Title: Lecture - Numerical Methods, PHYS 777

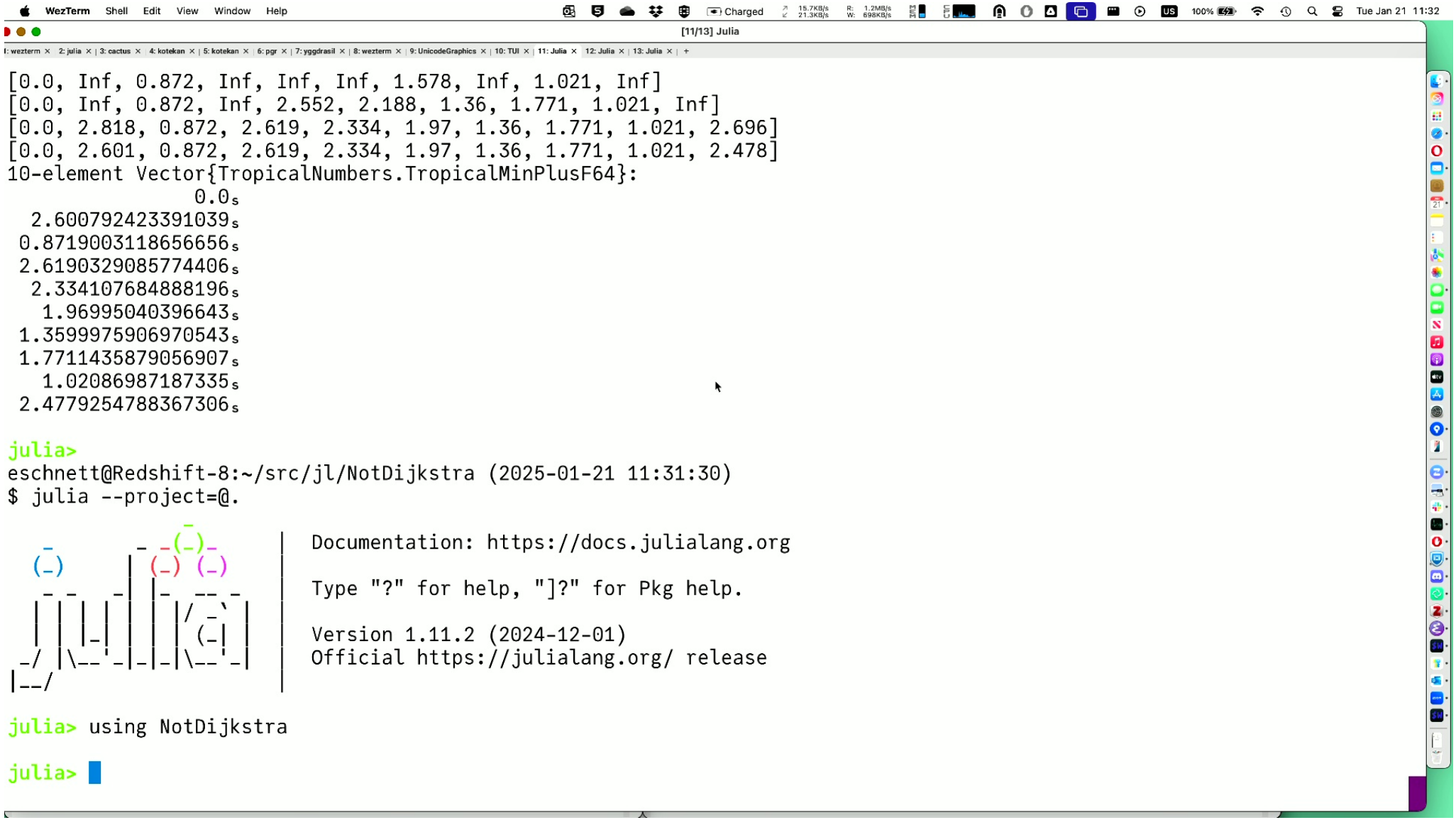
Speakers: Erik Schnetter, Dustin Lang

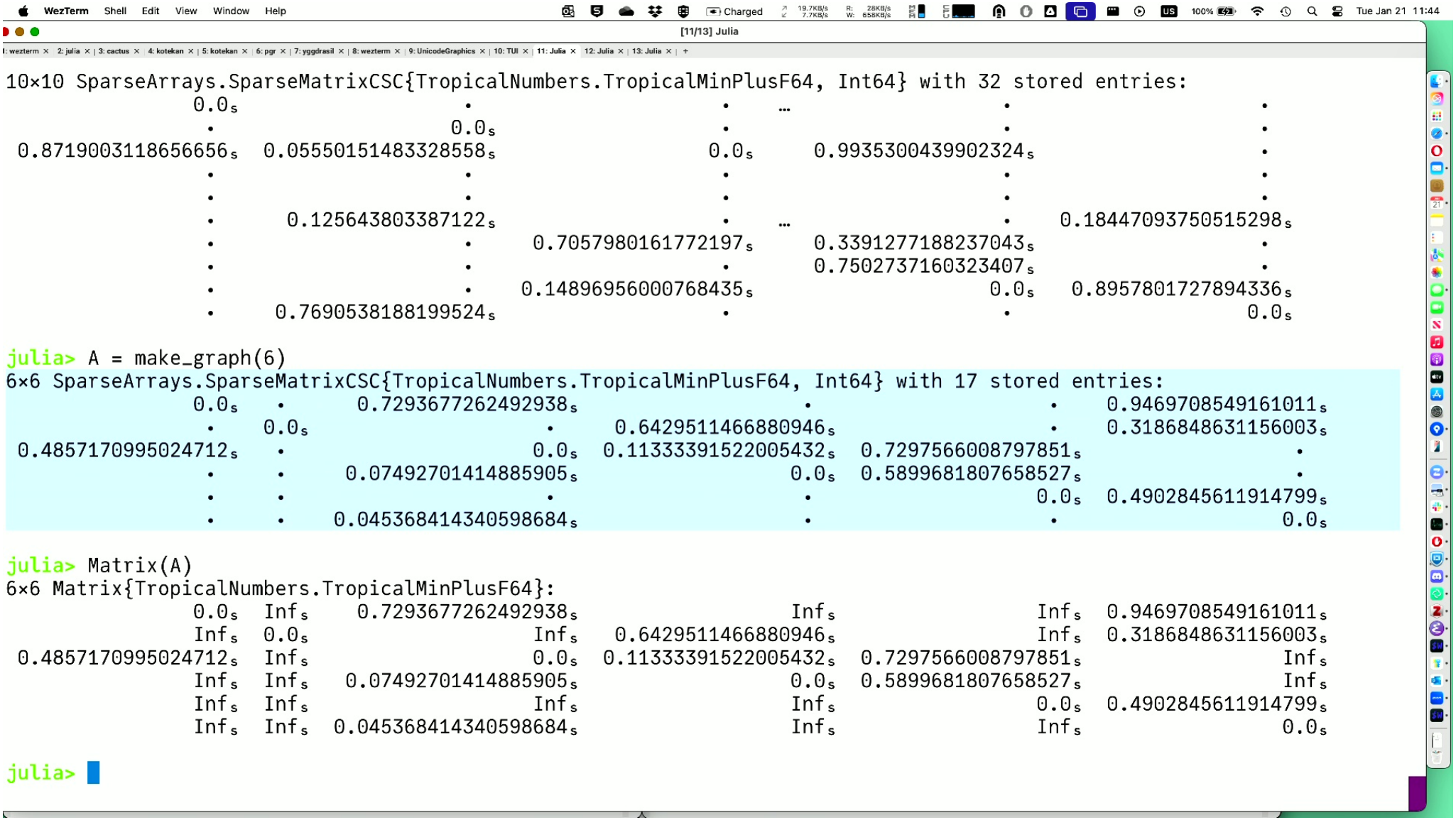
Collection/Series: Numerical Methods (Core), PHYS 777-, January 6 - February 5, 2025

