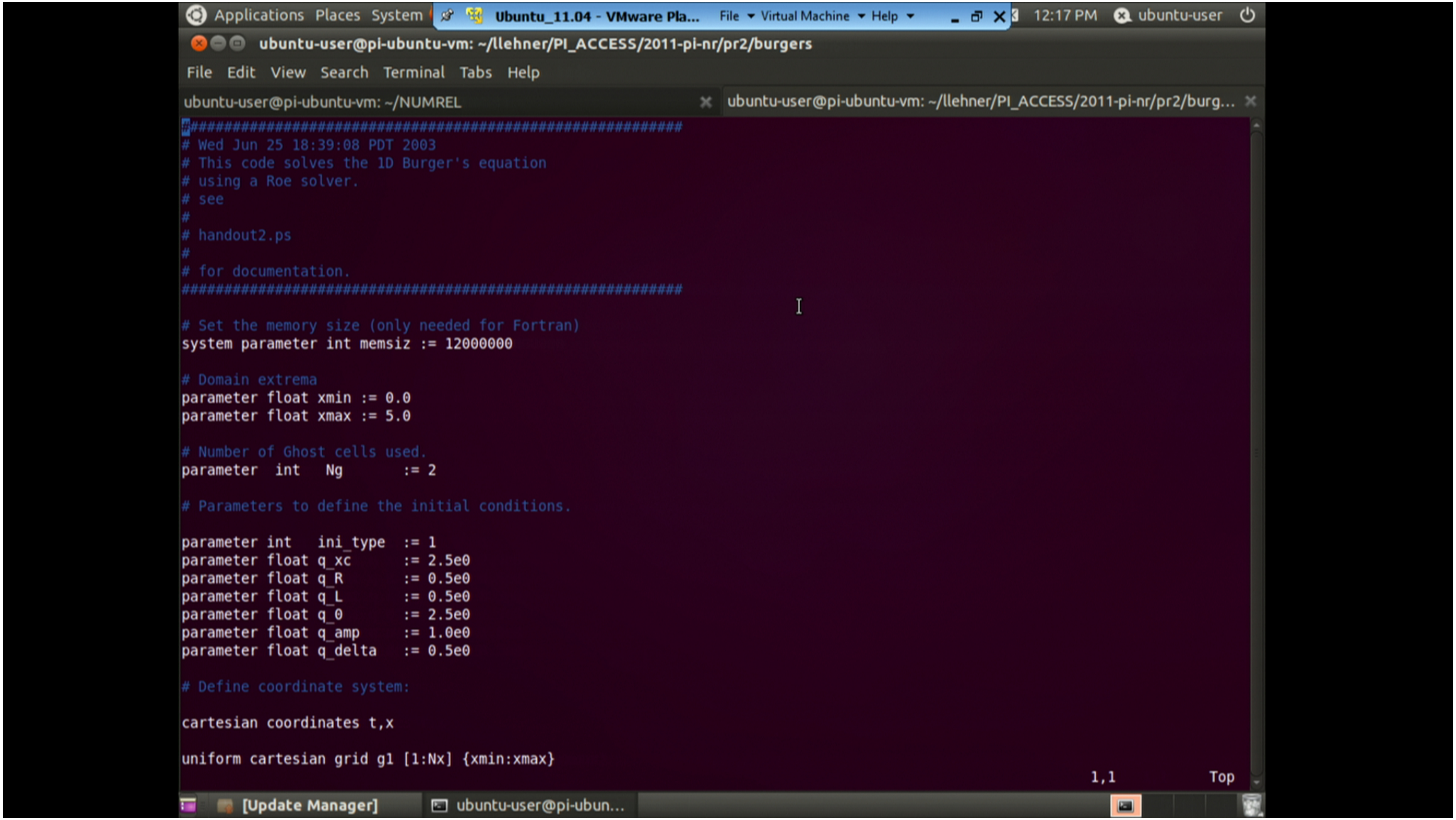


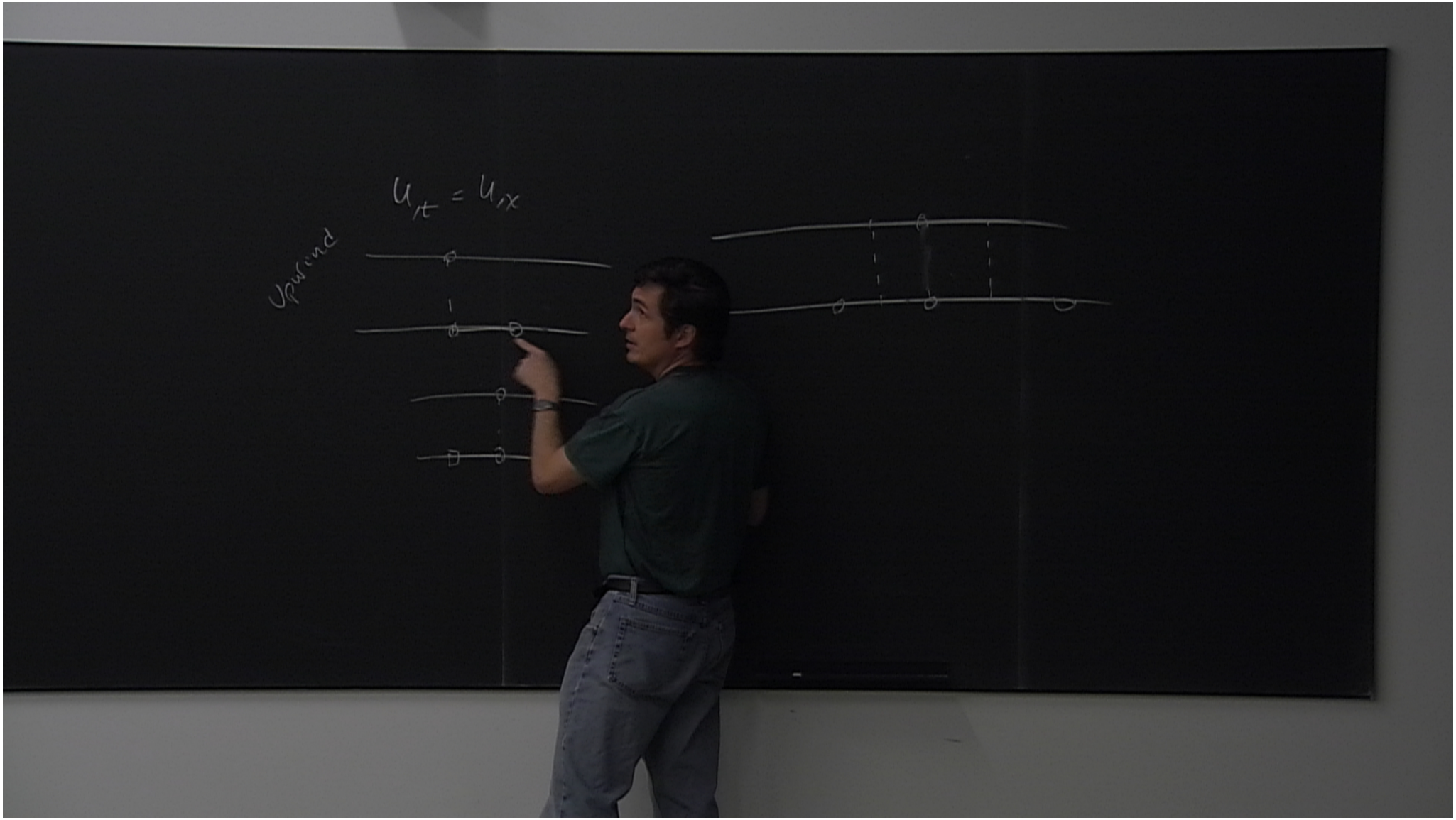

$\vec{E}$, $\vec{B}$, Ψ, Φ
$\leftarrow \nabla E - \nabla B = 0$
Ψ, ϕ
$E_{,t} = -\nabla \wedge B - a^i \Psi$
$B_{,t} = \nabla \wedge E - a^i \phi$
$\Psi_{,t} = -\nabla E - G_2 \Psi$
$\phi_{,t} = -\nabla B - G_2 \phi$

