Unit Checked Arithmetic via Unitful.jl

Chris Rackauckas

March 1, 2019

Units and dimensional analysis are standard tools across the sciences for checking the correctness of your equation. However, most ODE solvers only allow for the equation to be in dimensionless form, leaving it up to the user to both convert the equation to a dimensionless form, punch in the equations, and hopefully not make an error along the way.

DifferentialEquations.jl allows for one to use Unitful.jl to have unit-checked arithmetic natively in the solvers. Given the dispatch implementation of the Unitful, this has little overhead.

0.1 Using Unitful

To use Unitful, you need to have the package installed. Then you can add units to your variables. For example:

```
using Unitful
t = 1.0u"s"
```

```
1.0 s
```

Notice that `t` is a variable with units in seconds. If we make another value with seconds, they can add

```
t2 = 1.02u"s"
t+t2
```

```
2.02 s
```

and they can multiply:

```
t*t2
```

```
1.02 s^2
```

You can even do rational roots:

```
sqrt(t)
```

```
1.0 s^1/2
```

Many operations work. These operations will check to make sure units are correct, and will throw an error for incorrect operations:

```
t + sqrt(t)
```

```
Error: DimensionError: 1.0 s and 1.0 s^1/2 are not dimensionally compatible.
```

0.2 Using Unitful with DifferentialEquations.jl

Just like with other number systems, you can choose the units for your numbers by simply specifying the units of the initial condition and the timestep. For example, to solve the linear ODE where the variable has units of Newton's and `t` is in Seconds, we would use:

```
using DifferentialEquations
f = (y,p,t) -> 0.5*y
u0 = 1.5u"N"
prob = ODEProblem(f,u0,(0.0u"s",1.0u"s"))
sol = solve(prob,Tsit5())
```

```
Error: DimensionError: N s^-1 and 0.75 N are not dimensionally compatible.
```

Notice that we recieved a unit mismatch error. This is correctly so! Remember that for an ODE:

$$\frac{dy}{dt} = f(t,y)$$

we must have that `f` is a rate, i.e. `f` is a change in `y` per unit time. So we need to fix the units of `f` in our example to be `N/s`. Notice that we then do not receive an error if we do the following:

```
f = (y,p,t) -> 0.5*y/3.0u"s"
prob = ODEProblem(f,u0,(0.0u"s",1.0u"s"))
sol = solve(prob,Tsit5())
```

```
retcode: Success
Interpolation: specialized 4th order "free" interpolation
t: 3-element Array{Unitful.Quantity{Float64,𝐓,Unitful.FreeUnits{(s,),𝐓,nothing}},1}:
```

```
                0.0 s
 0.14311598261241779 s
                1.0 s
u: 3-element Array{Unitful.Quantity{Float64,L*M*T^-2,Unitful.FreeUnits{(N,),L*M*T^-2,
    nothing}},1}:
               1.5 N
 1.5362091208988309 N
 1.7720406194871123 N
```

