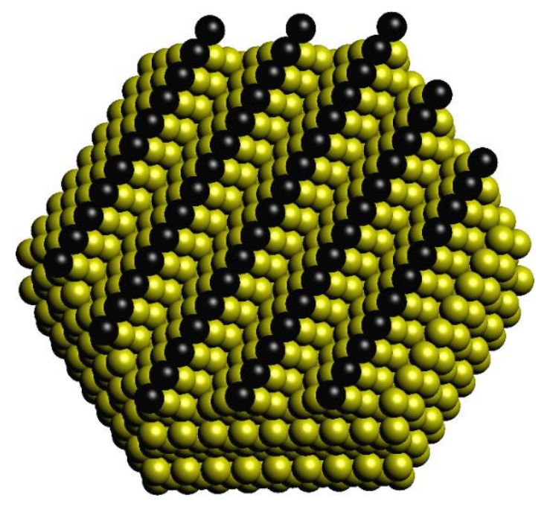
A Unified Approach and Computational Toolkit for Nanoscale Materials Design

Christopher L. Hanselman[a], David C. Miller[b], Chrysanthos E. Gounaris[a]
[a]Carnegie Mellon University, [b]National Energy Technology Laboratory

Introduction

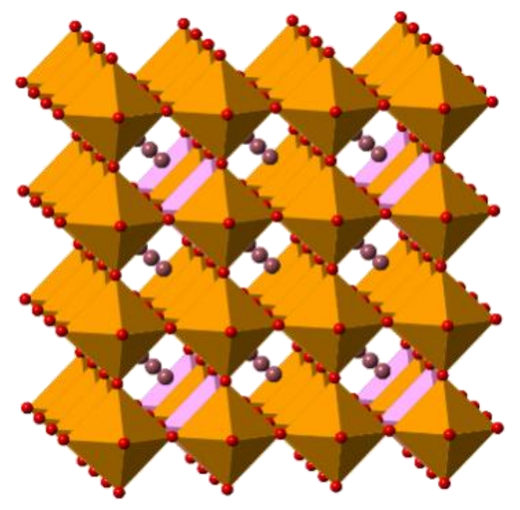
Optimization-Based Design Paradigm

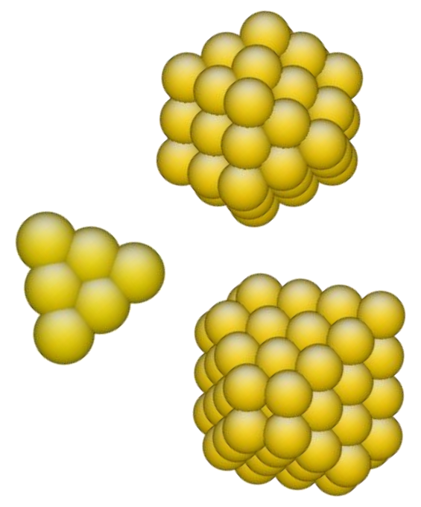
- Lots of work on ***how*** to fabricate complex structures, but no systematic approach as to ***which*** structures to fabricate
- Proposed paradigm: Design bottom-up, arranging matter via **mathematical optimization**
- Mathematical optimization provides a **rigorous, systematic** way to explore the entire design space


Adapted from [2]

Atomic-scale Material Optimization

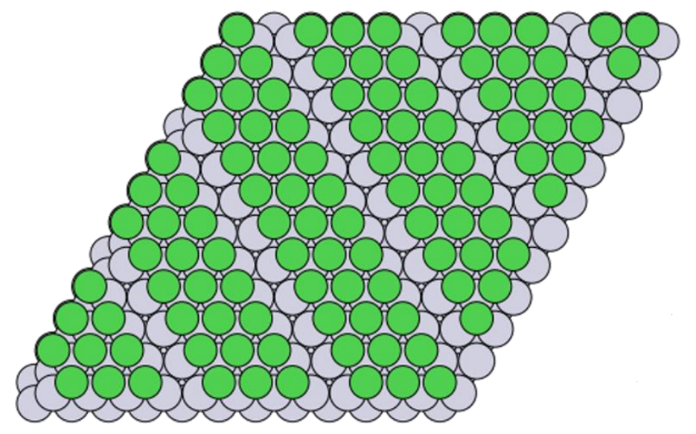
- Nanostructured surface defects [1,2]
- Dopant placement patterns [3]
- Nanocluster cohesive energy [4]
- Adaptive learning for microstructure optimization [5]