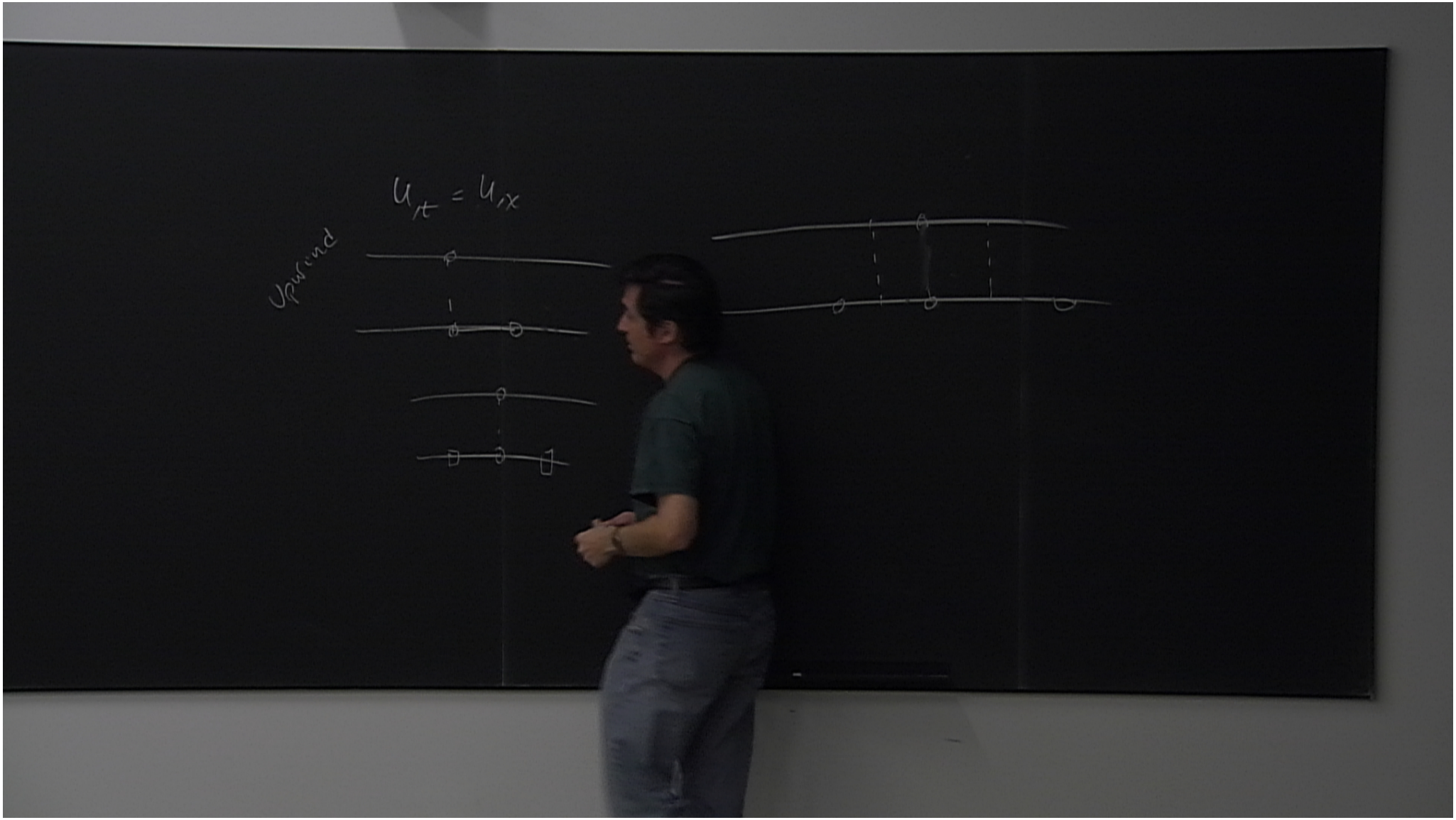

$\vec{E}, \vec{B}, \Psi, \vec{\Phi}$
$\leftarrow \nabla E = \nabla B = 0$
Ψ, Φ
$E_{,t} = -\nabla \times B - \partial^i \Psi$
$B_{,t} = \nabla \times E - \partial^i \Phi$
$\Psi_{,t} = -\nabla E - G_2 \Psi$
$\Phi_{,t} = -\nabla B - G_2 \Phi$