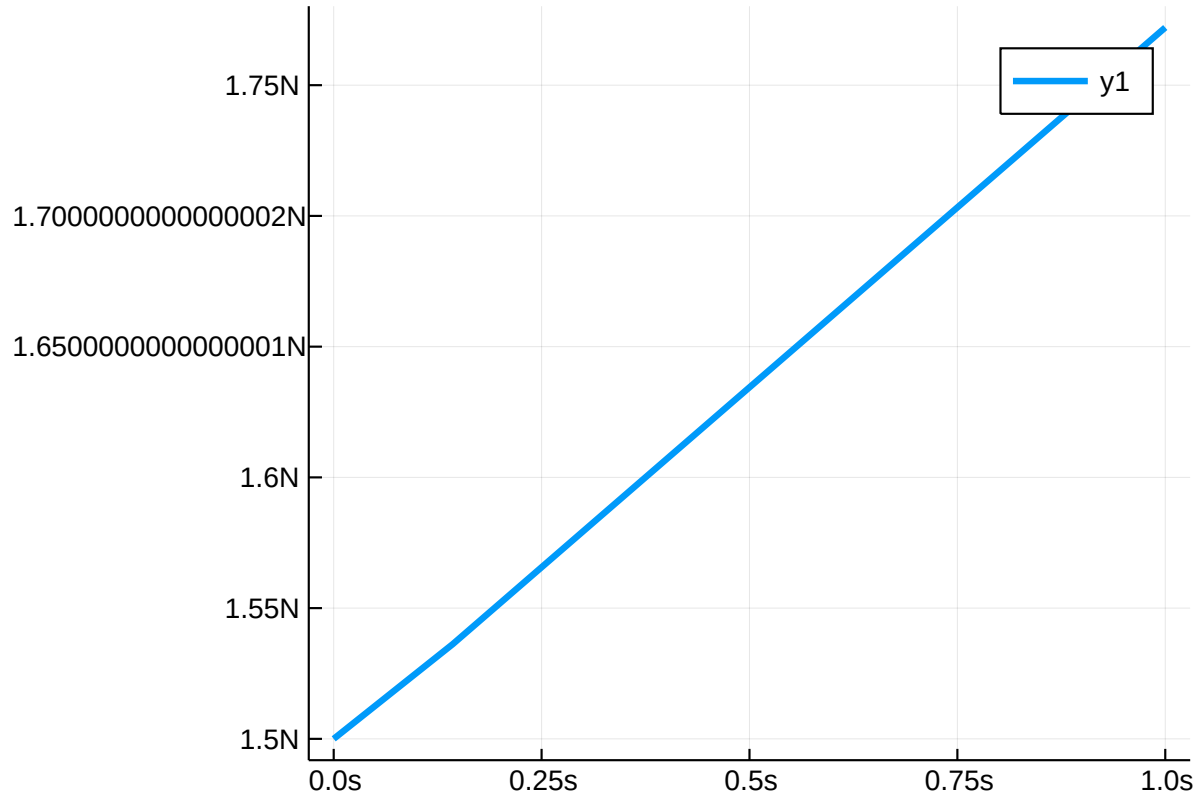
This gives a a normal solution object. Notice that the values are all with the correct units:

```
print(sol[:])
```

```
Unitful.Quantity{Float64,L*M*T^-2,Unitful.FreeUnits{(N,),L*M*T^-2,nothing}}[1.5 N,
    1.53621 N, 1.77204 N]
```

We can plot the solution using the plot recipe.

```
#Pkg.clone("https://github.com/ajkeller34/UnitfulPlots.jl")
using UnitfulPlots, Plots
gr()
plot(sol.t,sol[:],lw=3)
```



Notice that here we pulled the units for the label directly from the solution. Thus if the units change, the labels will change automatically.

```
using DiffEqTutorials
```

```
DiffEqTutorials.tutorial_footer(WEAVE_ARGS[:folder],WEAVE_ARGS[:file])
```

0.3 Appendix

These benchmarks are part of the DiffEqTutorials.jl repository, found at: https://github.com/JuliaDiffEq/

To locally run this tutorial, do the following commands:

```
using DiffEqTutorials
DiffEqTutorials.weave_file("type_handling","unitful.jmd")
```

Computer Information:

```
Julia Version 1.1.0
Commit 80516ca202 (2019-01-21 21:24 UTC)
Platform Info:
  OS: Windows (x86_64-w64-mingw32)
  CPU: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-6.0.1 (ORCJIT, skylake)
Environment:
  JULIA_EDITOR = "C:\Users\accou\AppData\Local\atom\app-1.34.0\atom.exe" -a
  JULIA_NUM_THREADS = 6
```

Package Information:

```
Status `C:\Users\accou\.julia\external\DiffEqTutorials.jl\src\Project.toml`
```