Python 🐍 and AMPL :

Build Prescriptive Analytics applications quickly with Pandas, Colab, Streamlit, and amplpy

Filipe Brandão, Robert Fourer

{filipe,4er@ampl.com}

AMPL Optimization Inc.
www.ampl.com — +1 773-336-AMPL

INFORMS 2023, Phoenix, Arizona — October 15-18, 2023, CC-North 231B,
Technology Tutorial, Tuesday, October 17, 2:55 - 3:30 pm

Outline

Part I:

- Quick introduction to AMPL and new modeling functionalities

Part II (live demos):

- Quick introduction to amplpy (our Python API)
- AMPL on Google Colab
    - AMPL Model Colaboratory (https://colab.ampl.com)
    - New book: **Hands-On Optimization with AMPL in Python** (https://mo-book.ampl.com)
- AMPL and solvers as python packages
- AMPL on Streamlit Cloud
- How to deploy large-scale optimization applications with AMPL

Formulating Models More Like You Think About Them

Describe an optimization problem

- In a form *you find natural or convenient*
- Using readily recognized expressions

Send it to a solver

- In a form *the solver will accept*
- Relying on the modeling software to translate

Get back a result

- In the form you originally used

Typical User Complaint

Thank you so much for replying.
Let me show my "if-then" constraint in a more clear way as follows:

```
set veh := {1..16 by 1};

param veh_ind {veh};
param theory_time {veh};
param UP := 400000;

var in_lane_veh {veh} integer >=1, <=2;
var in_in_time {veh} >=0, <=UP;
```

Note that "in_lane_veh {veh}" are integer variables which equal 1 or 2,
and "in_in_time {veh}" are continuous variables.

```
subject to IfConstr {i in 1..card(veh)-1, j in i+1..card(veh):
  veh_ind[i] = veh_ind[j] and theory_time[i] <= theory_time[j]}:

    in_lane_veh[i] = in_lane_veh[j] ==> in_in_time[j] >= in_in_time[i] + l_veh/V;
```

When I run my program, there appears the following statement:
CPLEX 20.1.0.0: logical constraint _slogcon[1] is not an indicator constraint.

Typical Reply

To reformulate this model in a way that your MIP solver would accept,
you could define some more binary variables,

```
var in_lane_same {veh,veh} binary;
```

with the idea that in_lane_same[i,j] should be 1 if and only if in_lane_veh[i] = in_lane_veh[j].
Then the desired relation could be written as two constraints:

```
in_lane_veh[i] = in_lane_veh[j] ==> in_lane_same[i,j] = 1
in_lane_same[i,j] = 1 ==> in_in_time[j] >= in_in_time[i] + l_veh/V;
```

The second one is an indicator constraint, but you would just need
to replace the first one by equivalent linear constraints.

Given that in_lan_veh can only be either 1 or 2, those constraints could be

```
in_lane_same[i,j] >= 3 - in_lane_veh[i] - in_lane_veh[j]
in_lane_same[i,j] >= in_lane_veh[i] + in_lane_veh[j] - 3
```

New Solver Interface Library (MP)

Design

- C++ library for building efficient, configurable solver drivers
- Support for features of current C interface library
- *Extensive toolset for problem recognition and transformation*

Motivation . . .

- AMPL has logical and “not linear” expressions
  for *writing models the way you think of them*
- Old interfaces have very limited support for these
- New interfaces, built with MP,
  allow these expressions to be used and combined freely

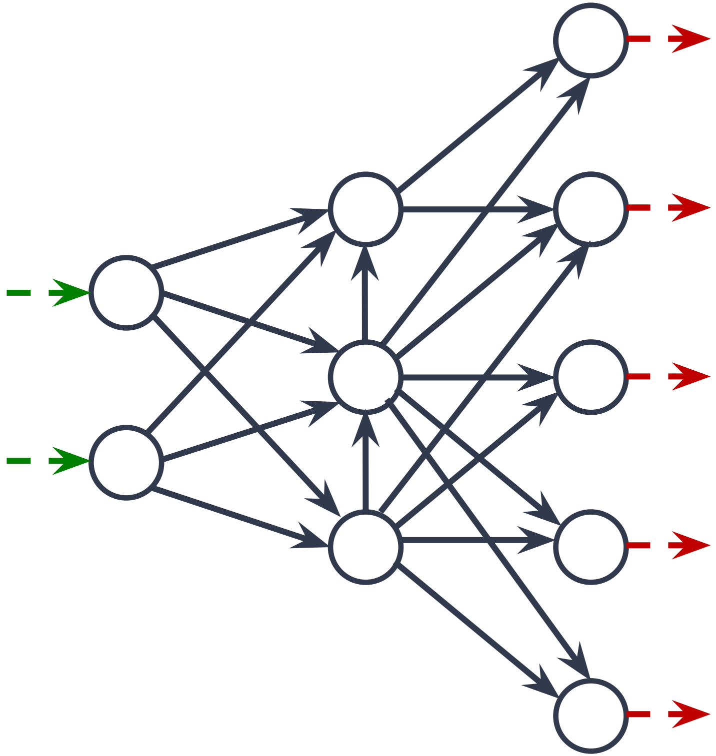
Example: Multi-Product Network Flow

Motivation

- Ship products efficiently to meet demands

Context

- a transportation network
  - Nodes representing cities
  - arcs representing roads
- supplies at nodes
- demands at nodes
- capacities on arcs
- shipping costs on arcs



