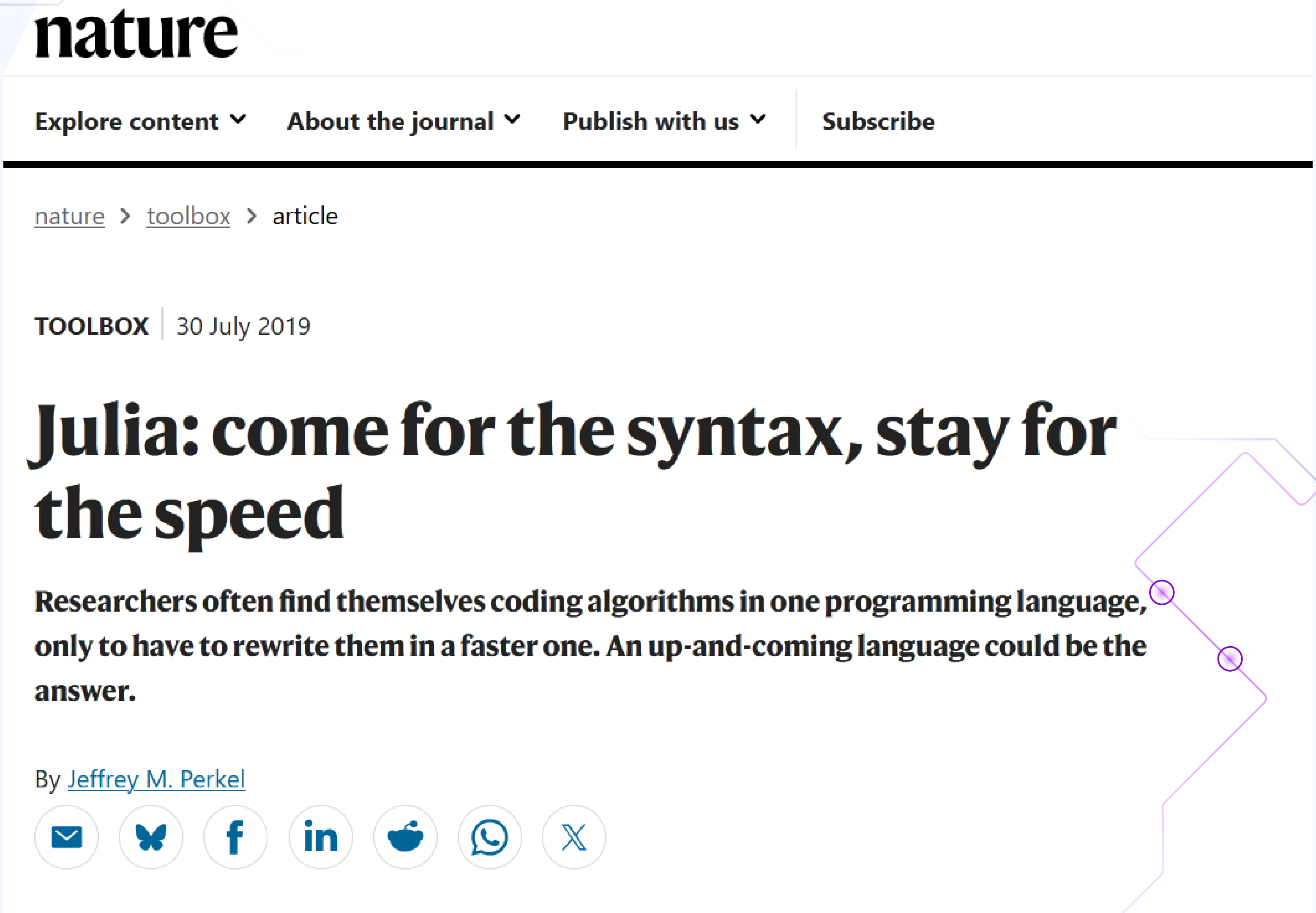
Physics in Julia: combining **Unitful.jl** and **DifferentialEquations.jl**

Daria Klimaszewska @agh.edu.pl
AGH University of Krakow, Poland
Departament of Physics and Applied Computer Science



TOOLBOX | 30 July 2019

Julia: come for the syntax, stay for the speed

Researchers often find themselves coding algorithms in one programming language, only to have to rewrite them in a faster one. An up-and-coming language could be the answer.

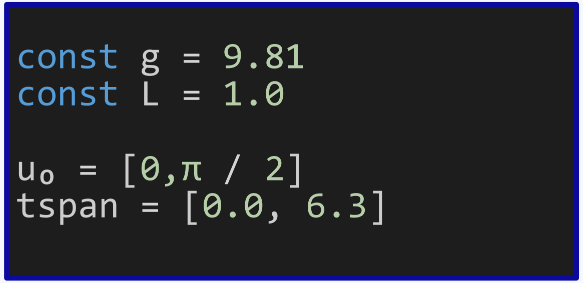
By Jeffrey M. Perkel

“For those who want some proof that physicists are human, the proof is in the idiocy of all the different units which they use for measuring energy”

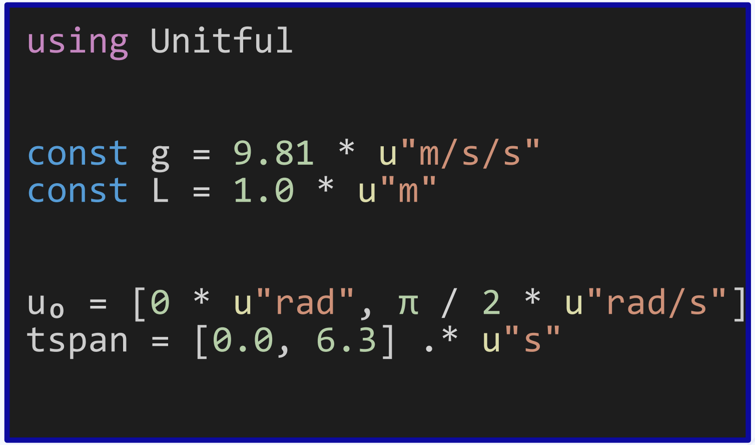
— Richard P. Feynman (Nobel’65), The Character of Physical Law

Unitful.jl

- **Code readability/maintainability**
- Conversions and prefixes
- Compile-time dimensionality check
- Tool for unit testing
- Automatic units on plots
- No runtime overhead



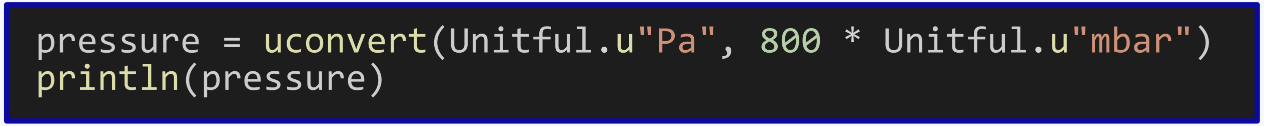
```julia
const g = 9.81
const L = 1.0

u₀ = [0,π / 2]
tspan = [0.0, 6.3]
```



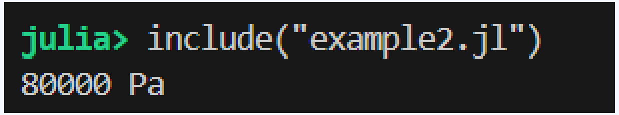
```julia
using Unitful


const g = 9.81 * u"m/s/s"
const L = 1.0 * u"m"


u₀ = [0 * u"rad", π / 2 * u"rad/s"]
tspan = [0.0, 6.3] .* u"s"
```

Unitful.jl

- Code readability/maintainability
- **Conversions and prefixes**
- Compile-time dimensionality check
- Tool for unit testing
- Automatic units on plots
- No runtime overhead



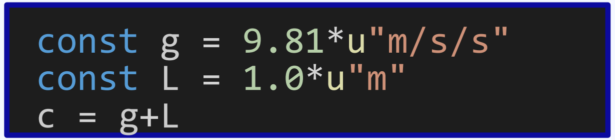
```julia
pressure = uconvert(Unitful.u"Pa", 800 * Unitful.u"mbar")
println(pressure)
```



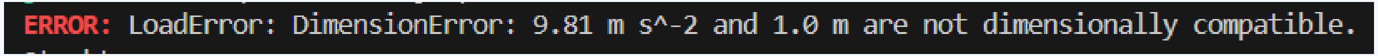
```
julia> include("example2.jl")
80000 Pa
```