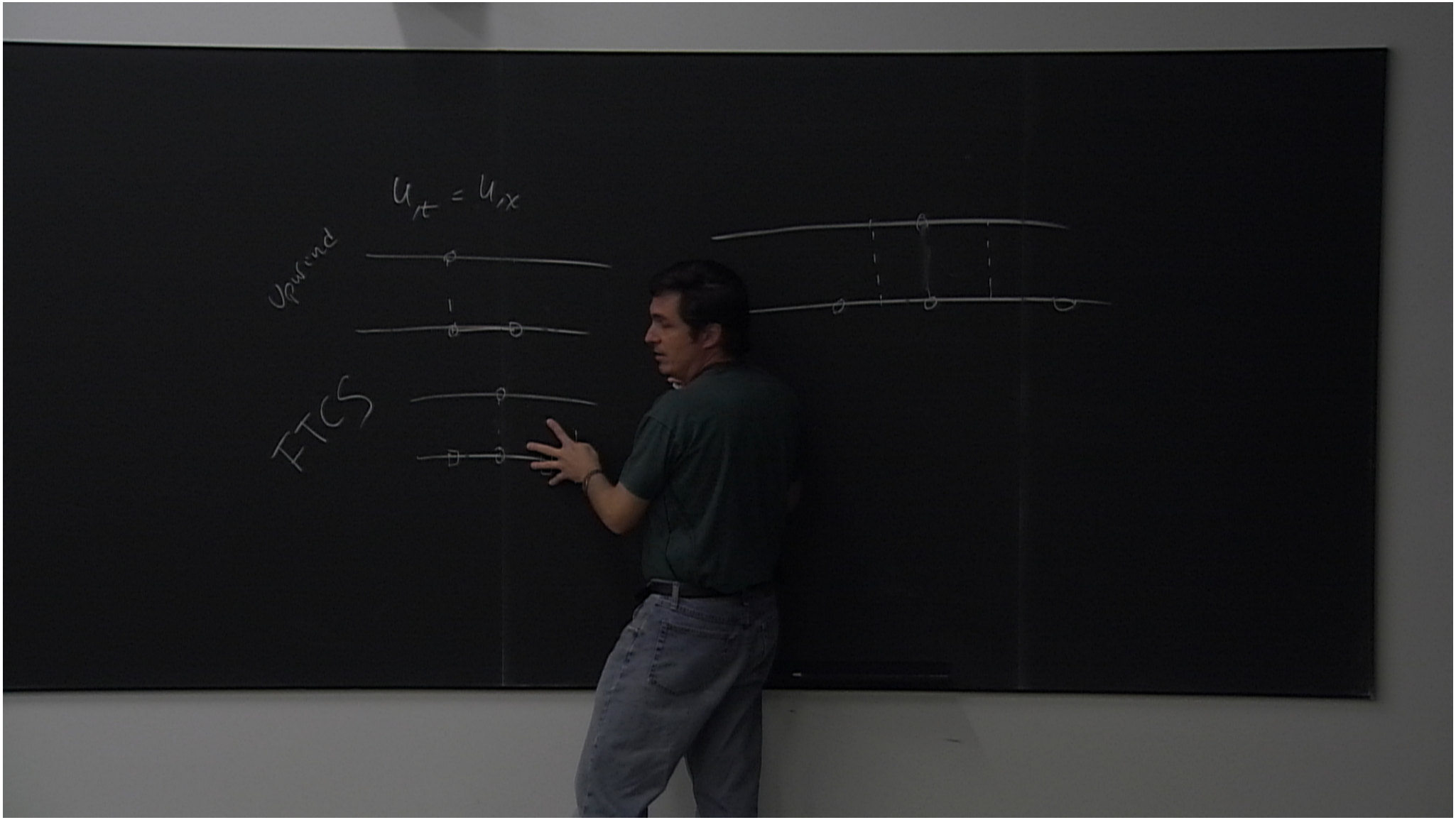

$= -\nabla \cdot B - \dot{a}\psi$
$= \nabla \cdot E - a\dot{\psi}$
$= -\nabla E - G_2 \psi$
$= -\nabla B - G_2 \phi$
$C_3 = \nabla B$
$\dot{C}_3 = (\nabla \dot{B}) = \nabla(\nabla \cdot E - a\dot{\phi}) = -\nabla^2 \phi$
$\phi_{tt} = -\nabla(\dot{B}) - G_2 \phi_t$

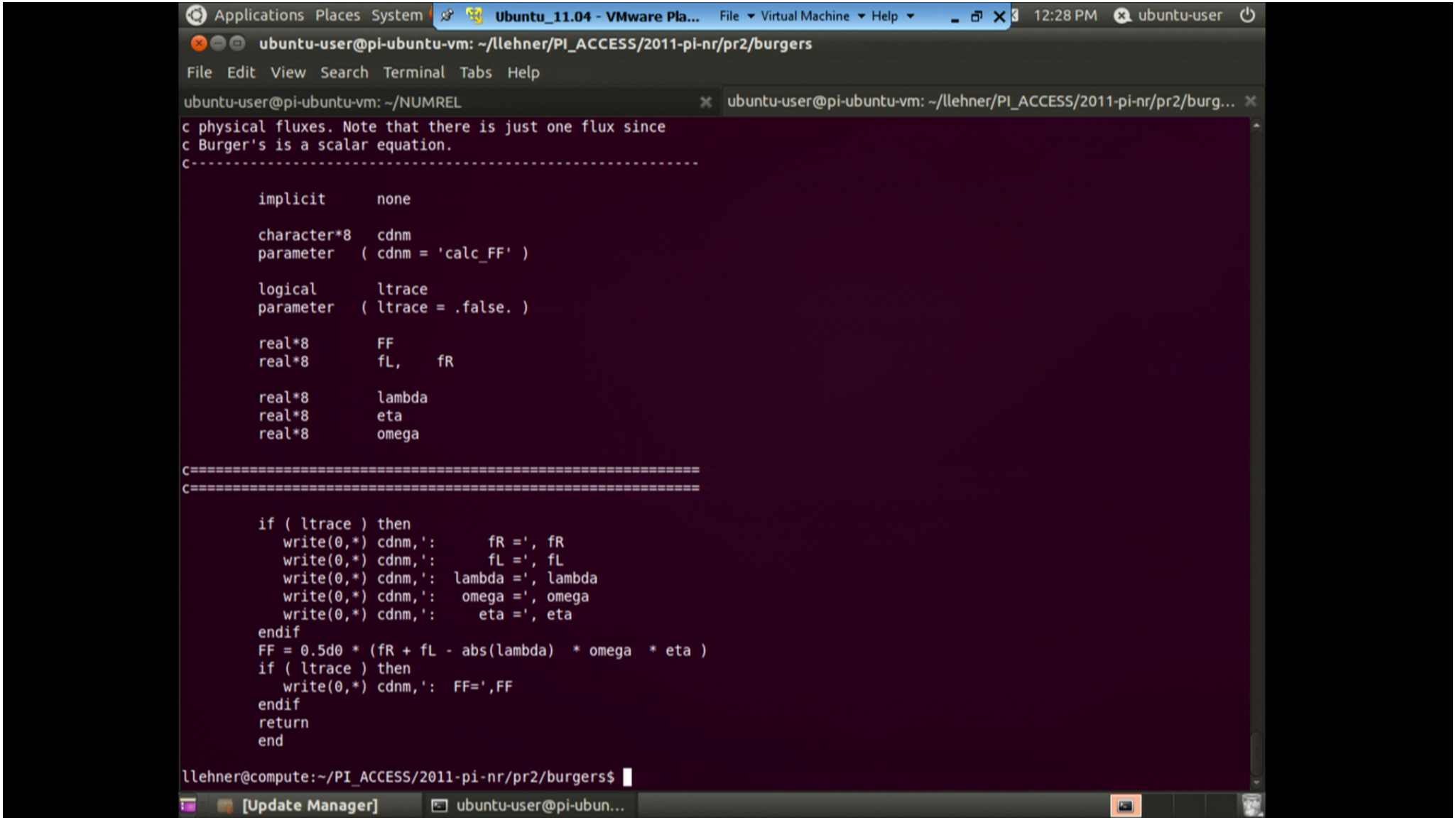

$= -\nabla\cdot B - a^i\psi$
$= \nabla\cdot E - a^i\phi$
$= -\nabla E - \sigma_2\psi$
$= -\nabla B - \sigma_2\phi$
$C_3 = \nabla B$
$\dot{C}_3 = (\nabla\dot{B}) = \nabla(\nabla\cdot E - a_i\phi) = -\nabla^2\phi$
$\phi_{,tt} = -\nabla(\dot{B}) - \sigma_2\phi_{,t} = \nabla^2\phi - \sigma_2\phi_{,t}$
$\Box\phi = -\sigma_2\phi_{,t}$