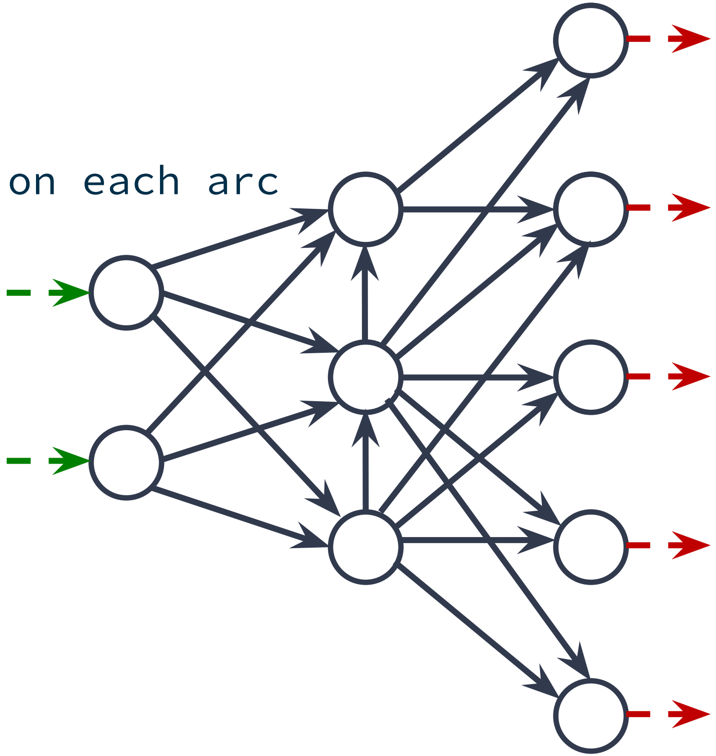
Example: Multi-Product Network Flow

Decide

- how much of each product to ship on each arc

So that

- shipping costs are kept low
- shipments on each arc respect capacity of the arc
- supplies, demands, and shipments are in balance at each node



AMPL Model for Multi-Product Network Flow

```
set PRODUCTS;
set NODES;
param net_inflow {PRODUCTS,NODES};

set ARCS within {NODES,NODES};
param capacity {ARCS} >= 0;

param var_cost {PRODUCTS,ARCS} >= 0;
var Flow {PRODUCTS,ARCS} >= 0;

minimize TotalCost:
   sum {p in PRODUCTS, (i,j) in ARCS} var_cost[p,i,j] * Flow[p,i,j];

subject to Capacity {(i,j) in ARCS}:
   sum {p in PRODUCTS} Flow[p,i,j] <= capacity[i,j];

subject to Conservation {p in PRODUCTS, j in NODES}:
   sum {(i,j) in ARCS} Flow[p,i,j] + net_inflow[p,j] =
   sum {(j,i) in ARCS} Flow[p,j,i];
```

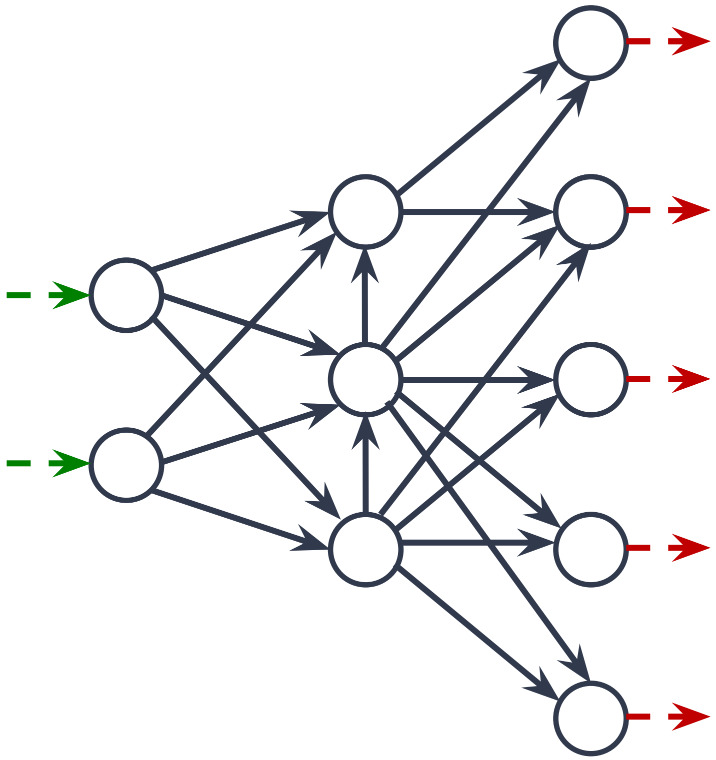
Example with conditions: Multi-Product Network Flow

Decide also

- whether to use each arc

So that

- variable costs plus fixed costs for shipping are kept low
- shipments are not too small
- not too many arcs are used



Positive Shipments Incur Fixed Costs

Linearization

```
param fix_cost {ARCS} >= 0;
var Use {ARCS} binary;

minimize TotalCost:
   sum {p in PRODUCTS, (i,j) in ARCS} var_cost[p,i,j] * Flow[p,i,j] +
   sum {(i,j) in ARCS} fix_cost[i,j] * Use[i,j];
```

How you think about it

```
param fix_cost {ARCS} >= 0;

minimize TotalCost:
   sum {p in PRODUCTS, (i,j) in ARCS} var_cost[p,i,j] * Flow[p,i,j] +
   sum {(i,j) in ARCS}
      if exists {p in PRODUCTS} Flow[p,i,j] > 0 then fix_cost[i,j];
```

