Who are we?

Joel Andersson

FMIOPT AS, Norway
joel@fmiopt.com

Joris Gillis

Yacoda BV, Belgium
joris@yacoda.com

yacoda
Let's do some coding...

Consulting

Teaching

FMI & Modelica

Pharmacology

HVAC

Diagnostics

Motorsport

Trains, Boats

Delays

Radiotherapy

Wind energy

Abstractions

What is CasADi?

"Define system" → "Define problem based on system" → "Solve/deploy problem"

static/kinematic/dynamic

Initial value problem

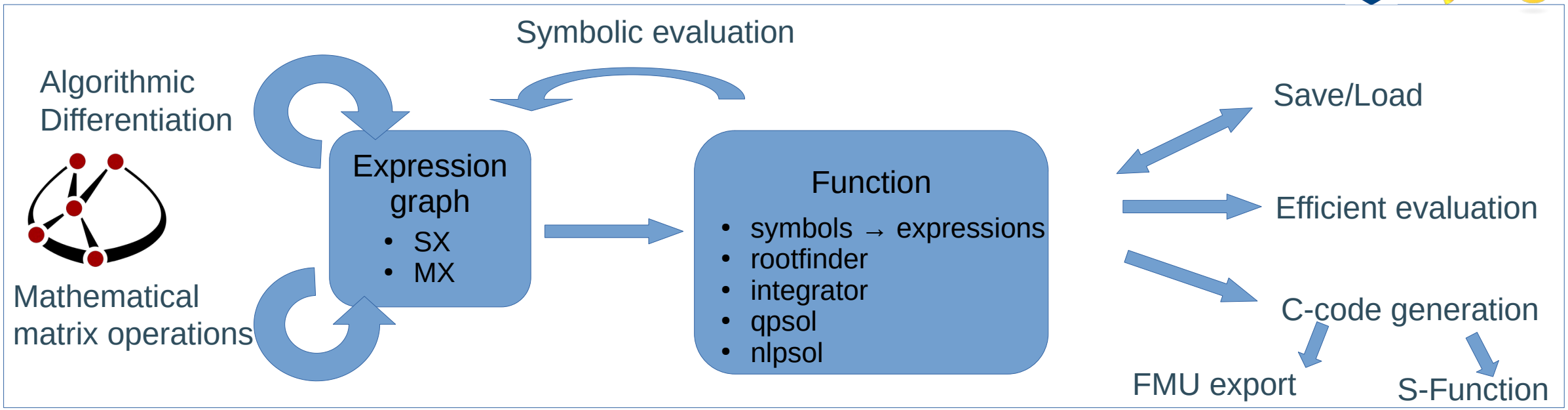
Rootfinding problem

Nonlinear constrained optimization

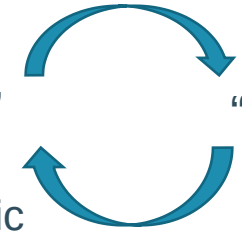
Ipopt, fatrop, SNOPT
Cvodes
OSQP, Highs
CPLEX,Gurobi
HPIPM...

Capabilities

Win, Linux, Mac



Usage patterns



"Define system"

"Define problem based on system"

"Solve/deploy problem"

static/kinematic/dynamic

- From scratch
- FMU Model-Exchange (e.g. modelica)
- Callback class: tensorflow, pytorch
- Virtual machine: mupad, sympy
- Parsers: pymoca, simscape

Initial value problem

Rootfinding problem

Nonlinear constrained optimization

Ipopt, fatrop, SNOPT
Cvodes
OSQP, Highs
CPLEX,Gurobi
HPIPM...