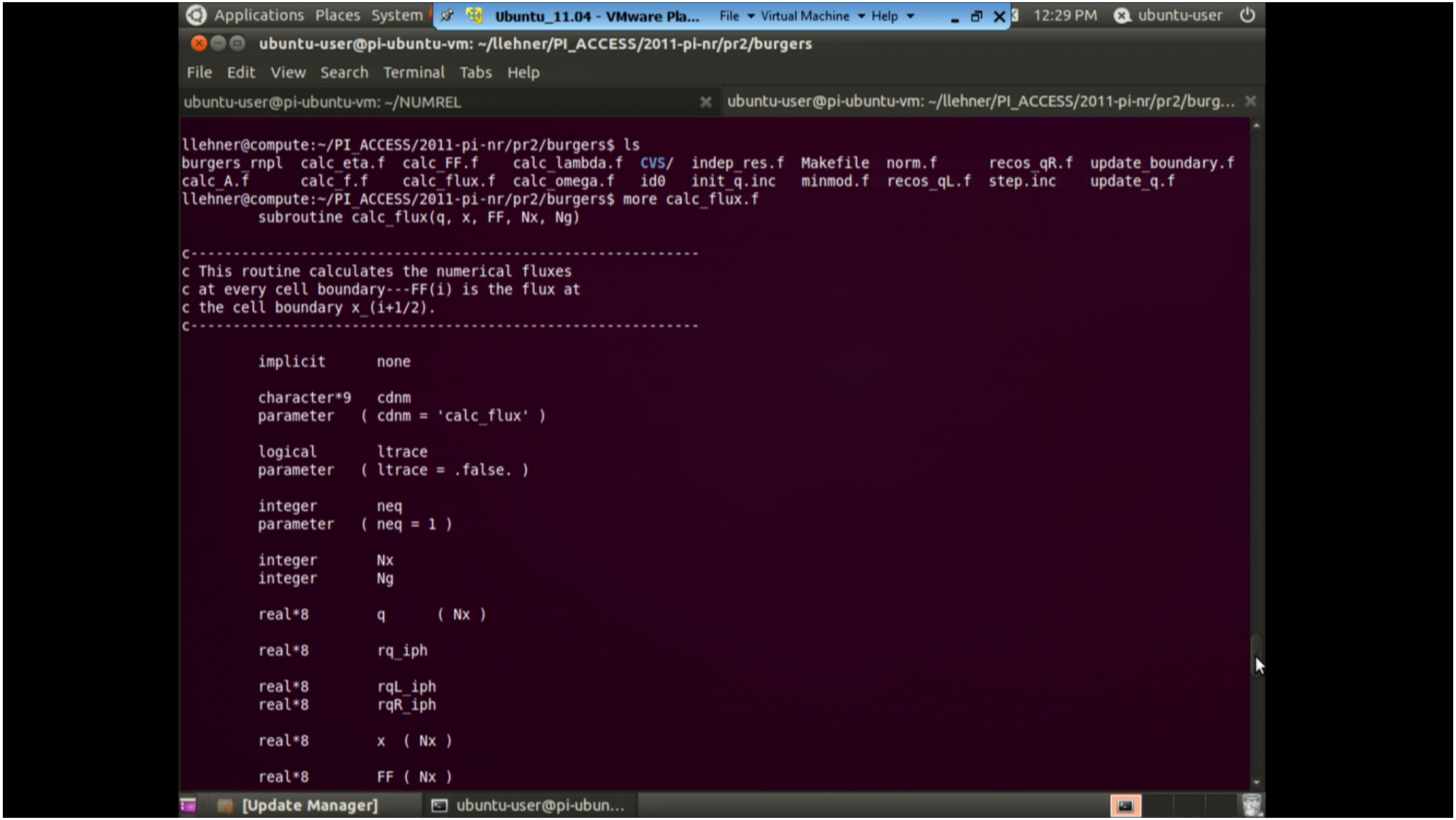

Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:17 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
##
Wed Jun 25 18:39:08 PDT 2003
This code solves the 1D Burger's equation
using a Roe solver.
see
#
handout2.ps
#
for documentation.
##
Set the memory size (only needed for Fortran)
system parameter int memsiz := 12000000
Domain extrema
parameter float xmin := 0.0
parameter float xmax := 5.0
Number of Ghost cells used.
parameter int Ng := 2
Parameters to define the initial conditions.
parameter int ini_type := 1
parameter float q_xc := 2.5e0
parameter float q_R := 0.5e0
parameter float q_L := 0.5e0
parameter float q_0 := 2.5e0
parameter float q_amp := 1.0e0
parameter float q_delta := 0.5e0
Define coordinate system:
cartesian coordinates t,x
uniform cartesian grid g1 [1:Nx] {xmin:xmax}
1,1 Top
[Update Manager]
ubuntu-user@pi-ubun...