Shipments Can't Be Too Small

Linearization

```
subject to Min_Shipment {(i,j) in ARCS}:
   sum {p in PRODUCTS} Flow[p,i,j] >= min_ship * Use[i,j];

subject to Capacity {(i,j) in ARCS}:
   sum {p in PRODUCTS} Flow[p,i,j] <= capacity[i,j] * Use[i,j];
```

How you think about it

```
subject to Shipment_Limits {(i,j) in ARCS}:
  sum {p in PRODUCTS} Flow[p,i,j] = 0 or
  min_ship <= sum {p in PRODUCTS} Flow[p,i,j] <= capacity[i,j];
```

Can't Use Too Many Arcs

Linearization

```
subject to Max_Used:
   sum {(i,j) in ARCS} Use[i,j] <= max_arcs;
```

How you think about it

```
subject to Limit_Used:
  atmost max_arcs {(i,j) in ARCS}
    (sum {p in PRODUCTS} Flow[p,i,j] > 0);
```

Linearization is Usually Not That Easy!

```
subject to IfConstr {i in 1..card(veh)-1, j in i+1..card(veh):
         veh_ind[i] = veh_ind[j] and theory_time[i] <= theory_time[j]}:
   in_lane_veh[i] = in_lane_veh[j]
      ==> in_in_time[j] >= in_in_time[i] + l_veh/V;
```

```
minimize total_fuelcost:
   sum{(i,j) in A} sum{k in V} X[i,j,k] *
      ((if H[i,k] <= 300 then dMor[i,j] else
        if H[i,k] <= 660 then dAft[i,j] else
        if H[i,k] <= 901 then dEve[i,j]) * 5 +
       (if H[i,k] <= 300 then tMor[i,j] else
        if H[i,k] <= 660 then tAft[i,j] else
        if H[i,k] <= 901 then tEve[i,j]) * 0.0504);
```

```
subject to NoPersonIsolated
      {l in TYPES['loc'], r in TYPES['rank'], j in 1..numberGrps}:
   sum {i in LOCRANK[l,r]} Assign[i,j] = 0 or
   sum {i in LOCRANK[l,r]} Assign[i,j] + sum {a in ADJACENT[r]} sum {i in LOCRANK[l,a]} Assign[i,j] >= 2;
```

Supported Extensions

Operators and functions

- Conditional: `if-then-else`; `==>`, `<==`, `<==>`
- Logical: `or`, `and`, `not`; `exists`, `forall`
- Piecewise linear: `abs`; `min`, `max`; `<<breakpoints; slopes>>`
- Counting: `count`; `atmost`, `atleast`, `exactly`; `numberof`
- Comparison: `>`, `<`, `!=`; `alldiff`
- Complementarity: `complements`
- Nonlinear: `*`, `/`, `^`; `exp`, `log`; `sin`, `cos`, `tan`; `sinh`, `cosh`, `tanh`
- Set membership: `in`

Expressions and constraints

- High-order polynomials
- Second-order and exponential cones (MOSEK driver!)

Supported Solvers

Solvers

- **Gurobi**, **Xpress**, **COPT**, **MOSEK**
- **HiGHS**, **CBC**, **SCIP**, **GCG**
- **CPLEX** soon

Modeling guide

- [*https://mp.ampl.com/model-guide.html*](https://mp.ampl.com/model-guide.html)

Examples using MP features

- [*https://colab.ampl.com*](https://colab.ampl.com)
- [*https://ampl.com/mo-book*](https://ampl.com/mo-book)

Quick introduction to amplpy! 🐍

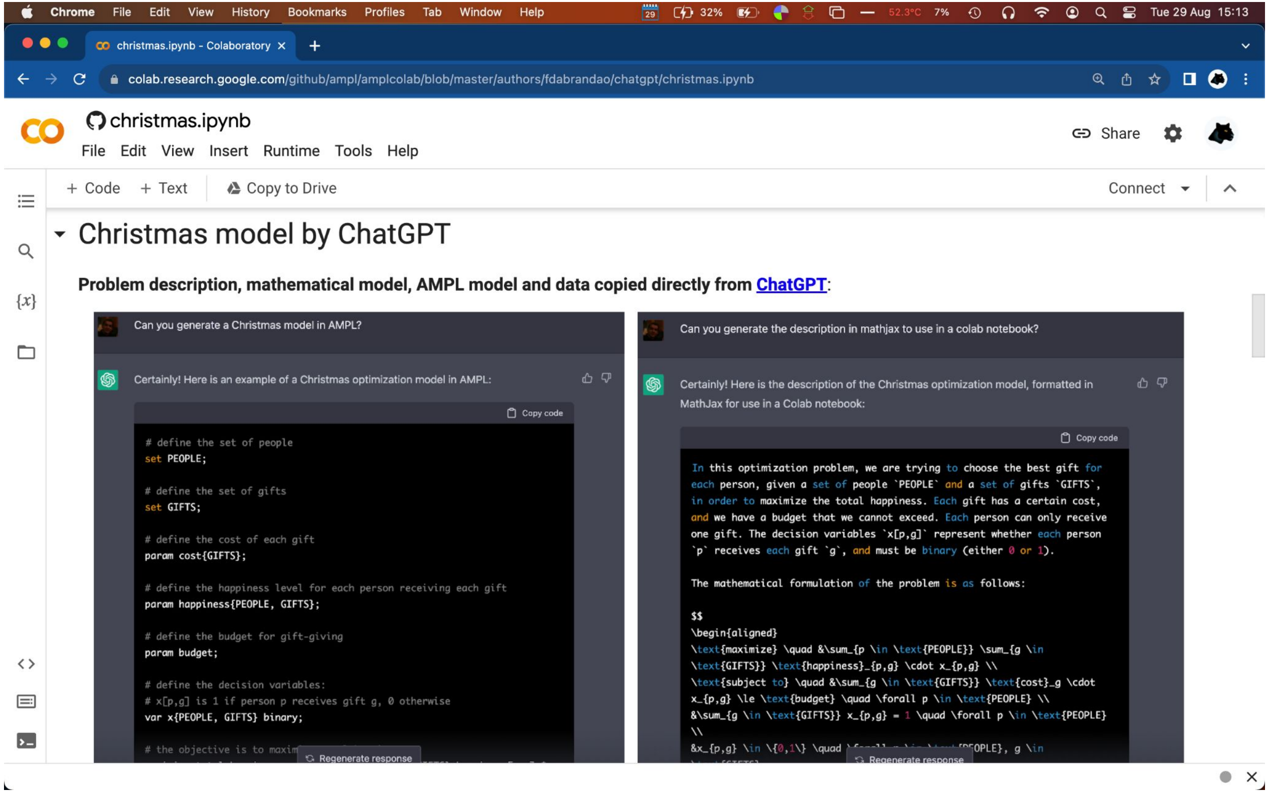
What do you need to know to use amplpy?

