Title: Numerical Methods Lecture - 230202

Speakers: Erik Schnetter

Collection: Numerical Methods (2022/2023)

Date: February 02, 2023 - 9:15 AM

URL: https://pirsa.org/23020000

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help Charged Thu Feb 2 09:19

symmetry.pi.local

JupyterLab Start Page

File Edit View Run Kernel Git Tabs Settings Help Mem:1.08 GB

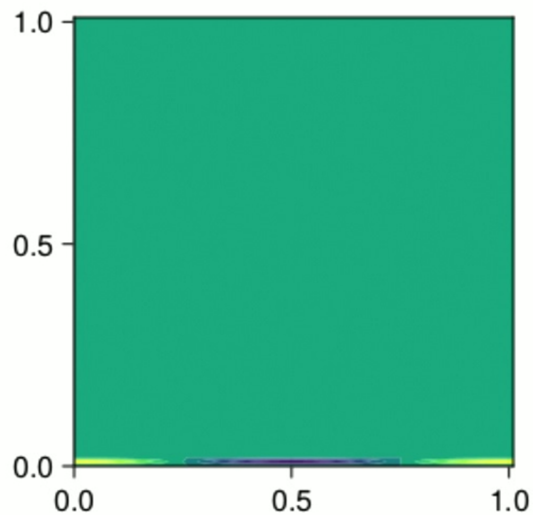
Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Code git Julia 1.8.5

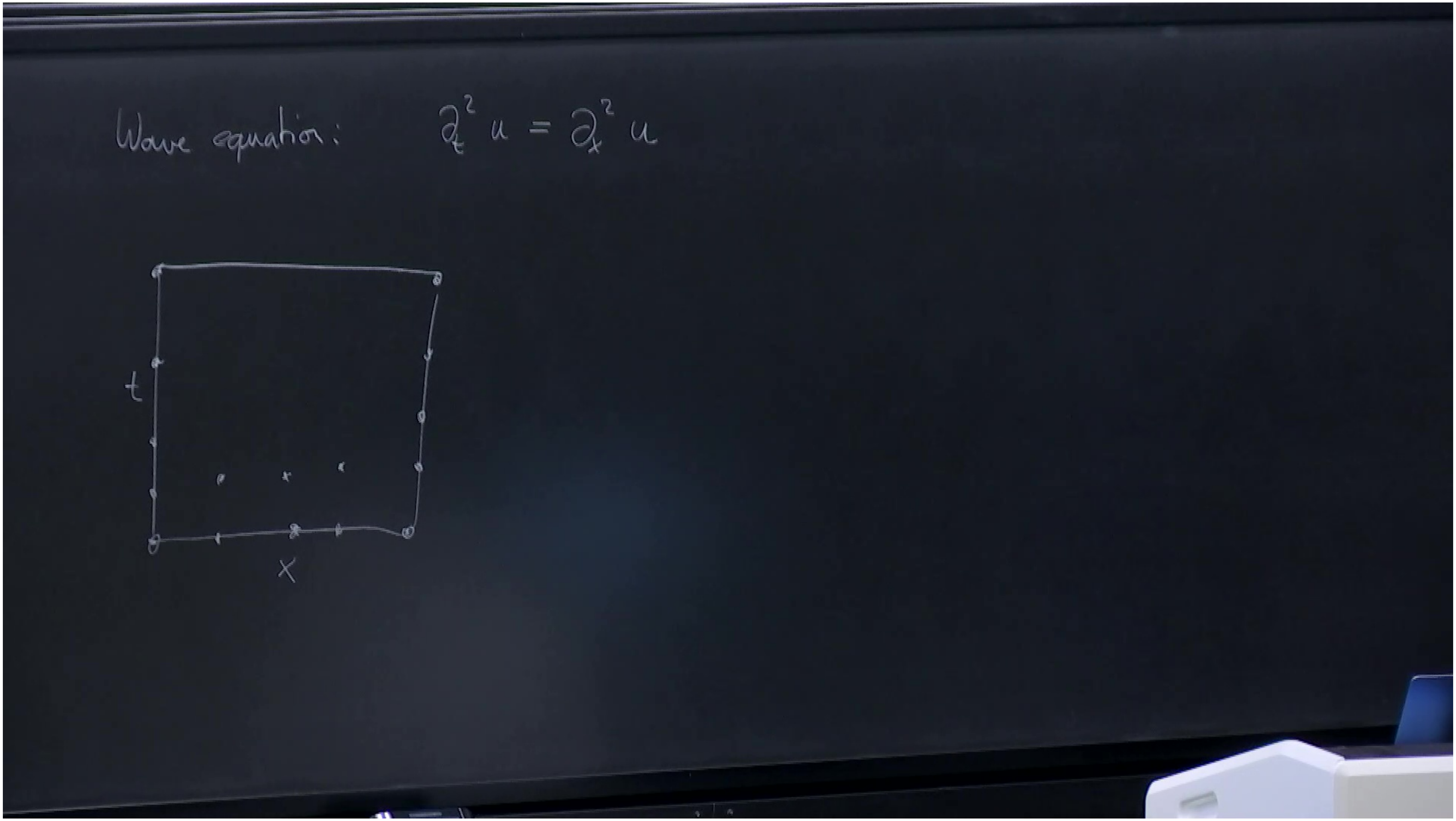
[10]:
```julia
using CairoMakie
```

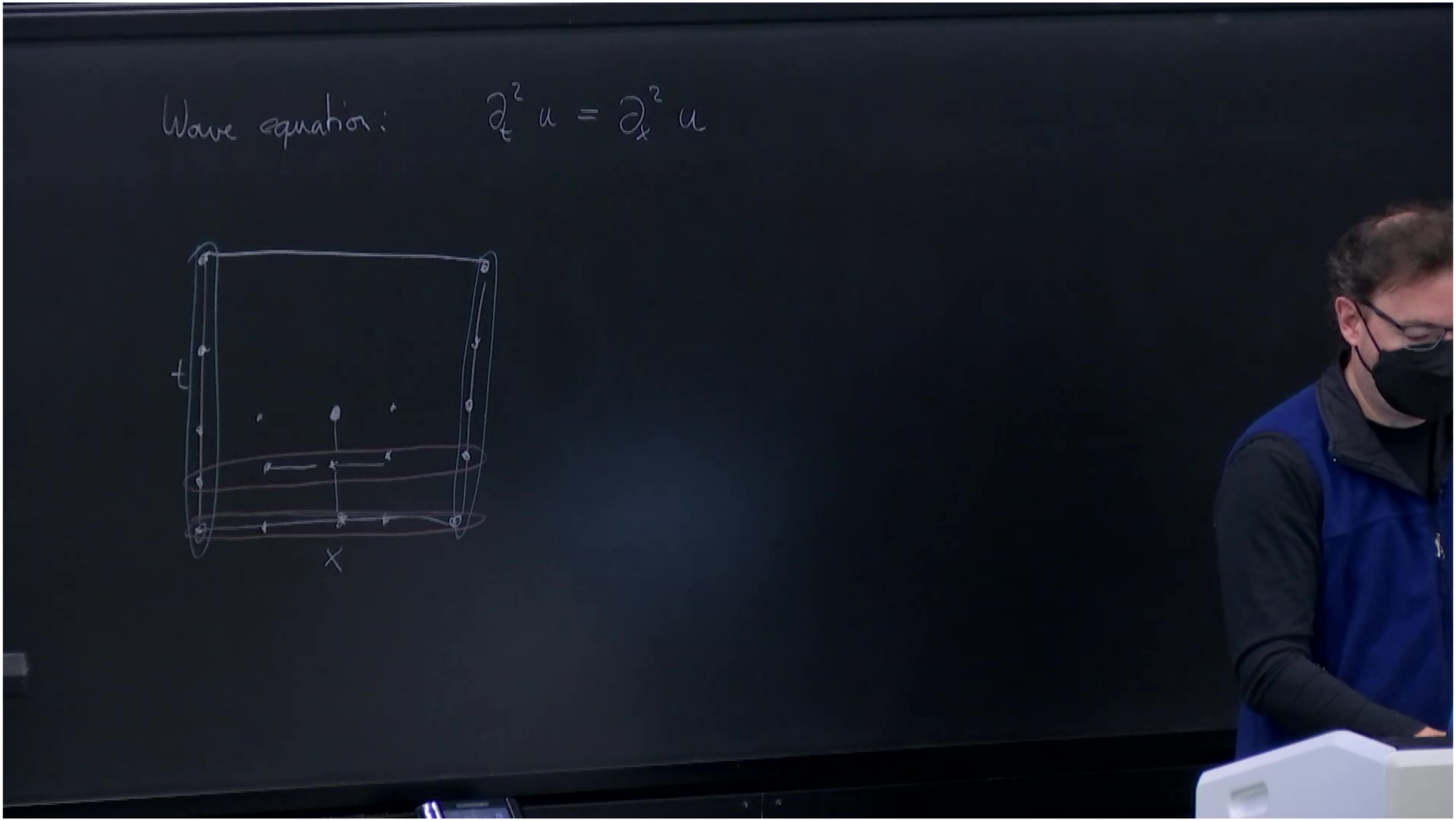
[11]:
```julia
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
contourf!(vec(x), vec(t), vec(u))
fig
```