Unitful.jl

- Code readability/maintainability
- Conversions and prefixes
- **Compile-time dimensionality check**
- Tool for unit testing
- Automatic units on plots
- No runtime overhead



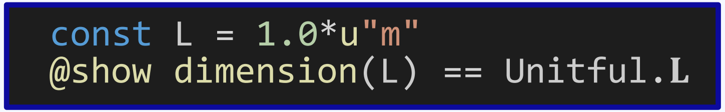
```julia
const g = 9.81*u"m/s/s"
const L = 1.0*u"m"
c = g+L
```



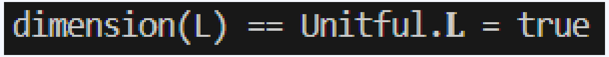
```
ERROR: LoadError: DimensionError: 9.81 m s^-2 and 1.0 m are not dimensionally compatible.
```

Unitful.jl

- Code readability/maintainability
- Conversions and prefixes
- Compile-time dimensionality check
- **Tool for unit testing**
- Automatic units on plots
- No runtime overhead



```julia
const L = 1.0*u"m"
@show dimension(L) == Unitful.L
```

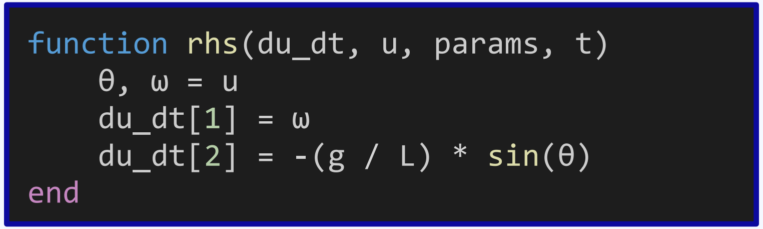


```
dimension(L) == Unitful.L = true
```

DifferentialEquations.jl

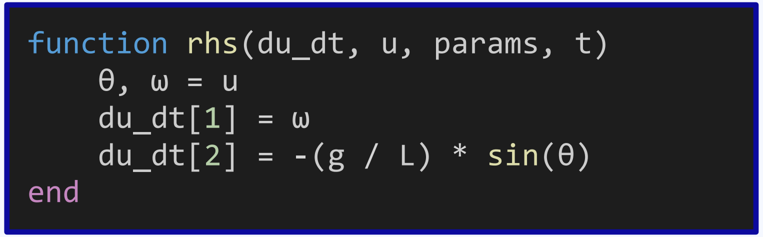
Image by Manfred Antranias Zimmer from Pixabay

$$\frac{d^2\theta}{dt^2} = \frac{g}{L} sin\theta$$



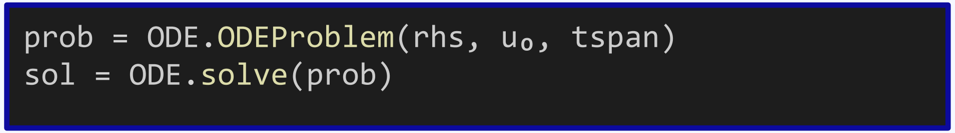
```julia
function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(g / L) * sin(θ)
end
```

DifferentialEquations.jl



```julia
function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(g / L) * sin(θ)
end
```

This package enables to solve the system of ordinary differential equations in just **two lines** of code:

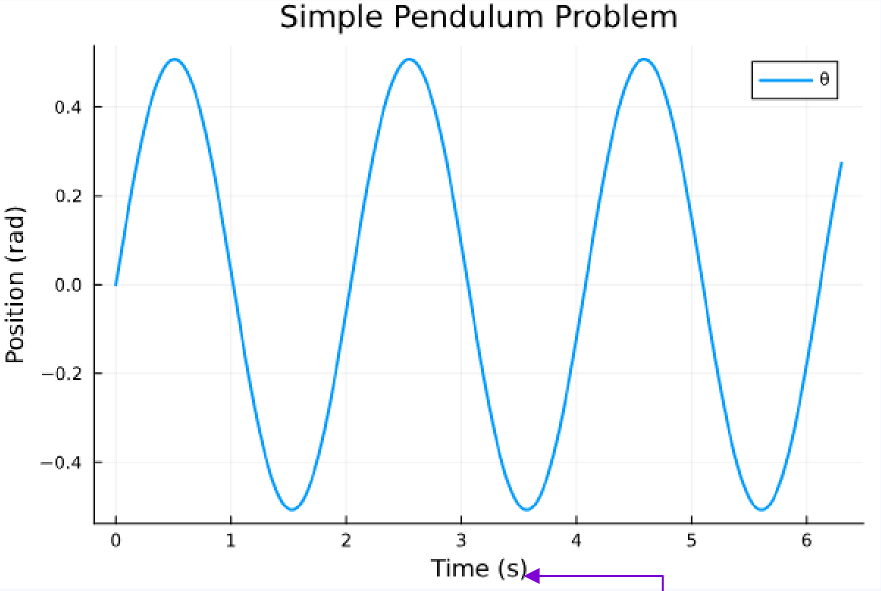


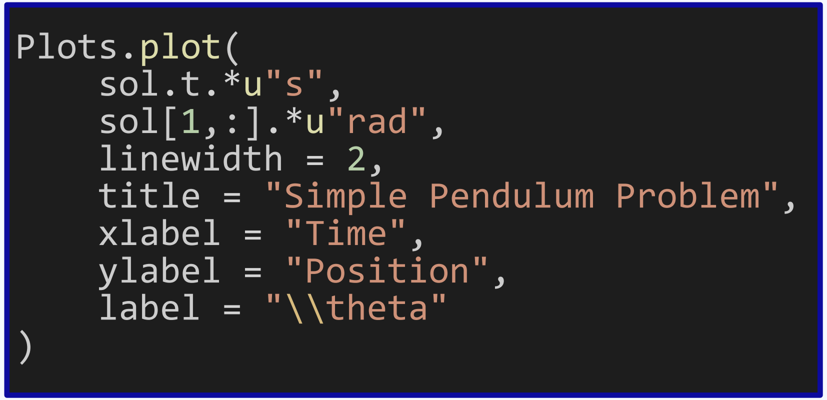
```julia
prob = ODE.ODEProblem(rhs, u₀, tspan)
sol = ODE.solve(prob)
```

← Create differentail equation problem

← Solve it!

DifferentialEquations.jl

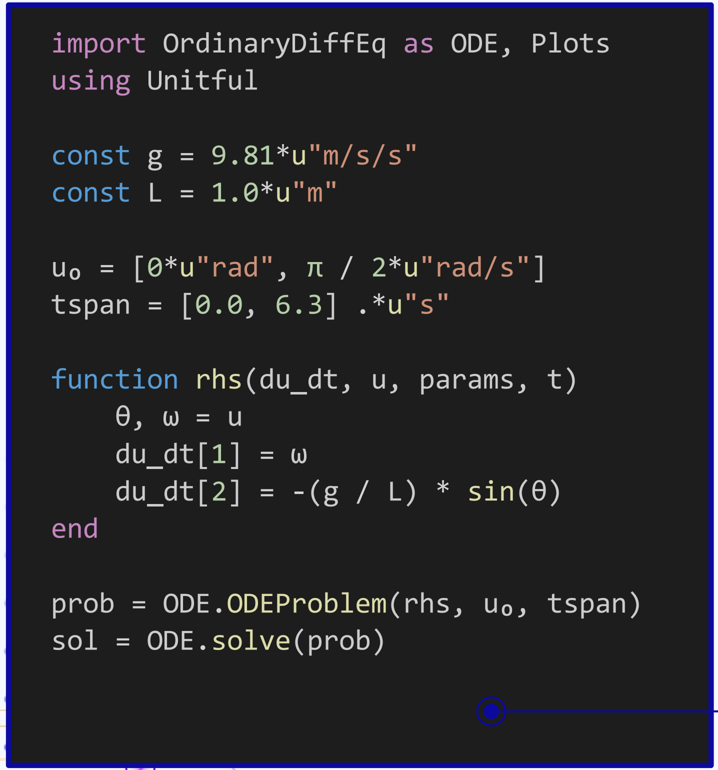




```
Plots.plot(
    sol.t.*u"s",
    sol[1,:].*u"rad",
    linewidth = 2,
    title = "Simple Pendulum Problem",
    xlabel = "Time",
    ylabel = "Position",
    label = "\\theta"
)
```

Automatic labeling!

