

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

=====
Traceback (most recent call last):

File "C:\Users\Vickram\Downloads\projpyth\pyjultransistor.py", line 81, in <module>
 prob_f = de.jit(prob_mm)
File "C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\jllwrap\any.jl", line 208, in __call__
 return self._jl_callmethod(\$(pyjl_methodnum(pyjlany_call)), args, kwargs)
juliacall.JuliaError: Non-permutation mass matrix is not supported.

Stacktrace:

```
[1] error(s::String)
    @ Base .\error.jl:35
[2] #782
    @
C:\Users\Vickram\.julia\packages\ModelingToolkit\Mxj1Q\src\systems\diffEQs\modeling
toolkitize.jl:40 [inlined]
[3] iterate
    @ .\generator.jl:47 [inlined]
[4] _collect(c::Vector{Symbolics.Num}, itr::Base.Generator{Vector{Symbolics.Num},
ModelingToolkit.var"#782#786"{Symbolics.Differential, Set{Symbolics.Num}}},
::Base.EltypeUnknown, isz::Base.HasShape{1})
    @ Base .\array.jl:854
[5] collect_similar
    @ .\array.jl:763 [inlined]
[6] map
    @ .\abstractarray.jl:3285 [inlined]
[7] modelingtoolkitize(prob::SciMLBase.ODEProblem{Vector{Float64}, Tuple{Float64,
Float64}, true, Tuple{Float64, Float64, Float64, Float64, Float64, Float64,
Float64, Float64, Float64, Float64, Float64, Float64, Int64},
SciMLBase.ODEFunction{true, SciMLBase.FullSpecialize,
ComposedFunction{typeof(SciMLBasePythonCallExt._pyconvert), Py}, Matrix{Float64},
Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing,
Nothing, Nothing, typeof(SciMLBase.DEFAULT_OBSERVED), Nothing,
SymbolicIndexingInterface.SymbolCache{Nothing, Nothing, Nothing, Dict{Symbol,
Union{Expr, Number, Symbol}}}, Nothing, Nothing}, @Kwargs{}},
SciMLBase.StandardODEProblem}; kwargs::@Kwargs{})
    @ ModelingToolkit
C:\Users\Vickram\.julia\packages\ModelingToolkit\Mxj1Q\src\systems\diffEQs\modeling
toolkitize.jl:34
[8] modelingtoolkitize
    @
C:\Users\Vickram\.julia\packages\ModelingToolkit\Mxj1Q\src\systems\diffEQs\modeling
toolkitize.jl:6 [inlined]
[9] jit(x::SciMLBase.ODEProblem{Vector{Float64}, Tuple{Float64, Float64}, true,
Tuple{Float64, Float64, Float64, Float64, Float64, Float64, Float64, Float64,
Float64, Float64, Float64, Float64, Int64}, SciMLBase.ODEFunction{true,
```

```
SciMLBase.FullSpecialize,  
ComposedFunction{typeof(SciMLBasePythonCallExt._pyconvert), Py}, Matrix{Float64},  
Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing,  
Nothing, Nothing, typeof(SciMLBase.DEFAULT_OBSERVED), Nothing,  
SymbolicIndexingInterface.SymbolCache{Nothing, Nothing, Nothing, Dict{Symbol,  
Union{Expr, Number, Symbol}}}, Nothing, Nothing}, @Kwargs{  
SciMLBase.StandardODEProblem})
```

```
@ Main .\none:1
```

```
[10] pyjlany_call(self::typeof(jit), args_::Py, kwargs_::Py)
```

```
@ PythonCall
```

```
C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\jlwrap\any.jl:37
```

```
[11] _pyjl_callmethod(f::Any, self_::Ptr{PythonCall.C.PyObject},  
args_::Ptr{PythonCall.C.PyObject}, nargs::Int64)
```

```
@ PythonCall
```

```
C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\jlwrap\base.jl:69
```

```
[12] _pyjl_callmethod(o::Ptr{PythonCall.C.PyObject},  
args::Ptr{PythonCall.C.PyObject})
```

```
@ PythonCall.C
```

```
C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\cpython\jlwrap.jl:47
```

```
=====  
Traceback (most recent call last):