[11]:



Simple 0 1 Julia 1.8.5 | Idle Mem: 1.08 GB Saving completed Mode: Command Ln 1, Col 1 Wave Equation 2023.ipynb


Wave equation: $\partial_t^2 u = \partial_x^2 u$
t
x


Wave equation: $\partial_t^2 u = \partial_x^2 u$
t
x

Safari File Edit View History Bookmarks Develop Window Help Thu Feb 2 09:23

symmetry.pi.local

JupyterLab | Start Page

File Edit View Run Kernel Git Tabs Settings Help Mem:1.08 GB

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Code git Julia 1.8.5

Solve the wave equation

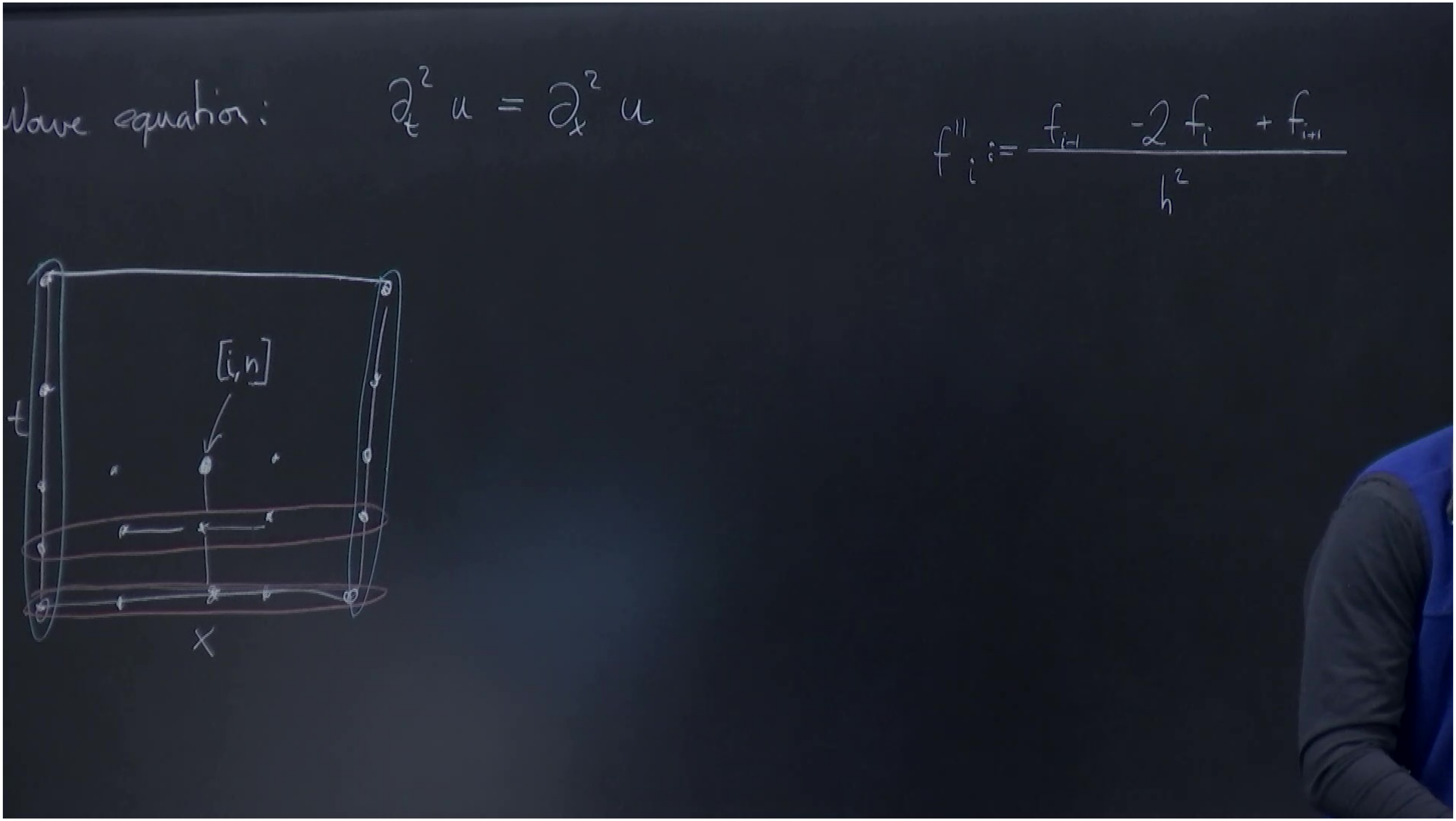
[1]:
```
# Define two grid functions `t`, `x` holding the c
function coords(nn, ni, dt, dx)
    t = [n * dt for i in 0:ni, n in 0:nn]
    x = [i * dx for i in 0:ni, n in 0:nn]
    return t, x
end
```

[1]:
```
coords (generic function with 1 method)
```

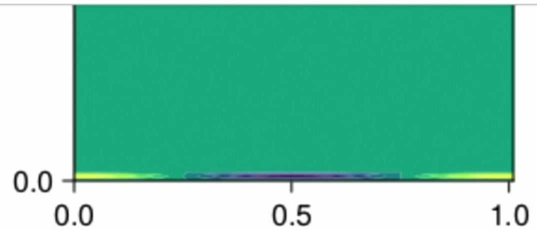
[7]:
```
t, x = coords(101, 101, 1/100, 1/100);
```

[8]:
```
function standing(t, x)
    ni, nn = size(t)
    u = zeros(ni, nn)
    # Initial conditions for first two points in time
    for n in 1:2, i in 1:ni
        u[i, n] = cos(2π * t[n, i]) * sin(2π * x[n, i])
    end
    return u
end
```

Simple 0 1 Julia 1.8.5 | Idle Mem: 1.08 GB Saving completed Mode: Edit Ln 1, Col 51 Wave Equation 2023.ipynb


Wave equation: $\partial_t^2 u = \partial_x^2 u$
$f''_i := \frac{f_{i-1} - 2 f_i + f_{i+1}}{h^2}$
[i,n]
t
x

Wave Equation 2023.ipynb




```julia
# Find the solution at one interior point u[i,n]
function evolve_point!(u, dt, dx, n, i)
    ni, nn = size(u)
    # Finite difference for second derivative in x direction at point [i,n-1]
    uxx = (u[i-1,n-1] - 2*u[i,n-1] + u[i+1,n-1]) / dx^2
    # Time derivative
    utt = (u[i,n-2] - 2*u[i,n-1] + u[i,n]) / dt^2
```

Safari File Edit View History Bookmarks Develop Window Help — Charged — Thu Feb 2 09:33

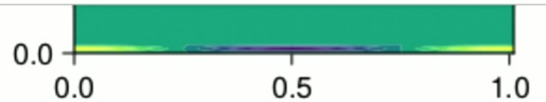
symmetry.pi.local

JupyterLab | Start Page

File Edit View Run Kernel Git Tabs Settings Help — Mem:1.08 GB

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Code — git — Julia 1.8.5



[15]:

```julia
# Find the solution at one interior point u[i,n]
function evolve_point!(u, dt, dx, n, i)
    ni, nn = size(u)
    # Finite difference for second derivative in x direction at point [i,n-1]
    uxx = (u[i-1,n-1] - 2*u[i,n-1] + u[i+1,n-1]) / dx^2
    # Time derivative
    # utt = (u[i,n-2] - 2*u[i,n-1] + u[i,n]) / dt^2
    # Wave equation
    # utt = uxx
    # Solve for u[i,n]
    u[i,n] = uxx * dt^2 - u[i,n-2] + 2*u[i,n-1]
    return
end
```