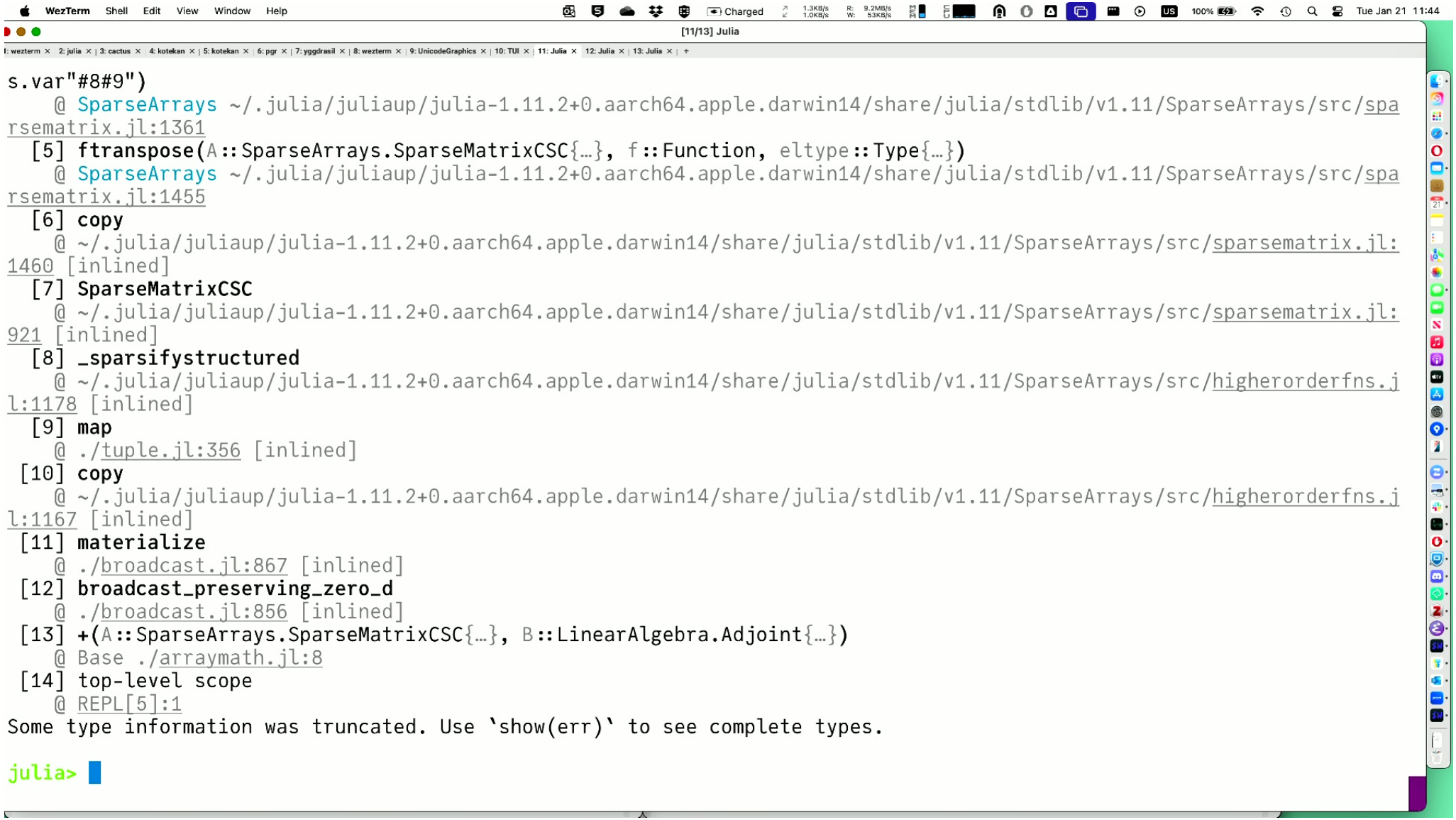
Subject: Other

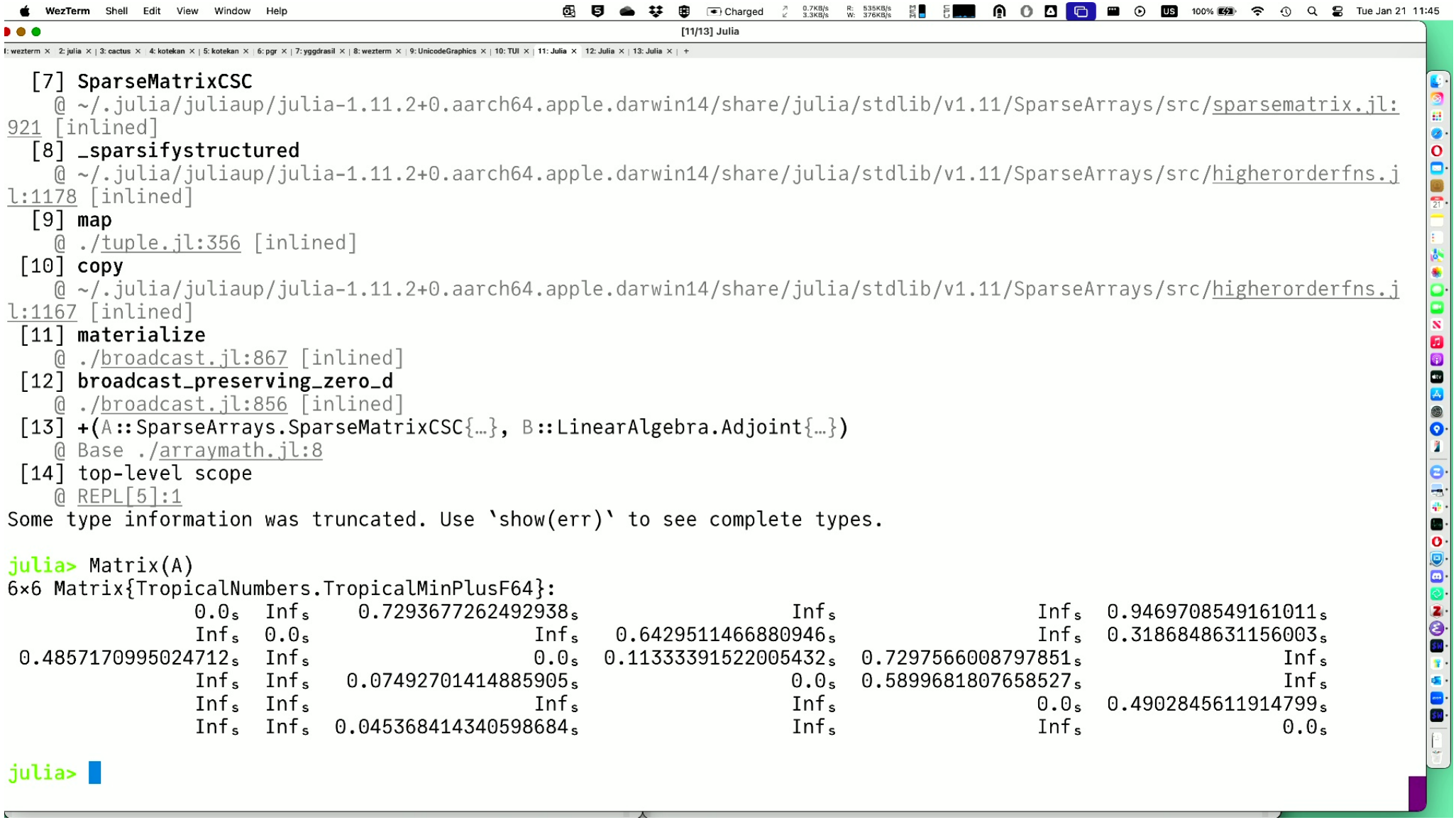
Date: January 21, 2025 - 11:30 AM

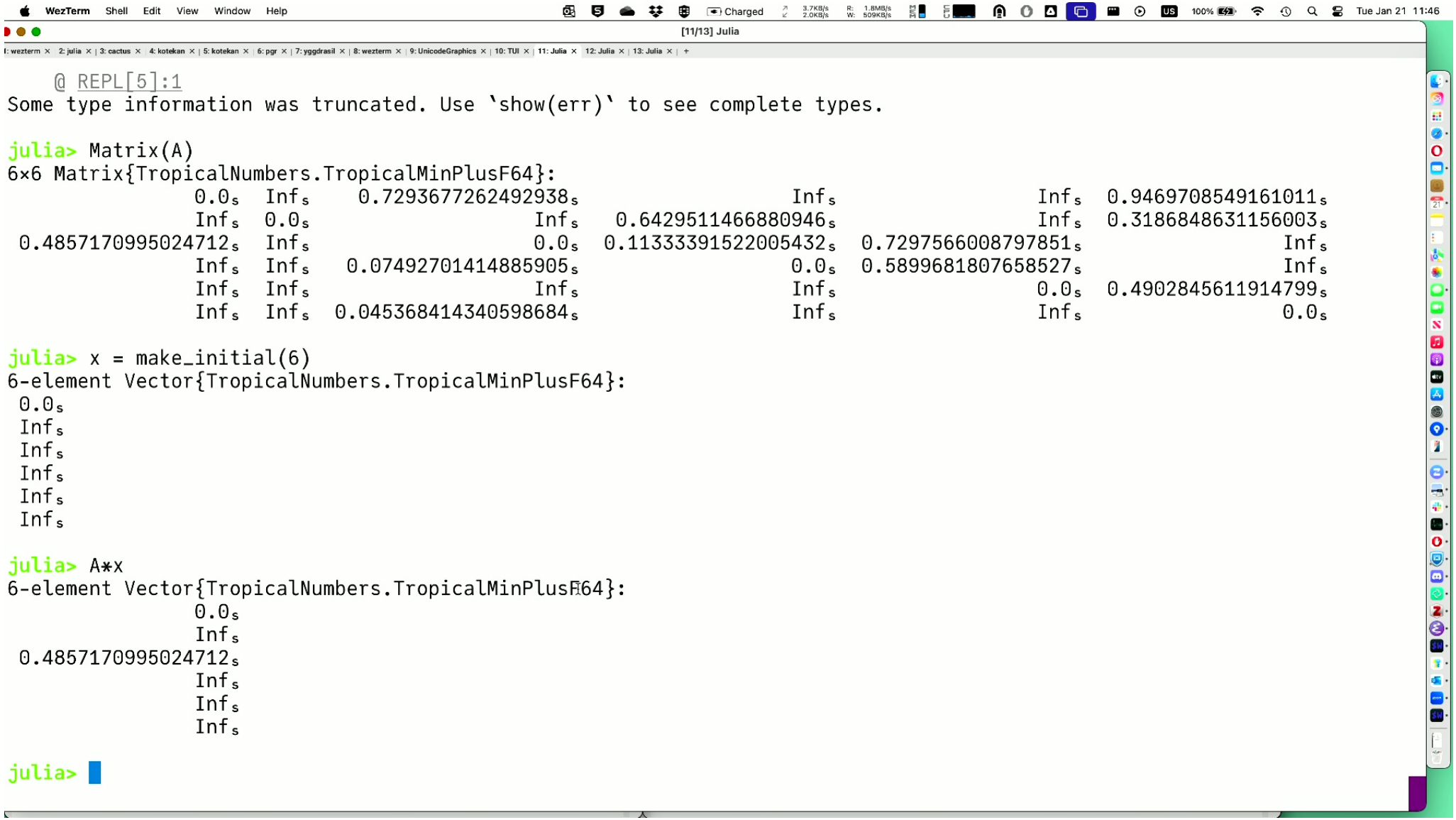
URL: https://pirsa.org/25010059

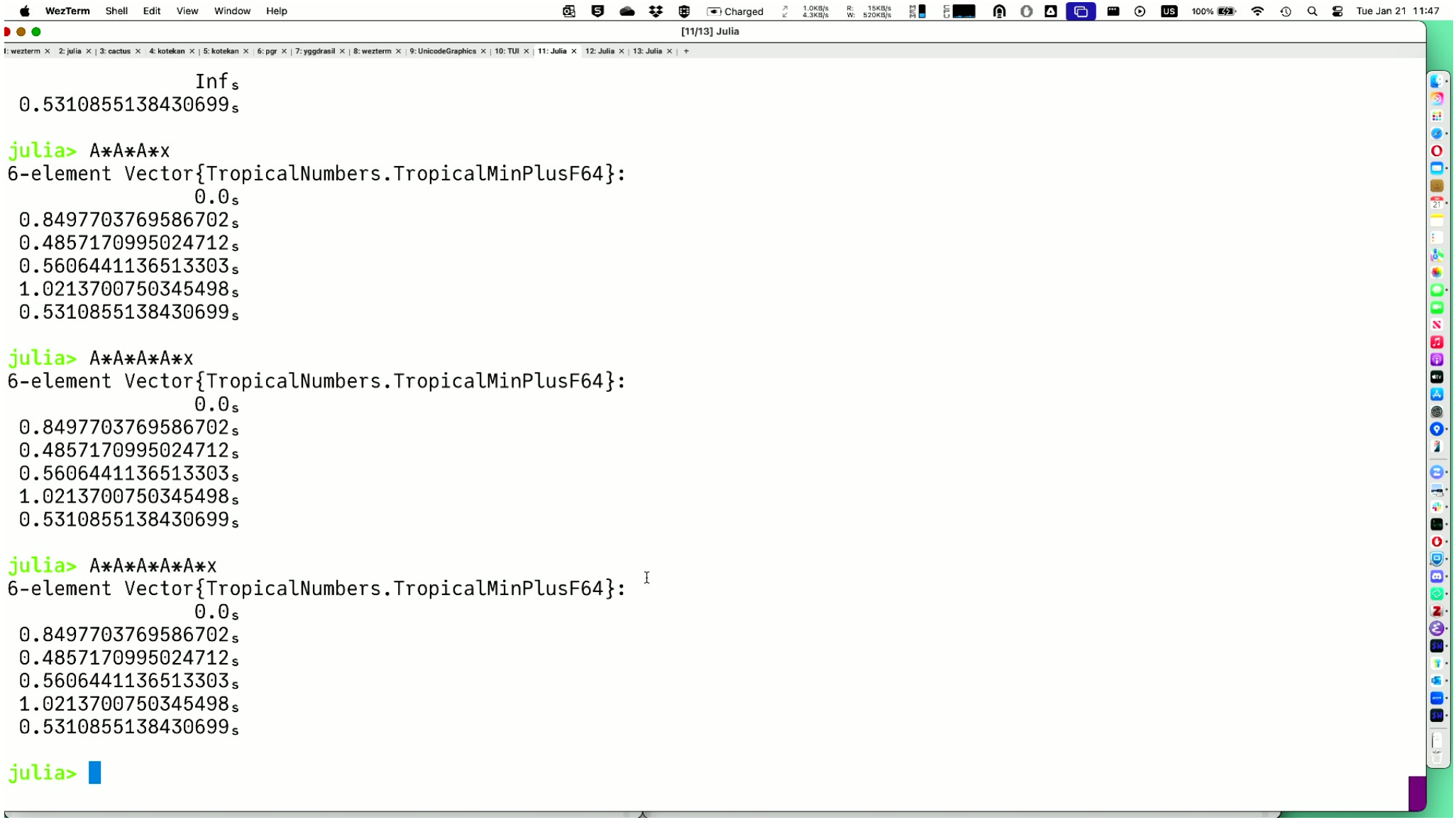

WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 11:32
[11/13] Julia
[0.0, Inf, 0.872, Inf, Inf, Inf, 1.578, Inf, 1.021, Inf]
[0.0, Inf, 0.872, Inf, 2.552, 2.188, 1.36, 1.771, 1.021, Inf]
[0.0, 2.818, 0.872, 2.619, 2.334, 1.97, 1.36, 1.771, 1.021, 2.696]
[0.0, 2.601, 0.872, 2.619, 2.334, 1.97, 1.36, 1.771, 1.021, 2.478]
10-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0s
2.600792423391039s
0.8719003118656656s
2.6190329085774406s
2.334107684888196s
1.96995040396643s
1.3599975906970543s
1.7711435879056907s
1.02086987187335s
2.4779254788367306s
julia>
eschnett@Redshift-8:~/src/jl/NotDijkstra (2025-01-21 11:31:30)
$ julia --project=@.
Documentation: https://docs.julialang.org
Type "?" for help, "]?" for Pkg help.
Version 1.11.2 (2024-12-01)
Official https://julialang.org/ release
julia> using NotDijkstra
julia>

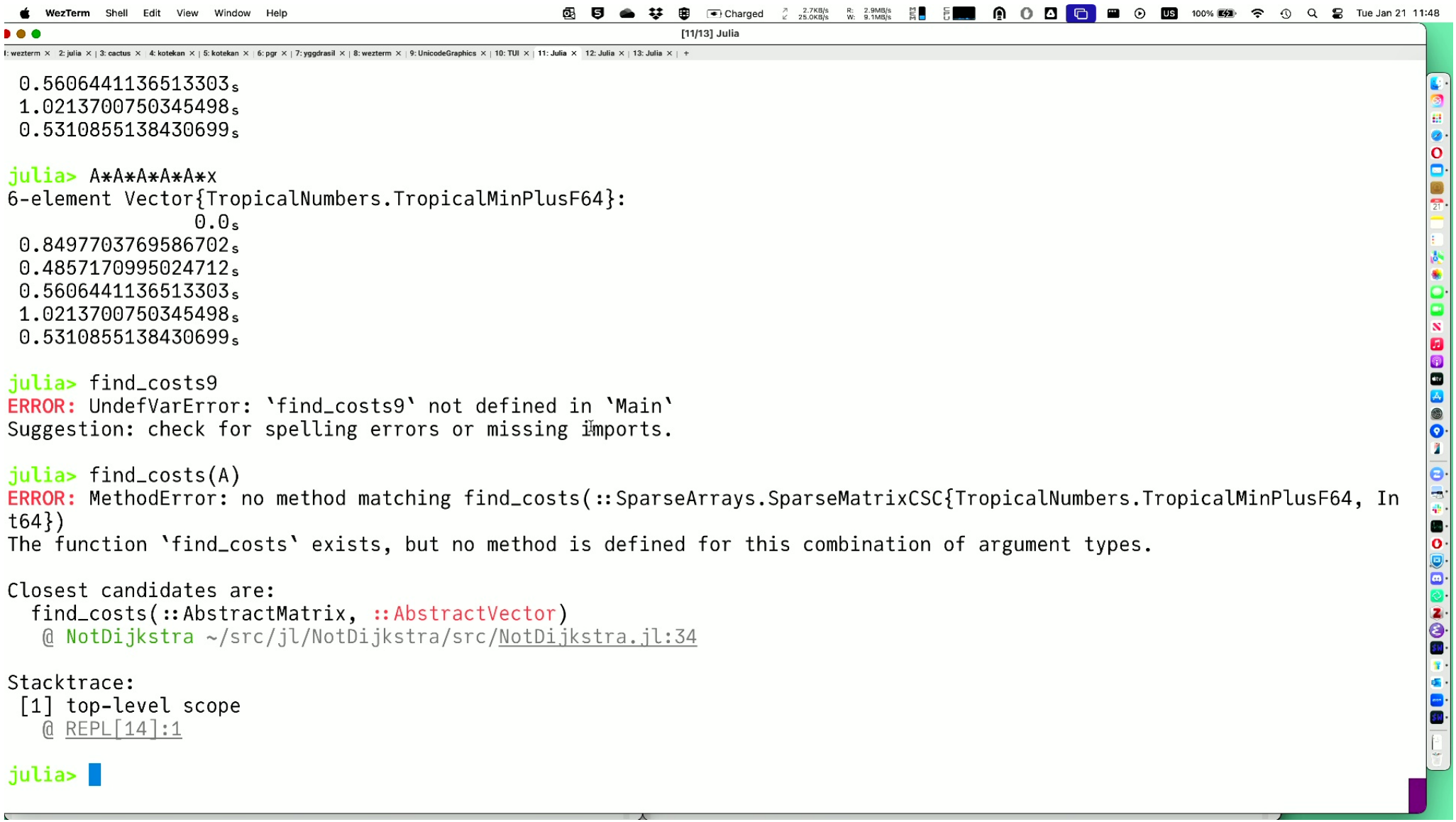

WezTerm Shell Edit View Window Help
Tue Jan 21 11:44
[11/13] Julia
10×10 SparseArrays.SparseMatrixCSC{TropicalNumbers.TropicalMinPlusF64, Int64} with 32 stored entries:
0.0s • • … • •
• 0.0s • • •
0.8719003118656656s 0.05550151483328558s 0.0s 0.9935300439902324s •
• • • • •
• • • • •
• 0.125643803387122s • … • 0.18447093750515298s
• • 0.7057980161772197s 0.3391277188237043s •
• • • 0.7502737160323407s •
• • 0.14896956000768435s 0.0s 0.8957801727894336s
• 0.7690538188199524s • • 0.0s
julia> A = make_graph(6)
6×6 SparseArrays.SparseMatrixCSC{TropicalNumbers.TropicalMinPlusF64, Int64} with 17 stored entries:
0.0s • 0.7293677262492938s • • 0.9469708549161011s
• 0.0s • 0.6429511466880946s • 0.3186848631156003s
0.4857170995024712s • 0.0s 0.11333391522005432s 0.7297566008797851s •
• • 0.07492701414885905s 0.0s 0.5899681807658527s •
• • • • 0.0s 0.4902845611914799s
• • 0.045368414340598684s • • 0.0s
julia> Matrix(A)
6×6 Matrix{TropicalNumbers.TropicalMinPlusF64}:
0.0s Infs 0.7293677262492938s Infs Infs 0.9469708549161011s
Infs 0.0s Infs 0.6429511466880946s Infs 0.3186848631156003s
0.4857170995024712s Infs 0.0s 0.11333391522005432s 0.7297566008797851s Infs
Infs Infs 0.07492701414885905s 0.0s 0.5899681807658527s Infs
Infs Infs Infs Infs 0.0s 0.4902845611914799s
Infs Infs 0.045368414340598684s Infs Infs 0.0s
julia>