Adapted from [3]


Adapted from [4]

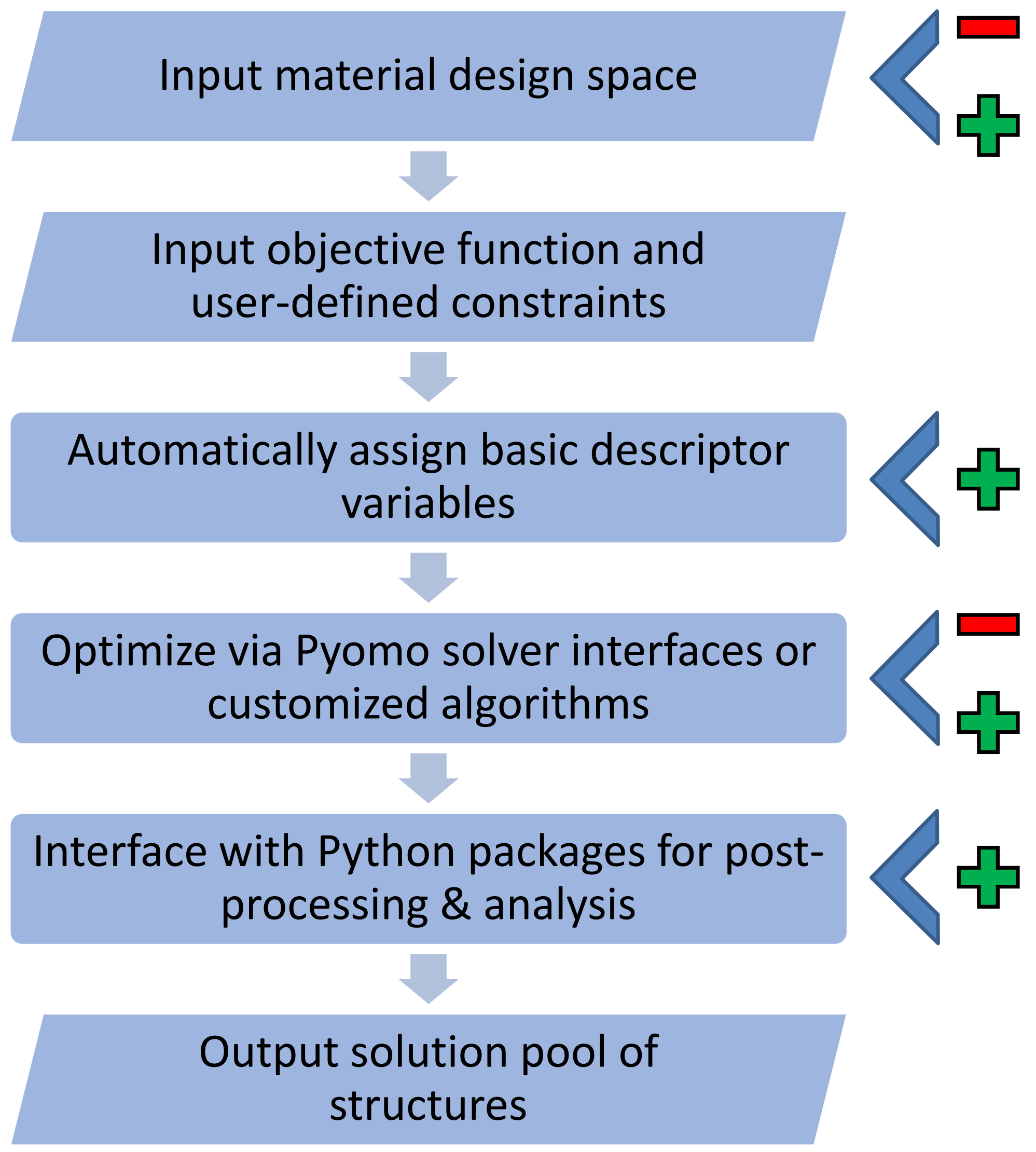
Common Modeling Components

- Building blocks (e.g., atoms)
- Neighbor interactions (e.g., bonds)
- Neighbor counts (e.g., coordination number)
- Conformations (e.g., edge sites, facet sites)


Adapted from [5]

Key Question:
How to automatically instantiate and optimize atomic-scale material design problems?