[15]:

```
evolve_point! (generic function with 1 method)
```

[ ]:

Simple 0 1 Julia 1.8.5 | Idle Mem: 1.08 GB Saving completed Mode: Edit Ln 12, Col 11 Wave Equation 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help · Charged · Thu Feb 2 09:36

symmetry.pi.local

JupyterLab · Start Page

File Edit View Run Kernel Git Tabs Settings Help · Mem:1.14 GB

Untitled10.ipynb · Convergence 2023 b.ipynb · Wave Equation 2023.ipynb

Code · git · Julia 1.8.5

```
    uxx = (u[i-1,n-1] - 2*u[i,n-1] + u[i+1,n-1]) / dx^2
    # Time derivative
    # utt = (u[i,n-2] - 2*u[i,n-1] + u[i,n]) / dt^2
    # Wave equation
    # utt = uxx
    # Solve for u[i,n]
    u[i,n] = uxx * dt^2 - u[i,n-2] + 2*u[i,n-1]
    return
end
```

[16]: evolve_point! (generic function with 1 method)

[ ]:

```
# Find the solution for one point in time `n`
function evolve_step!(u, dt, dx, n)
    ni, nn = size(u)
    # Apply boundary condition at left boundary
    u[1,n] = 0
    # Evolve interior
    for i in 2:ni-1
        evolve_point!(u, dt, dx, n, i)
    end
    # Apply boundary condition at right boundary
    u[ni,n] = 0
    return
end
```

Simple · 0 · 1 · Julia 1.8.5 | Idle · Mem: 1.14 GB · Saving completed · Mode: Edit · Ln 13, Col 4 · Wave Equation 2023.ipynb

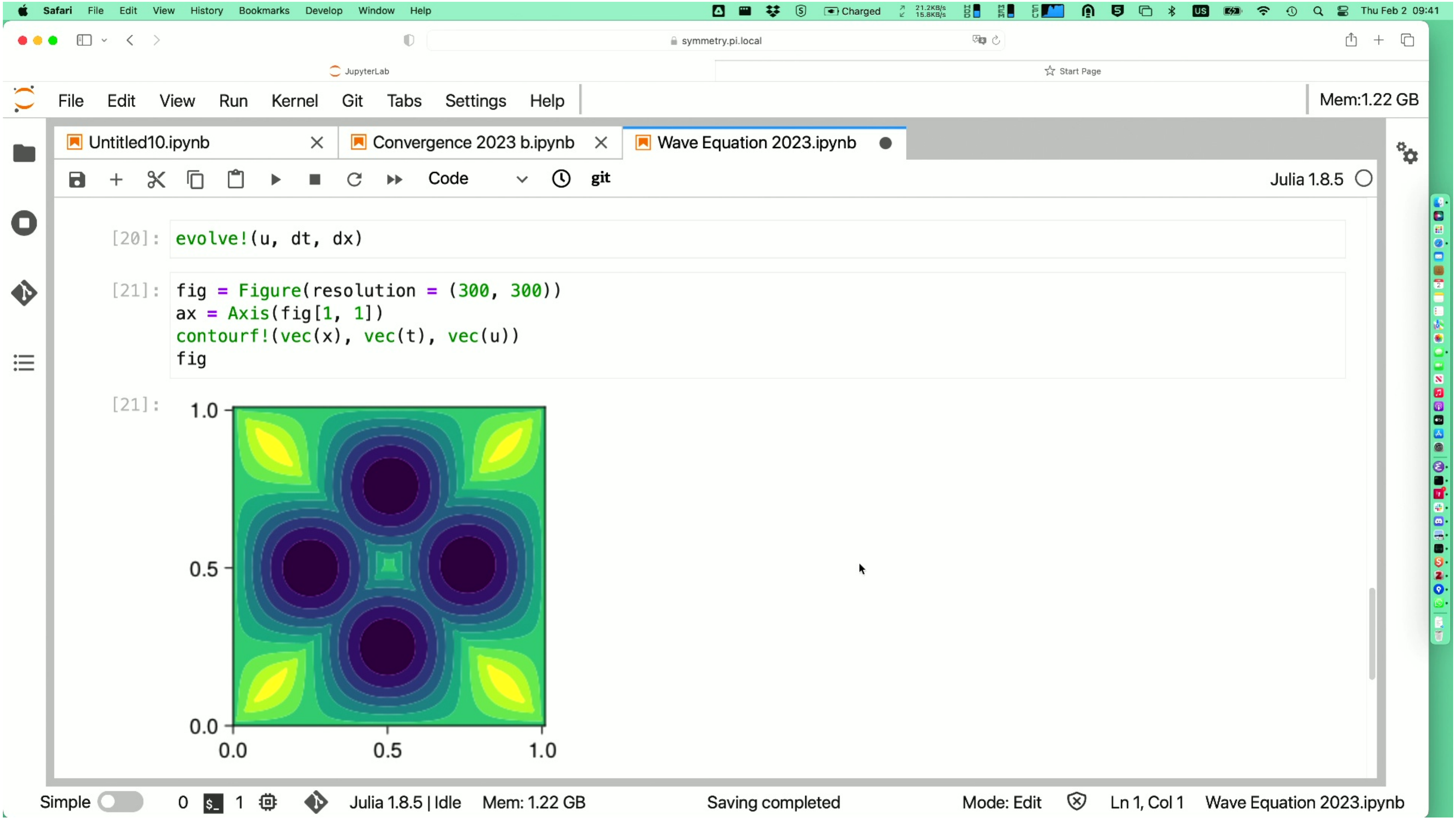
Solve the wave equation

[12]:
```
# Define two grid functions `t`, `x` holding the coordinates
function coords(nn, ni, dt, dx)
    t = [n * dt for i in 0:ni, n in 0:nn]
    x = [i * dx for i in 0:ni, n in 0:nn]
    return t, x
end
```

[12]:
```
coords (generic function with 1 method)
```

[7]:
```
nn = 101
ni = 101
dt = 1 / (nn-1)
dx = 1 / (ni-1)
t, x = coords(101, 101, 1/100, 1/100);
```

[13]:
```
# Define initial conditions for a standing wave
function standing(t, x)
    ni, nn = size(t)
    u = zeros(ni, nn)
    # Initial conditions for first two points in time
    for n in 1:2, i in 1:ni
```


Safari File Edit View History Bookmarks Develop Window Help
Charged
Thu Feb 2 09:41
symmetry.pi.local
JupyterLab
Start Page
File Edit View Run Kernel Git Tabs Settings Help
Mem:1.22 GB
Untitled10.ipynb
Convergence 2023 b.ipynb
Wave Equation 2023.ipynb
Code
git
Julia 1.8.5
[20]: evolve!(u, dt, dx)
[21]: fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
contourf!(vec(x), vec(t), vec(u))
fig
[21]:
1.0
0.5
0.0
0.0
0.5
1.0
Simple
0
1
Julia 1.8.5 | Idle
Mem: 1.22 GB
Saving completed
Mode: Edit
Ln 1, Col 1
Wave Equation 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help Thu Feb 2 09:46

symmetry.pi.local

JupyterLab Start Page

File Edit View Run Kernel Git Tabs Settings Help Mem:1.08 GB

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Code git Julia 1.8.5

```
dx = 1 / (ni-1)
t, x = coords(nn, ni, dt, dx);
```

[3]:
```
# Define initial conditions for a standing wave
function standing(t, x)
    ni, nn = size(t)
    u = zeros(ni, nn)
    # Initial conditions for first two points in time
    for n in 1:2, i in 1:ni
        u[i, n] = cos(2π * t[n, i]) * sin(2π * x[n, i])
    end
    return u
end
```

[3]:
```
standing (generic function with 1 method)
```

[4]:
```
u = standing(t, x);
```

[5]:
```
using CairoMakie
```

[6]:
```
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
contourf!(vec(x), vec(t), vec(u))
fig
```