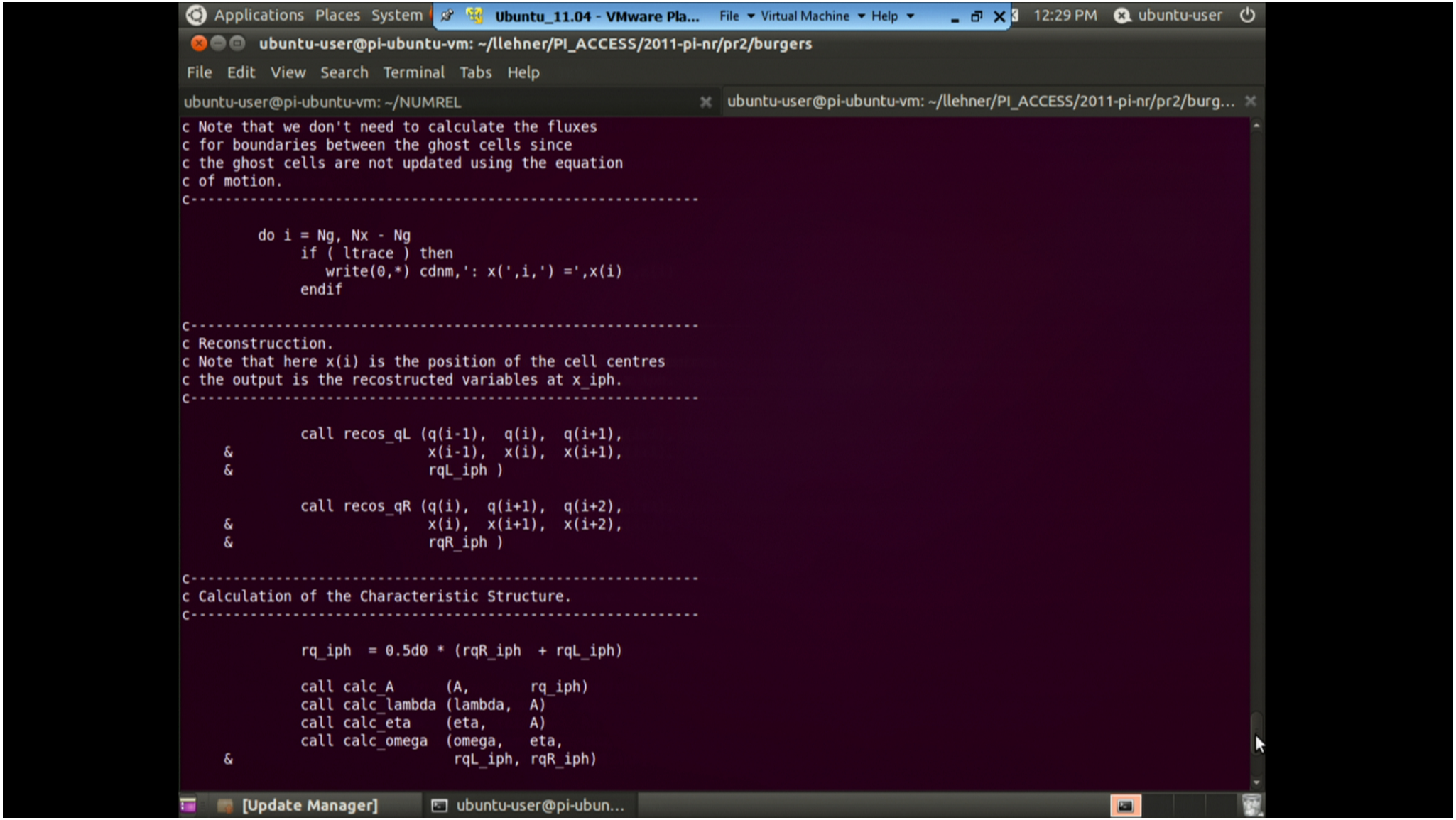

$u_t = u_x$
Upwind

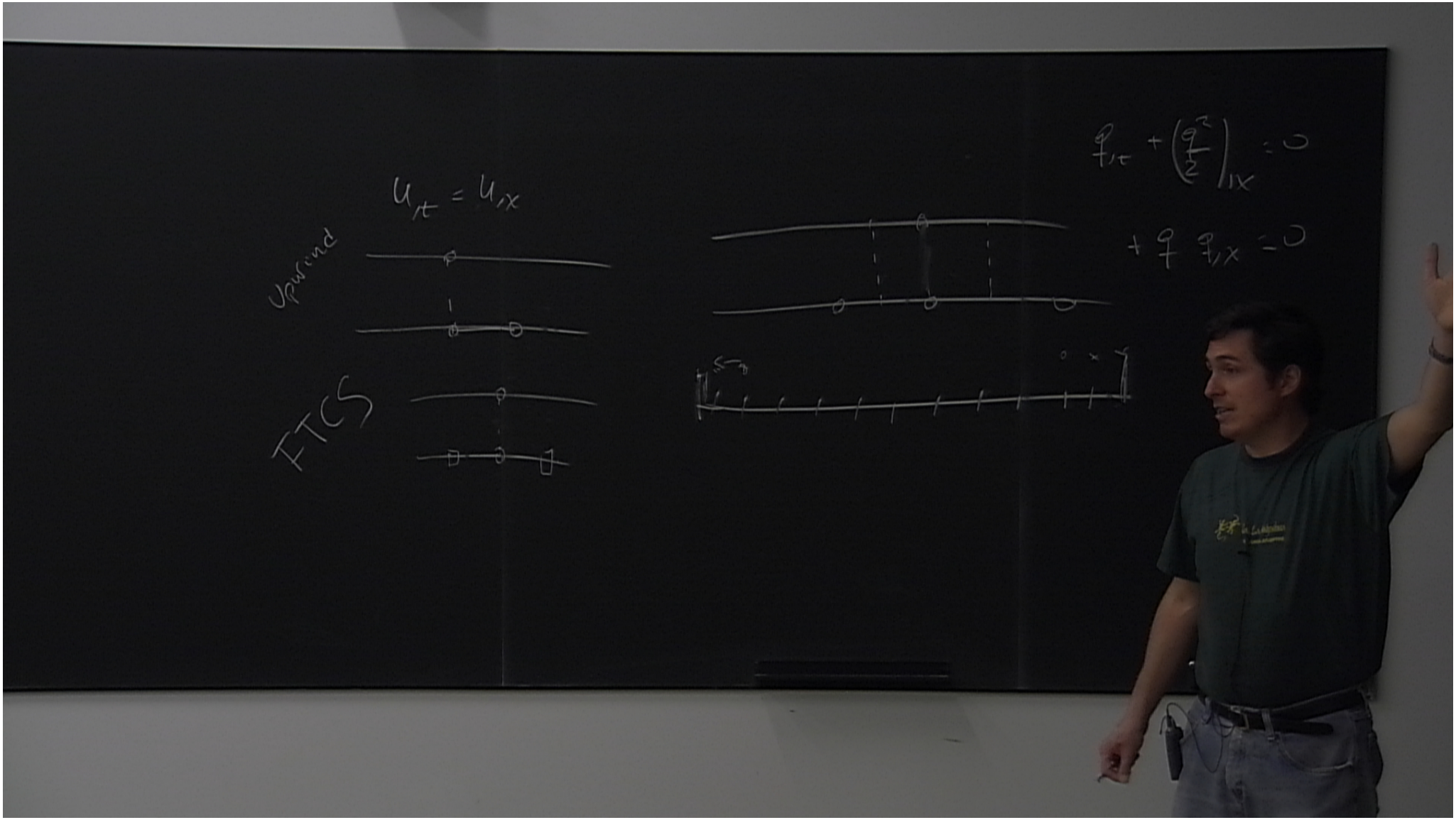

$u_t = u_x$
upwind

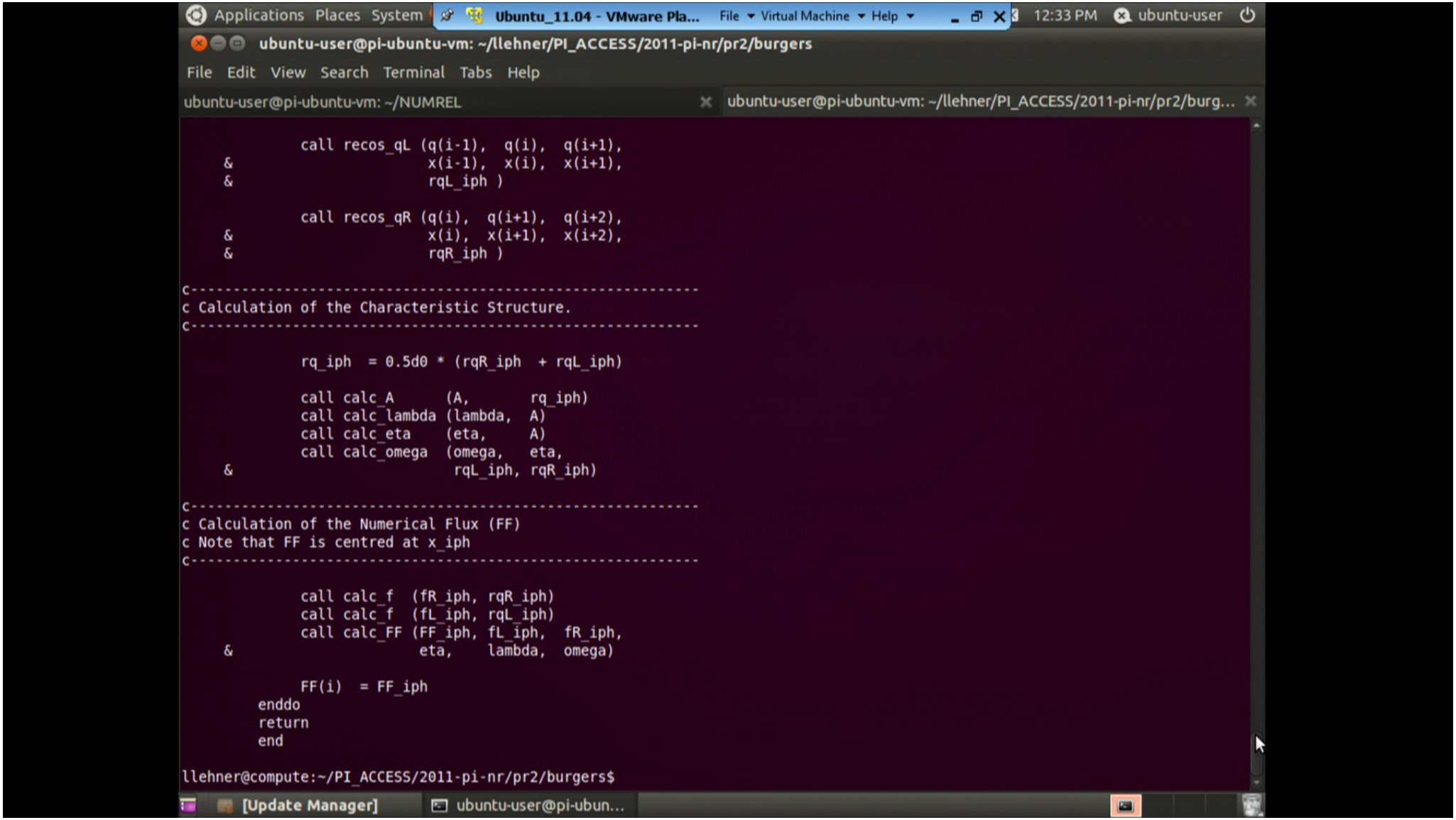

$u_t = u_x$
upwind
FTCS

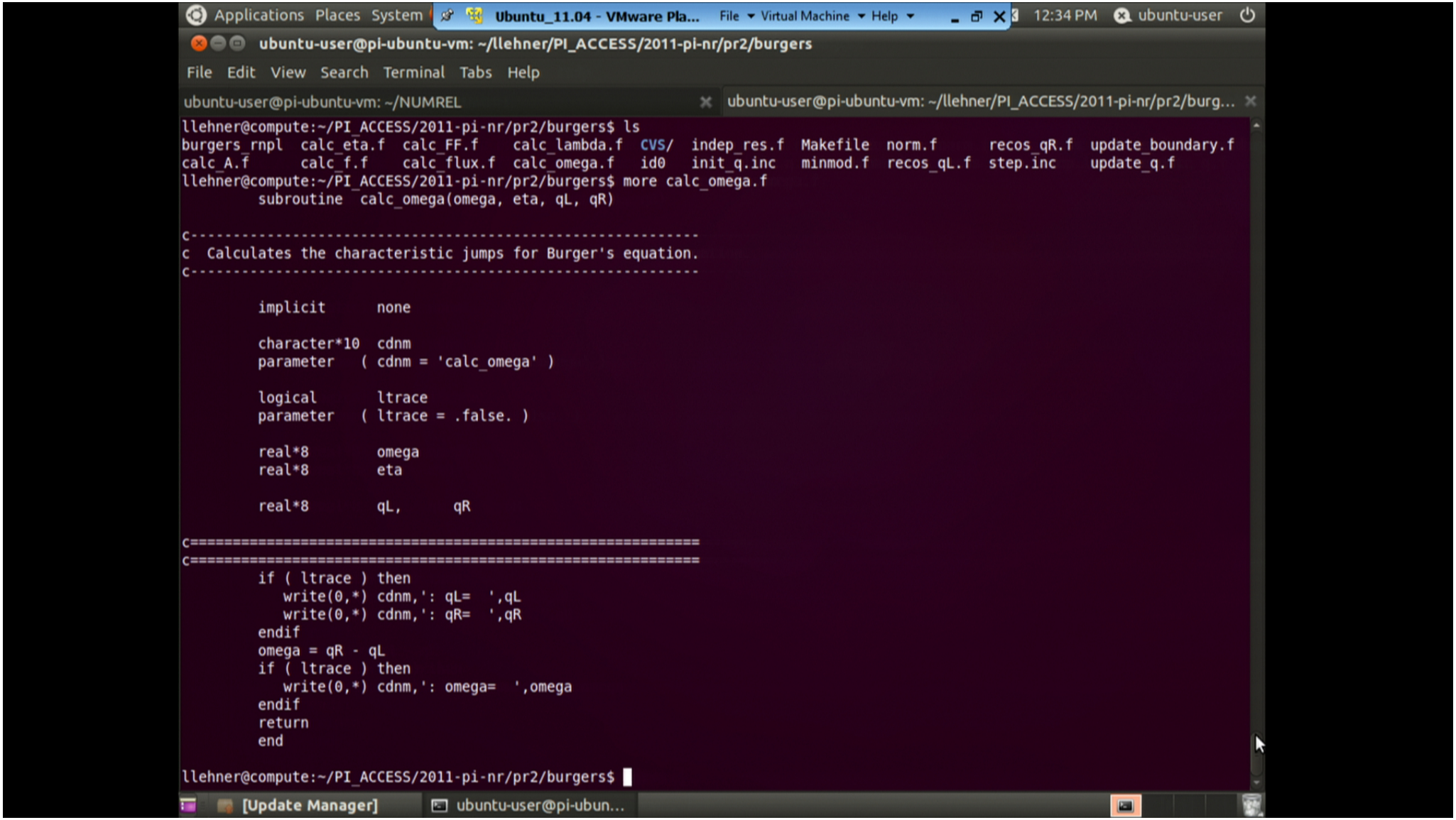

Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:28 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
c physical fluxes. Note that there is just one flux since
c Burger's is a scalar equation.
c---
implicit none
character*8 cdnm
parameter (cdnm = 'calc_FF')
logical ltrace
parameter (ltrace = .false.)
real*8 FF
real*8 fL, fR
real*8 lambda
real*8 eta
real*8 omega
c===
c===
if (ltrace) then
write(0,*) cdnm,': fR =', fR
write(0,*) cdnm,': fL =', fL
write(0,*) cdnm,': lambda =', lambda
write(0,*) cdnm,': omega =', omega
write(0,*) cdnm,': eta =', eta
endif
FF = 0.5d0 * (fR + fL - abs(lambda) * omega * eta)
if (ltrace) then
write(0,*) cdnm,': FF=',FF
endif
return
end
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$