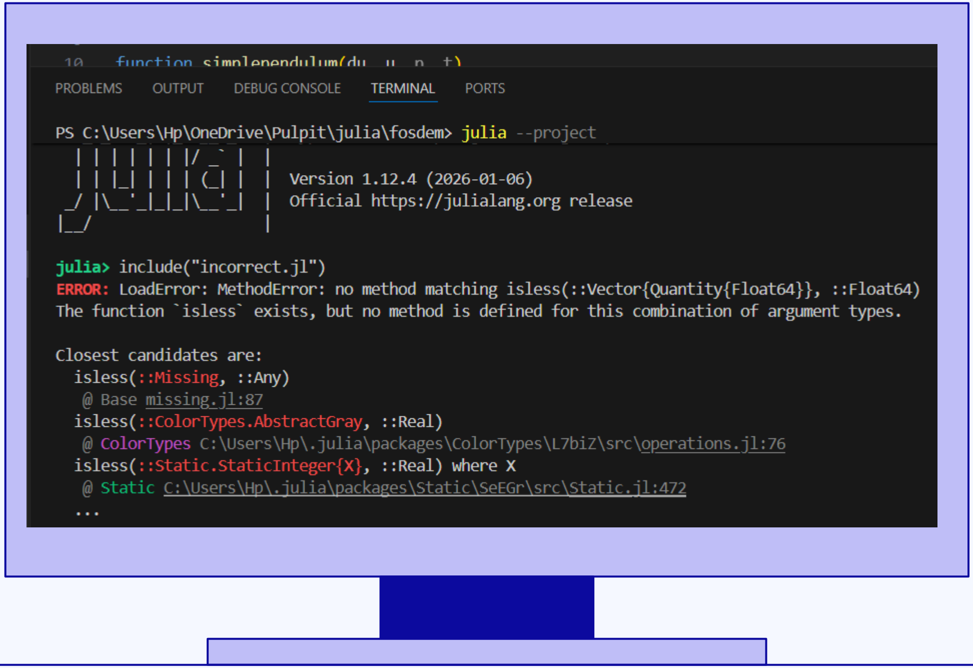
Let’s try together: Unitful.jl + DiffEq.jl



```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const g = 9.81*u"m/s/s"
const L = 1.0*u"m"

u₀ = [0*u"rad", π / 2*u"rad/s"]
tspan = [0.0, 6.3] .*u"s"

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(g / L) * sin(θ)
end

prob = ODE.ODEProblem(rhs, u₀, tspan)
sol = ODE.solve(prob)
```

Let's try together: Unitful.jl + DiffEq.jl

```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const g = 9.81*u"m/s/s"
const L = 1.0*u"m"

u₀ = [0*u"rad", π / 2*u"rad/s"]
tspan = [0.0, 6.3] .*u"s"

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(g / L) * sin(θ)
end

prob = ODE.ODEProblem(rhs, u₀, tspan)
sol = ODE.solve(prob)
```

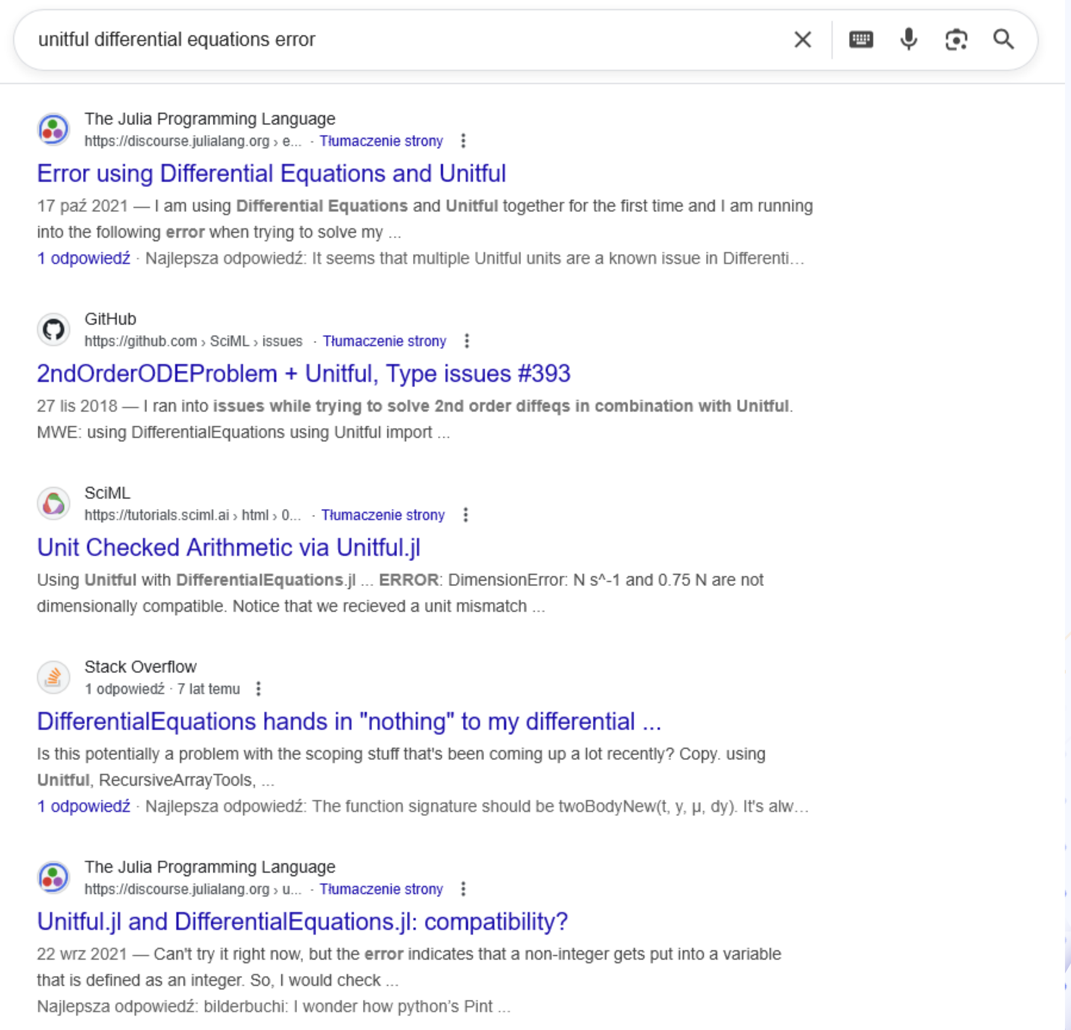


```
function simplependulum(du, u, p, t)
```

```
PS C:\Users\Hp\OneDrive\Pulpit\julia\fosdem> julia --project
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.12.4 (2026-01-06)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org release
|__/                   |

julia> include("incorrect.jl")
ERROR: LoadError: MethodError: no method matching isless(::Vector{Quantity{Float64}}, ::Float64)
The function `isless` exists, but no method is defined for this combination of argument types.

Closest candidates are:
  isless(::Missing, ::Any)
   @ Base missing.jl:87
  isless(::ColorTypes.AbstractGray, ::Real)
   @ ColorTypes C:\Users\Hp\.julia\packages\ColorTypes\L7biZ\src\operations.jl:76
  isless(::Static.StaticInteger{X}, ::Real) where X
   @ Static C:\Users\Hp\.julia\packages\Static\SeEGr\src\Static.jl:472
  ...
```

What does the community say?

- errors inevitable
- „solution”: drop units
- only basic solvers compatible



unitful differential equations error

The Julia Programming Language
https://discourse.julialang.org › e... · Tłumaczenie strony

Error using Differential Equations and Unitful

17 paź 2021 — I am using **Differential Equations** and **Unitful** together for the first time and I am running into the following **error** when trying to solve my ...

1 odpowiedź · Najlepsza odpowiedź: It seems that multiple Unitful units are a known issue in Differenti...

GitHub
https://github.com › SciML › issues · Tłumaczenie strony

2ndOrderODEProblem + Unitful, Type issues #393

27 lis 2018 — I ran into **issues while trying to solve 2nd order diffeqs in combination with Unitful**. MWE: using DifferentialEquations using Unitful import ...