Simple 0 1 Julia 1.8.5 | Idle Mem: 1.08 GB Saving completed Mode: Command Ln 3, Col 34 Wave Equation 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help

symmetry.pi.local

JupyterLab | Start Page

File Edit View Run Kernel Git Tabs Settings Help | Mem:1.10 GB

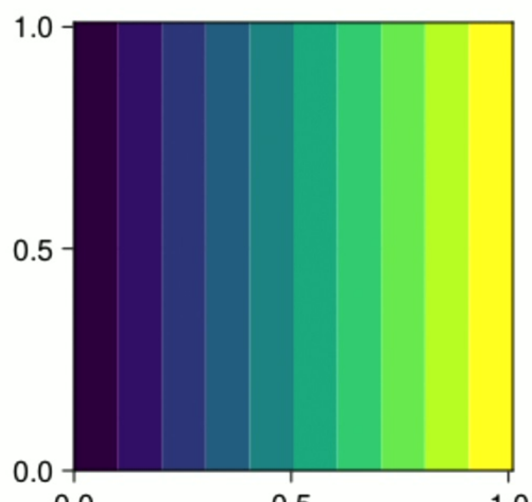
Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Code | git | Julia 1.8.5

[14]:
```julia
using CairoMakie
```

[21]:
```julia
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
# contourf!(vec(x), vec(t), vec(u))
contourf!(vec(x), vec(t), vec(x))
fig
```

[21]:



Simple 0 1 Julia 1.8.5 | Idle Mem: 1.10 GB Saving completed Mode: Command Ln 1, Col 51 Wave Equation 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help — Calculating... — Thu Feb 2 09:47

symmetry.pi.local

JupyterLab — Start Page

File Edit View Run Kernel Git Tabs Settings Help — Mem:1.10 GB

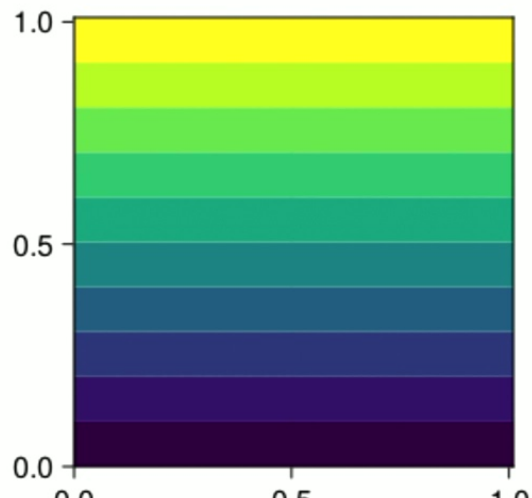
Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Code — git — Julia 1.8.5

[14]:
```
using CairoMakie
```

[22]:
```
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
# contourf!(vec(x), vec(t), vec(u))
contourf!(vec(x), vec(t), vec(t))
fig
```

[22]:



Simple 0 1 Julia 1.8.5 | Idle Mem: 1.10 GB Saving completed Mode: Command Ln 1, Col 51 Wave Equation 2023.ipynb

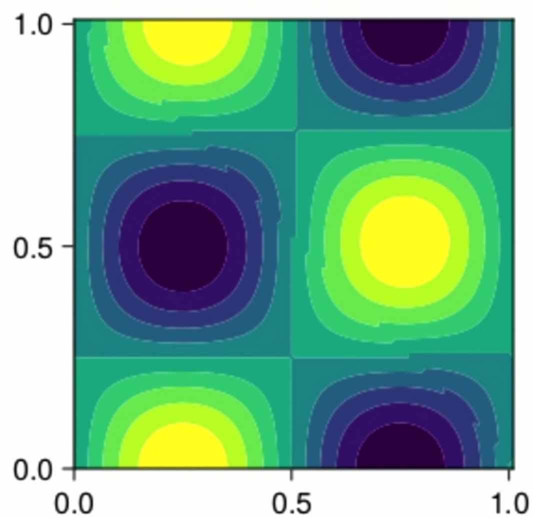
File Edit View Run Kernel Git Tabs Settings Help

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

[14]:
```julia
using CairoMakie
```

[23]:
```julia
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
contourf!(vec(x), vec(t), vec(u))
fig
```

[23]:



Simple 0 1 Julia 1.8.5 | Idle Mem: 1.10 GB Saving completed Mode: Command Ln 1, Col 51 Wave Equation 2023.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Julia 1.8.5

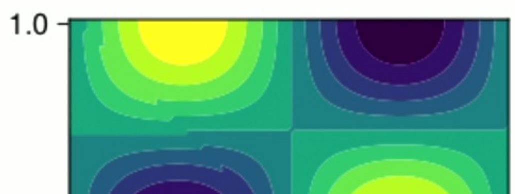
[30]:
```julia
# Find the solution everywhere
function evolve!(u, dt, dx)
    ni, nn = size(u)
    # Loop over all times
    for n in 3:nn
        evolve_step!(u, dt, dx, n)
    end
    return
end
```

[30]:
```
evolve! (generic function with 1 method)
```

[31]:
```julia
evolve!(u, dt, dx)
```

[32]:
```julia
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
contourf!(vec(x), vec(t), vec(u))
fig
```

[32]:



Simple 0 1 Julia 1.8.5 | Idle Mem: 1.10 GB Saving completed Mode: Command Ln 2, Col 21 Wave Equation 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help Thu Feb 2 09:51

symmetry.pi.local

JupyterLab Start Page

File Edit View Run Kernel Git Tabs Settings Help Mem:1.27 GB

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Code git Julia 1.8.5

```
[34]: t1, x1 = coords(2nn-1, 2ni-1, dt/2, dx/2);
```

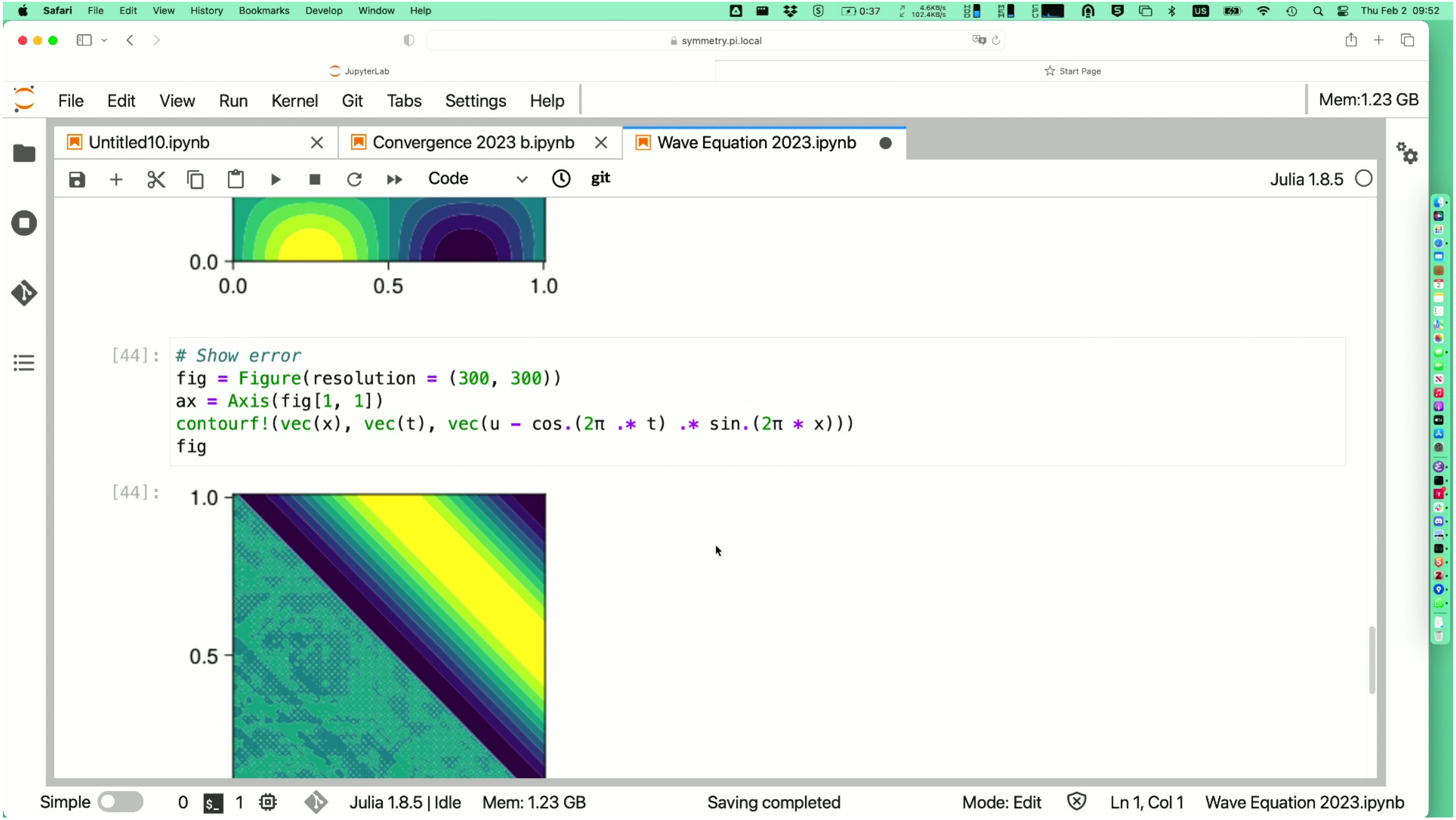
```
[35]: u1 = standing(t1, x1);
```

```
[37]: evolve!(u1, dt/2, dx/2)
```

```
[42]: fig = Figure(resolution = (300, 300))
ax = Axis(fig[1,1])
contourf!(fig[1,1], vec(x), vec(t), vec(u))
contourf!(fig[1,2], vec(x1), vec(t1), vec(u1))
fig
```

```
There needs to be a single axis-like object at GridLayoutBase.Span(1:1, 2:2), GridLayoutBase.Inner() to plot in to.
Use a non-mutating plotting command to create an axis implicitly.

Stacktrace:
 [1] error(s::String)
   @ Base ./error.jl:35
 [2] plot!(::Type{Combined{Makie.contourf}}, ::GridPosition, ::Vector{Float64}, ::Vararg{Vector{Float64}}; kwargs::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
   @ Makie ~/.julia/packages/Makie/Za3LL/src/figureplotting.jl:112
 [3] plot!(::Type{Combined{Makie.contourf}}, ::GridPosition, ::Vector{Float64}, ::Vector{Float64}, ::Vector{Float64})
```