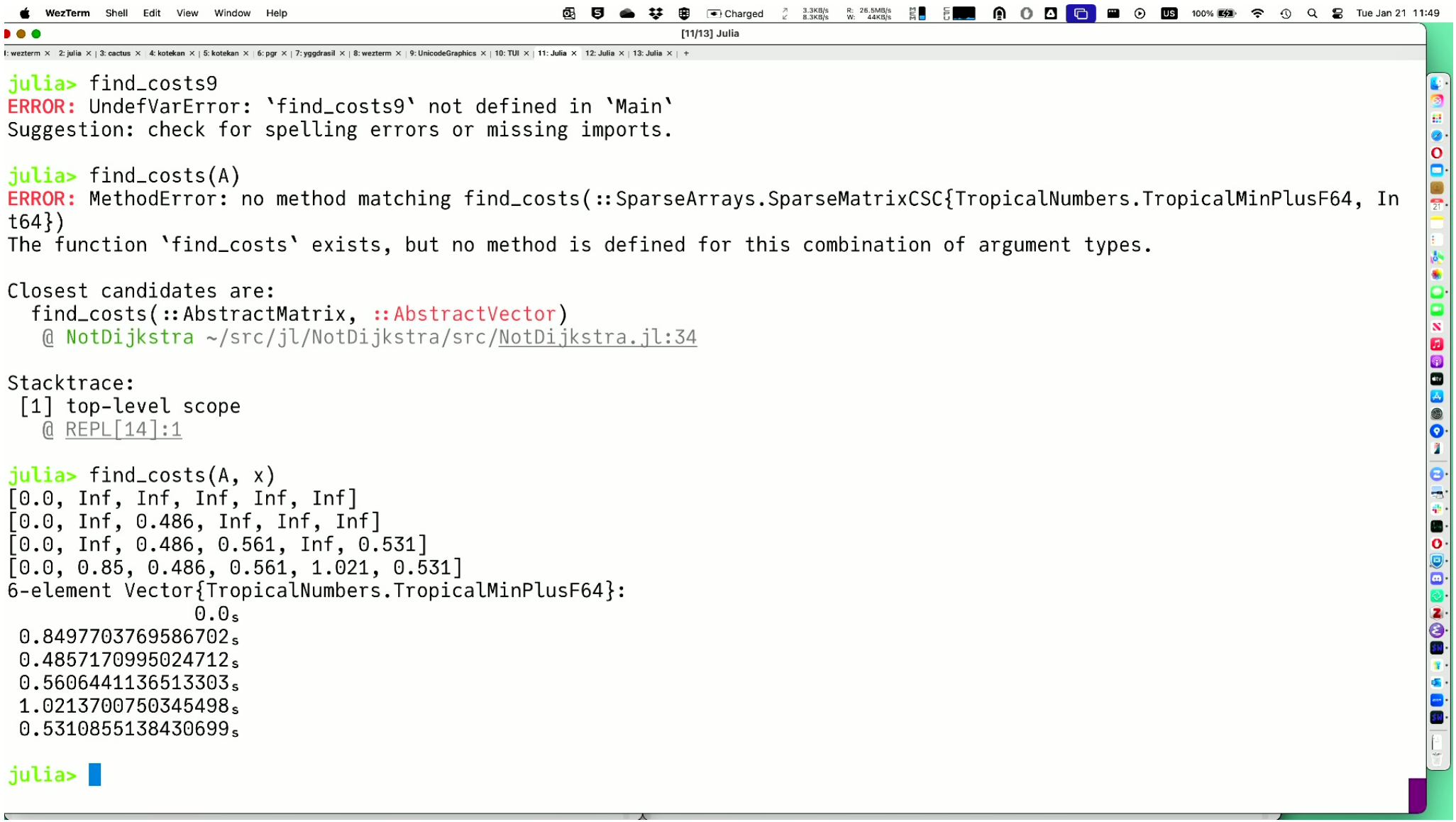

WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 11:44
[11/13] Julia
1: wezterm × | 2: julia × | 3: cactus × | 4: kotekan × | 5: kotekan × | 6: pgr × | 7: yggdrasil × | 8: wezterm × | 9: UnicodeGraphics × | 10: TUI × | 11: Julia × | 12: Julia × | 13: Julia × | +
s.var"#8#9")
@ SparseArrays ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/sparsematrix.jl:1361
[5] ftranspose(A::SparseArrays.SparseMatrixCSC{…}, f::Function, eltype::Type{…})
@ SparseArrays ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/sparsematrix.jl:1455
[6] copy
@ ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/sparsematrix.jl:1460 [inlined]
[7] SparseMatrixCSC
@ ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/sparsematrix.jl:921 [inlined]
[8] _sparsifystructured
@ ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/higherorderfns.jl:1178 [inlined]
[9] map
@ ./tuple.jl:356 [inlined]
[10] copy
@ ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/higherorderfns.jl:1167 [inlined]
[11] materialize
@ ./broadcast.jl:867 [inlined]
[12] broadcast_preserving_zero_d
@ ./broadcast.jl:856 [inlined]
[13] +(A::SparseArrays.SparseMatrixCSC{…}, B::LinearAlgebra.Adjoint{…})
@ Base ./arraymath.jl:8
[14] top-level scope
@ REPL[5]:1
Some type information was truncated. Use `show(err)` to see complete types.
julia>

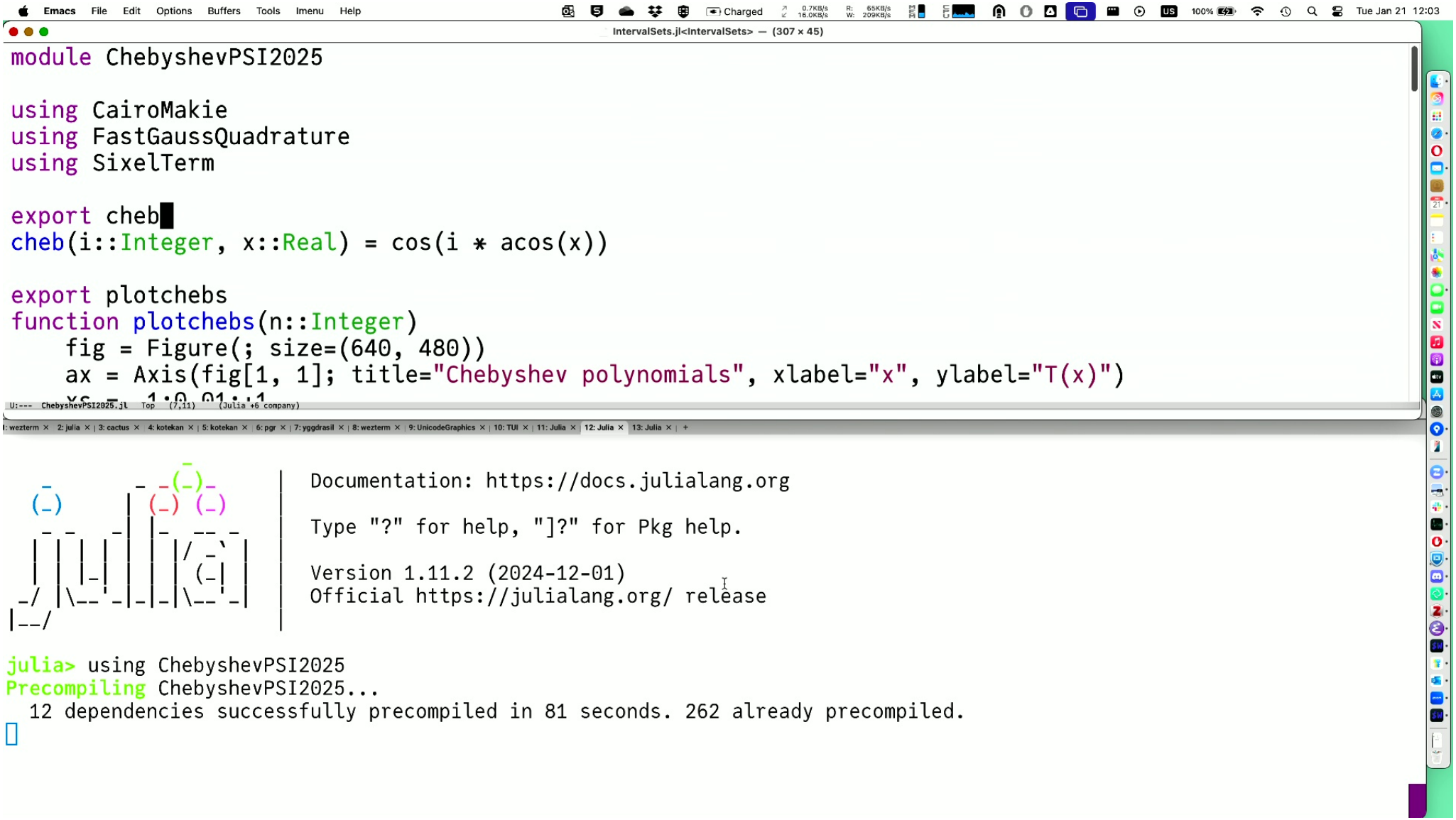

WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 11:45
[11/13] Julia
1: wezterm × 2: julia × 3: cactus × 4: kotekan × 5: kotekan × 6: pgr × 7: yggdrasil × 8: wezterm × 9: UnicodeGraphics × 10: TUI × 11: Julia × 12: Julia × 13: Julia ×
[7] SparseMatrixCSC
@ ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/sparsematrix.jl:921 [inlined]
[8] _sparsifystructured
@ ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/higherorderfns.jl:1178 [inlined]
[9] map
@ ./tuple.jl:356 [inlined]
[10] copy
@ ~/.julia/juliaup/julia-1.11.2+0.aarch64.apple.darwin14/share/julia/stdlib/v1.11/SparseArrays/src/higherorderfns.jl:1167 [inlined]
[11] materialize
@ ./broadcast.jl:867 [inlined]
[12] broadcast_preserving_zero_d
@ ./broadcast.jl:856 [inlined]
[13] +(A::SparseArrays.SparseMatrixCSC{…}, B::LinearAlgebra.Adjoint{…})
@ Base ./arraymath.jl:8
[14] top-level scope
@ REPL[5]:1
Some type information was truncated. Use `show(err)` to see complete types.
julia> Matrix(A)
6×6 Matrix{TropicalNumbers.TropicalMinPlusF64}:
0.0s Infs 0.7293677262492938s Infs Infs 0.9469708549161011s
Infs 0.0s Infs 0.6429511466880946s Infs 0.3186848631156003s
0.4857170995024712s Infs 0.0s 0.11333391522005432s 0.7297566008797851s Infs
Infs Infs 0.07492701414885905s 0.0s 0.5899681807658527s Infs
Infs Infs Infs Infs 0.0s 0.4902845611914799s
Infs Infs 0.045368414340598684s Infs Infs 0.0s
julia>


WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 11:46
[11/13] Julia
@ REPL[5]:1
Some type information was truncated. Use `show(err)` to see complete types.
julia> Matrix(A)
6×6 Matrix{TropicalNumbers.TropicalMinPlusF64}:
0.0s Infs 0.7293677262492938s Infs Infs 0.9469708549161011s
Infs 0.0s Infs 0.6429511466880946s Infs 0.3186848631156003s
0.4857170995024712s Infs 0.0s 0.11333391522005432s 0.7297566008797851s Infs
Infs Infs 0.07492701414885905s 0.0s 0.5899681807658527s Infs
Infs Infs Infs Infs 0.0s 0.4902845611914799s
Infs Infs 0.045368414340598684s Infs Infs 0.0s
julia> x = make_initial(6)
6-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0s
Infs
Infs
Infs
Infs
Infs
julia> A*x
6-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0s
Infs
0.4857170995024712s
Infs
Infs
Infs
julia>