SciML
https://tutorials.sciml.ai › html › 0... · Tłumaczenie strony

Unit Checked Arithmetic via Unitful.jl

Using **Unitful** with **DifferentialEquations**.jl ... **ERROR**: DimensionError: N s^-1 and 0.75 N are not dimensionally compatible. Notice that we recieved a unit mismatch ...

Stack Overflow
1 odpowiedź · 7 lat temu

DifferentialEquations hands in "nothing" to my differential ...

Is this potentially a problem with the scoping stuff that's been coming up a lot recently? Copy. using **Unitful**, RecursiveArrayTools, ...

1 odpowiedź · Najlepsza odpowiedź: The function signature should be twoBodyNew(t, y, μ, dy). It's alw...

The Julia Programming Language
https://discourse.julialang.org › u... · Tłumaczenie strony

Unitful.jl and DifferentialEquations.jl: compatibility?

22 wrz 2021 — Can't try it right now, but the **error** indicates that a non-integer gets put into a variable that is defined as an integer. So, I would check ...

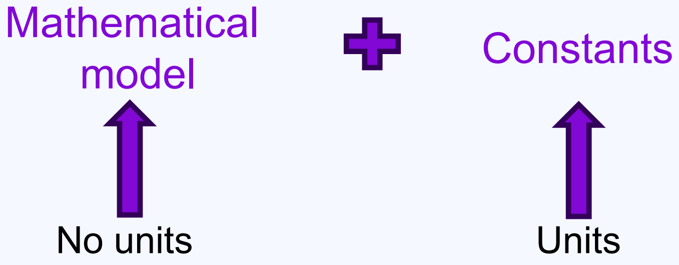
Najlepsza odpowiedź: bilderbuchi: I wonder how python's Pint ...

"… the astute observer (Fourier was the first astute observer) notices that the equations … **remain true when** the size of the fundamental **units is changed**"

— Percy W. Bridgman (Nobel'46), Dimensional Analysis

Separation of Concerns

Physics:



Mathematical model + Constants

No units → Mathematical model

Units → Constants

Software engineering:

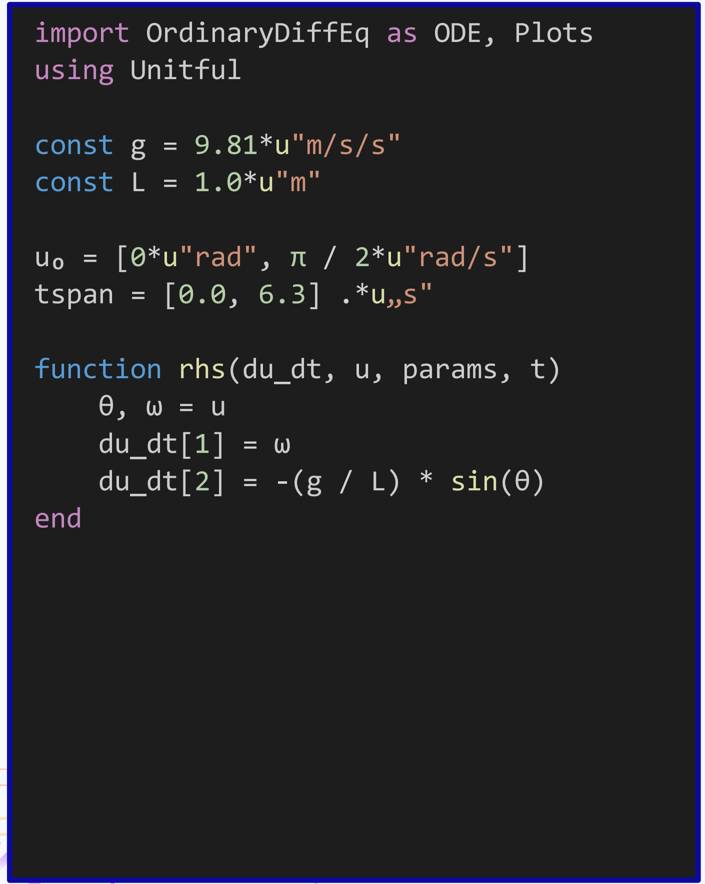
- Make **injectible dependency** holding all constants
- Use **dependency with units** for dimensionality calculations
- Inject **dependecy stripped** of units for numerical calculations

What to change so it does work?

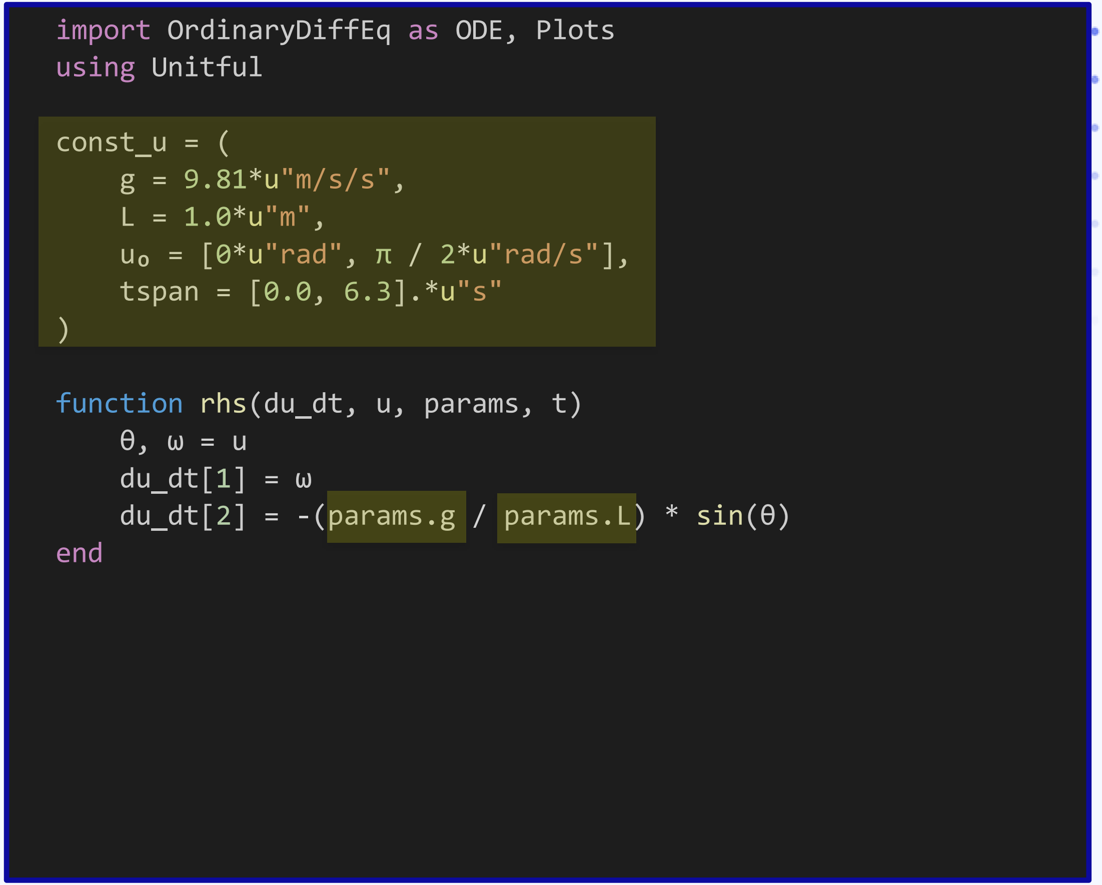
```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const g = 9.81*u"m/s/s"
const L = 1.0*u"m"

u₀ = [0*u"rad", π / 2*u"rad/s"]
tspan = [0.0, 6.3] .*u„s"
```

```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const_u = (
    g = 9.81*u"m/s/s",
    L = 1.0*u"m",
    u₀ = [0*u"rad", π / 2*u"rad/s"],
    tspan = [0.0, 6.3].*u"s"
)
```

What to change so it does work?



```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const g = 9.81*u"m/s/s"
const L = 1.0*u"m"

u₀ = [0*u"rad", π / 2*u"rad/s"]
tspan = [0.0, 6.3] .*u„s"

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(g / L) * sin(θ)
end
```