Simple 0 1 Julia 1.8.5 | Idle Mem: 1.27 GB Saving completed Mode: Edit Ln 4, Col 1 Wave Equation 2023.ipynb


Safari File Edit View History Bookmarks Develop Window Help
Thu Feb 2 09:52
symmetry.pi.local
JupyterLab
Start Page
File Edit View Run Kernel Git Tabs Settings Help
Mem:1.23 GB
Untitled10.ipynb
Convergence 2023 b.ipynb
Wave Equation 2023.ipynb
Code
git
Julia 1.8.5
0.0
0.0 0.5 1.0
[44]:
# Show error
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
contourf!(vec(x), vec(t), vec(u - cos.(2π .* t) .* sin.(2π * x)))
fig
[44]:
1.0
0.5
Simple 0 1 Julia 1.8.5 | Idle Mem: 1.23 GB Saving completed Mode: Edit Ln 1, Col 1 Wave Equation 2023.ipynb

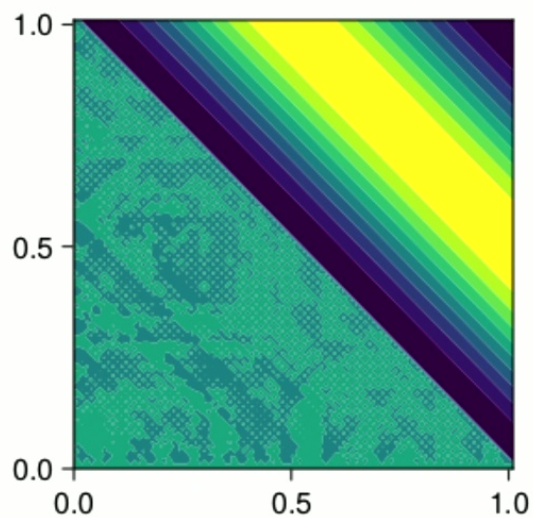
File Edit View Run Kernel Git Tabs Settings Help

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb

Julia 1.8.5

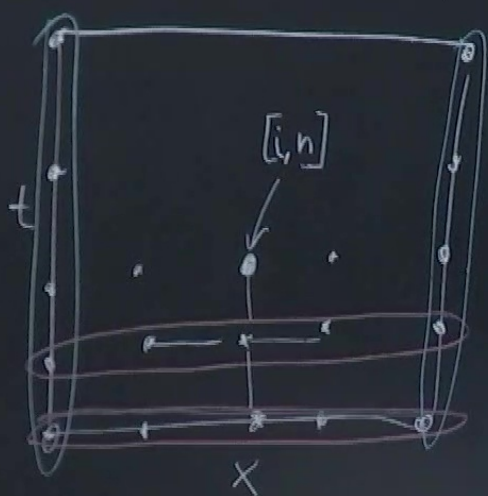
[48]:
```
# Show error
fig = Figure(resolution = (300, 300))
ax = Axis(fig[1, 1])
contourf!(vec(x), vec(t), vec(u - cos.(2π * t) .* sin.(2π * x)))
fig
```

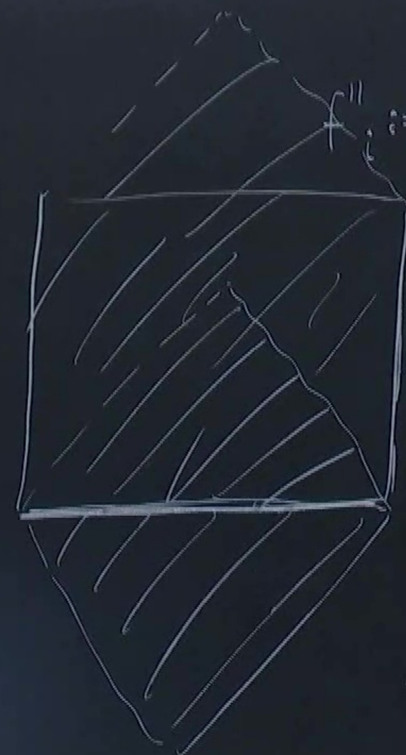
[48]:



Simple 0 1 Julia 1.8.5 | Idle Mem: 1.24 GB Saving completed Mode: Command Ln 4, Col 4 Wave Equation 2023.ipynb

Wave equation: $\partial_t^2 u = \partial_x^2 u$





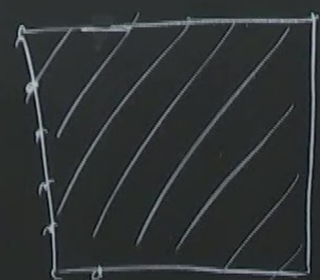
$$f''_i := \frac{f_{i-1} - 2 f_i + f_{i+1}}{h^2}$$