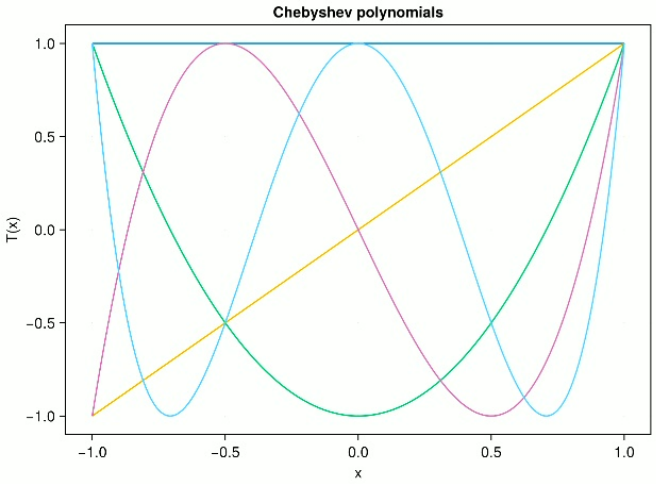

WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 11:47
[11/13] Julia
1: wezterm 2: julia 3: cactus 4: kotekan 5: kotekan 6: pgr 7: yggdrasil 8: wezterm 9: UnicodeGraphics 10: TUI 11: Julia 12: Julia 13: Julia
Inf s
0.5310855138430699 s
julia> A*A*A*x
6-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0 s
0.8497703769586702 s
0.4857170995024712 s
0.5606441136513303 s
1.0213700750345498 s
0.5310855138430699 s
julia> A*A*A*A*x
6-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0 s
0.8497703769586702 s
0.4857170995024712 s
0.5606441136513303 s
1.0213700750345498 s
0.5310855138430699 s
julia> A*A*A*A*A*x
6-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0 s
0.8497703769586702 s
0.4857170995024712 s
0.5606441136513303 s
1.0213700750345498 s
0.5310855138430699 s
julia>


WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 11:48
[11/13] Julia
1: wezterm 2: julia 3: cactus 4: kotekan 5: kotekan 6: pgr 7: yggdrasil 8: wezterm 9: UnicodeGraphics 10: TUI 11: Julia 12: Julia 13: Julia
0.5606441136513303s
1.0213700750345498s
0.5310855138430699s
julia> A*A*A*A*A*x
6-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0s
0.8497703769586702s
0.4857170995024712s
0.5606441136513303s
1.0213700750345498s
0.5310855138430699s
julia> find_costs9
ERROR: UndefVarError: `find_costs9` not defined in `Main`
Suggestion: check for spelling errors or missing imports.
julia> find_costs(A)
ERROR: MethodError: no method matching find_costs(::SparseArrays.SparseMatrixCSC{TropicalNumbers.TropicalMinPlusF64, Int64})
The function `find_costs` exists, but no method is defined for this combination of argument types.
Closest candidates are:
find_costs(::AbstractMatrix, ::AbstractVector)
@ NotDijkstra ~/src/jl/NotDijkstra/src/NotDijkstra.jl:34
Stacktrace:
[1] top-level scope
@ REPL[14]:1
julia>

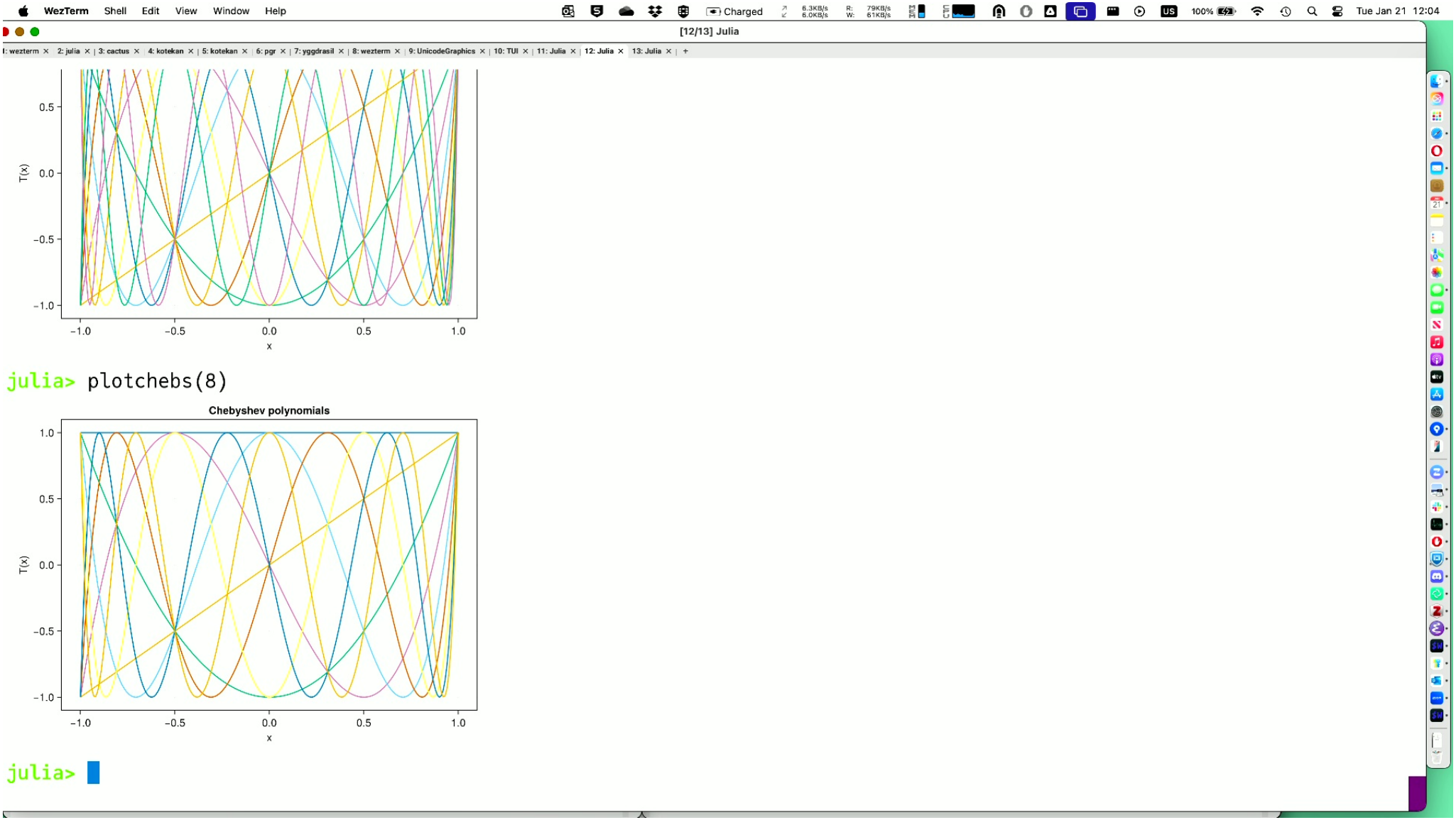

WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 11:49
[11/13] Julia
julia> find_costs9
ERROR: UndefVarError: `find_costs9` not defined in `Main`
Suggestion: check for spelling errors or missing imports.
julia> find_costs(A)
ERROR: MethodError: no method matching find_costs(::SparseArrays.SparseMatrixCSC{TropicalNumbers.TropicalMinPlusF64, Int64})
The function `find_costs` exists, but no method is defined for this combination of argument types.
Closest candidates are:
find_costs(::AbstractMatrix, ::AbstractVector)
@ NotDijkstra ~/src/jl/NotDijkstra/src/NotDijkstra.jl:34
Stacktrace:
[1] top-level scope
@ REPL[14]:1
julia> find_costs(A, x)
[0.0, Inf, Inf, Inf, Inf, Inf]
[0.0, Inf, 0.486, Inf, Inf, Inf]
[0.0, Inf, 0.486, 0.561, Inf, 0.531]
[0.0, 0.85, 0.486, 0.561, 1.021, 0.531]
6-element Vector{TropicalNumbers.TropicalMinPlusF64}:
0.0ₛ
0.8497703769586702ₛ
0.4857170995024712ₛ
0.5606441136513303ₛ
1.0213700750345498ₛ
0.5310855138430699ₛ
julia>


Emacs File Edit Options Buffers Tools Imenu Help
Charged
Tue Jan 21 12:03
IntervalSets.jl<IntervalSets> — (307 × 45)
module ChebyshevPSI2025
using CairoMakie
using FastGaussQuadrature
using SixelTerm
export cheb
cheb(i::Integer, x::Real) = cos(i * acos(x))
export plotchebs
function plotchebs(n::Integer)
fig = Figure(; size=(640, 480))
ax = Axis(fig[1, 1]; title="Chebyshev polynomials", xlabel="x", ylabel="T(x)")
U:--- ChebyshevPSI2025.jl Top (7,11) (Julia +6 company)
1: wezterm 2: julia 3: cactus 4: kotekan 5: kotekan 6: pgr 7: yggdrasil 8: wezterm 9: UnicodeGraphics 10: TUI 11: Julia 12: Julia 13: Julia
Documentation: https://docs.julialang.org
Type "?" for help, "]?" for Pkg help.
Version 1.11.2 (2024-12-01)
Official https://julialang.org/ release
julia> using ChebyshevPSI2025
Precompiling ChebyshevPSI2025...
12 dependencies successfully precompiled in 81 seconds. 262 already precompiled.

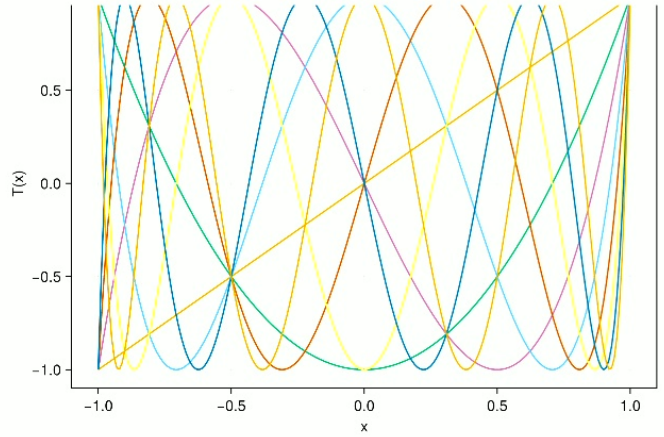
```
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.11.2 (2024-12-01)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> using ChebyshevPSI2025
Precompiling ChebyshevPSI2025...
  12 dependencies successfully precompiled in 81 seconds. 262 already precompiled.

julia> plotchebs(4)
```




```
julia>
```

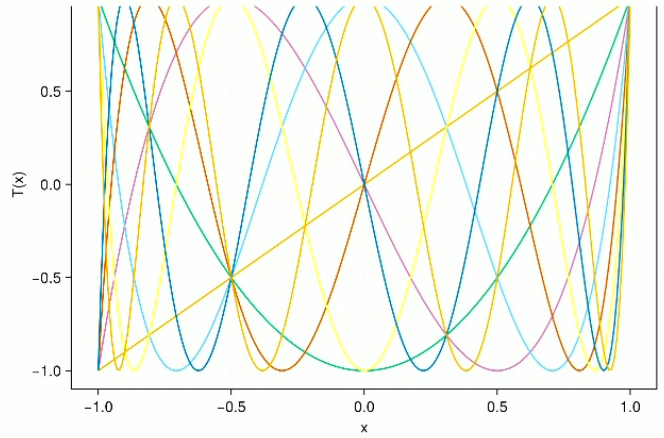

WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 12:04
[12/13] Julia
1: wezterm
2: julia
3: cactus
4: kotekan
5: kotekan
6: pgr
7: yggdrasil
8: wezterm
9: UnicodeGraphics
10: TUI
11: Julia
12: Julia
13: Julia
T(x)
x
julia> plotchebs(8)
Chebyshev polynomials
julia>

```
module ChebyshevPSI2025

using CairoMakie
using FastGaussQuadrature
using SixelTerm

export cheb
cheb(i::Integer, x::Real) = cos(i * acos(x))

export plotchebs
function plotchebs(n::Integer)
    fig = Figure(; size=(640, 480))
    ax = Axis(fig[1, 1]; title="Chebyshev polynomials", xlabel="x", ylabel="T(x)")
```



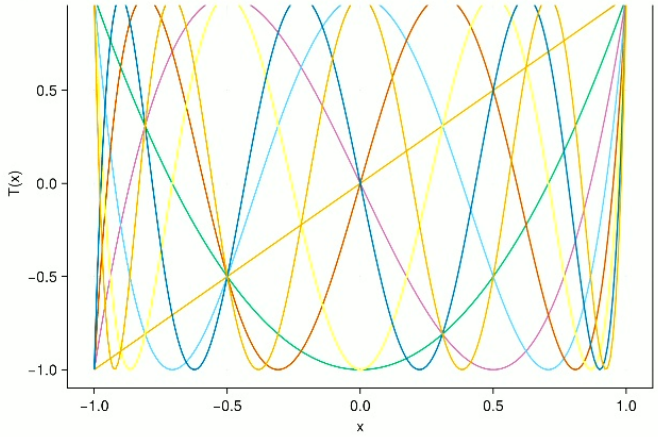

```
julia>
```

```
    for i in 0:n
        lines!(xs, cheb.(i, xs))
    end
    return fig
end

# We promise we won't use more than that many Chebyshev coefficients
const num_quadrature_points = 100
const chebdot_points, chebdot_weights = gausschebyshevt(num_quadrature_points)
const legdot_points, legdot_weights = gausslegendre(num_quadrature_points)

export chebdot
function chebdot(f, g)
```