```

```
File "C:\Users\Vickram\Downloads\projpyth\pyjultransistor.py", line 81, in  
<module>
```

```
    prob_f = de.jit(prob_mm)
```

```
File "C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\jlwrap\any.jl", line  
208, in __call__
```

```
    return self._jl_callmethod($(pyjl_methodnum(pyjlany_call)), args, kwargs)
```

```
juliacall.JuliaError: Non-permutation mass matrix is not supported.
```

```
Stacktrace:
```

```
[1] error(s::String)
```

```
@ Base .\error.jl:35
```

```
[2] #782
```

```
@
```

```
C:\Users\Vickram\.julia\packages\ModelingToolkit\Mxj1Q\src\systems\diffeqs\modeling  
toolkitize.jl:40 [inlined]
```

```
[3] iterate
```

```
@ .\generator.jl:47 [inlined]
```

```
[4] _collect(c::Vector{Symbolics.Num}, itr::Base.Generator{Vector{Symbolics.Num},  
ModelingToolkit.var"#782#786"{Symbolics.Differential, Set{Symbolics.Num}}},  
::Base.EltypeUnknown, isz::Base.HasShape{1})
```

```
@ Base .\array.jl:854
```

```
[5] collect_similar
```

```
@ .\array.jl:763 [inlined]
```

```
[6] map
```

```
@ .\abstractarray.jl:3285 [inlined]
```

```
[7] modelingtoolkitize(prob::SciMLBase.ODEProblem{Vector{Float64}, Tuple{Float64,  
Float64}, true, Tuple{Float64, Float64, Float64, Float64, Float64, Float64,  
Float64, Float64, Float64, Float64, Float64, Float64, Int64},  
SciMLBase.ODEFunction{true, SciMLBase.FullSpecialize,
```

```

ComposedFunction{typeof(SciMLBasePythonCallExt._pyconvert), Py}, Matrix{Float64},
Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing,
Nothing, Nothing, typeof(SciMLBase.DEFAULT_OBSERVED), Nothing,
SymbolicIndexingInterface.SymbolCache{Nothing, Nothing, Nothing, Dict{Symbol,
Union{Expr, Number, Symbol}}}, Nothing, Nothing}, @Kwargs{ },
SciMLBase.StandardODEProblem}; kwargs::@Kwargs{ })
    @ ModelingToolkit
C:\Users\Vickram\.julia\packages\ModelingToolkit\Mxj1Q\src\systems\diffEQs\modeling
toolkitize.jl:34
    [8] modelingtoolkitize
        @
C:\Users\Vickram\.julia\packages\ModelingToolkit\Mxj1Q\src\systems\diffEQs\modeling
toolkitize.jl:6 [inlined]
    [9] jit(x::SciMLBase.ODEProblem{Vector{Float64}, Tuple{Float64, Float64}, true,
Tuple{Float64, Float64, Float64, Float64, Float64, Float64, Float64, Float64,
Float64, Float64, Float64, Float64, Float64, Int64}, SciMLBase.ODEFunction{true,
SciMLBase.FullSpecialize,
ComposedFunction{typeof(SciMLBasePythonCallExt._pyconvert), Py}, Matrix{Float64},
Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing,
Nothing, Nothing, typeof(SciMLBase.DEFAULT_OBSERVED), Nothing,
SymbolicIndexingInterface.SymbolCache{Nothing, Nothing, Nothing, Dict{Symbol,
Union{Expr, Number, Symbol}}}, Nothing, Nothing}, @Kwargs{ },
SciMLBase.StandardODEProblem})
        @ Main .\none:1
    [10] pyjlany_call(self::typeof(jit), args_::Py, kwargs_::Py)
        @ PythonCall
C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\jlwrap\any.jl:37
    [11] _pyjl_callmethod(f::Any, self_::Ptr{PythonCall.C.PyObject},
args_::Ptr{PythonCall.C.PyObject}, nargs::Int64)
        @ PythonCall
C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\jlwrap\base.jl:69
    [12] _pyjl_callmethod(o::Ptr{PythonCall.C.PyObject},
args_::Ptr{PythonCall.C.PyObject})
        @ PythonCall.C
C:\Users\Vickram\.julia\packages\PythonCall\wXfah\src\cpython\jlwrap.jl:47

```