- Basic Python features (lists, dictionaries, etc.)

- Data manipulation with Pandas dataframes

- How to model in AMPL (or how to ask Chat GPT to write AMPL models for you!)

Example: Christmas model (https://colab.ampl.com)

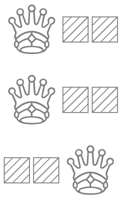


Example: N-Queens (https://colab.ampl.com)

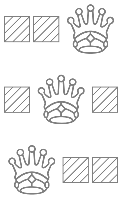
How can *n* queens be placed on an *n×n* chessboard so that no two of them attack each other?

Constraint **alldiff** enforces a set of integer variables to take distinct values. Using alldiff, we can model N-Queens as follows:

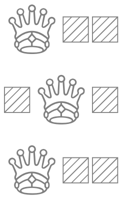
```
param n integer > 0; # N-queens
var Row {1..n} integer >= 1 <= n;
s.t. row_attacks: alldiff ({j in 1..n} Row[j]);
s.t. diag_attacks: alldiff ({j in 1..n} Row[j]+j);
s.t. rdiag_attacks: alldiff ({j in 1..n} Row[j]-j);
```



Row[1] == Row[2]
1 == 1

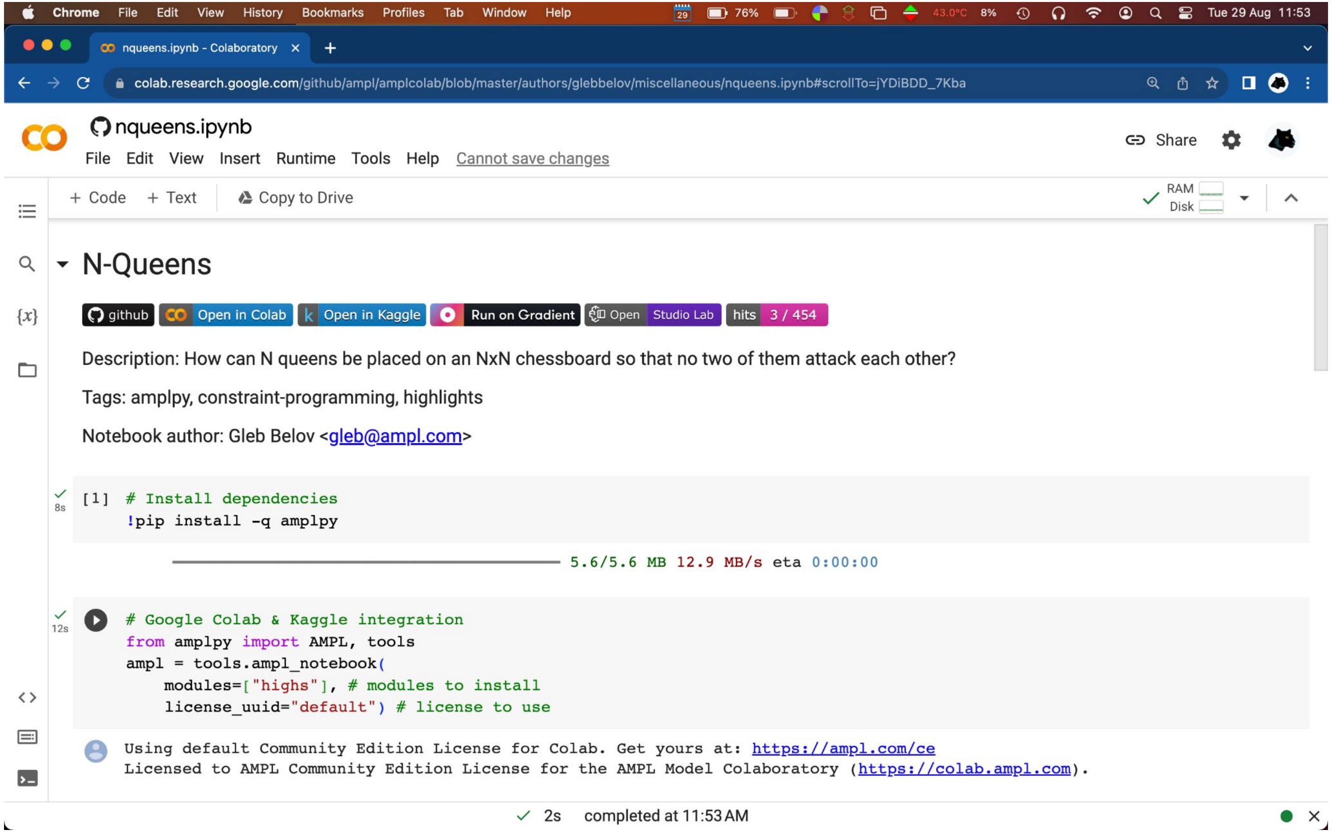


Row[1]+1 == Row[2]+2
3+1 == 2+2



Row[1]-1 == Row[2]-2
1-1 == 2-2

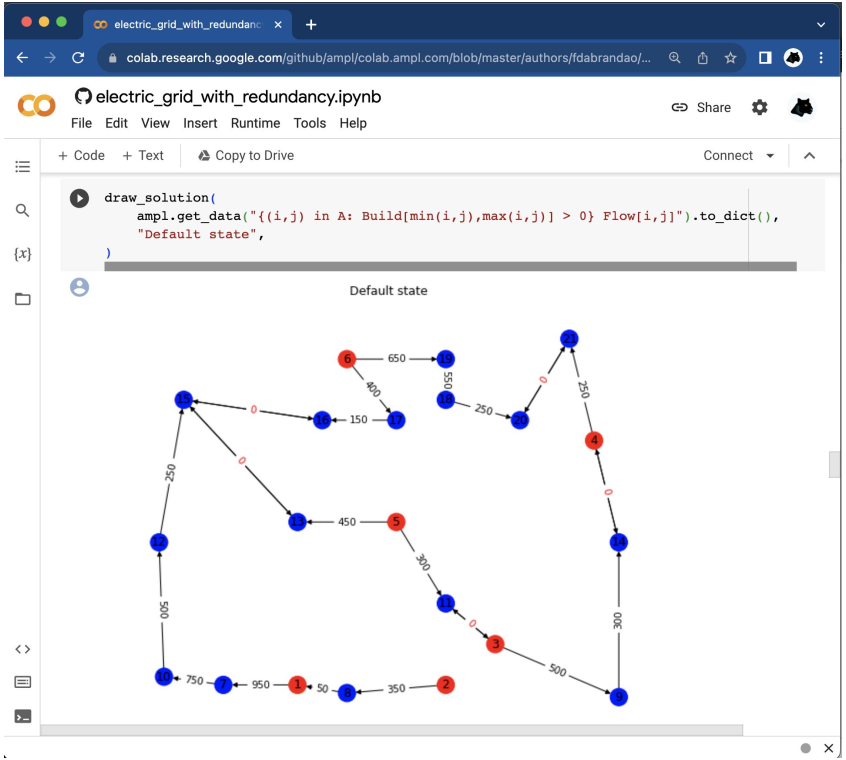
Example: N-Queens (https://colab.ampl.com)



Example: Network design with redundancy (https://colab.ampl.com)