[Update Manager]
ubuntu-user@pi-ubun...

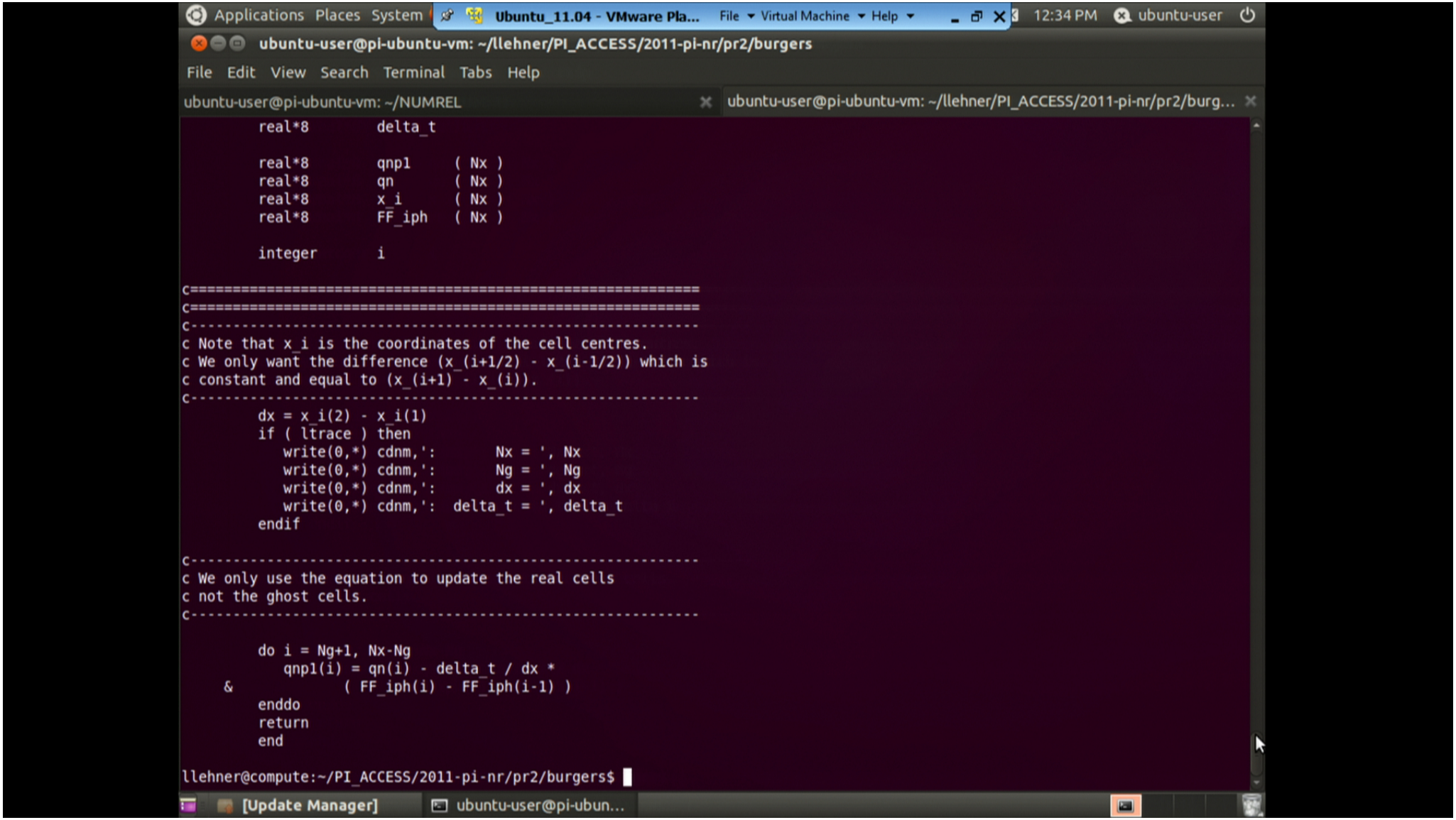

Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:29 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$ ls
burgers_rnpl calc_eta.f calc_FF.f calc_lambda.f CVS/ indep_res.f Makefile norm.f recos_qR.f update_boundary.f
calc_A.f calc_f.f calc_flux.f calc_omega.f id0 init_q.inc minmod.f recos_qL.f step.inc update_q.f
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$ more calc_flux.f
subroutine calc_flux(q, x, FF, Nx, Ng)
c---
c This routine calculates the numerical fluxes
c at every cell boundary---FF(i) is the flux at
c the cell boundary x_(i+1/2).
c---
implicit none
character*9 cdnm
parameter (cdnm = 'calc_flux')
logical ltrace
parameter (ltrace = .false.)
integer neq
parameter (neq = 1)
integer Nx
integer Ng
real*8 q (Nx)
real*8 rq_iph
real*8 rqL_iph
real*8 rqR_iph
real*8 x (Nx)
real*8 FF (Nx)
[Update Manager]
ubuntu-user@pi-ubun...

