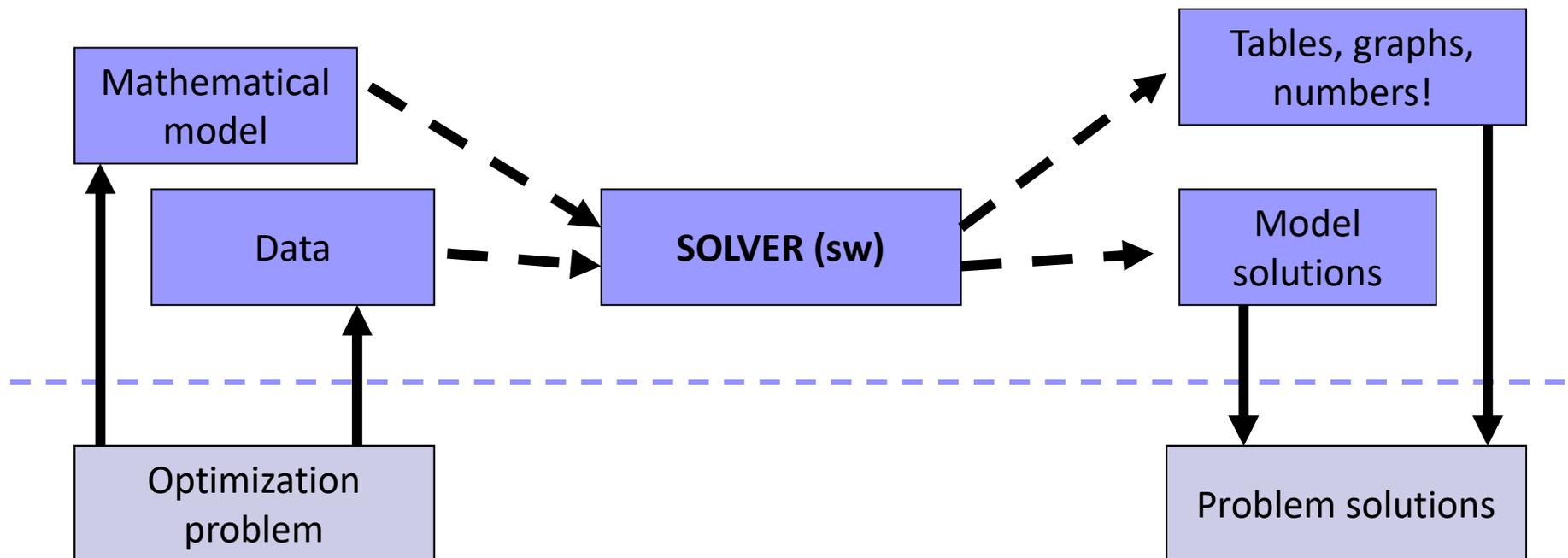


# Solvers for Mathematical Programming

# Solvers (optimizing engines)

A **solver** is a software application that takes the description of an optimization problem as **input** and provides the solution of the model (and related information) as **output**.