electric_grid_with_redundancy.ipynb

```
param cost{L};      # cost to build a link
param capacity;     # capacity of links

var Build {L} binary;    # Build[i,j] = 1 iff link btw i & j is built
var Flow {A} >= 0;       # Flow[i,j] is flow from i to j

minimize TotalBuildCost:
    sum {(i,j) in L} cost[i,j] * Build[i,j];

subject to Balance {i in N}:
    sum {(j,i) in A} Flow[j,i] - sum{(i,j) in A} Flow[i,j] >= demsup[i];

subject to ArcExists1 {(i,j) in L}:
    Flow[i,j] <= Build[i,j] * capacity;
subject to ArcExists2 {(i,j) in L}:
    Flow[j,i] <= Build[i,j] * capacity;

var FlRm {A,N} >= 0; # FlRm[i,j,rn] is flow from i to j when node rn is removed

subject to RemoveNode {rm in N}:
    sum {(i,rm) in A} FlRm[i,rm,rm] + sum {(rm,j) in A} FlRm[rm,j,rm] = 0;

subject to BalanceRm {i in N, rm in N: i != rm}:
    sum {(j,i) in A} FlRm[j,i,rm] - sum {(i,j) in A} FlRm[i,j,rm] >= demsup[i];

subject to ArcExistsRm1 {(i,j) in L, rm in N}:
    FlRm[i,j,rm] <= Build[i,j] * capacity;
subject to ArcExistsRm2 {(i,j) in L, rm in N}:
    FlRm[j,i,rm] <= Build[i,j] * capacity;
```



Wait a minute. How are AMPL & solvers running on Google Colab integrated with Python 🐍?