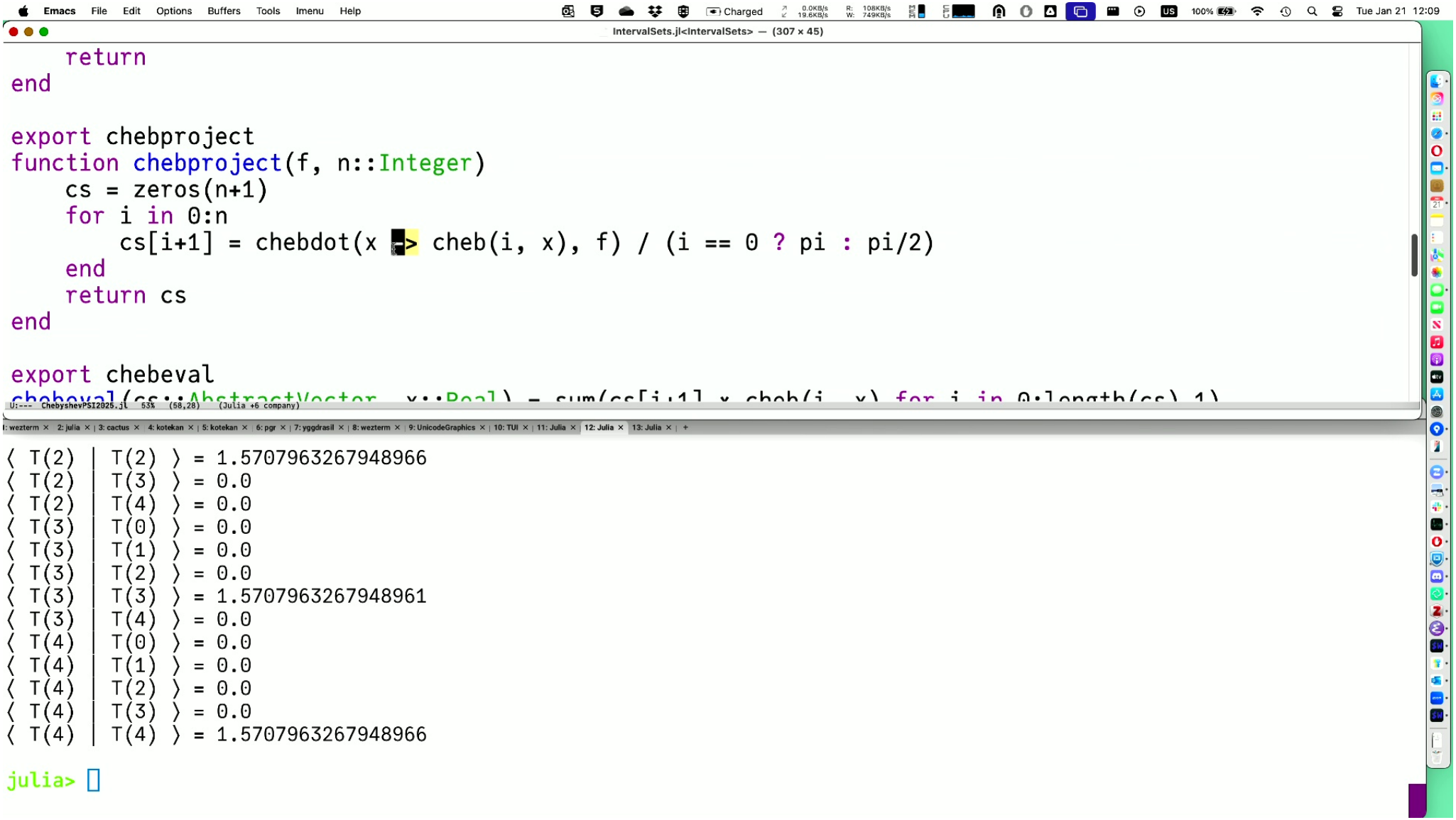

```
julia>
```

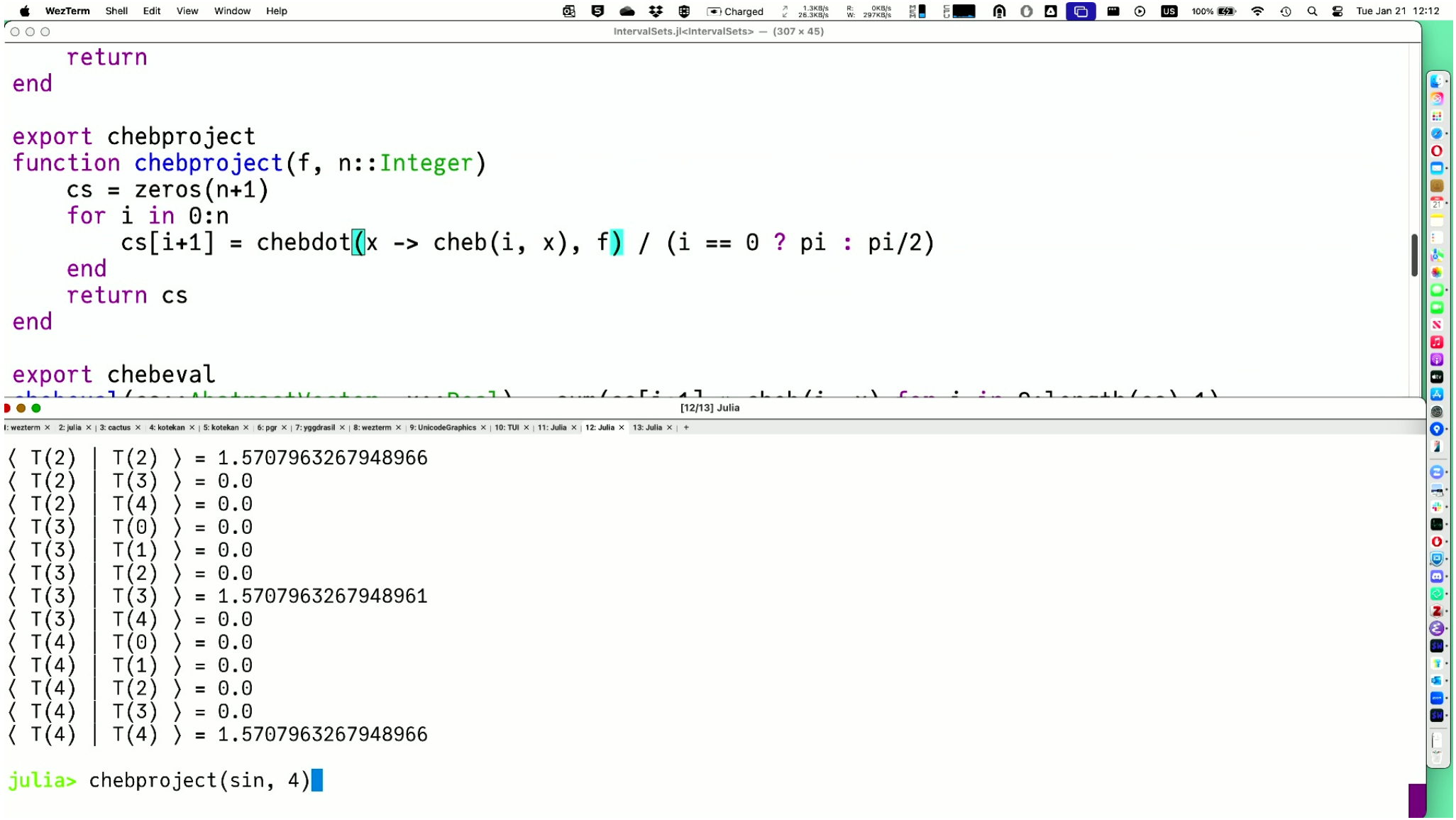
```
export check_chebdot
function check_chebdot(n::Integer)
    chop(x) = abs(x) < 10*eps(typeof(x)) ? zero(x) : x
    for i in 0:n, j in 0:n
        val = chebdot(x -> cheb(i, x), x -> cheb(j, x))
        println("⟨ T($i) | T($j) ⟩ = $(chop(val))")
    end
    return
end

export chebproject
function chebproject(f, n::Integer)
    cc = zeros(n+1)
```

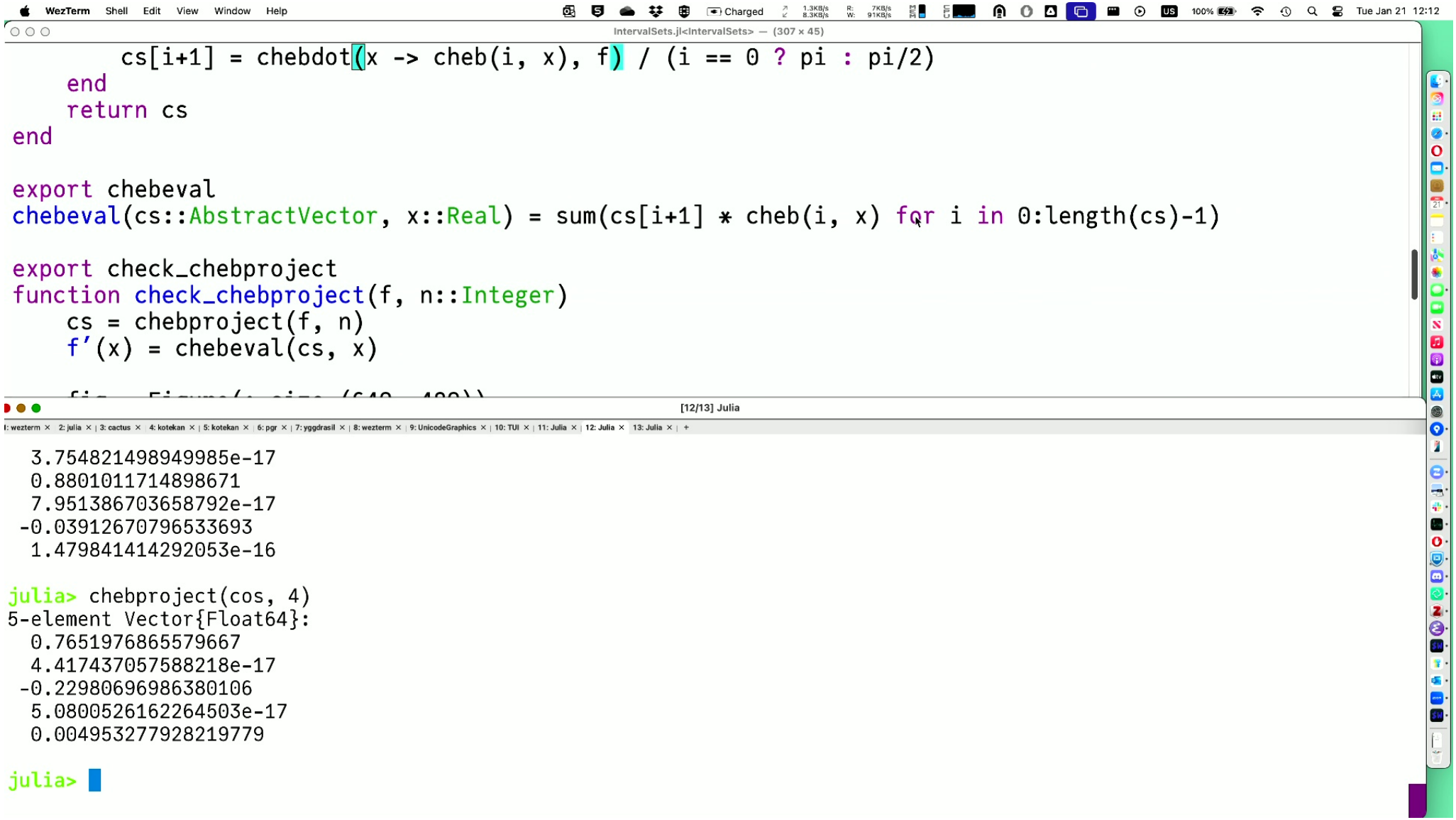



```
julia> check_chebdot(4)
```

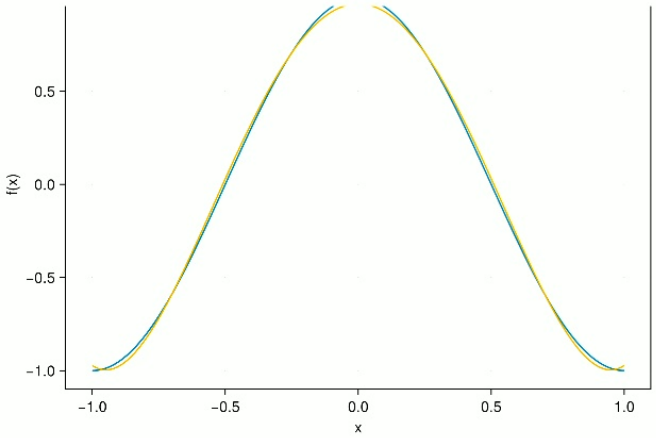

```
julia> check_chebdot(4)
⟨ T(0) | T(0) ⟩ = 3.1415926535897922
⟨ T(0) | T(1) ⟩ = 0.0
⟨ T(0) | T(2) ⟩ = 0.0
⟨ T(0) | T(3) ⟩ = 0.0
⟨ T(0) | T(4) ⟩ = 0.0
⟨ T(1) | T(0) ⟩ = 0.0
⟨ T(1) | T(1) ⟩ = 1.5707963267948966
⟨ T(1) | T(2) ⟩ = 0.0
⟨ T(1) | T(3) ⟩ = 0.0
⟨ T(1) | T(4) ⟩ = 0.0
⟨ T(2) | T(0) ⟩ = 0.0
⟨ T(2) | T(1) ⟩ = 0.0
⟨ T(2) | T(2) ⟩ = 1.5707963267948966
⟨ T(2) | T(3) ⟩ = 0.0
⟨ T(2) | T(4) ⟩ = 0.0
⟨ T(3) | T(0) ⟩ = 0.0
⟨ T(3) | T(1) ⟩ = 0.0
⟨ T(3) | T(2) ⟩ = 0.0
⟨ T(3) | T(3) ⟩ = 1.5707963267948961
⟨ T(3) | T(4) ⟩ = 0.0
⟨ T(4) | T(0) ⟩ = 0.0
⟨ T(4) | T(1) ⟩ = 0.0
⟨ T(4) | T(2) ⟩ = 0.0
⟨ T(4) | T(3) ⟩ = 0.0
⟨ T(4) | T(4) ⟩ = 1.5707963267948966

julia>
```