Proposed Approach


Input material design space
Input objective function and user-defined constraints
Automatically assign basic descriptor variables
Optimize via Pyomo solver interfaces or customized algorithms
Interface with Python packages for post-processing & analysis
Output solution pool of structures

Challenges & Opportunities:

− Potentially hard to specify
+ Fully flexible modeling object hierarchy provided

+ Simplifies user input & avoids modelling errors

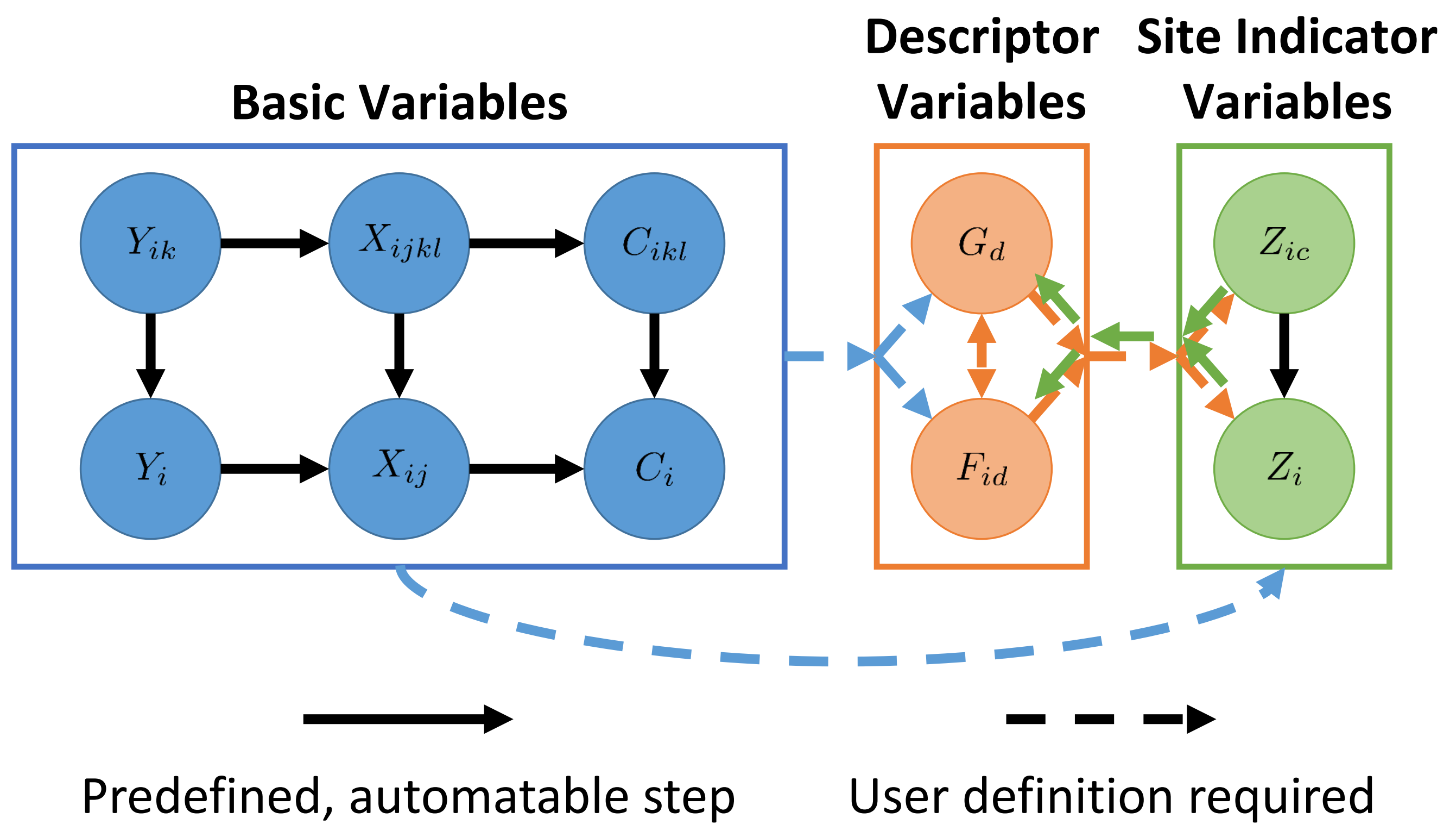
− Naïve formulations are often difficult to solve
+ Customized approaches are easy to test and develop