AMPL and all Solvers are now available as Python Packages

AMPL and all solvers are now available as python packages for **Windows**, **Linux** (**X86_64**, **aarch64**, **ppc64le**), and **macOS**.

```
# Install Python API for AMPL
$ python -m pip install amplpy --upgrade

# Install solver modules (e.g., HiGHS, CBC, Gurobi)
$ python -m amplpy.modules install highs cbc gurobi

# Activate your license (e.g., free https://ampl.com/ce license)
$ python -m amplpy.modules activate <license-uuid>

# Import in Python
$ python
>>> from amplpy import AMPL
>>> ampl = AMPL() # instantiate AMPL object
```

> https://ampl.com/python/

AMPL is Free on Google Colab

> https://dev.ampl.com/ampl/python/colab.html

> https://try.ampl.com (quickly access to AMPL on Colab)

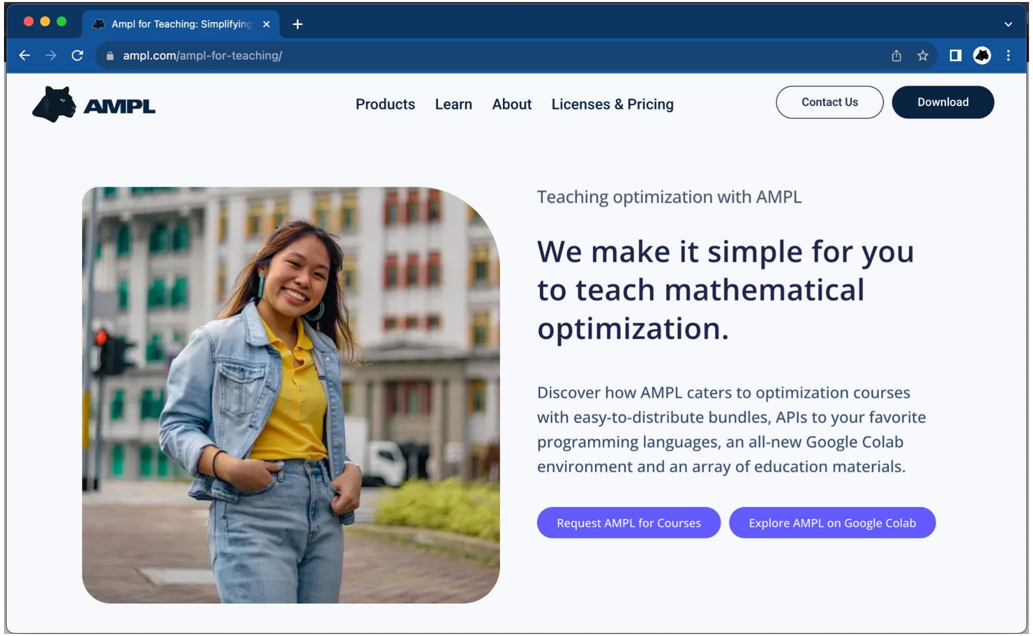
You can install AMPL on Google Colab (where it is free by default) as follows:

```
# Install dependencies
%pip install -q amplpy
```

```
# Google Colab & Kaggle integration
from amplpy import AMPL, tools
ampl = tools.ampl_notebook(
    modules=["gurobi", "coin", "highs", "gokestrel"], # modules to install
    license_uuid="default") # license to use
```

Free licenses to use on Google Colab (and locally!)

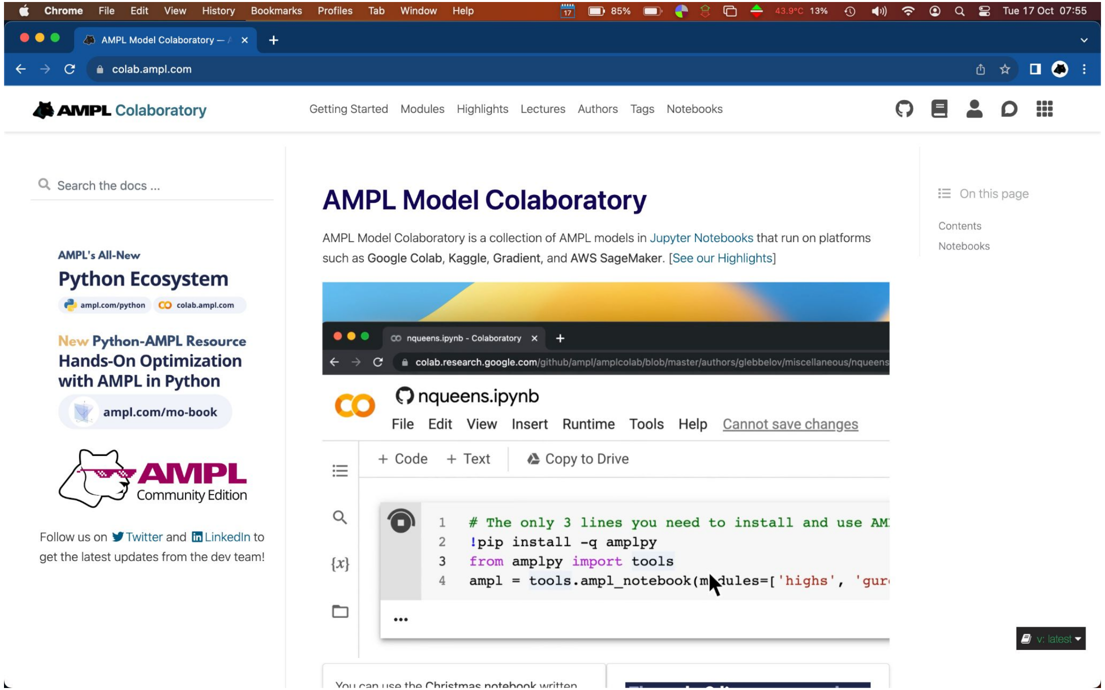
- **ampl.com/ce**
  - For personal use
  - Immediate access without approvals required
  - No size-limits
  - Includes access to:
    - Open-source solvers
    - Commercial solver trials
- **ampl.com/courses**
  - For teaching
  - No size-limits
  - **Full access to all solvers!**
  - All students can use the license during the course.



