

Sum-of-Squares optimization with JuMP in Julia

Benoît Legat

September, 2021

POEMA Learning Week 2

Background

Examples

Extensions

Background

Nonnegative polynomial into sum of squares (SOS)

$$(x^3, x^2, x) \quad \text{not unique}$$
$$p(x) = X^T Q X$$

$$x^6 - 2x^4 + 2x^2 = X^T \begin{pmatrix} 1 & 0 & -1 \\ . & 0 & 0 \\ . & . & 2 \end{pmatrix} X$$

$$p(x) \geq 0 \quad \forall x \iff Q \geq 0 \quad \downarrow \text{cholesky}$$

$$x^2 + (-x^3 + x)^2 \longleftarrow X^T \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}^T \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} X$$

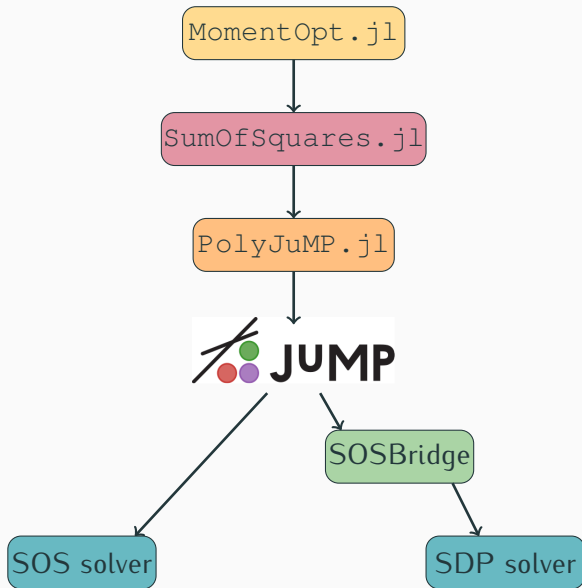
Polynomial optimization in Julia

- `ahmadreza-marandi/Polyopt.jl`
- `rdeits/Mayday.jl`
- `ameliaperry/SumOfSquaresOptimization.jl`
- `JulieSliwak/MathProgComplex.jl`
- `jump-dev/PolyJuMP.jl`,
`jump-dev/SumOfSquares.jl`
- `lanl-ansi/MomentOpt.jl`
- `tweisser/PolyPowerModels`,
`tweisser/PolyModels`
- `wangjie212/TSSOS`, `wangjie212/NCTSSOS`
- `wangjie212/SONCSOCP`
- `blegat/CondensedMatterSOS.jl`

History

- Started in August 2016: SOSTOOLS demos.
- October 2016: Amelia Perry, Joey Huchette, Robin Deits and Chris Coey **join**, package **split** into MultivariatePolynomials, PolyJuMP and SumOfSquares.
- April 2017, DynamicPolynomials, TypedPolynomials,
- August 2017 MultivariateMoments and SemialgebraicSets **factored out** of SumOfSquares.
- 2017–2019 MathOptInterface (MOI) → **rewrite** of JuMP, PolyJuMP and SumOfSquares.
- 2019–2020 MutableArithmetics for **performance**.
- 2019–2020 Tillmann Teweiser **joins**, develop MomentOpt.
- 2020–2021 Marek Kaluba **joins**, develop SymbolicWedderburn for symmetry reduction.

Overview



Examples

`https://jump.dev/SumOfSquares.jl/stable/
generated/Polynomial%20Optimization/
polynomial_optimization/`

Noncommutative variables

[https://jump.dev/SumOfSquares.jl/stable/
generated/Noncommutative%20and%20Hermitian/
noncommutative_variables/](https://jump.dev/SumOfSquares.jl/stable/generated/Noncommutative%20and%20Hermitian/noncommutative_variables/)

Symmetry reduction

- https://jump.dev/SumOfSquares.jl/stable/generated/Symmetry/permutation_symmetry/
- https://jump.dev/SumOfSquares.jl/stable/generated/Symmetry/even_reduction/
- https://jump.dev/SumOfSquares.jl/stable/generated/Symmetry/dihedral_symmetry_of_the_robinson_form/

Extensions

New polynomial implementation

`https://blegat.github.io/CondensedMatterSOS.jl/dev/generated/Ising/`

New certificate

```
https://jump.dev/SumOfSquares.jl/stable/  
generated/Extension/certificate/
```

If certificate not **flexible** enough: define new `MathOptInterface` **bridge**.

`https://jump.dev/SumOfSquares.jl/stable/
generated/Extension/univariate_solver/`

PolyJuMP extension

What does this **means** ?

```
1 @polyvar x y
2 @constraint(model, x^4 + y^4 >= x * y - 1)
```

PolyJuMP needs a **polymodule** (shortcut: `SOSModel`):

```
1 model = Model()
2 setpolymodule!(model, SumOfSquares)
```

With `SumOfSquares.jl`, it is **interpreted** as:

```
1 @constraint(model, x^4 + y^4 - x * y + 1 in SOSCone())
```

New bases

MultivariateBases provides:

- FixedPolynomialBasis: Any vector of polynomials.
- Monomial bases: MonomialBasis and ScaledMonomialBasis.
- Orthogonal bases:
 - Hermite bases: ProbabilistsHermiteBasis and PhysicistsHermiteBasis.
 - LaguerreBasis.
 - Gegenbauer bases:
 - LegendreBasis.
 - Chebyshev bases: ChebyshevBasisFirstKind and ChebyshevBasisSecondKind

Any other basis implementing the MultivariateBases API can be used with SumOfSquares!

See `https://jump.dev/SumOfSquares.jl/stable/generated/Extension/hypercube/`.

Conclusion

- `MathOptInterface.jl` (MOI) is extensible by design: Defining new set and bridges is the way to go.
- `JuMP.jl` macros are designed for extensions as well:
- `PolyJuMP.jl` extends `JuMP` macros for polynomials, defines MOI set and bridges for polynomial equality in algebraic set.
- `SumOfSquares.jl` is a `PolyJuMP`'s polymodule, defines MOI set and bridges (**parametrized** by **certificates**) for SOS constraints in basic semialgebraic set.