B

付録B

# amplpy API の書き方リファレンス

Python から AMPL を動かす API の典型的な書き方をまとめる

# この章の内容

1. インストールと準備
2. モデルの読み込み
3. データの読み込み
4. 求解
5. 結果の取り出し
6. データを変えて解き直す
7. よく使う書き方

B.1

# インストールと準備

amplpy 本体・ソルバーモジュール・ライセンスの 3 つをそろえる

# インストール

```bash
# AMPL の Python API をインストールする
python -m pip install amplpy --upgrade

# ソルバーモジュールをインストールする
# （必要なものだけ入れればよい）
python -m amplpy.modules install highs gurobi xpress cplex

# ライセンスを有効化する
# （無償ライセンスは ampl.com/ce または ampl.com/courses）
python -m amplpy.modules activate <your-license-uuid>
```

# AMPL オブジェクトを作る

solve.py

```
from amplpy import AMPL
import pandas as pd

ampl = AMPL()

# amplpy.modules を使わず、AMPL がパスに
# 入っていない場合
from amplpy import AMPL, add_to_path
add_to_path(r"full path to the AMPL installation directory")
ampl = AMPL()
```

# B.2 モデルの読み込み

標準は ampl.read()。ampl.eval() は手早い試作向け

# ファイルから、またはインラインで

solve.py

```
# ファイルから（推奨）
ampl.read("model.mod")

# インラインで（手早い試作になら可）
ampl.eval(r"""
    set CITIES;
    param distance {CITIES, CITIES} >= 0;
    var X {CITIES} binary;
    minimize Total:
        sum {i in CITIES, j in CITIES}
            distance[i,j] * X[i] * X[j];
""")
```

- 標準的なやり方は ampl.read()
- ampl.eval() は短いインライン定義向け
  - ノートブック環境など ファイルが不便なとき

# B.3 データの読み込み

pandas の DataFrame からも、Python の辞書・リストからも渡せる

# DataFrame から読み込む

solve.py

```
# 書き方 1: ampl.set_data(df, "SET_NAME")
food_df = pd.DataFrame([
    ("BEEF", 3.19, 2, 10),
    ("CHK",  2.59, 2, 10),
], columns=["FOOD", "cost", "f_min", "f_max"]).set_index("FOOD")
ampl.set_data(food_df, "FOOD")

# 書き方 2: ampl.get_parameter("p").set_values(df)
import numpy as np
amt_df = pd.DataFrame(
    np.array([[60, 8], [20, 0]]),
    columns=["BEEF", "CHK"],       # FOOD の添字
    index=["A", "C"],              # NUTR の添字
)
ampl.get_parameter("amt").set_values(amt_df)
```

# Python の辞書・リストから読み込む

solve.py

```
# 集合
ampl.set["FOOD"] = ["BEEF", "CHK", "FISH"]

# 1 つの集合で添字付けされたパラメータ
ampl.param["cost"] = {"BEEF": 3.19, "CHK": 2.59, "FISH": 2.29}

# 複数の集合で添字付けされたパラメータ
ampl.param["amt"] = {
    ("A", "BEEF"): 60,
    ("A", "CHK"):  8,
    ("C", "BEEF"): 20,
    ("C", "CHK"):  0,
}

# 特定のパラメータ値だけを更新する
cost = ampl.get_parameter("cost")
cost.set_values({"BEEF": 5.01, "FISH": 4.55})
```

# 性能に関する注意

amplpy は数百万行の DataFrame も効率よく扱える。1000 万行を超えるならテーブルハンドラを検討

### 1 amplcsv

CSV ファイルを Python を経由せずに直接読み込む

### 2 amplxl

Excel の XLSX ファイルを直接読み込む

### 3 eodbc

ODBC 経由でデータベース（MySQL、PostgreSQL など）から読み込む

# B.4 求解

ソルバーを指定して解き、求解の状態をつねに確認する

# ソルバーの指定と求解

solve.py

```
# ソルバーを指定する
ampl.option["solver"] = "highs"

# ソルバー固有のオプションを指定する
ampl.option["highs_options"] = "outlev=1 timelimit=300"

# ソルバーオプションの一般形:
# ampl.option[f"{solver}_options"] = "option_string"

# 解く
ampl.solve()

# 求解の状態をつねに確認する
assert ampl.solve_result == "solved"
# 他の値: "infeasible", "unbounded", "limit" など
```

# ソルバーオプションの例

ampl.option["<ソルバー名>_options"] に文字列で渡す

| ソルバー | オプション名 | 設定例 |
|---|---|---|
| HiGHS | highs_options | outlev=1 timelimit=300 |
| Gurobi | gurobi_options | outlev=1 timelimit=300 mipgap=0.01 threads=8 |
| CPLEX | cplex_options | outlev=1 timelimit=300 mipgap=0.01 |
| COPT | copt_options | outlev=1 timelimit=300 |

# B.5 結果の取り出し

目的関数値・変数・制約・任意の式を取り出す

# 結果を取り出す

solve.py

```
# 目的関数値
total_cost = ampl.get_objective("Total_Cost")
print(total_cost.value())

# 変数の値を pandas の DataFrame として
buy = ampl.get_variable("Buy")
df = buy.get_values().to_pandas()

# 制約に関する情報
diet = ampl.get_constraint("Diet")
df = diet.get_values().to_pandas()

# 任意の AMPL の式を評価する
df = ampl.get_data(
    "{j in FOOD} 100*Buy[j]/Buy[j].ub").to_pandas()
```

- get_objective で目的関数値
- get_variable で変数の値
- get_constraint で制約（双対変数）
- get_data で任意の式を DataFrame に

B.6

# データを変えて解き直す

モデルを読み直さずに、パラメータだけを更新できる

# パラメータを更新して解き直す

solve.py

```
# 特定のパラメータ値を更新する
ampl.get_parameter("cost").set_values({"BEEF": 5.01, "HAM": 4.55})

# 短く書くならこちら
ampl.param["cost"] = {"BEEF": 5.01, "HAM": 4.55}

# 解き直す
ampl.solve()
assert ampl.solve_result == "solved"
print(f"New objective: {ampl.get_objective('Total_Cost').value()}")
```

B.7

# よく使う書き方

パラメータを変えながら繰り返し解く

# 感度分析のループと複数のシナリオ

solve.py

```
# 感度分析のループ
results = {}
for budget in [100, 200, 300, 400, 500]:
    ampl.param["max_budget"] = budget
    ampl.solve()
    if ampl.solve_result == "solved":
        results[budget] = ampl.get_objective("Profit").value()

results_df = pd.DataFrame(list(results.items()),
                          columns=["Budget", "Profit"])

# 複数のシナリオ
for scenario_name, scenario_data in scenarios.items():
    ampl.get_parameter("demand").set_values(scenario_data)
    ampl.solve()
    if ampl.solve_result == "solved":
        cost = ampl.get_objective('Cost').value()
        print(f"{scenario_name}: {cost:.2f}")
```

# まとめ

- ✓ モデルは ampl.read() で読み込む
- ✓ データは set_data・set_values・辞書で渡す
- ✓ solve 後は solve_result を必ず確認する
- ✓ 結果は get_* で DataFrame として取り出す
- ✓ パラメータを更新して何度でも解き直せる