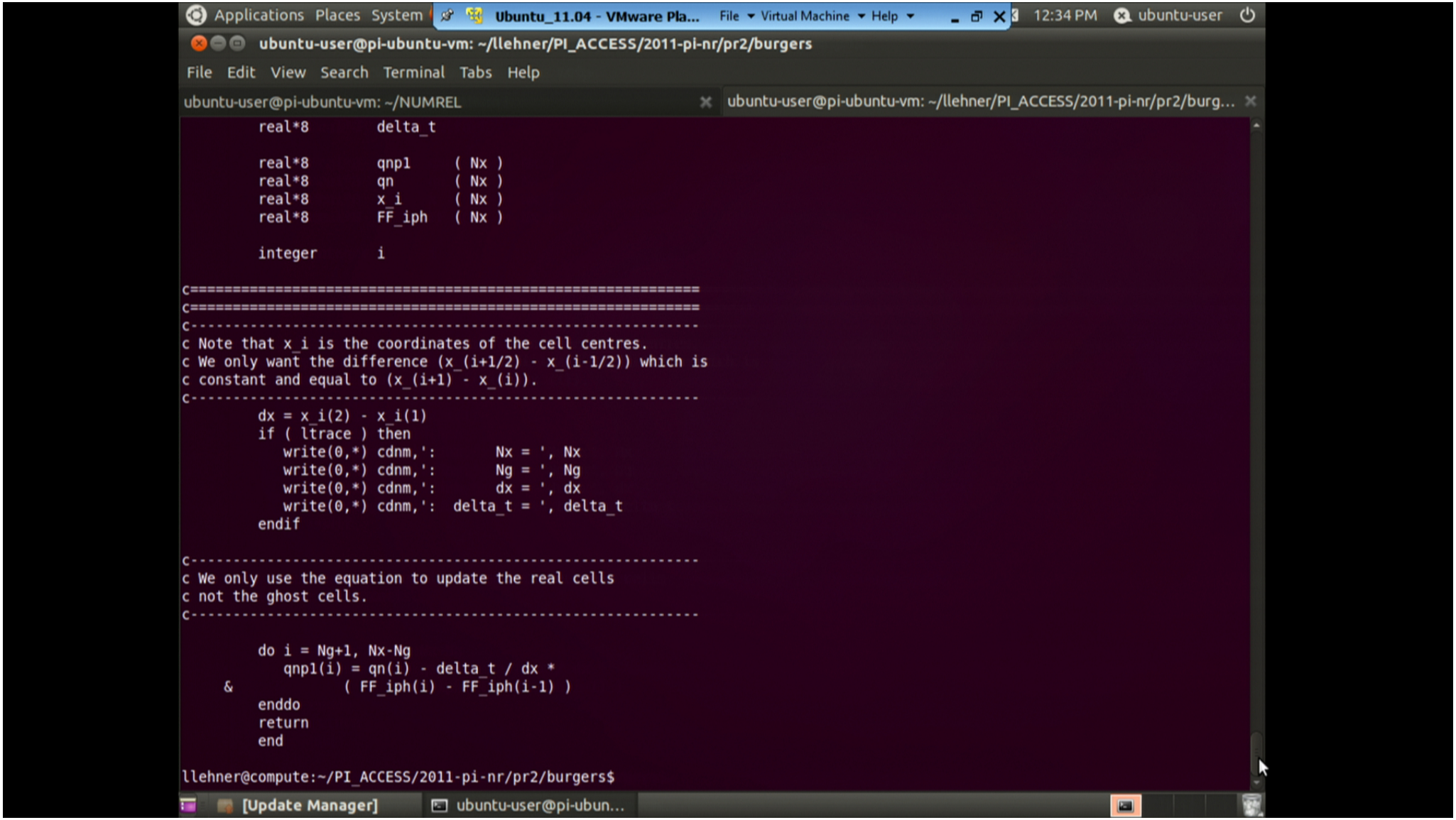

Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:29 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
c Note that we don't need to calculate the fluxes
c for boundaries between the ghost cells since
c the ghost cells are not updated using the equation
c of motion.
c---
do i = Ng, Nx - Ng
if (ltrace) then
write(0,*) cdnm,': x(',i,') =',x(i)
endif
c---
c Reconstrucction.
c Note that here x(i) is the position of the cell centres
c the output is the recostructed variables at x_iph.
c---
call recos_qL (q(i-1), q(i), q(i+1),
& x(i-1), x(i), x(i+1),
& rqL_iph)
call recos_qR (q(i), q(i+1), q(i+2),
& x(i), x(i+1), x(i+2),
& rqR_iph)
c---
c Calculation of the Characteristic Structure.
c---
rq_iph = 0.5d0 * (rqR_iph + rqL_iph)
call calc_A (A, rq_iph)
call calc_lambda (lambda, A)
call calc_eta (eta, A)
call calc_omega (omega, eta,
& rqL_iph, rqR_iph)
[Update Manager]
ubuntu-user@pi-ubun...