Emacs File Edit Options Buffers Tools Imenu Help
Charged
Tue Jan 21 12:09
IntervalSets.jl<IntervalSets> — (307 × 45)
return
end
export chebproject
function chebproject(f, n::Integer)
cs = zeros(n+1)
for i in 0:n
cs[i+1] = chebdot(x -> cheb(i, x), f) / (i == 0 ? pi : pi/2)
end
return cs
end
export chebeval
U:--- ChebyshevPSI2025.jl 53% (58,28) (Julia +6 company)
1: wezterm 2: julia 3: cactus 4: kotekan 5: kotekan 6: pgr 7: yggdrasil 8: wezterm 9: UnicodeGraphics 10: TUI 11: Julia 12: Julia 13: Julia
⟨ T(2) | T(2) ⟩ = 1.5707963267948966
⟨ T(2) | T(3) ⟩ = 0.0
⟨ T(2) | T(4) ⟩ = 0.0
⟨ T(3) | T(0) ⟩ = 0.0
⟨ T(3) | T(1) ⟩ = 0.0
⟨ T(3) | T(2) ⟩ = 0.0
⟨ T(3) | T(3) ⟩ = 1.5707963267948961
⟨ T(3) | T(4) ⟩ = 0.0
⟨ T(4) | T(0) ⟩ = 0.0
⟨ T(4) | T(1) ⟩ = 0.0
⟨ T(4) | T(2) ⟩ = 0.0
⟨ T(4) | T(3) ⟩ = 0.0
⟨ T(4) | T(4) ⟩ = 1.5707963267948966
julia>


WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 12:12
IntervalSets.jl<IntervalSets> — (307 × 45)
return
end
export chebproject
function chebproject(f, n::Integer)
cs = zeros(n+1)
for i in 0:n
cs[i+1] = chebdot(x -> cheb(i, x), f) / (i == 0 ? pi : pi/2)
end
return cs
end
export chebeval
[12/13] Julia
1: wezterm 2: julia 3: cactus 4: kotekan 5: kotekan 6: pgr 7: yggdrasil 8: wezterm 9: UnicodeGraphics 10: TUI 11: Julia 12: Julia 13: Julia
⟨ T(2) | T(2) ⟩ = 1.5707963267948966
⟨ T(2) | T(3) ⟩ = 0.0
⟨ T(2) | T(4) ⟩ = 0.0
⟨ T(3) | T(0) ⟩ = 0.0
⟨ T(3) | T(1) ⟩ = 0.0
⟨ T(3) | T(2) ⟩ = 0.0
⟨ T(3) | T(3) ⟩ = 1.5707963267948961
⟨ T(3) | T(4) ⟩ = 0.0
⟨ T(4) | T(0) ⟩ = 0.0
⟨ T(4) | T(1) ⟩ = 0.0
⟨ T(4) | T(2) ⟩ = 0.0
⟨ T(4) | T(3) ⟩ = 0.0
⟨ T(4) | T(4) ⟩ = 1.5707963267948966
julia> chebproject(sin, 4)

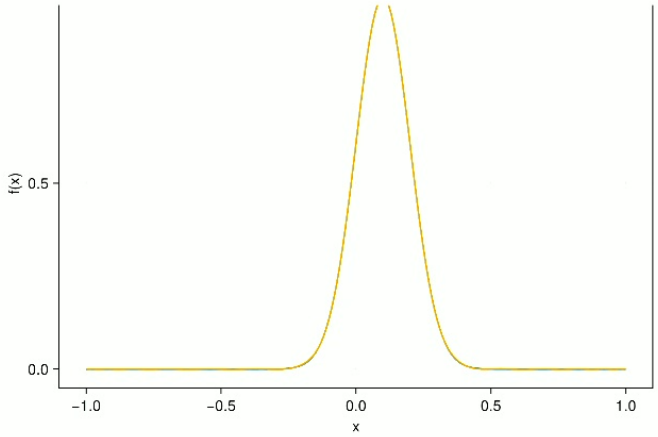

WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 12:12
IntervalSets.jl<IntervalSets> — (307 × 45)
cs[i+1] = chebdot(x -> cheb(i, x), f) / (i == 0 ? pi : pi/2)
end
return cs
end
export chebeval
chebeval(cs::AbstractVector, x::Real) = sum(cs[i+1] * cheb(i, x) for i in 0:length(cs)-1)
export check_chebproject
function check_chebproject(f, n::Integer)
cs = chebproject(f, n)
f′(x) = chebeval(cs, x)
[12/13] Julia
1: wezterm 2: julia 3: cactus 4: kotekan 5: kotekan 6: pgr 7: yggdrasil 8: wezterm 9: UnicodeGraphics 10: TUI 11: Julia 12: Julia 13: Julia
3.754821498949985e-17
0.8801011714898671
7.951386703658792e-17
-0.03912670796533693
1.479841414292053e-16
julia> chebproject(cos, 4)
5-element Vector{Float64}:
0.7651976865579667
4.417437057588218e-17
-0.22980696986380106
5.0800526162264503e-17
0.004953277928219779
julia>

```
    ax = Axis(fig[1, 1]; title="Chebyshev expansion (n=$n)", xlabel="x", ylabel="f(x)")
    xs = -1:0.01:+1
    lines!(xs, f.(xs))
    lines!(xs, f'.(xs))
    return fig
end

export gaussian
gaussian(x) = exp(-1/2 * ((x-0.1)/0.1)^2)

export check_convergence
function check_convergence(f, nmax::Integer)
    errors = zeros(nmax+1)
```

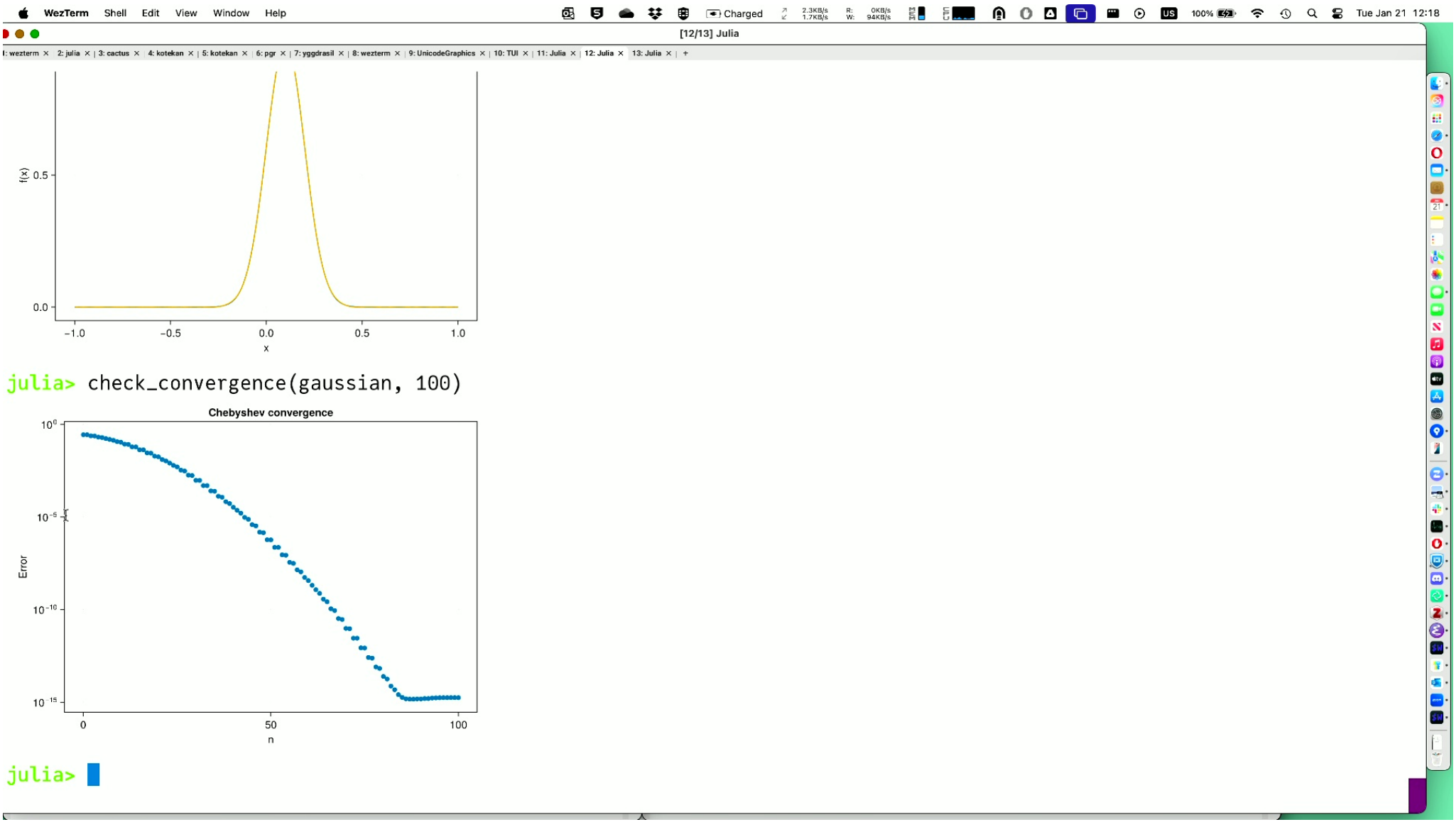


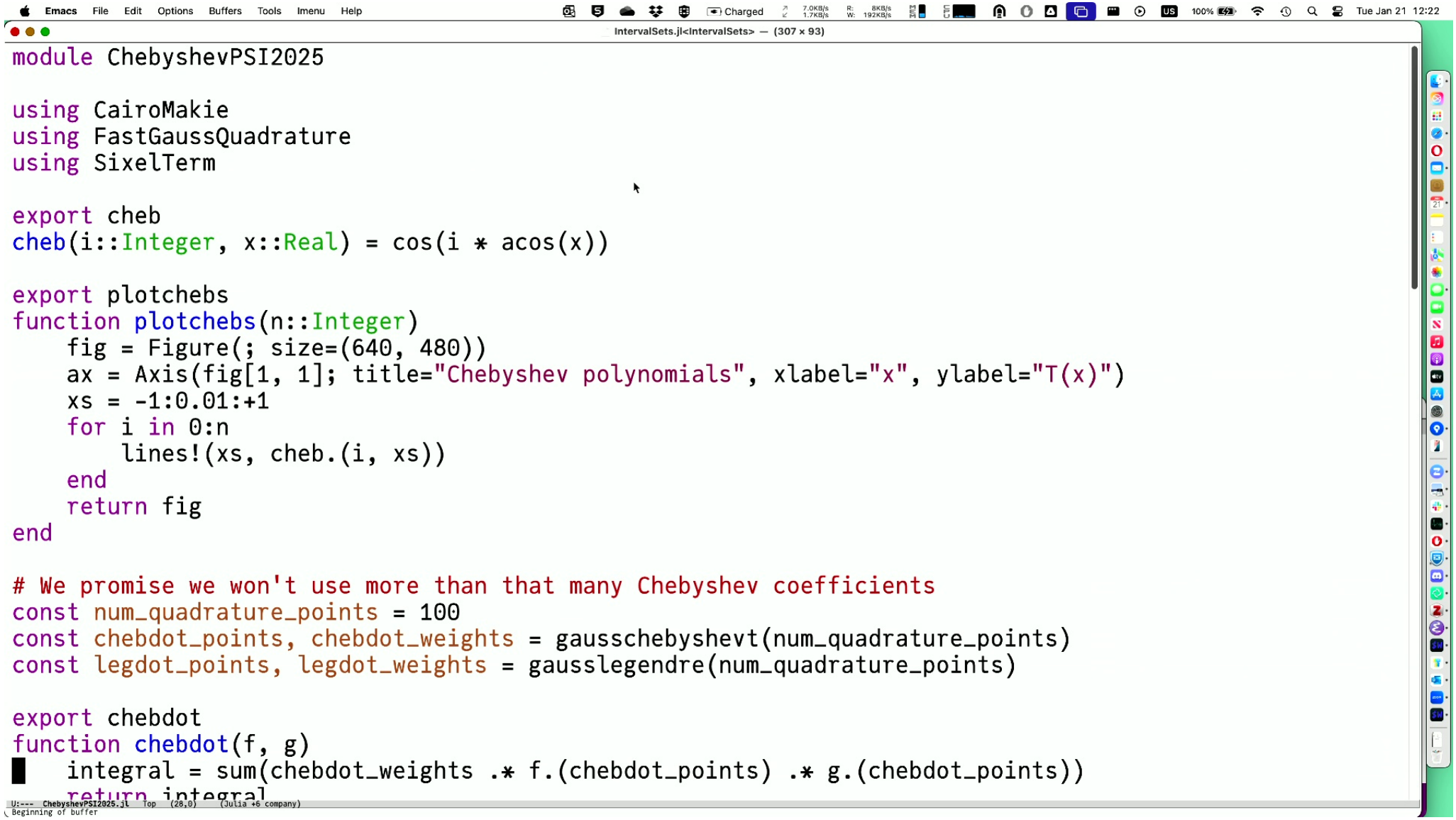

```
julia> check_chebproject(gaussian, 4)f
```