The Python-first 🐍 approach to learn and model with AMPL!

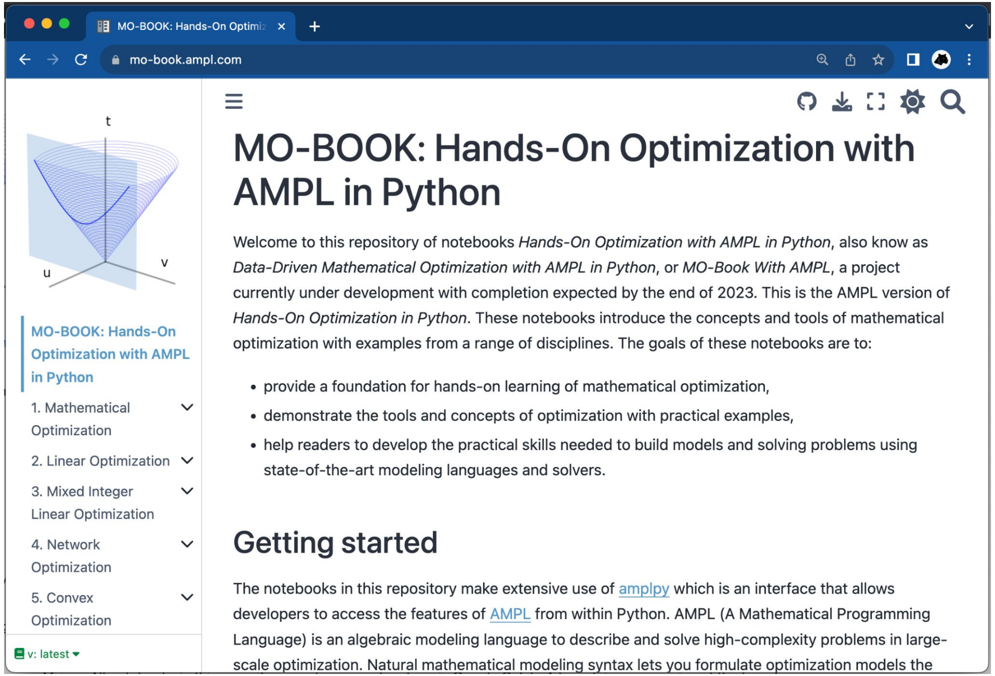
AMPL Model Colaboratory (https://colab.ampl.com)

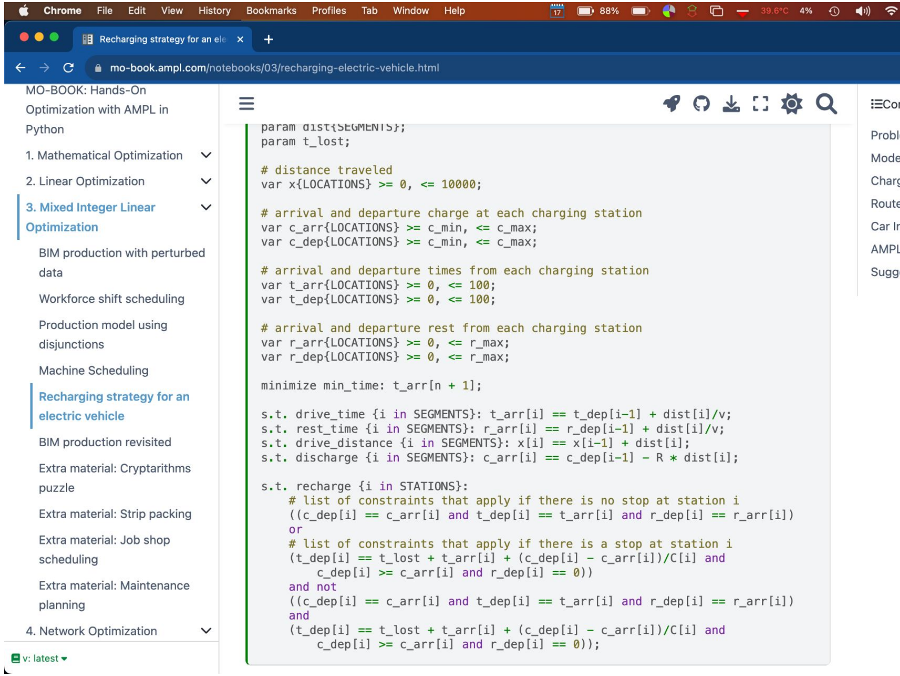
> Many examples: https://colab.ampl.com (live demo)



Data-Driven Mathematical Optimization with AMPL in Python

> New Book: https://ampl.com/mo-book (live demo)