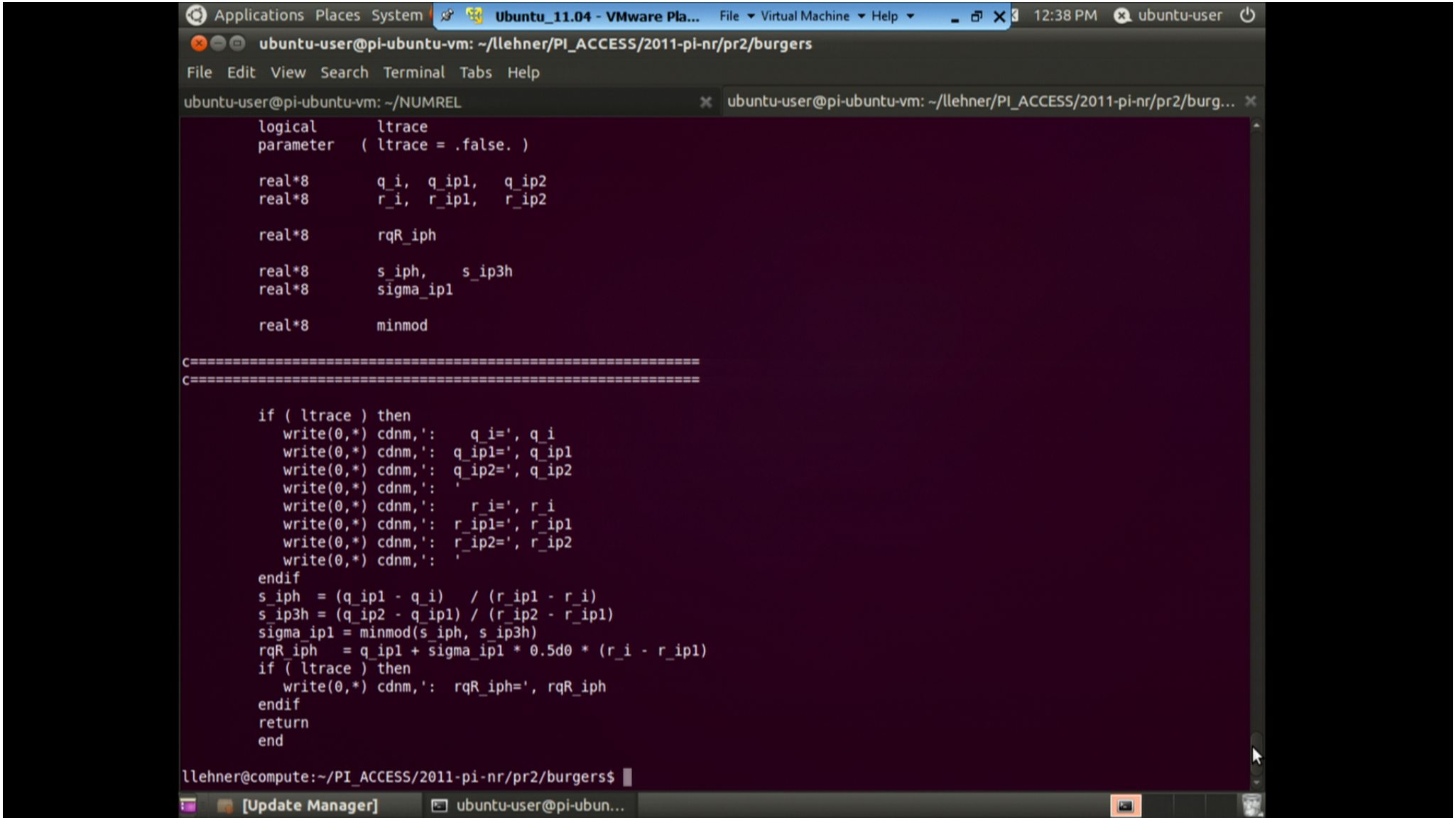

$u_t = u_x$
upwind
FTCS
$q_t + \left(\frac{q^2}{2}\right)_x = 0$
$+ q\, q_x = 0$


Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:33 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
call recos_qL (q(i-1), q(i), q(i+1),
& x(i-1), x(i), x(i+1),
& rqL_iph)
call recos_qR (q(i), q(i+1), q(i+2),
& x(i), x(i+1), x(i+2),
& rqR_iph)
c---
c Calculation of the Characteristic Structure.
c---
rq_iph = 0.5d0 * (rqR_iph + rqL_iph)
call calc_A (A, rq_iph)
call calc_lambda (lambda, A)
call calc_eta (eta, A)
call calc_omega (omega, eta,
& rqL_iph, rqR_iph)
c---
c Calculation of the Numerical Flux (FF)
c Note that FF is centred at x_iph
c---
call calc_f (fR_iph, rqR_iph)
call calc_f (fL_iph, rqL_iph)
call calc_FF (FF_iph, fL_iph, fR_iph,
& eta, lambda, omega)
FF(i) = FF_iph
enddo
return
end
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$
[Update Manager]
ubuntu-user@pi-ubun...


Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:34 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$ ls
burgers_rnpl calc_eta.f calc_FF.f calc_lambda.f CVS/ indep_res.f Makefile norm.f recos_qR.f update_boundary.f
calc_A.f calc_f.f calc_flux.f calc_omega.f id0 init_q.inc minmod.f recos_qL.f step.inc update_q.f
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$ more calc_omega.f
subroutine calc_omega(omega, eta, qL, qR)
c---
c Calculates the characteristic jumps for Burger's equation.
c---
implicit none
character*10 cdnm
parameter (cdnm = 'calc_omega')
logical ltrace
parameter (ltrace = .false.)
real*8 omega
real*8 eta
real*8 qL, qR
c===
c===
if (ltrace) then
write(0,*) cdnm,': qL= ',qL
write(0,*) cdnm,': qR= ',qR
endif
omega = qR - qL
if (ltrace) then
write(0,*) cdnm,': omega= ',omega
endif
return
end
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$
[Update Manager]
ubuntu-user@pi-ubun...


Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:34 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
real*8 delta_t
real*8 qnp1 (Nx)
real*8 qn (Nx)
real*8 x_i (Nx)
real*8 FF_iph (Nx)
integer i
c Note that x_i is the coordinates of the cell centres.
c We only want the difference (x_(i+1/2) - x_(i-1/2)) which is
c constant and equal to (x_(i+1) - x_(i)).
dx = x_i(2) - x_i(1)
if (ltrace) then
write(0,*) cdnm,': Nx = ', Nx
write(0,*) cdnm,': Ng = ', Ng
write(0,*) cdnm,': dx = ', dx
write(0,*) cdnm,': delta_t = ', delta_t
endif
c We only use the equation to update the real cells
c not the ghost cells.
do i = Ng+1, Nx-Ng
qnp1(i) = qn(i) - delta_t / dx *
& (FF_iph(i) - FF_iph(i-1))
enddo
return
end
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$
[Update Manager]
ubuntu-user@pi-ubun...


Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:34 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
real*8 delta_t
real*8 qnp1 (Nx)
real*8 qn (Nx)
real*8 x_i (Nx)
real*8 FF_iph (Nx)
integer i
c Note that x_i is the coordinates of the cell centres.
c We only want the difference (x_(i+1/2) - x_(i-1/2)) which is
c constant and equal to (x_(i+1) - x_(i)).
dx = x_i(2) - x_i(1)
if (ltrace) then
write(0,*) cdnm,': Nx = ', Nx
write(0,*) cdnm,': Ng = ', Ng
write(0,*) cdnm,': dx = ', dx
write(0,*) cdnm,': delta_t = ', delta_t
endif
c We only use the equation to update the real cells
c not the ghost cells.
do i = Ng+1, Nx-Ng
qnp1(i) = qn(i) - delta_t / dx *
& (FF_iph(i) - FF_iph(i-1))
enddo
return
end
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$
[Update Manager]
ubuntu-user@pi-ubun...


Applications Places System
Ubuntu_11.04 - VMware Pla...
File Virtual Machine Help
12:38 PM
ubuntu-user
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burgers
File Edit View Search Terminal Tabs Help
ubuntu-user@pi-ubuntu-vm: ~/NUMREL
ubuntu-user@pi-ubuntu-vm: ~/llehner/PI_ACCESS/2011-pi-nr/pr2/burg...
logical ltrace
parameter (ltrace = .false.)
real*8 q_i, q_ip1, q_ip2
real*8 r_i, r_ip1, r_ip2
real*8 rqR_iph
real*8 s_iph, s_ip3h
real*8 sigma_ip1
real*8 minmod
c===
c===
if (ltrace) then
write(0,*) cdnm,': q_i=', q_i
write(0,*) cdnm,': q_ip1=', q_ip1
write(0,*) cdnm,': q_ip2=', q_ip2
write(0,*) cdnm,': '
write(0,*) cdnm,': r_i=', r_i
write(0,*) cdnm,': r_ip1=', r_ip1
write(0,*) cdnm,': r_ip2=', r_ip2
write(0,*) cdnm,': '
endif
s_iph = (q_ip1 - q_i) / (r_ip1 - r_i)
s_ip3h = (q_ip2 - q_ip1) / (r_ip2 - r_ip1)
sigma_ip1 = minmod(s_iph, s_ip3h)
rqR_iph = q_ip1 + sigma_ip1 * 0.5d0 * (r_i - r_ip1)
if (ltrace) then
write(0,*) cdnm,': rqR_iph=', rqR_iph
endif
return
end
llehner@compute:~/PI_ACCESS/2011-pi-nr/pr2/burgers$
[Update Manager]
ubuntu-user@pi-ubun...