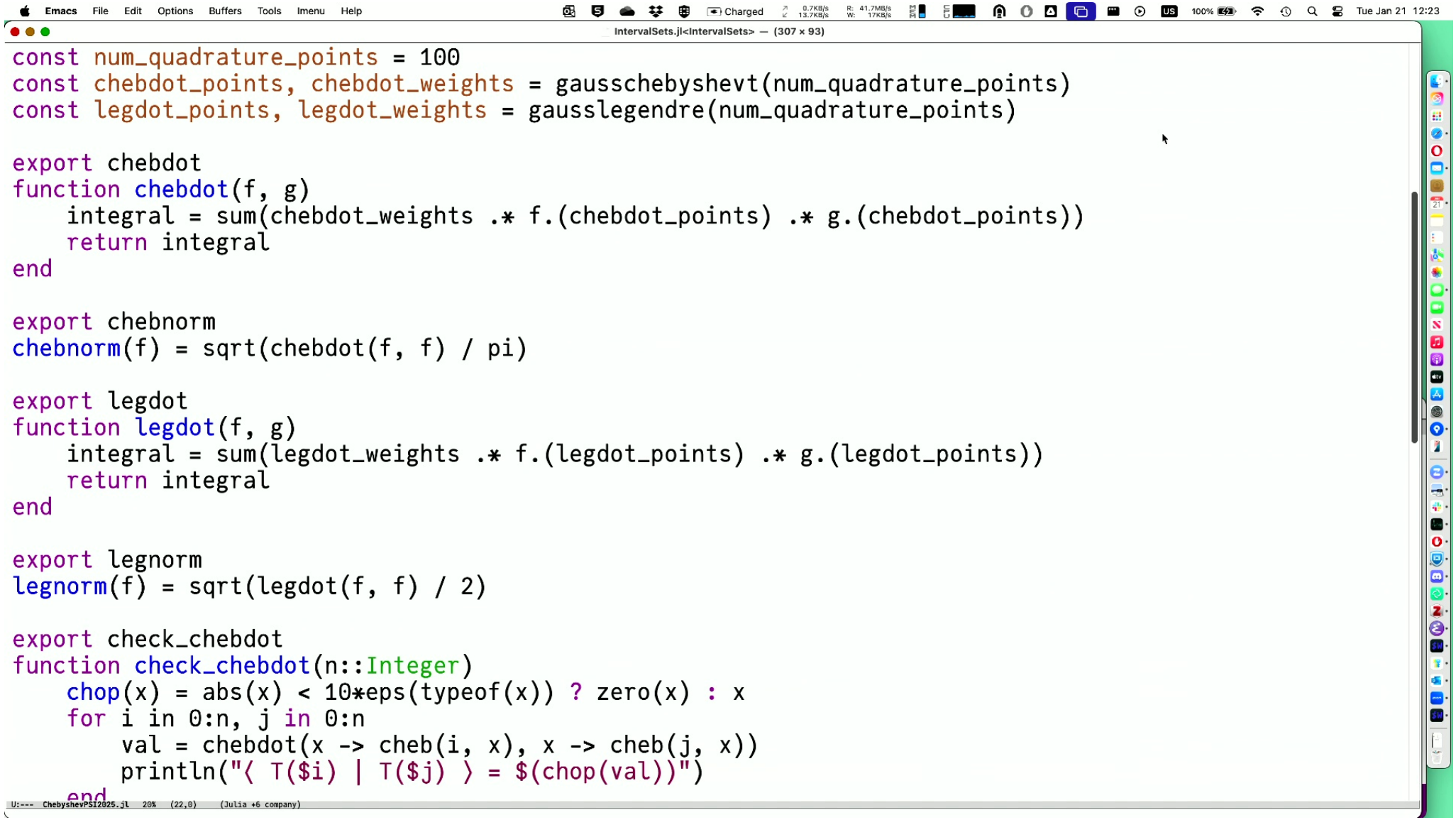
```
    ax = Axis(fig[1, 1]; title="Chebyshev expansion (n=$n)", xlabel="x", ylabel="f(x)")
    xs = -1:0.01:+1
    lines!(xs, f.(xs))
    lines!(xs, f'.(xs))
    return fig
end

export gaussian
gaussian(x) = exp(-1/2 * ((x-0.1)/0.1)^2)

export check_convergence
function check_convergence(f, nmax::Integer)
    errors = zeros(nmax+1)
```

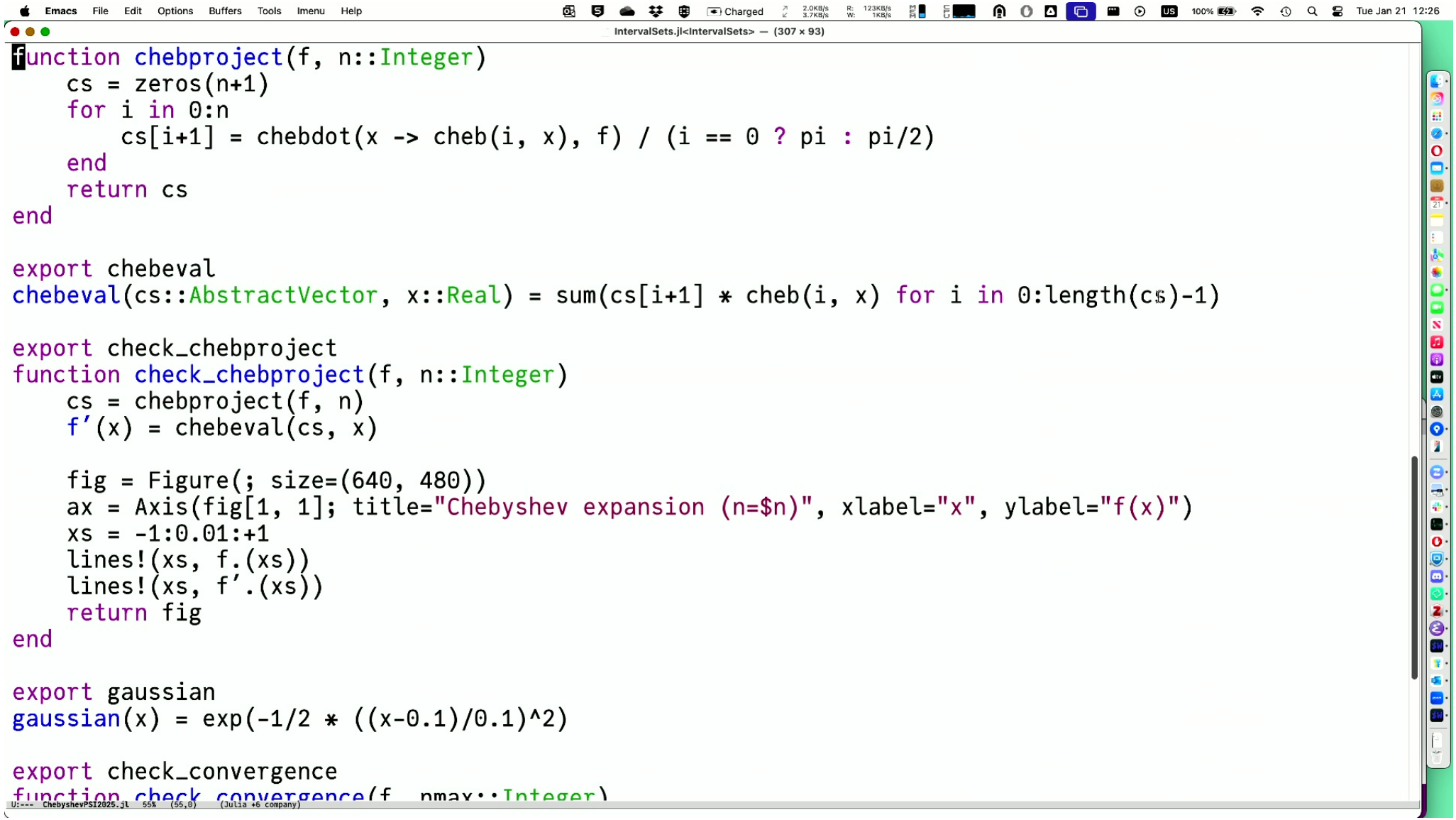


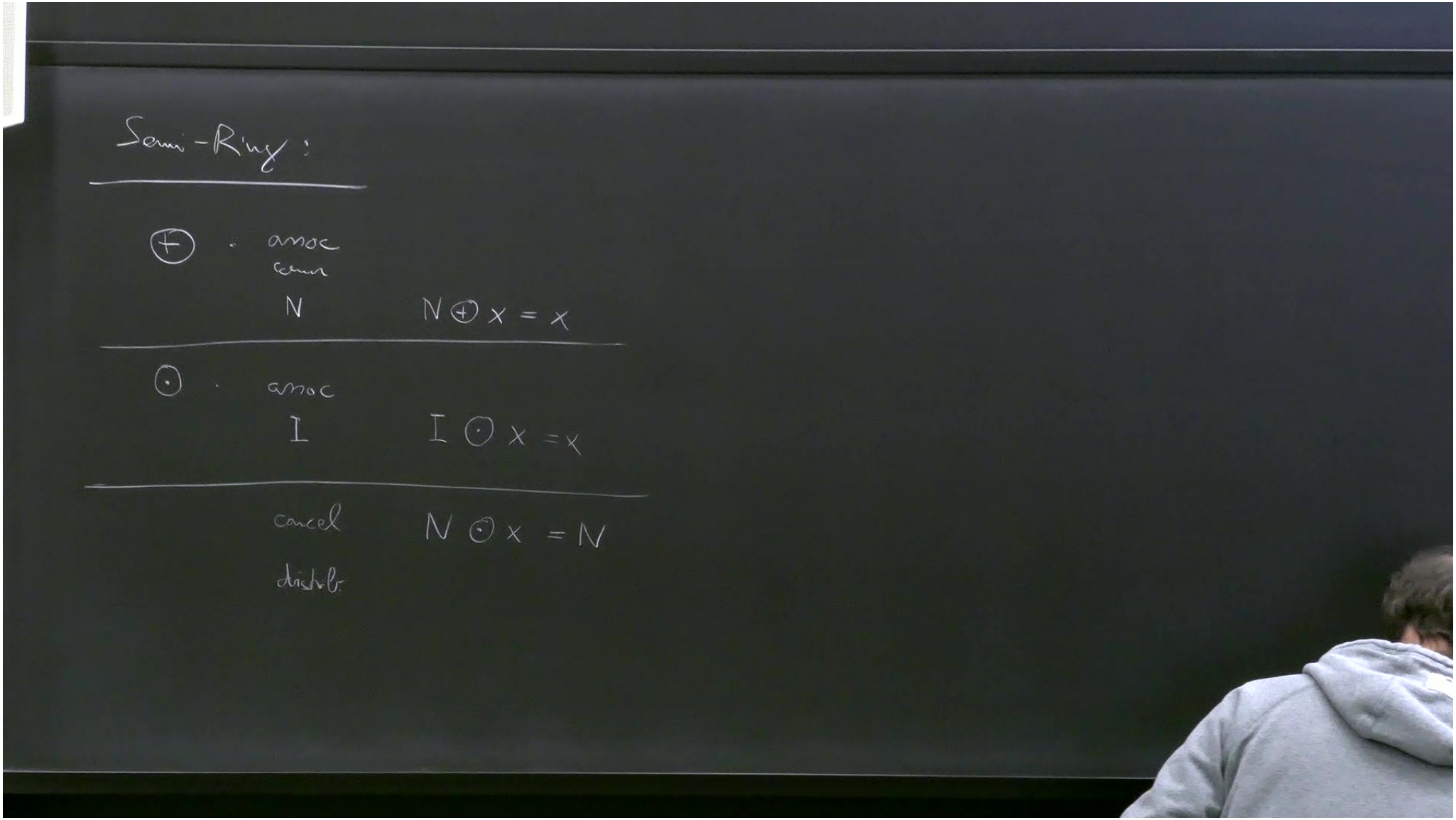

```
julia> check_convergence(gaussian,
```


WezTerm Shell Edit View Window Help
Charged
Tue Jan 21 12:18
[12/13] Julia
1: wezterm 2: julia 3: cactus 4: kotekan 5: kotekan 6: pgr 7: yggdrasil 8: wezterm 9: UnicodeGraphics 10: TUI 11: Julia 12: Julia 13: Julia
f(x)
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
x
julia> check_convergence(gaussian, 100)
Chebyshev convergence
Error
n
julia>


Emacs File Edit Options Buffers Tools Imenu Help
Charged
Tue Jan 21 12:22
IntervalSets.jl<IntervalSets> — (307 × 93)
module ChebyshevPSI2025
using CairoMakie
using FastGaussQuadrature
using SixelTerm
export cheb
cheb(i::Integer, x::Real) = cos(i * acos(x))
export plotchebs
function plotchebs(n::Integer)
fig = Figure(; size=(640, 480))
ax = Axis(fig[1, 1]; title="Chebyshev polynomials", xlabel="x", ylabel="T(x)")
xs = -1:0.01:+1
for i in 0:n
lines!(xs, cheb.(i, xs))
end
return fig
end
We promise we won't use more than that many Chebyshev coefficients
const num_quadrature_points = 100
const chebdot_points, chebdot_weights = gausschebyshevt(num_quadrature_points)
const legdot_points, legdot_weights = gausslegendre(num_quadrature_points)
export chebdot
function chebdot(f, g)
integral = sum(chebdot_weights .* f.(chebdot_points) .* g.(chebdot_points))
return integral
U:--- ChebyshevPSI2025.jl Top (28,0) (Julia +6 company)
Beginning of buffer


Emacs File Edit Options Buffers Tools Imenu Help
Charged
Tue Jan 21 12:23
IntervalSets.jl<IntervalSets> — (307 × 93)
const num_quadrature_points = 100
const chebdot_points, chebdot_weights = gausschebyshevt(num_quadrature_points)
const legdot_points, legdot_weights = gausslegendre(num_quadrature_points)
export chebdot
function chebdot(f, g)
integral = sum(chebdot_weights .* f.(chebdot_points) .* g.(chebdot_points))
return integral
end
export chebnorm
chebnorm(f) = sqrt(chebdot(f, f) / pi)
export legdot
function legdot(f, g)
integral = sum(legdot_weights .* f.(legdot_points) .* g.(legdot_points))
return integral
end
export legnorm
legnorm(f) = sqrt(legdot(f, f) / 2)
export check_chebdot
function check_chebdot(n::Integer)
chop(x) = abs(x) < 10*eps(typeof(x)) ? zero(x) : x
for i in 0:n, j in 0:n
val = chebdot(x -> cheb(i, x), x -> cheb(j, x))
println("⟨ T($i) | T($j) ⟩ = $(chop(val))")
end
U:--- ChebyshevPSI2025.jl 20% (22,0) (Julia +6 company)