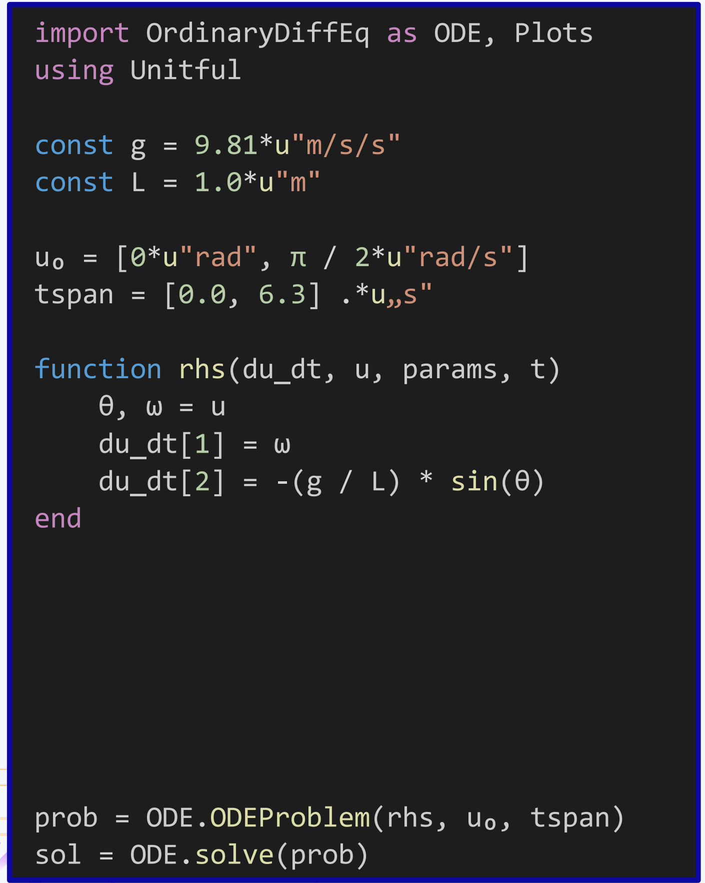
```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const_u = (
    g = 9.81*u"m/s/s",
    L = 1.0*u"m",
    u₀ = [0*u"rad", π / 2*u"rad/s"],
    tspan = [0.0, 6.3].*u"s"
)

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(params.g / params.L) * sin(θ)
end
```

What to change so it does work?

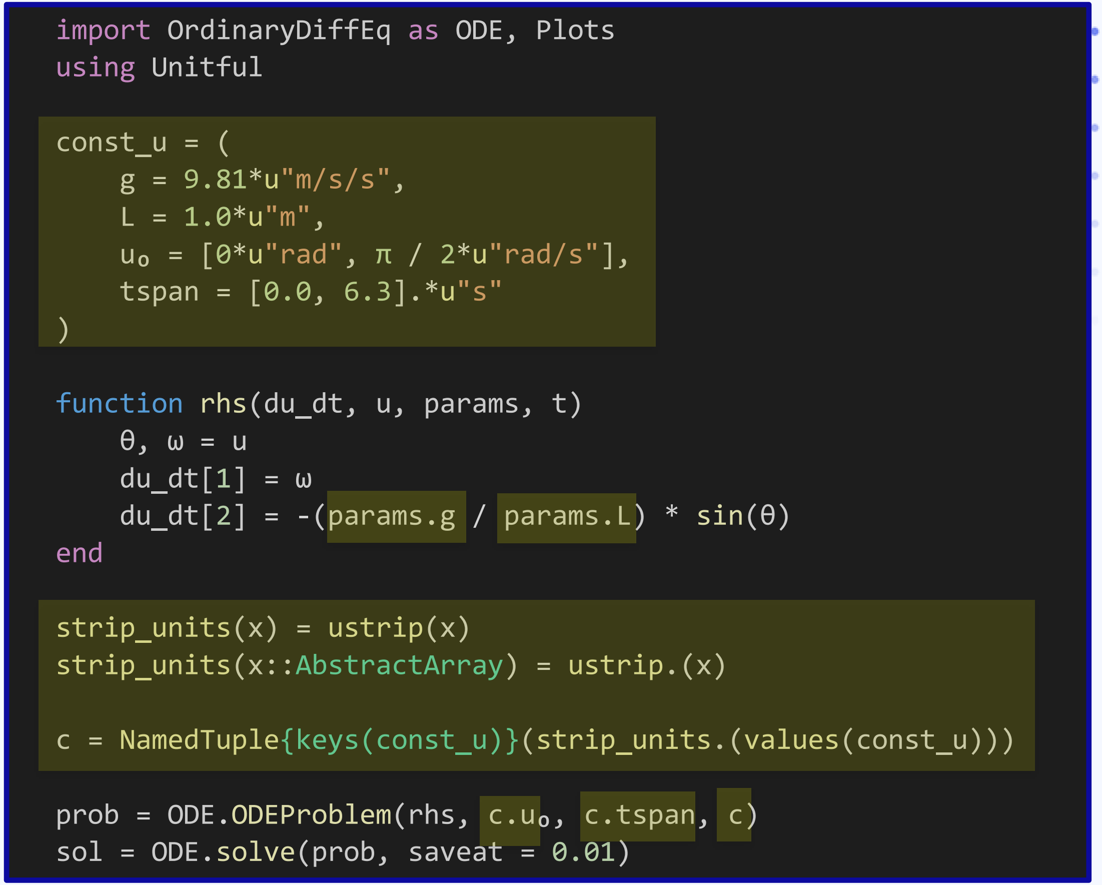
```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const g = 9.81*u"m/s/s"
const L = 1.0*u"m"

u₀ = [0*u"rad", π / 2*u"rad/s"]
tspan = [0.0, 6.3] .*u„s"

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(g / L) * sin(θ)
end
```

```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const_u = (
    g = 9.81*u"m/s/s",
    L = 1.0*u"m",
    u₀ = [0*u"rad", π / 2*u"rad/s"],
    tspan = [0.0, 6.3].*u"s"
)

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(params.g / params.L) * sin(θ)
end

strip_units(x) = ustrip(x)
strip_units(x::AbstractArray) = ustrip.(x)

c = NamedTuple{keys(const_u)}(strip_units.(values(const_u)))
```

What to change so it does work?



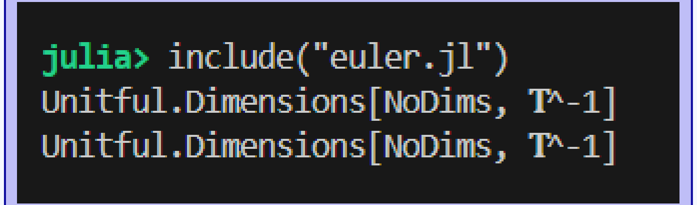
```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const g = 9.81*u"m/s/s"
const L = 1.0*u"m"

u₀ = [0*u"rad", π / 2*u"rad/s"]
tspan = [0.0, 6.3] .*u„s"

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(g / L) * sin(θ)
end




prob = ODE.ODEProblem(rhs, u₀, tspan)
sol = ODE.solve(prob)
```



```julia
import OrdinaryDiffEq as ODE, Plots
using Unitful

const_u = (
    g = 9.81*u"m/s/s",
    L = 1.0*u"m",
    u₀ = [0*u"rad", π / 2*u"rad/s"],
    tspan = [0.0, 6.3].*u"s"
)

function rhs(du_dt, u, params, t)
    θ, ω = u
    du_dt[1] = ω
    du_dt[2] = -(params.g / params.L) * sin(θ)
end

strip_units(x) = ustrip(x)
strip_units(x::AbstractArray) = ustrip.(x)

c = NamedTuple{keys(const_u)}(strip_units.(values(const_u)))

prob = ODE.ODEProblem(rhs, c.u₀, c.tspan, c)
sol = ODE.solve(prob, saveat = 0.01)
```