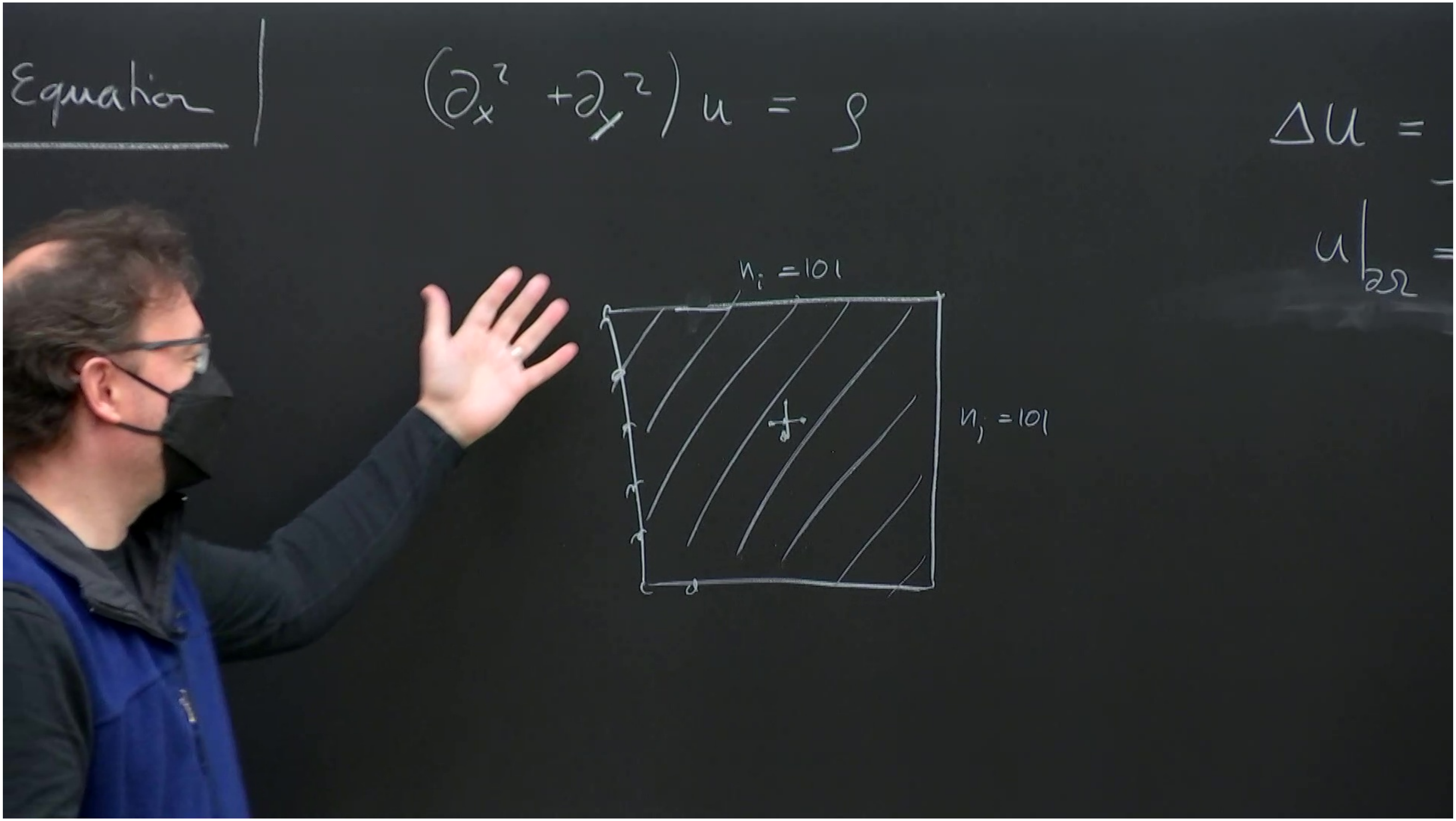
Poisson Equation

$$(\partial_x^2 + \partial_y^2)\, u = \rho$$

$$\Delta u = \rho \qquad u = \Delta^{-1} \rho$$

$$u\big|_{\partial\Omega} = 0$$




Equation
$(\partial_x^2 + \partial_y^2) u = \rho$
$n_i = 101$
$n_j = 101$
$\Delta u =$
$u|_{\partial\Omega} =$