Emacs File Edit Options Buffers Tools Imenu Help
Charged
Tue Jan 21 12:26
IntervalSets.jl<IntervalSets> — (307 × 93)
function chebproject(f, n::Integer)
cs = zeros(n+1)
for i in 0:n
cs[i+1] = chebdot(x -> cheb(i, x), f) / (i == 0 ? pi : pi/2)
end
return cs
end
export chebeval
chebeval(cs::AbstractVector, x::Real) = sum(cs[i+1] * cheb(i, x) for i in 0:length(cs)-1)
export check_chebproject
function check_chebproject(f, n::Integer)
cs = chebproject(f, n)
f′(x) = chebeval(cs, x)
fig = Figure(; size=(640, 480))
ax = Axis(fig[1, 1]; title="Chebyshev expansion (n=$n)", xlabel="x", ylabel="f(x)")
xs = -1:0.01:+1
lines!(xs, f.(xs))
lines!(xs, f′.(xs))
return fig
end
export gaussian
gaussian(x) = exp(-1/2 * ((x-0.1)/0.1)^2)
export check_convergence
function check_convergence(f, nmax::Integer)
U:--- ChebyshevPSI2025.jl 55% (55,0) (Julia +6 company)


Semi-Ring:
⊕ : assoc comm
N N ⊕ x = x
⊙ : assoc
I I ⊙ x = x
cancel N ⊙ x = N
distrib

Semi-Ring

$\oplus$: assoc
comm
N $\quad N \oplus x = x$

$\odot$: assoc
I $\quad I \odot x = x$

$N \odot x = N$

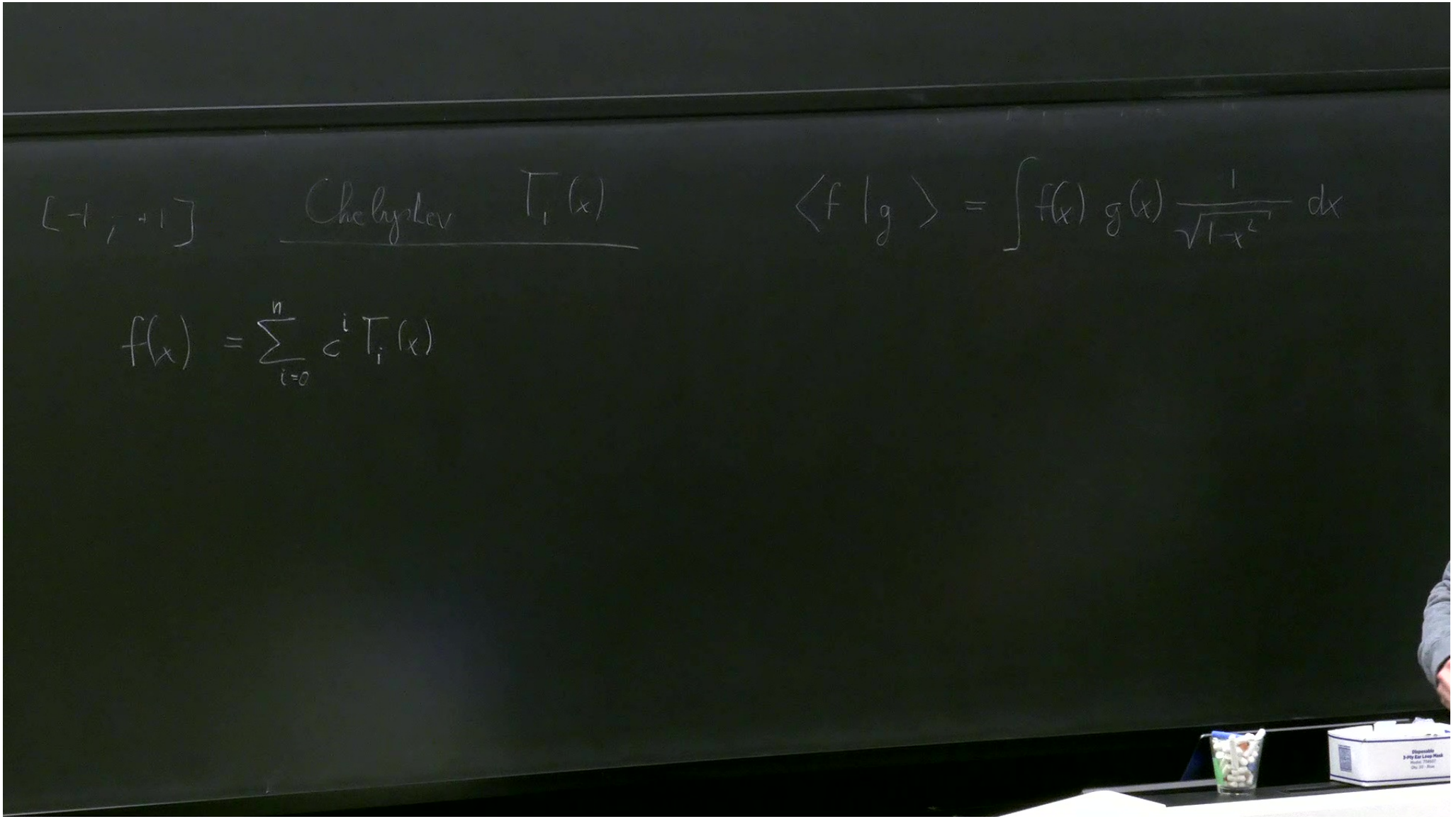
distrib

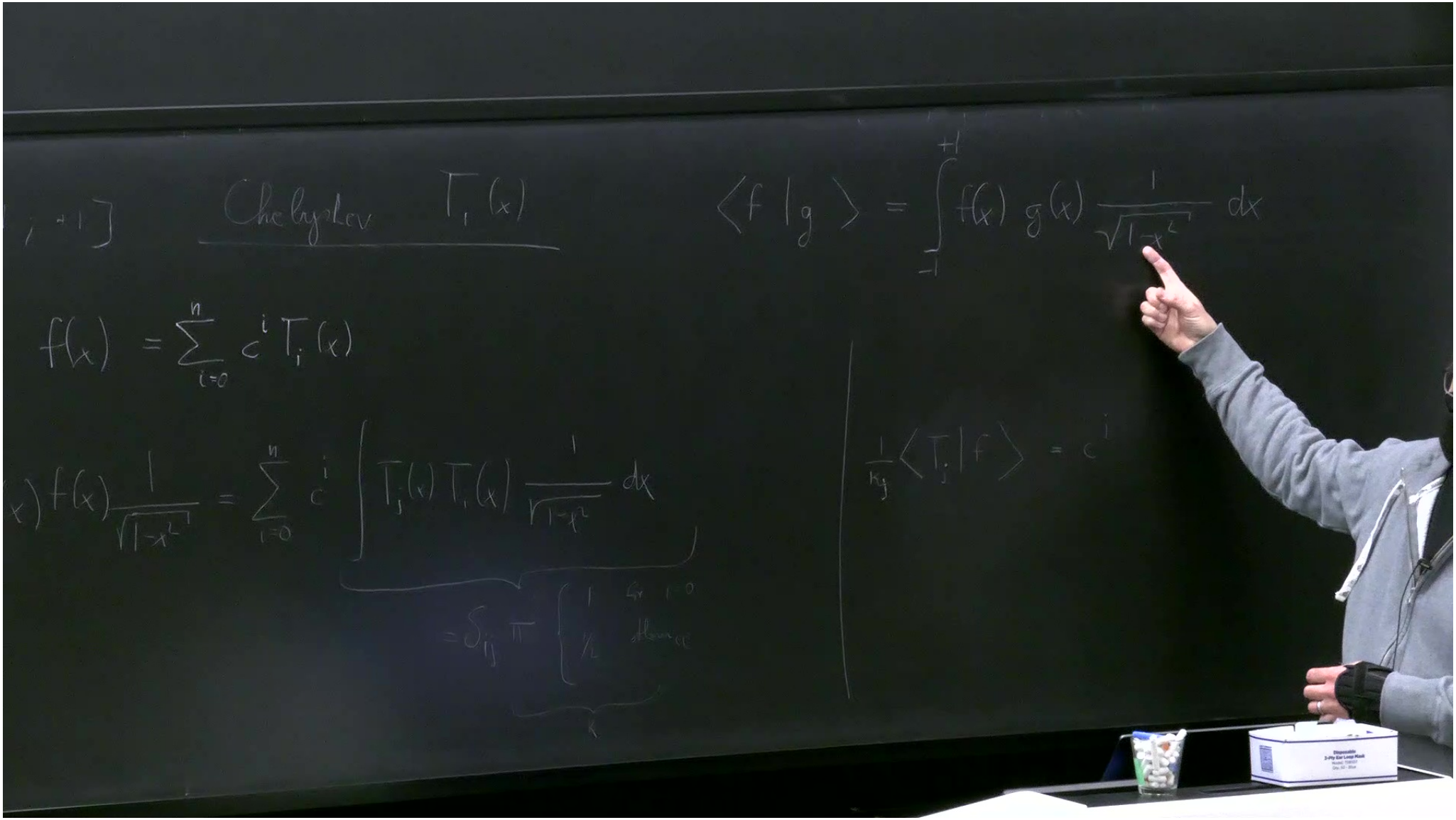
Tropical Numbers

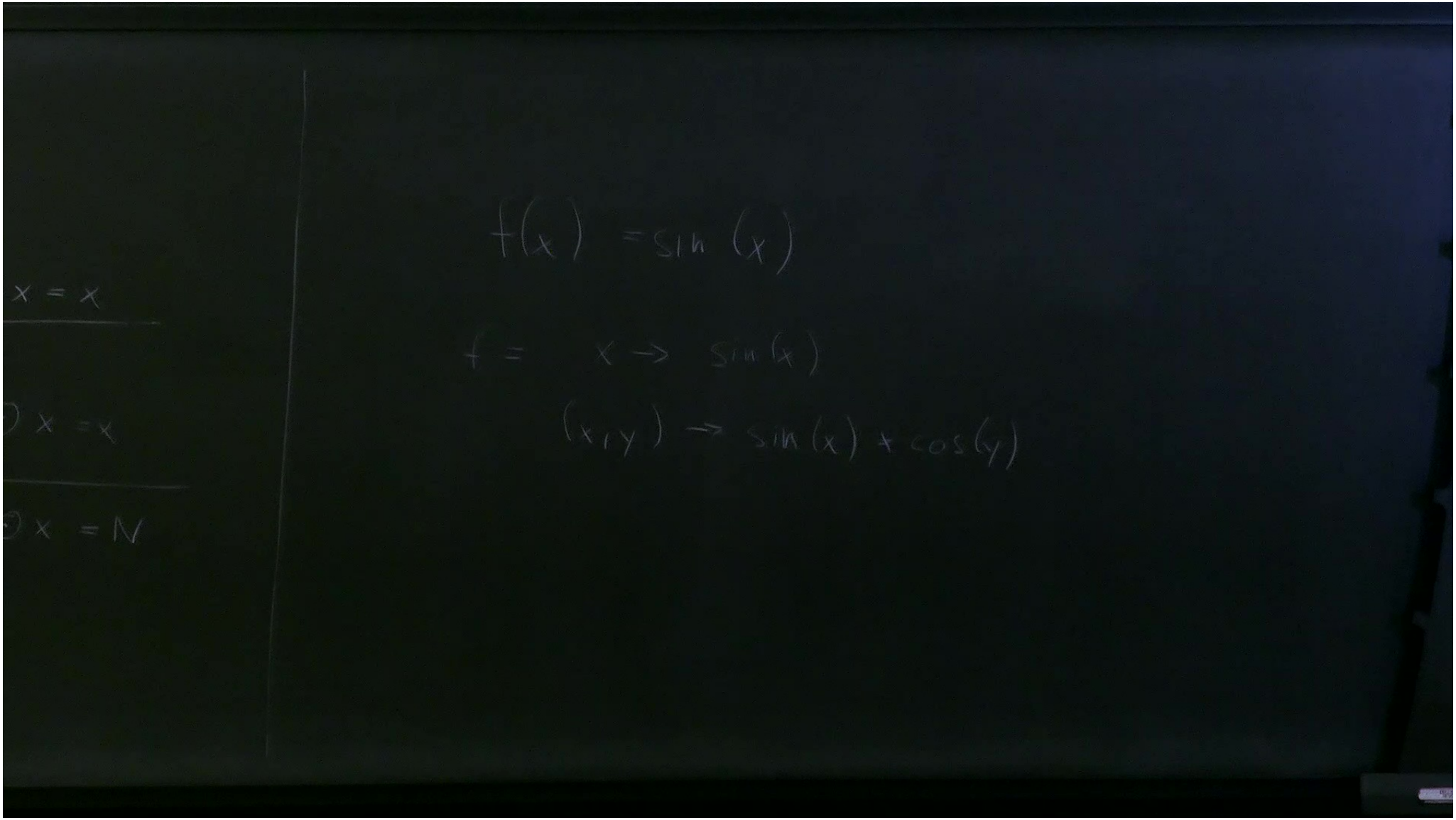
min − plus $\quad \mathbb{R}_{\geq 0} \cup \{\infty\}$

$\oplus$ $\quad \min$
$N = \infty$

$\odot$ $\quad +$
$I = 0$


[-1, +1]
Chebyshev $T_i(x)$
$\langle f | g \rangle = \int f(x)\, g(x) \frac{1}{\sqrt{1-x^2}}\, dx$
$f(x) = \sum_{i=0}^{n} c^i T_i(x)$


Chebyshev $T_i(x)$
$\langle f | g \rangle = \int_{-1}^{+1} f(x)\, g(x) \frac{1}{\sqrt{1-x^2}}\, dx$
$f(x) = \sum_{i=0}^{n} c^i T_i(x)$
$\frac{1}{K_j} \langle T_j | f \rangle = c^j$


x = x
x = x
x = N
f(x) = sin(x)
f = x -> sin(x)
(x,y) -> sin(x) + cos(y)