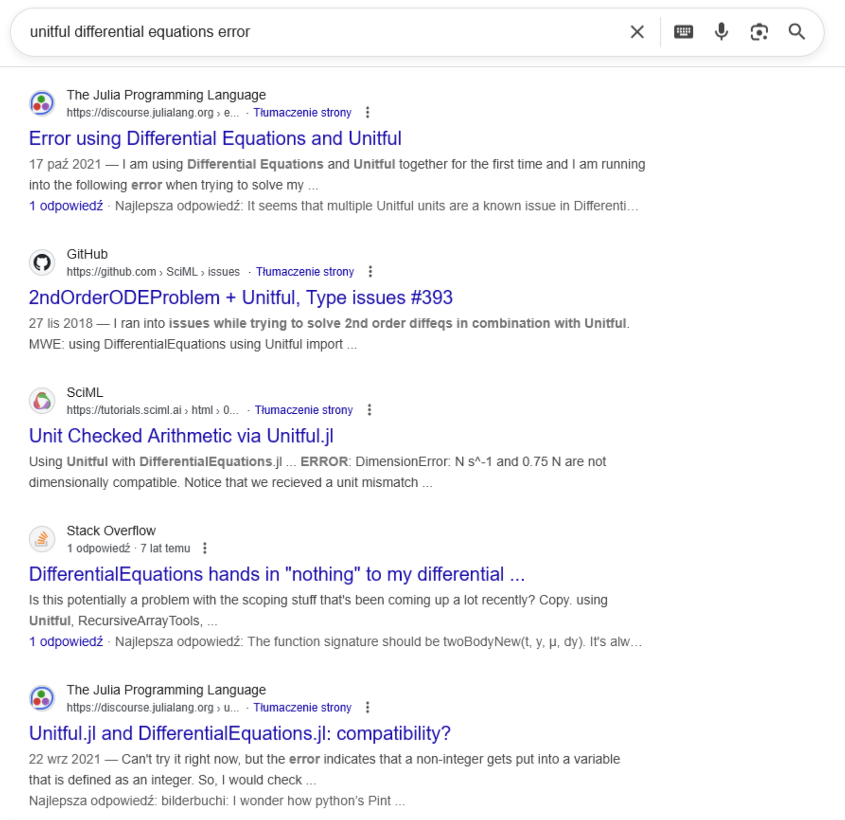
Note: the function works with units too

```julia
du_dt = similar(const_u.u₀)

rhs(du_dt, const_u.u₀, const_u, 0.1u"s")

u = const_u.u₀ + du_dt * 0.1u"s"



@test dimension.(u) == dimension.(const_u.u₀)

println(dimension.(u))

println(dimension.(const_u.u₀))
```



```
julia> include("euler.jl")
Unitful.Dimensions[NoDims, 𝐓^-1]
Unitful.Dimensions[NoDims, 𝐓^-1]
```

Take-home message:
there is no Unitful.jl + DiffEq.jl problem!



unitful differential equations error

The Julia Programming Language
https://discourse.julialang.org › e... · Tłumaczenie strony

Error using Differential Equations and Unitful

17 paź 2021 — I am using **Differential Equations** and **Unitful** together for the first time and I am running into the following **error** when trying to solve my ...

1 odpowiedź · Najlepsza odpowiedź: It seems that multiple Unitful units are a known issue in Differenti...

GitHub
https://github.com › SciML › issues · Tłumaczenie strony

2ndOrderODEProblem + Unitful, Type issues #393

27 lis 2018 — I ran into **issues while trying to solve 2nd order diffeqs in combination with Unitful**. MWE: using DifferentialEquations using Unitful import ...

SciML
https://tutorials.sciml.ai › html › 0... · Tłumaczenie strony

Unit Checked Arithmetic via Unitful.jl

Using **Unitful** with **DifferentialEquations**.jl ... **ERROR**: DimensionError: N s^-1 and 0.75 N are not dimensionally compatible. Notice that we recieved a unit mismatch ...

Stack Overflow
1 odpowiedź · 7 lat temu

DifferentialEquations hands in "nothing" to my differential ...

Is this potentially a problem with the scoping stuff that's been coming up a lot recently? Copy. using **Unitful**, RecursiveArrayTools, ...

1 odpowiedź · Najlepsza odpowiedź: The function signature should be twoBodyNew(t, y, μ, dy). It's alw...

The Julia Programming Language
https://discourse.julialang.org › u... · Tłumaczenie strony

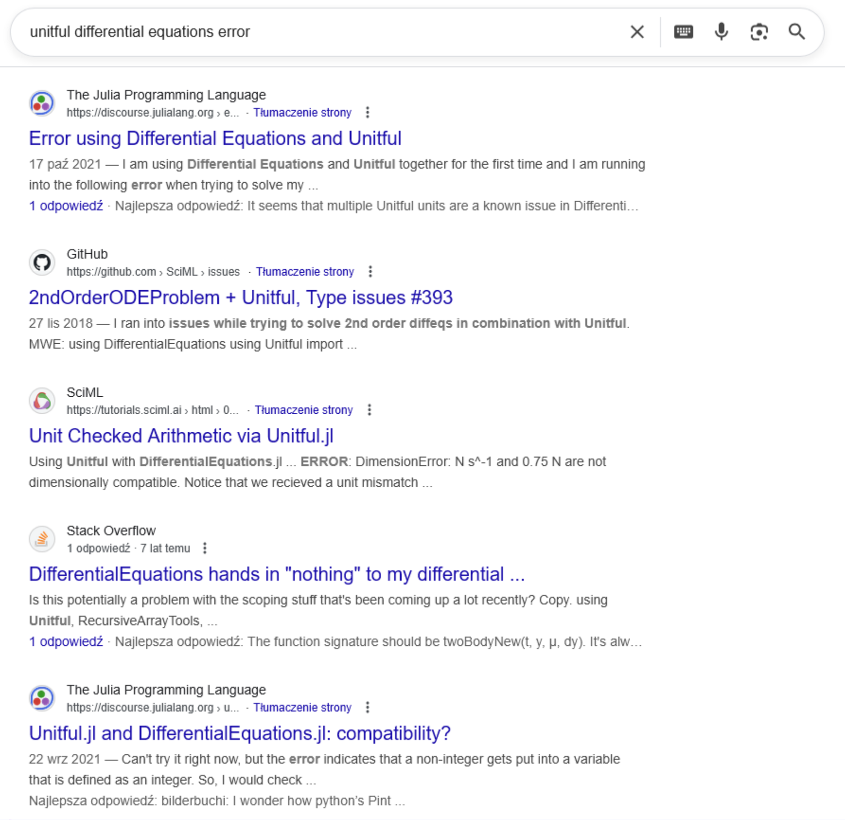
Unitful.jl and DifferentialEquations.jl: compatibility?

22 wrz 2021 — Can't try it right now, but the **error** indicates that a non-integer gets put into a variable that is defined as an integer. So, I would check ...

Najlepsza odpowiedź: bilderbuchi: I wonder how python's Pint ...

- ODE definition should be **invariant to units**

Take-home message: there is no Unitful.jl + DiffEq.jl problem!



unitful differential equations error

The Julia Programming Language
https://discourse.julialang.org › e... · Tłumaczenie strony

Error using Differential Equations and Unitful

17 paź 2021 — I am using **Differential Equations** and **Unitful** together for the first time and I am running into the following **error** when trying to solve my ...

1 odpowiedź · Najlepsza odpowiedź: It seems that multiple Unitful units are a known issue in Differenti...

GitHub
https://github.com › SciML › issues · Tłumaczenie strony

2ndOrderODEProblem + Unitful, Type issues #393

27 lis 2018 — I ran into **issues while trying to solve 2nd order diffeqs in combination with Unitful**. MWE: using DifferentialEquations using Unitful import ...

SciML
https://tutorials.sciml.ai › html › 0... · Tłumaczenie strony

Unit Checked Arithmetic via Unitful.jl

Using **Unitful** with **DifferentialEquations**.jl ... **ERROR**: DimensionError: N s^-1 and 0.75 N are not dimensionally compatible. Notice that we recieved a unit mismatch ...

Stack Overflow
1 odpowiedź · 7 lat temu

DifferentialEquations hands in "nothing" to my differential ...

Is this potentially a problem with the scoping stuff that's been coming up a lot recently? Copy. using **Unitful**, RecursiveArrayTools, ...

1 odpowiedź · Najlepsza odpowiedź: The function signature should be twoBodyNew(t, y, μ, dy). It's alw...

