Introduction

Figure 1: JuMP logo

Welcome to the documentation for JuMP.

!!! note This documentation is also available in md format: index.md.

What is JuMP?

JuMP is a domain-specific modeling language for mathematical optimization embedded in Julia.

JuMP makes it easy to formulate and solve a range of problem classes, including linear programs, integer programs, conic programs, semidefinite programs, and constrained nonlinear programs. Here's an example:

```
julia> using JuMP, Ipopt

julia> function solve_constrained_least_squares_regression(A::Matrix, b::Vector)
           m, n = size(A)
           model = Model(Ipopt.Optimizer)
           set_silent(model)
           @variable(model, x[1:n])
           @variable(model, residuals[1:m])
           @constraint(model, residuals == A * x - b)
           @constraint(model, sum(x) == 1)
           @objective(model, Min, sum(residuals.^2))
           optimize!(model)
           return value.(x)
       end
solve_constrained_least_squares_regression (generic function with 1 method)

julia> A, b = rand(10, 3), rand(10);

julia> x = solve_constrained_least_squares_regression(A, b)
3-element Vector{Float64}:
 0.4137624719002825
```

```
0.09707679853084578
0.48916072956887174
```

!!! tip If you aren't sure if you should use JuMP, read Should you use JuMP?.

Resources for getting started

There are a few ways to get started with JuMP:

- Read the Installation Guide.
- Read the introductory tutorials Getting started with Julia and Getting started with JuMP.
- Browse some of our modeling tutorials, including classics such as The diet problem, or the Maximum likelihood estimation problem using nonlinear programming.

!!! tip Need help? Join the community forum to search for answers to commonly asked questions.

```
Before asking a question, make sure to read the post [make it easier to help you]
(https://discourse.julialang.org/t/psa-make-it-easier-to-help-you/14757),
which contains a number of tips on how to ask a good question.
```

How the documentation is structured

Having a high-level overview of how this documentation is structured will help you know where to look for certain things.

- **Tutorials** contain worked examples of solving problems with JuMP. Start here if you are new to JuMP, or you have a particular problem class you want to model.
- The **Manual** contains short code-snippets that explain how to achieve specific tasks in JuMP. Look here if you want to know how to achieve a particular task, such as how to Delete a variable or how to Modify an objective coefficient.
- The **API Reference** contains a complete list of the functions you can use in JuMP. Look here if you want to know how to use a particular function.
- The **Background information** section contains background reading material to provide context to JuMP. Look here if you want an understanding of what JuMP is and why we created it, rather than how to use it.

- The **Developer docs** section contains information for people contributing to JuMP development or writing JuMP extensions. Don't worry about this section if you are using JuMP to formulate and solve problems as a user.
- The **MathOptInterface** section is a self-contained copy of the documentation for MathOptInterface. Look here for functions and constants beginning with `MOI.`, as well as for general information on how MathOptInterface works.

Citing JuMP

If you find JuMP useful in your work, we kindly request that you cite the following paper (preprint):

```
@article{Lubin2023,
    author = {Miles Lubin and Oscar Dowson and Joaquim {Dias Garcia} and
              Joey Huchette and Beno{\^i}t Legat and Juan Pablo Vielma},
    title = {{JuMP} 1.0: {R}ecent improvements to a modeling language
             for mathematical optimization},
    journal = {Mathematical Programming Computation},
    year = {2023},
    doi = {10.1007/s12532-023-00239-3}
}
```

NumFOCUS

Figure 2: NumFOCUS logo

JuMP is a Sponsored Project of NumFOCUS, a 501(c)(3) nonprofit charity in the United States. NumFOCUS provides JuMP with fiscal, legal, and administrative support to help ensure the health and sustainability of the project. Visit numfocus.org for more information.

You can support JuMP by donating.

Donations to JuMP are managed by NumFOCUS. For donors in the United States, your gift is tax-deductible to the extent provided by law. As with any donation, you should consult with your tax adviser about your particular tax situation.

JuMP's largest expense is the annual JuMP-dev workshop. Donations will help us provide travel support for JuMP-dev attendees and take advantage of other opportunities that arise to support JuMP development.

License

JuMP is licensed under the MPL-2.0 software license. Consult the license and the Mozilla FAQ for more information. In addition, JuMP is typically used in conjunction with solver packages and extensions which have their own licences. Consult their package repositories for the specific licenses that apply.