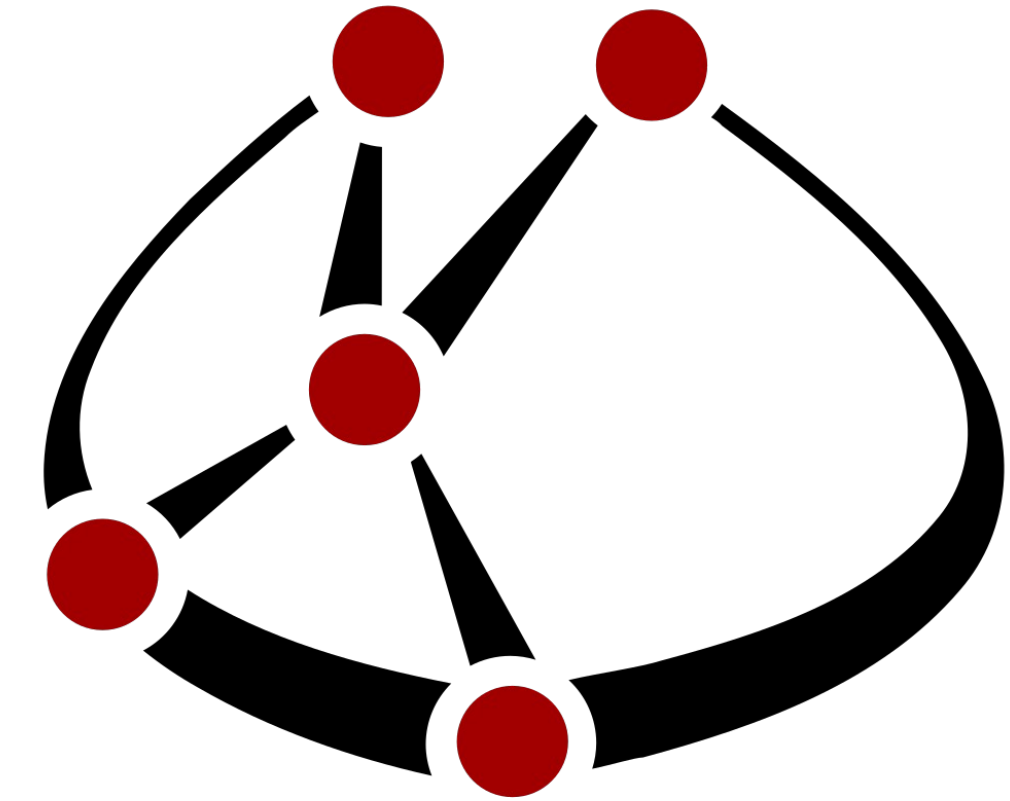
Aim

Support mathematical engineering

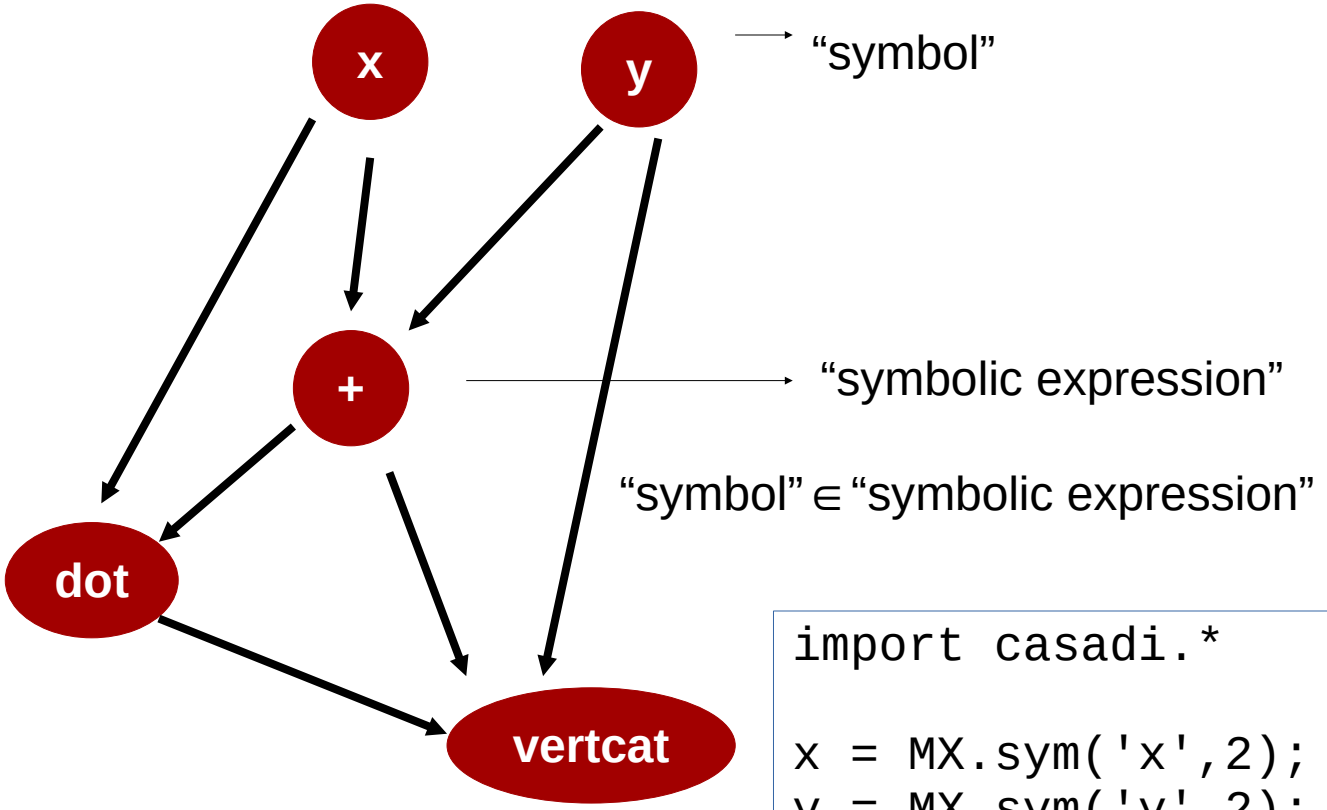
Open-Source

Foundation to build upon

Syntax: symbolic expressions



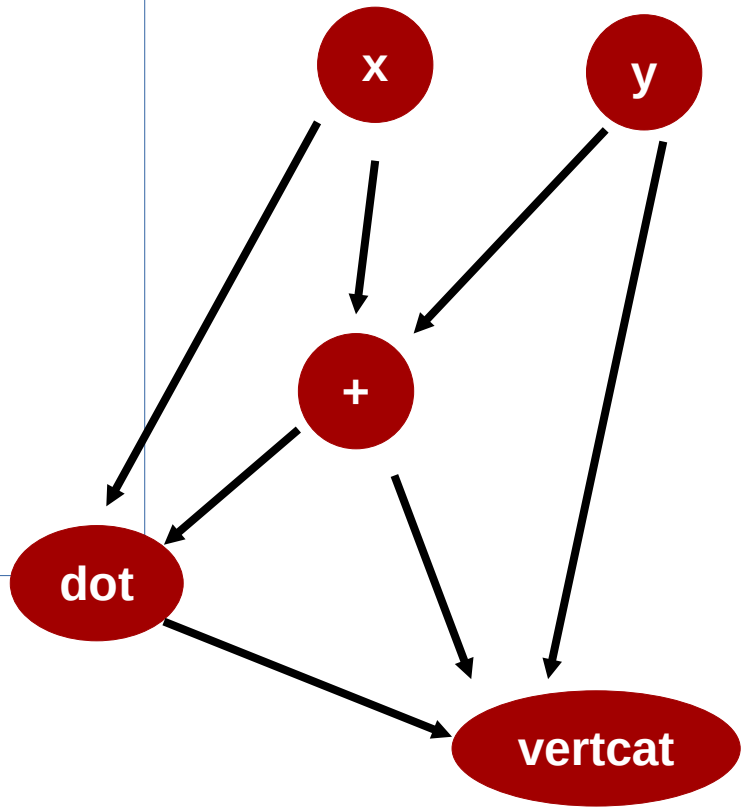
CasADi




```
import casadi.*

x = MX.sym('x',2);
y = MX.sym('y',2);

w1 = x+y;
w2 = dot(x,w1);
w3 = [y;w1;w2];
size(w3)
5  1
```

Syntax: Functions

```
F = Function('f',{x,y},{w3},{'x','y'},{'z'});
f:(x[2],y[2])->(z[5]) MXFunction

F([1,2],[3,4])
[3, 4, 4, 6, 16]

F.generate('F.c')

F('y',[3,4],'x',[1,2])
  struct with fields:

    z: [5×1 casadi.DM]
```

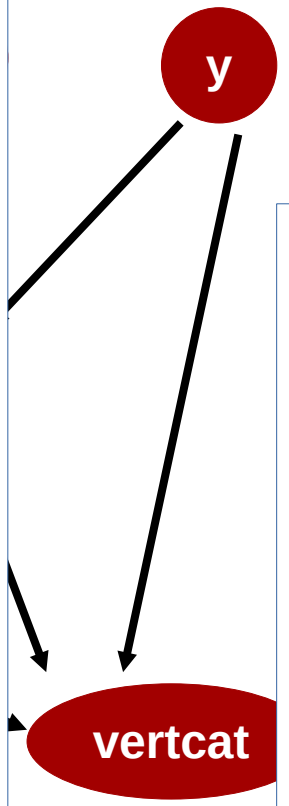



```
import casadi.*

x = MX.sym('x',2);
y = MX.sym('y',2);

w1 = x+y;
w2 = dot(x,w1);
w3 = [y;w1;w2];
size(w3)
5  1
```

Syntax: algorithmic differentiation and sparsity

```
F = Function('f',{x,y},{w3},{'x','y'},{'z'});
```

```
class(w3)
'casadi.MX'

J = jacobian(w3,y);
class(J)
'casadi.MX'

spy(sparsity(J))
*.
.*
*.
.*
**

Fd = Function('Fd',{x,y},{w3,J});
Fd:(i0[2],i1[2])->(o0[5],o1[5x2,6nz]) MXFunction

Fd = F.factory('Fd',{'x','y'},{'z','jac:z:y'});
Fd:(x[2],y[2])->(z[5],jac_z_y[5x2,6nz]) MXFunction

[A,B] = Fd([1 2],[3 4])
```




```
class(A)
'casadi.DM'
class(B)
'casadi.DM'

B
B =


[[1, 00],
 [00, 1],
 [1, 00],
 [00, 1],
 [1, 2]]

full(A)
sparse(B)
```