The Julia Programming Language
https://discourse.julialang.org › u... · Tłumaczenie strony

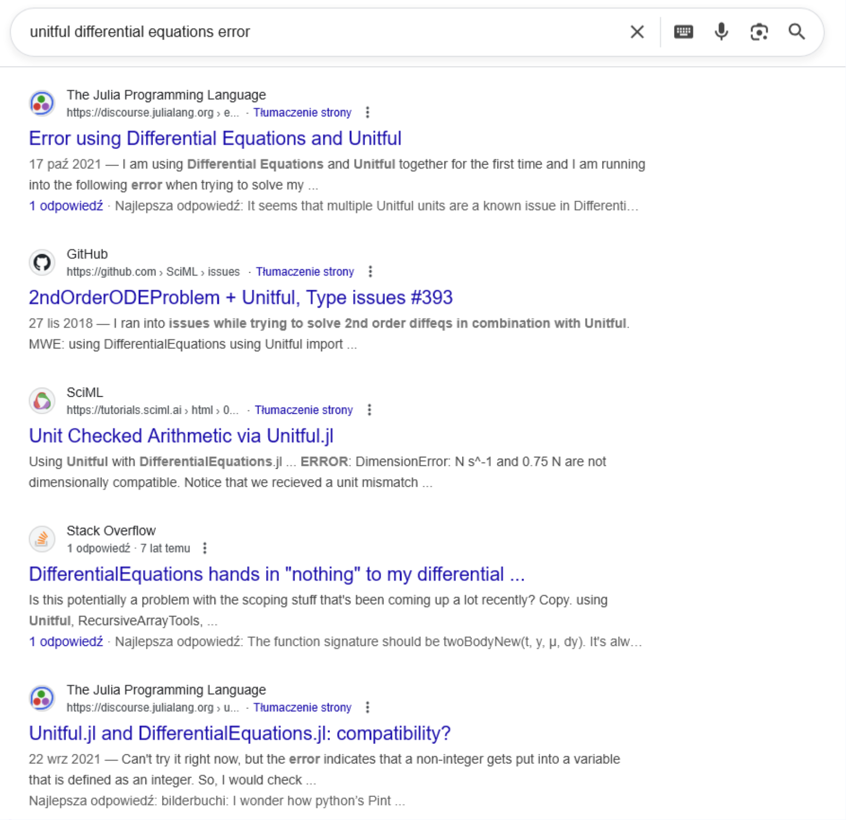
Unitful.jl and DifferentialEquations.jl: compatibility?

22 wrz 2021 — Can't try it right now, but the **error** indicates that a non-integer gets put into a variable that is defined as an integer. So, I would check ...

Najlepsza odpowiedź: bilderbuchi: I wonder how python's Pint ...

- ODE definition should be **invariant to units**
- **units** injected through constants only

Take-home message: there is no Unitful.jl + DiffEq.jl problem!



unitful differential equations error

The Julia Programming Language
https://discourse.julialang.org › e... · Tłumaczenie strony

Error using Differential Equations and Unitful

17 paź 2021 — I am using **Differential Equations** and **Unitful** together for the first time and I am running into the following **error** when trying to solve my ...

1 odpowiedź · Najlepsza odpowiedź: It seems that multiple Unitful units are a known issue in Differenti...

GitHub
https://github.com › SciML › issues · Tłumaczenie strony

2ndOrderODEProblem + Unitful, Type issues #393

27 lis 2018 — I ran into **issues while trying to solve 2nd order diffeqs in combination with Unitful**. MWE: using DifferentialEquations using Unitful import ...

SciML
https://tutorials.sciml.ai › html › 0... · Tłumaczenie strony

Unit Checked Arithmetic via Unitful.jl

Using **Unitful** with **DifferentialEquations**.jl ... **ERROR**: DimensionError: N s^-1 and 0.75 N are not dimensionally compatible. Notice that we recieved a unit mismatch ...

Stack Overflow
1 odpowiedź · 7 lat temu

DifferentialEquations hands in "nothing" to my differential ...

Is this potentially a problem with the scoping stuff that's been coming up a lot recently? Copy. using **Unitful**, RecursiveArrayTools, ...

1 odpowiedź · Najlepsza odpowiedź: The function signature should be twoBodyNew(t, y, μ, dy). It's alw...

The Julia Programming Language
https://discourse.julialang.org › u... · Tłumaczenie strony

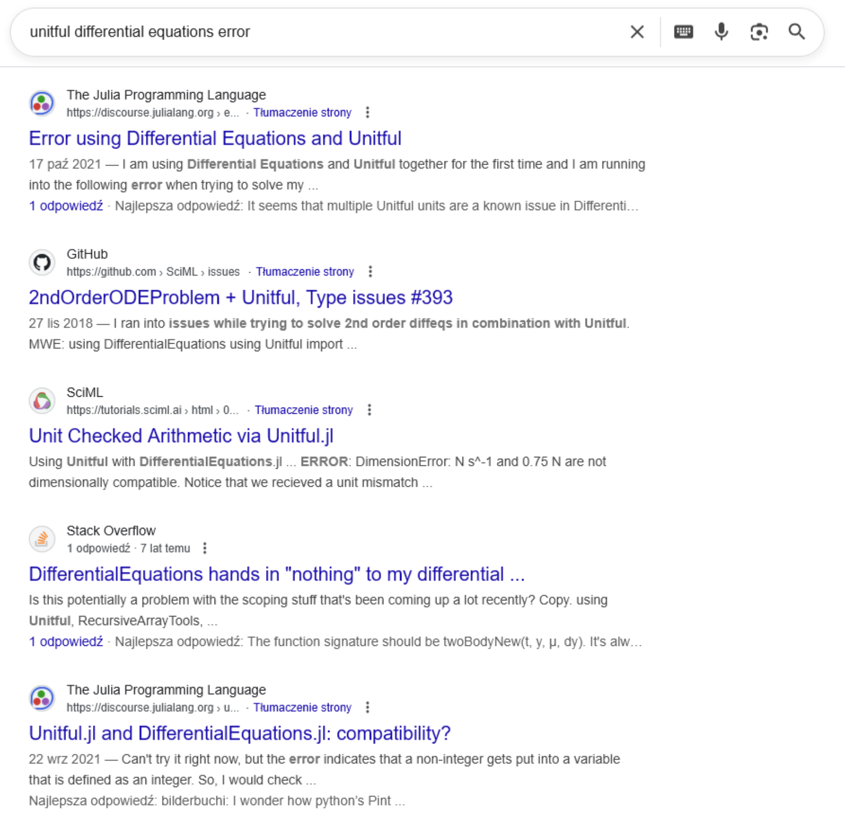
Unitful.jl and DifferentialEquations.jl: compatibility?

22 wrz 2021 — Can't try it right now, but the **error** indicates that a non-integer gets put into a variable that is defined as an integer. So, I would check ...

Najlepsza odpowiedź: bilderbuchi: I wonder how python's Pint ...

- ODE definition should be **invariant to units**
- **units** injected through constants only
- Applies to any templated/JIT-compiled tech (e.g., **Python + Numba + Pint,** where it also solves performance overhead issue!)

Take-home message: there is no Unitful.jl + DiffEq.jl problem!



unitful differential equations error

The Julia Programming Language
https://discourse.julialang.org › e... · Tłumaczenie strony

Error using Differential Equations and Unitful

17 paź 2021 — I am using **Differential Equations** and **Unitful** together for the first time and I am running into the following **error** when trying to solve my ...

1 odpowiedź · Najlepsza odpowiedź: It seems that multiple Unitful units are a known issue in Differenti...

GitHub
https://github.com › SciML › issues · Tłumaczenie strony

2ndOrderODEProblem + Unitful, Type issues #393

27 lis 2018 — I ran into **issues while trying to solve 2nd order diffeqs in combination with Unitful**. MWE: using DifferentialEquations using Unitful import ...

SciML
https://tutorials.sciml.ai › html › 0... · Tłumaczenie strony

Unit Checked Arithmetic via Unitful.jl

Using **Unitful** with **DifferentialEquations**.jl ... **ERROR**: DimensionError: N s^-1 and 0.75 N are not dimensionally compatible. Notice that we recieved a unit mismatch ...

Stack Overflow
1 odpowiedź · 7 lat temu

DifferentialEquations hands in "nothing" to my differential ...

Is this potentially a problem with the scoping stuff that's been coming up a lot recently? Copy. using **Unitful**, RecursiveArrayTools, ...

1 odpowiedź · Najlepsza odpowiedź: The function signature should be twoBodyNew(t, y, μ, dy). It's alw...