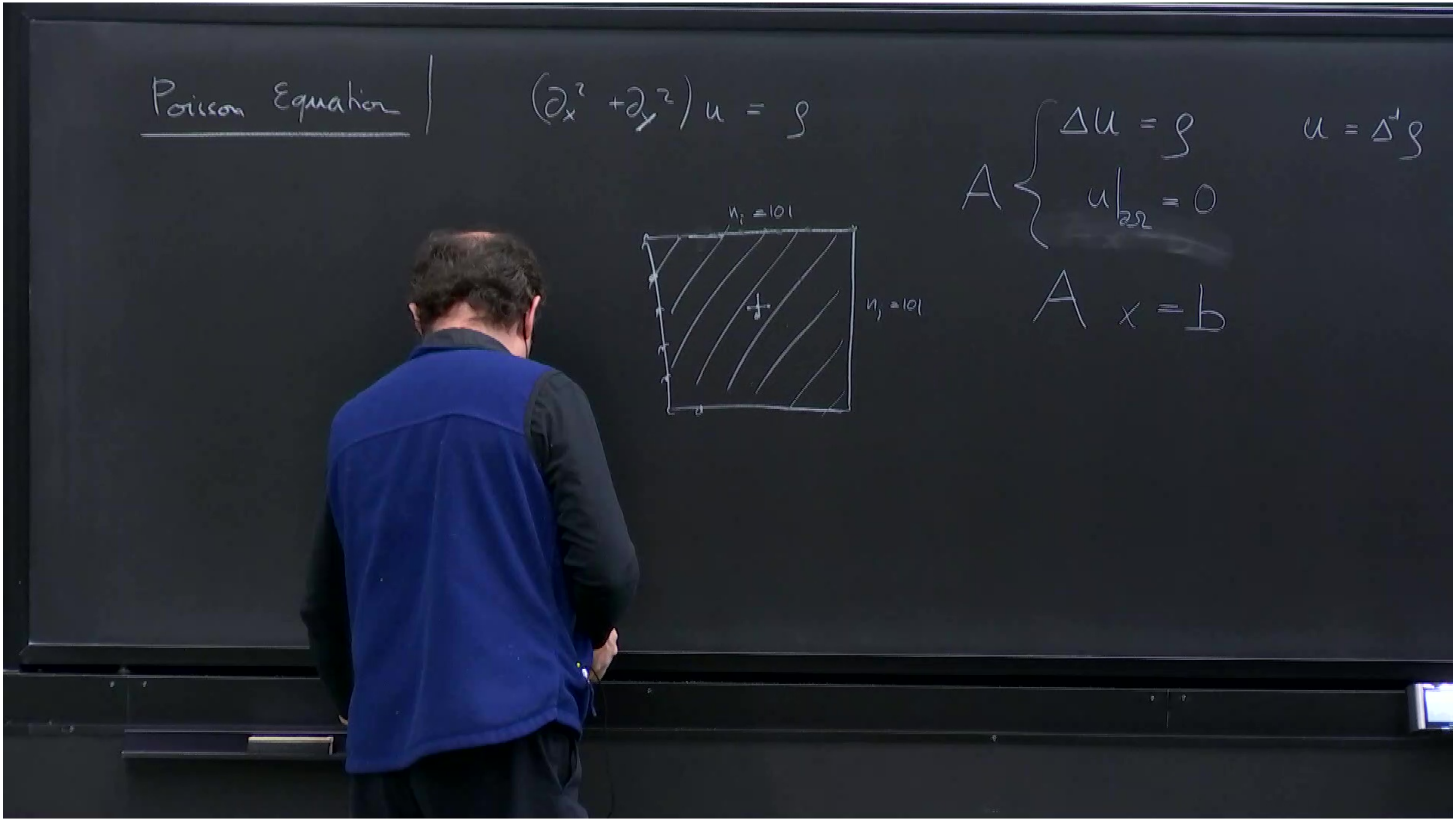
Solve Poisson equation

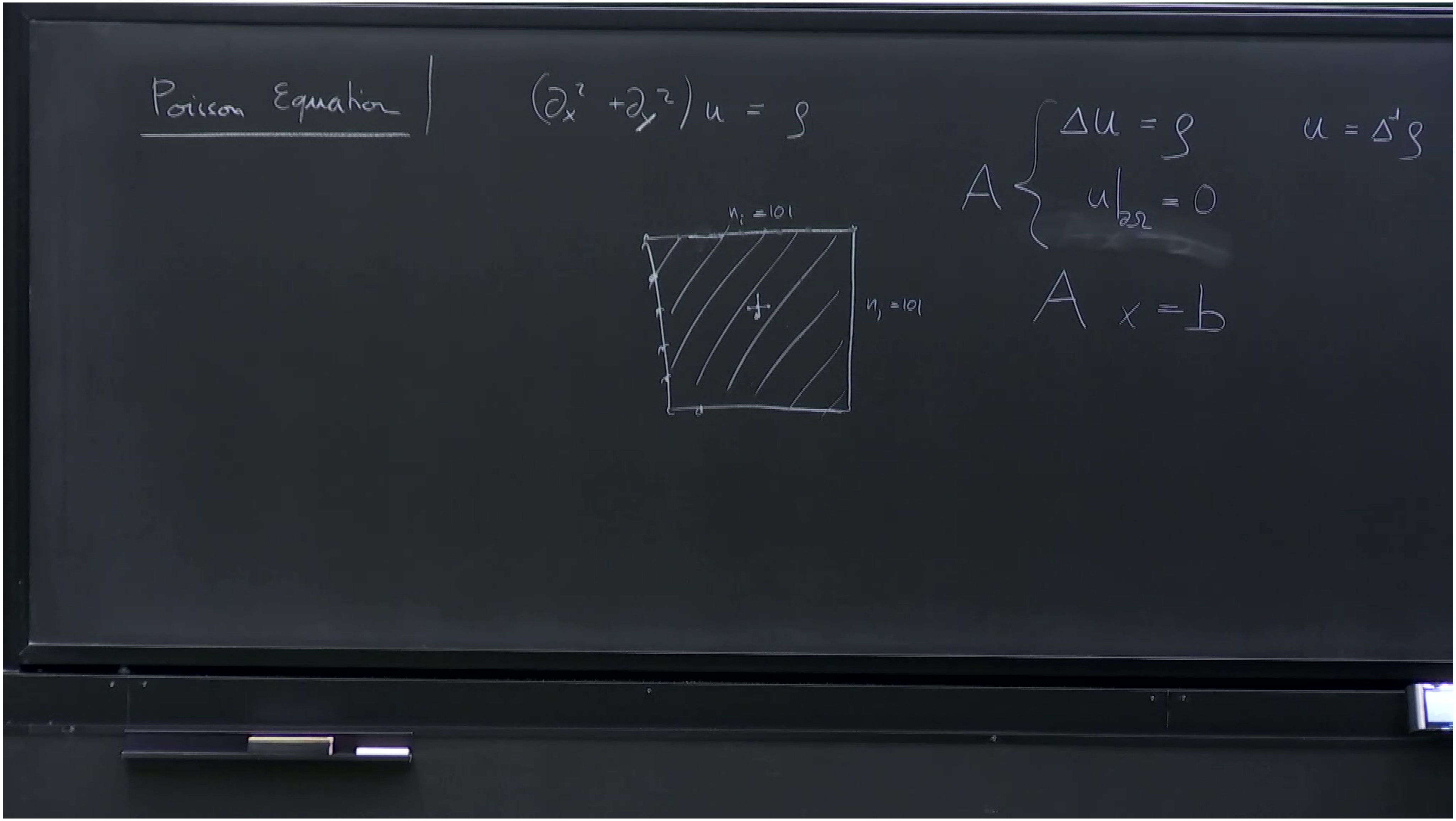
[2]:
```
# (∂ₜ² + ∂ₓ²) u = ρ
# u = 0   on boundary (Dirichlet boundary conditions)

# Using finite differencing
```

[4]:
```
# Choose problem size
ni = 5
nj = 5
dx = 1/(ni-1)
dy = 1/(nj-1);
```

[6]:
```
# Choose RHS (right hand side)
ρ = zeros(ni, nj)
ρ[2,2] = 1;
```

[ ]:


Poisson Equation
$(\partial_x^2 + \partial_y^2) u = \rho$
$n_i = 101$
$n_j = 101$
$A \begin{cases} \Delta u = \rho \\ u|_{\partial\Omega} = 0 \end{cases}$
$u = \Delta^{-1} \rho$
$A x = b$


Poisson Equation
$(\partial_x^2 + \partial_y^2) u = \rho$
$A \begin{cases} \Delta u = \rho \\ u|_{\partial\Omega} = 0 \end{cases}$
$u = \Delta^{-1} \rho$
$n_i = 101$
$n_j = 101$
$A x = b$

File Edit View Run Kernel Git Tabs Settings Help | Mem:1.56 GB

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb | Poisson Equation 2023.ipynb

Code | git | Julia 1.8.5

```julia
using SparseArrays
```

[9]:

```julia
# Define the non-zero elements
I = Int[]
J = Int[]
V = Float64[];
```

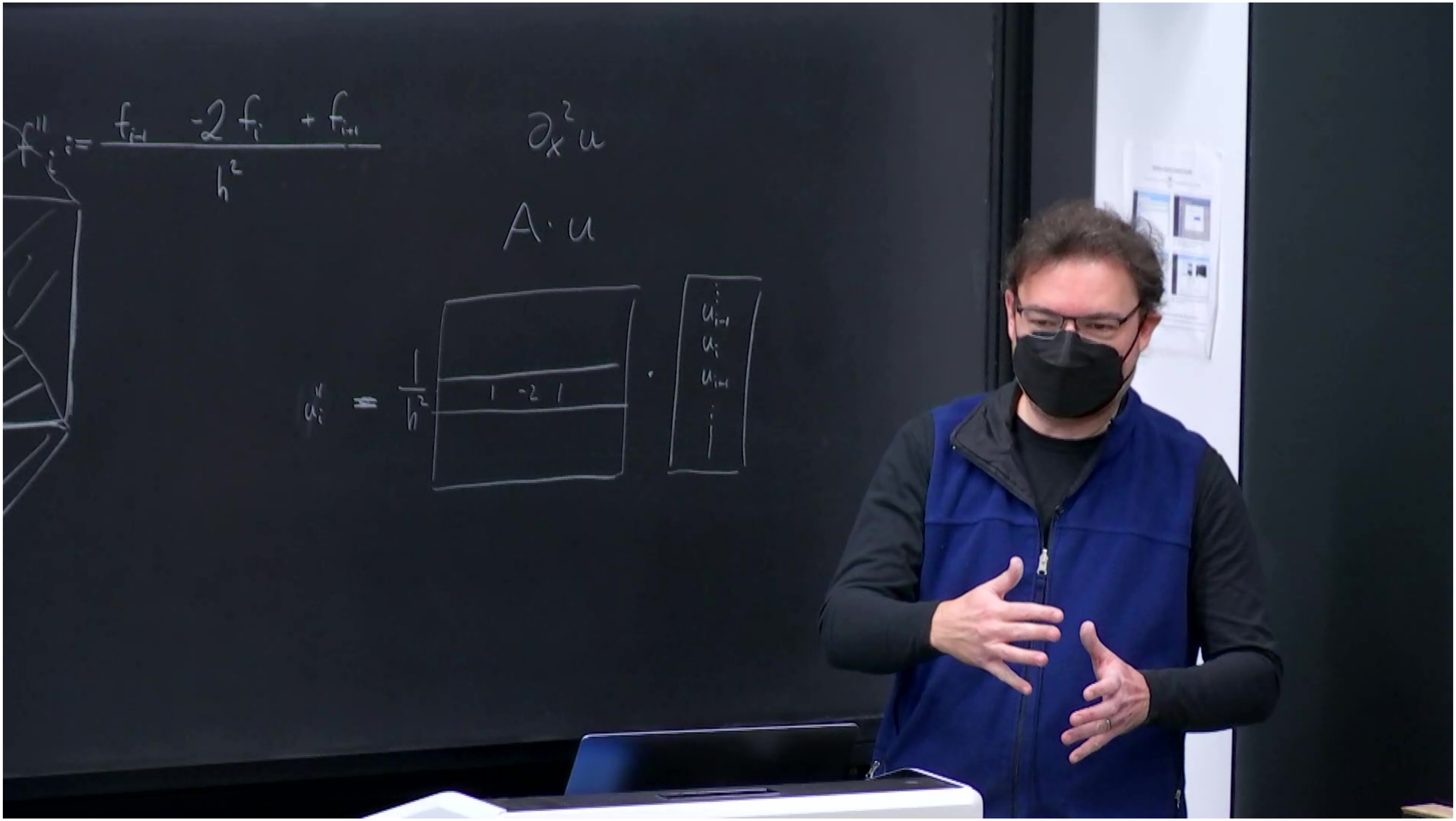
[13]:

```julia
# Boundary conditions (the easy part)

# left boundary
# A[1,1] = 1
push!(I, 1)
push!(J, 1)
push!(V, 1)

# right boundary
# A[ni,ni] = 1
push!(I, ni)
push!(J, ni)
push!(V, 1)
;
```

[ ]:

```julia
# Interior: Laplace operator
for i in 2:ni-1
end
```

Simple 0 2 Julia 1.8.5 | Idle Mem: 1.56 GB Saving completed Mode: Command Ln 3, Col 19 Poisson Equation 2023.ipynb


$f''_i := \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2}$
$\partial_x^2 u$
$A \cdot u$
$u''_i = \frac{1}{h^2}$
1 -2 1
u_{i-1}
u_i
u_{i+1}