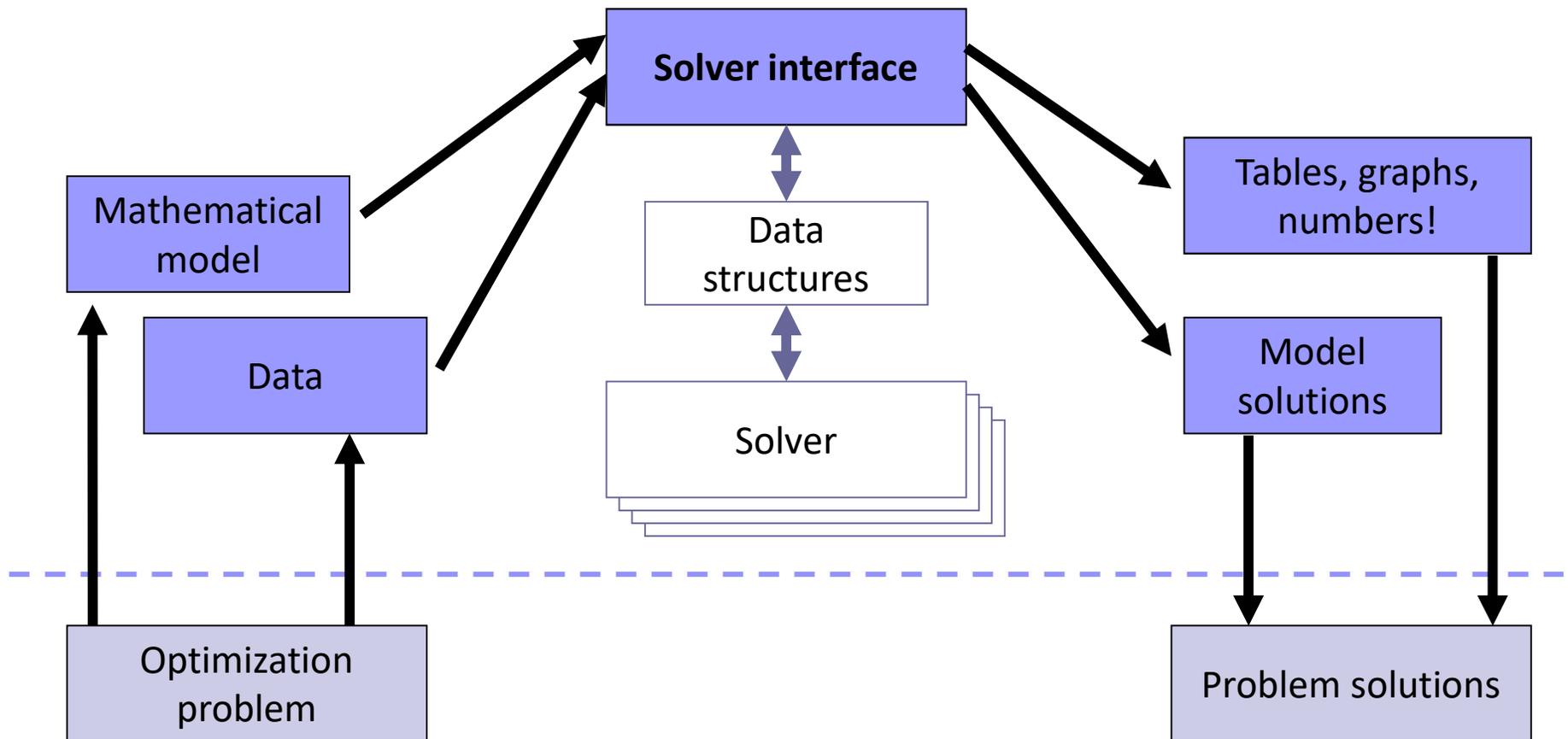


## MILP solvers

- Most used in practice:
  - very efficient
  - numerical stability
  - easy to use or embed
- **1 000 000 000 speed-up** in the last 15 years
  - hardware speed-up: x 1000
  - simplex improvements: x 1000
  - branch-and-cut improvement: x 1000
- e.g. Cplex, Gurobi, Xpress, Scip, Lindo, GLPK etc.

# Solver interfaces

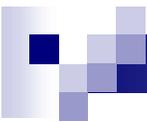
A solver can be accessed via modelling languages or **general-purpose-language libraries**





# IBM Ilog Cplex

- One of the first MILP solvers
- Includes **state-of-the-art** technology
- (One of) the best solvers available (Gurobi, Xpress)
- Possible interfaces
  - Interactive optimizer
  - **C – API libraries (Callable libraries)**
  - C++ libraries (Concert technologies)
  - Python / Java / .Net wrapper libraries
  - Matlab / Excel plugins
  - OPL / AMPL / ZIMPL ... algebraic modelling language



## Cplex Callable Libraries

- C API towards *LP/QP/MIP/MIQP* algorithms
- Basic objects: **Environment** and **Problem**
- **Environment**: license, optimization parameters ...
- **Problem**: contains problem information: variables, constraints ...)
- (at least one) environment and problem must be created

**CPXENVptr** **CPXopenCPLEX** / **CPXcloseCPLEX**

**CPXLPptr** **CPXcreateprob** / **CPXfreeprob**

## Cplex API functions

- The two objects can be accessed (e.g. to add variables or constraints, or to solve a problem) via the functions provided by the API
- (Almost) all the API functions can be called as

```
int CPXfuncName (environment [,problem] ,... ) ;
```

Error code (0=ok)  
CPXgetErrorstring returns a  
description of the error

Basic objects

Parameters

# Sparse matrix representation

- Sparse matrix: many zero entries
- Compact representation:
  - Explicit representation of “nonzeroes”
  - Linearization into indexes (**idx**) and values (**val**) vectors
  - A third vector to indicate where rows begins (**beg**)