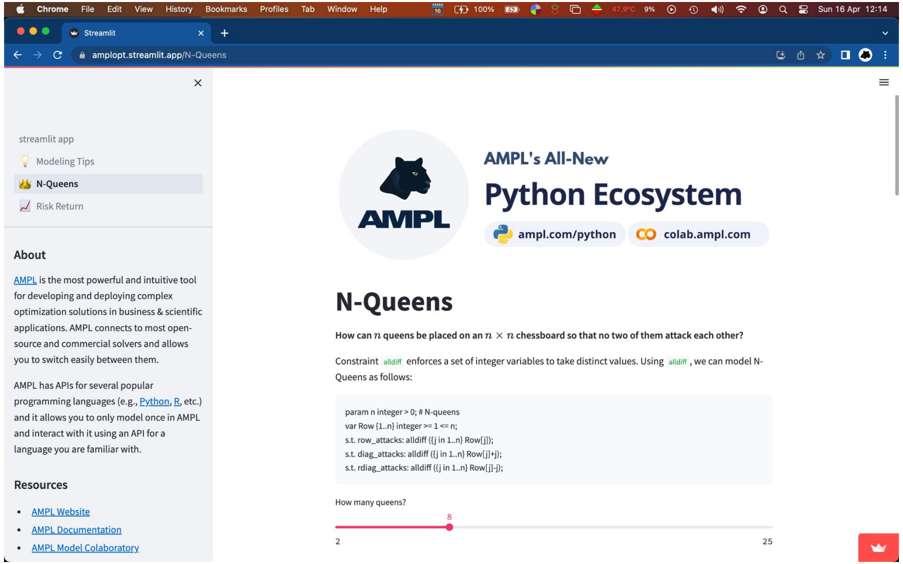


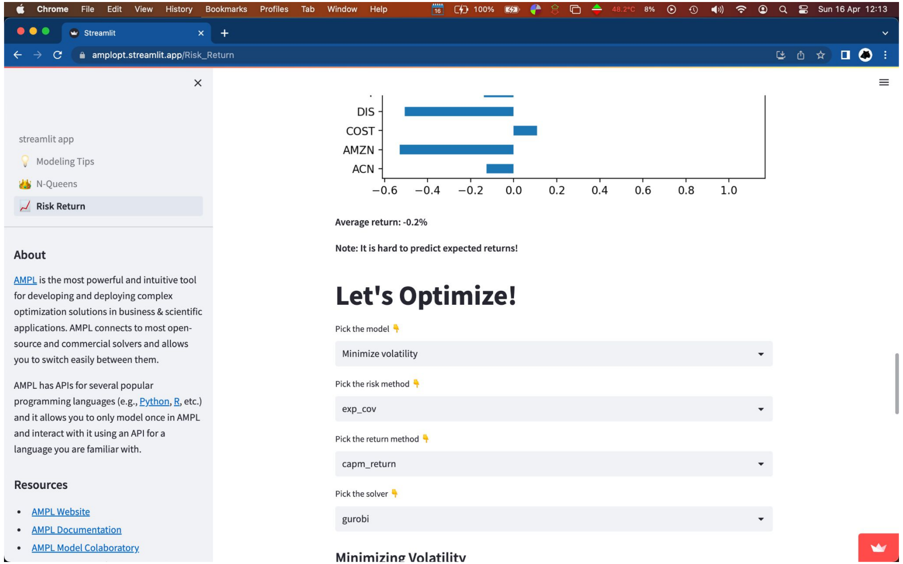


Deploying optimization applications quickly and easily using AMPL with Python

AMPL on Streamlit

> https://ampl.com/streamlit (live demo)





Deploy anywhere with Docker

> https://dev.ampl.com/ampl/docker/

AMPL can be easily used on Docker containers and deployed anywhere.

```
# Use any image as base image with python installed
FROM python:3.9-slim-bullseye

# Install amplpy and all necessary amplpy.modules:
RUN python -m pip install amplpy --no-cache-dir # Install amplpy
RUN python -m amplpy.modules install highs gurobi --no-cache-dir # Install modules
```

Example project showing how to deploy applications

> https://amplpyfinance.ampl.com/

- How to use AMPL with Docker Containers:
  - A basic Docker Compose template for orchestrating a **Flask** application & a **Celery** queue with **Redis**.
  - https://github.com/ampl/amplpyfinance/tree/master/deployment/docker

- The same Docker images can be deployed to **Kubernetes Clusters**

- How to use AMPL in Continuous Integration Systems
  - This project uses **Azure Pipelines** and **GitHub Actions** for **CI/CD**
  - https://dev.ampl.com/ampl/cicd/

Continuous Integration Systems

- How to use AMPL in Continuous Integration Systems
  - This project uses **Azure Pipelines** and **GitHub Actions** for **CI/CD**
  - https://dev.ampl.com/ampl/cicd/

```yaml
jobs:
  Test:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: ["3.10"]

    steps:
      - uses: actions/checkout@v3
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v4
        with:
          python-version: ${{ matrix.python-version }}
      - name: Install dependencies
        run: |
          set -ex
          python -m pip install -r requirements.txt
          python -m pip install amplpy
          python -m amplpy.modules install <solver1> <solver2>
          python -m amplpy.activate <license-uuid>
      - name: Install package
        run: |
          python -m pip install .
      - name: Test package
        run: |
          python -m <package-name>.tests
```

What about licenses for AMPL and Commercial Solvers?

Dynamic Licensing System

Free licenses to use on Google Colab (and locally!)

- **ampl.com/ce**
  - For personal use
  - **Immediate access without approvals required!**
  - No size-limits
  - Includes access to:
    - Open-source solvers
    - Commercial solver trials
- **ampl.com/courses**
  - For teaching
  - No size-limits
  - **Full access to all solvers!**
  - All students can use the license during the course.