Safari File Edit View History Bookmarks Develop Window Help Charged Thu Feb 2 10:26

symmetry.pi.local

JupyterLab Start Page

File Edit View Run Kernel Git Tabs Settings Help Mem:1.60 GB

Untitled10.ipynb Convergence 2023 b.ipynb Wave Equation 2023.ipynb Poisson Equation 2023.ipynb

Code git Julia 1.8.5

```
 ·     ·    1.0  -2.0  1.0
 ·     ·     ·     ·   1.0
```

[21]:
```julia
# Define RHS
b = [0; ρ[2:ni-1]; 0]
```

[21]:
```
5-element Vector{Float64}:
 0.0
 1.0
 1.0
 1.0
 0.0
```

[22]:
```julia
u = A \ b
```

[22]:
```
5-element Vector{Float64}:
  0.0
 -1.5
 -2.0
 -1.5
  0.0
```

[ ]:

Simple 0 2 Julia 1.8.5 | Idle Mem: 1.60 GB Saving completed Mode: Edit Ln 1, Col 1 Poisson Equation 2023.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb | Poisson Equation 2023.ipynb

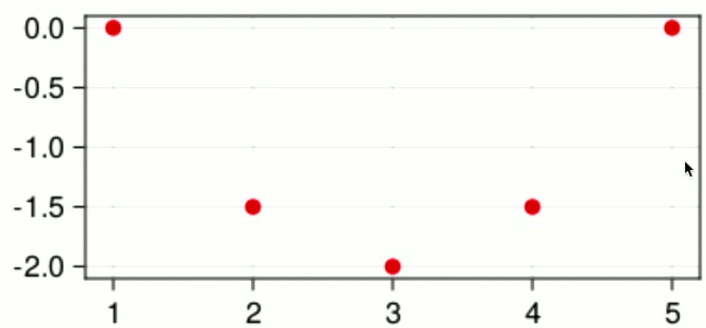
Julia 1.8.5

```
0.0
```

[23]:
```julia
using CairoMakie
```

[26]:
```julia
fig = Figure(resolution=(400, 200))
ax = Axis(fig[1, 1])
plot!(u; color=:red)
fig
```

[26]:



[ ]:

Simple 0 2 Julia 1.8.5 | Idle Mem: 2.04 GB Saving completed Mode: Command Ln 1, Col 1 Poisson Equation 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help Charged Thu Feb 2 10:34

symmetry.pi.local

JupyterLab Start Page

File Edit View Run Kernel Git Tabs Settings Help Mem:2.00 GB

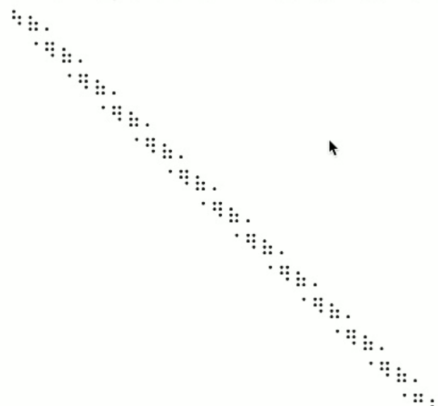
Untitled10.ipynb | Convergence 2023 b.ipynb | Wave Equation 2023.ipynb | Poisson Equation 2023.ipynb

Code git Julia 1.8.5

```julia
    # right stencil point
    push!(I, i)      # row
    push!(J, i+1)    # column
    push!(V, 1/dx^2)     # value
end
```

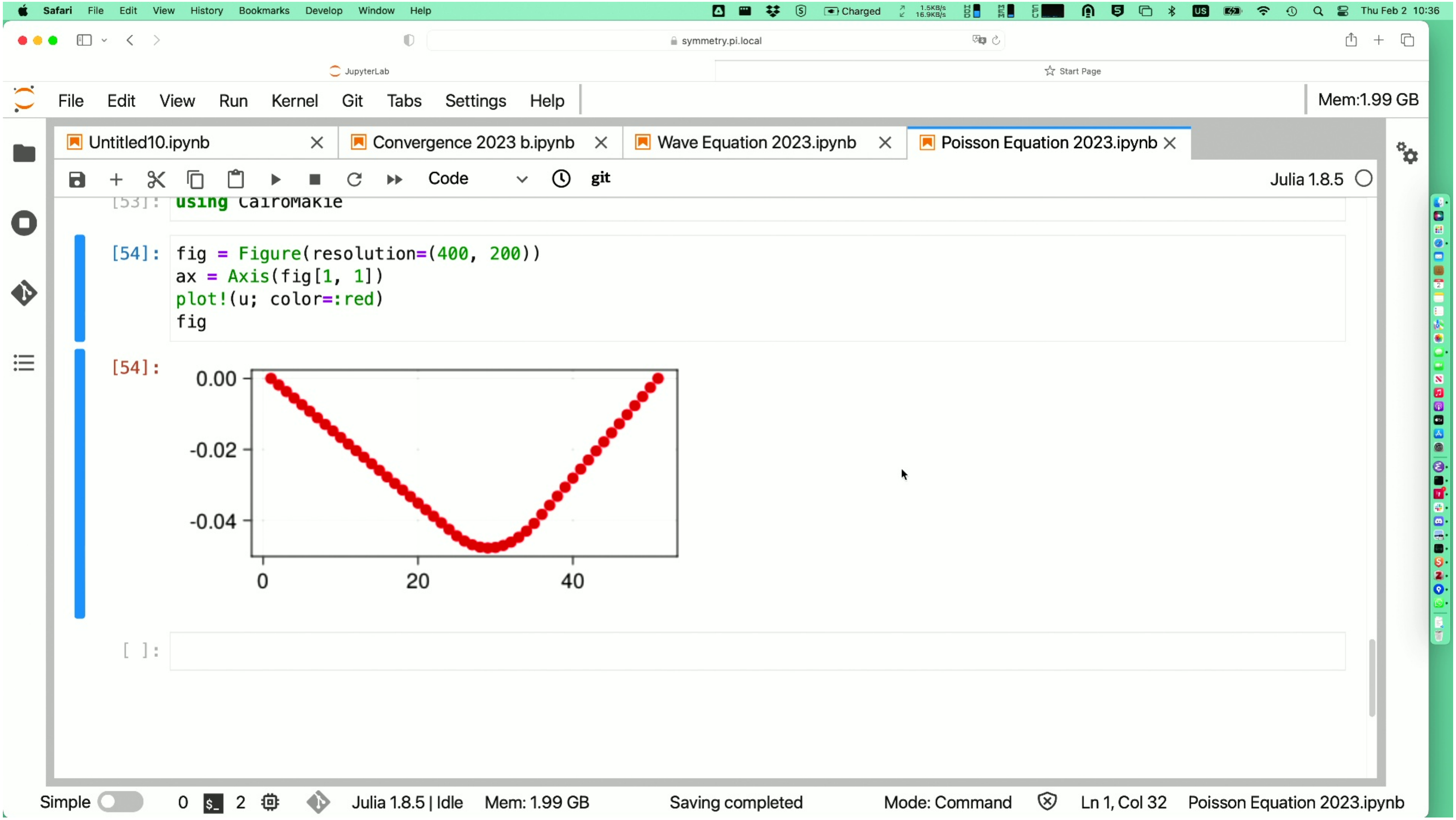
[39]:
```julia
A = sparse(I, J, V, ni, ni)
```

[39]:
```
51×51 SparseMatrixCSC{Float64, Int64} with 149 stored entries:
```



[40]:
```julia
# Define RHS
b = [0; ρ[2:ni-1]; 0]
```

Simple 0 2 Julia 1.8.5 | Idle Mem: 2.00 GB Saving completed Mode: Command Ln 1, Col 10 Poisson Equation 2023.ipynb


symmetry.pi.local
JupyterLab
Start Page
File Edit View Run Kernel Git Tabs Settings Help
Mem:1.99 GB
Untitled10.ipynb
Convergence 2023 b.ipynb
Wave Equation 2023.ipynb
Poisson Equation 2023.ipynb
Code
git
Julia 1.8.5
[53]: using CairoMakie
[54]: fig = Figure(resolution=(400, 200))
ax = Axis(fig[1, 1])
plot!(u; color=:red)
fig
[54]:
0.00
-0.02
-0.04
0
20
40
[ ]:
Simple
Julia 1.8.5 | Idle
Mem: 1.99 GB
Saving completed
Mode: Command
Ln 1, Col 32
Poisson Equation 2023.ipynb