The Julia Programming Language
https://discourse.julialang.org › u... · Tłumaczenie strony

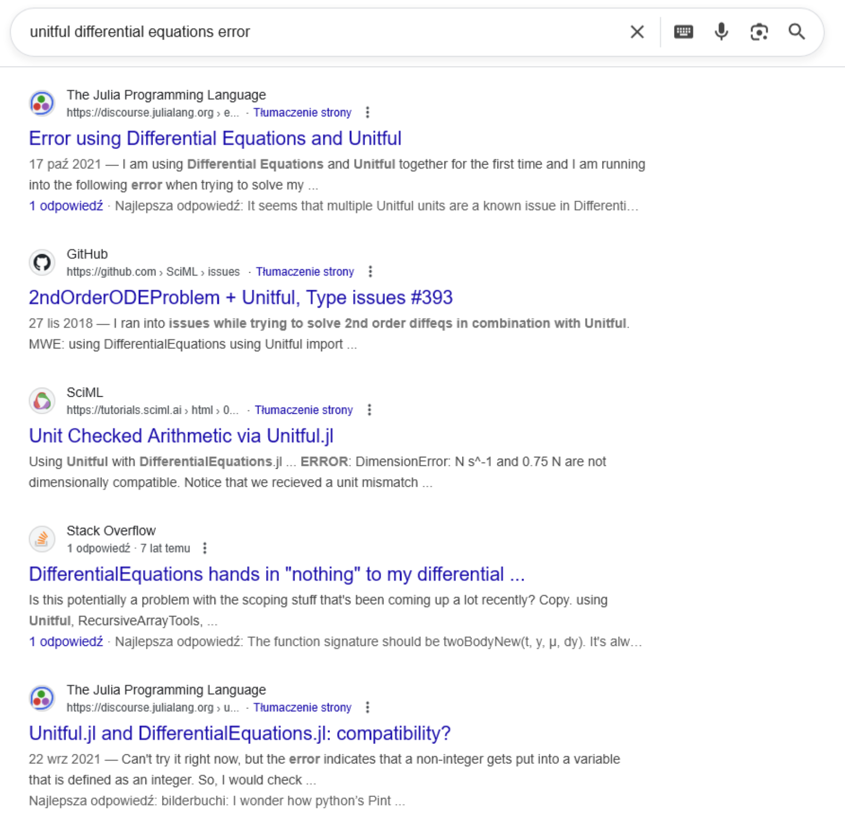
Unitful.jl and DifferentialEquations.jl: compatibility?

22 wrz 2021 — Can't try it right now, but the **error** indicates that a non-integer gets put into a variable that is defined as an integer. So, I would check ...

Najlepsza odpowiedź: bilderbuchi: I wonder how python's Pint ...

- ODE definition should be **invariant to units**
- **units** injected through constants only
- Applies to any templated/JIT-compiled tech (e.g., **Python + Numba + Pint,** where it also solves performance overhead issue!)
- unit-aware testing & unit-agnostic numerics (using a single codebase)

Take-home message: there is no Unitful.jl + DiffEq.jl problem!



unitful differential equations error

The Julia Programming Language
https://discourse.julialang.org › e... · Tłumaczenie strony

Error using Differential Equations and Unitful

17 paź 2021 — I am using **Differential Equations** and **Unitful** together for the first time and I am running into the following **error** when trying to solve my ...

1 odpowiedź · Najlepsza odpowiedź: It seems that multiple Unitful units are a known issue in Differenti...

GitHub
https://github.com › SciML › issues · Tłumaczenie strony

2ndOrderODEProblem + Unitful, Type issues #393

27 lis 2018 — I ran into **issues while trying to solve 2nd order diffeqs in combination with Unitful**. MWE: using DifferentialEquations using Unitful import ...

SciML
https://tutorials.sciml.ai › html › 0... · Tłumaczenie strony

Unit Checked Arithmetic via Unitful.jl

Using **Unitful** with **DifferentialEquations**.jl ... **ERROR**: DimensionError: N s^-1 and 0.75 N are not dimensionally compatible. Notice that we recieved a unit mismatch ...

Stack Overflow
1 odpowiedź · 7 lat temu

DifferentialEquations hands in "nothing" to my differential ...

Is this potentially a problem with the scoping stuff that's been coming up a lot recently? Copy. using **Unitful**, RecursiveArrayTools, ...

1 odpowiedź · Najlepsza odpowiedź: The function signature should be twoBodyNew(t, y, μ, dy). It's alw...

The Julia Programming Language
https://discourse.julialang.org › u... · Tłumaczenie strony

Unitful.jl and DifferentialEquations.jl: compatibility?

22 wrz 2021 — Can't try it right now, but the **error** indicates that a non-integer gets put into a variable that is defined as an integer. So, I would check ...

Najlepsza odpowiedź: bilderbuchi: I wonder how python's Pint ...

- ODE definition should be **invariant to units**
- **units** injected through constants only
- Applies to any templated/JIT-compiled tech (e.g., **Python + Numba + Pint,** where it also solves performance overhead issue!)
- unit-aware testing & unit-agnostic numerics (using a single codebase)
- **It's not a workaround, it's coding physics thinking like physicist!**

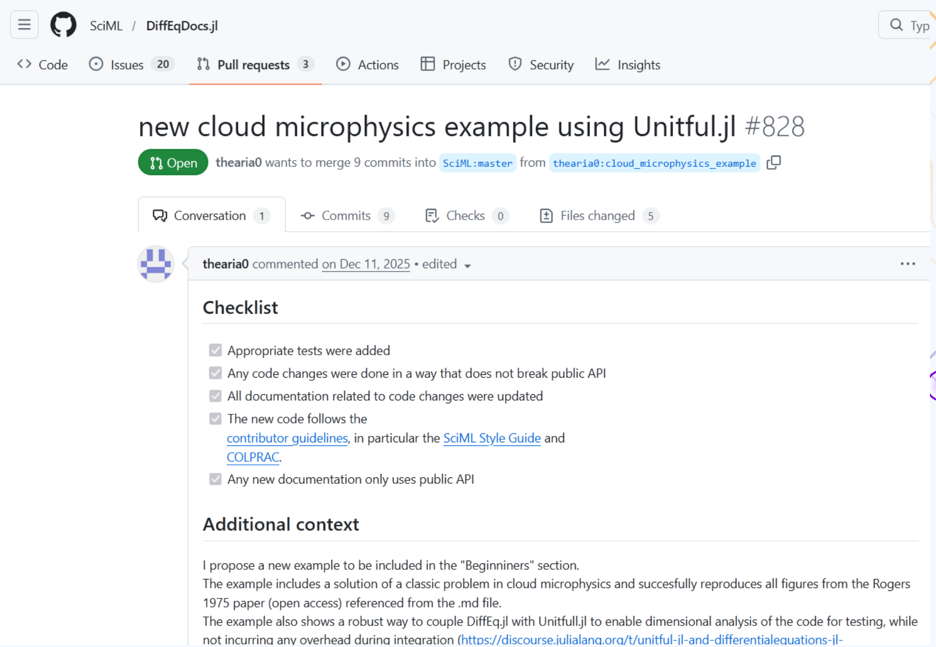
Thanks!

klimaszewska@student.agh.edu.pl

…/daria-klimaszewska

@thearia0

DiffEqDocs.jl PR



new cloud microphysics example using Unitful.jl #828

thearia0 wants to merge 9 commits into SciML:master from thearia0:cloud_microphysics_example

thearia0 commented on Dec 11, 2025 • edited

Checklist

- ☑ Appropriate tests were added
- ☑ Any code changes were done in a way that does not break public API
- ☑ All documentation related to code changes were updated
- ☑ The new code follows the contributor guidelines, in particular the SciML Style Guide and COLPRAC.
- ☑ Any new documentation only uses public API

Additional context

I propose a new example to be included in the "Beginniners" section.
The example includes a solution of a classic problem in cloud microphysics and succesfully reproduces all figures from the Rogers 1975 paper (open access) referenced from the .md file.
The example also shows a robust way to couple DiffEq.jl with Unitfull.jl to enable dimensional analysis of the code for testing, while not incurring any overhead during integration (https://discourse.julialang.org/t/unitful-jl-and-differentialequations-jl-

/ SciML / DiffEqDocs.jl / pull / 828

CREDITS: This presentation template was created by Slidesgo, and includes icons, infographics & images by Freepik