+ Easy integration into existing analysis/scripts

General Model Formulations

Basic Modeling Automation

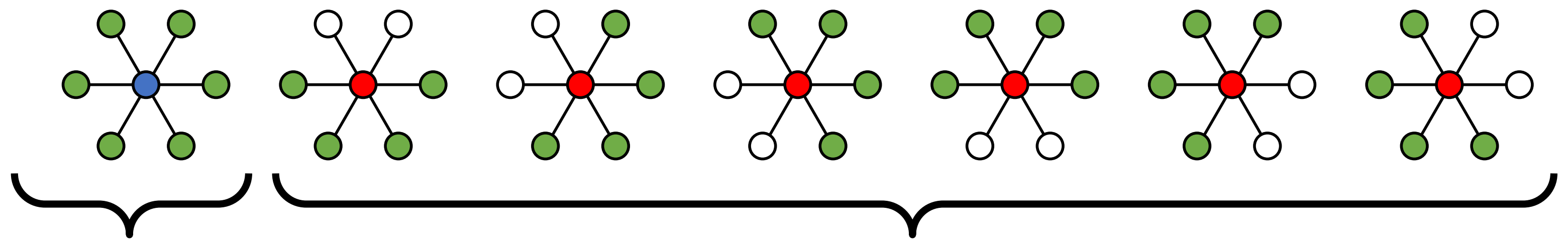
- Common modelling components can be instantiated automatically as needed
- Simplifies user code and eliminates potential errors
- Predefined options allow rapid model development and testing (e.g., symmetry breaking, alternative formulations)



Example: Bifunctional Surface Model

Model Bifunctional Catalyst: Ammonia Decomposition [5]

- Ni on Pt “patchy” catalyst, less than one monolayer
- Assume simple model for reactivity:
  - One “unit” of reactivity for each edge site next to a facet site


Facet Sites
NH_3 Dissociation*

Edge Sites
N Association*

User Defined:
Max. sum of reactive sites
Definition of reactive site

$$
\begin{aligned}
\max_{Y_i, Z_{ic}, \phi_{ij}} \quad & \sum_{i \in I} \sum_{j \in N_i} \phi_{ij} \\
\text{s.t.} \quad & \phi_{ij} \le \sum_{c \in C_A} Z_{ic} && \forall i \in I \; \forall j \in N_i \\
& \phi_{ij} \le \sum_{c \in C_B} Z_{jc} && \forall i \in I \; \forall j \in N_i
\end{aligned}
$$

Automatically Generated:
Indication of conformations
Variable design space

$$
\begin{aligned}
& \sum_{c \in M_{ij}^{+}} Z_{ic} \le Y_j && \forall i \in I \; \forall j \in N_i \\
& \sum_{c \in M_{ij}^{-}} Z_{ic} \le 1 - Y_j && \forall i \in I \; \forall j \in N_i \\
& Z_{ic} \ge 1 - \sum_{j \in N_i^{+}} (1 - Y_j) - \sum_{j \in N_i^{-}} (Y_j) && \forall i \in I \; \forall c \in C \\
& \phi_{ij} \in \{0,1\} && \forall i \in I \; \forall j \in N_i \\
& Z_{ic} \in \{0,1\} && \forall i \in I \; \forall c \in C \\
& Y_i \in \{0,1\} && \forall i \in I
\end{aligned}
$$

Example Jupyter Notebook & Results

Common Features:

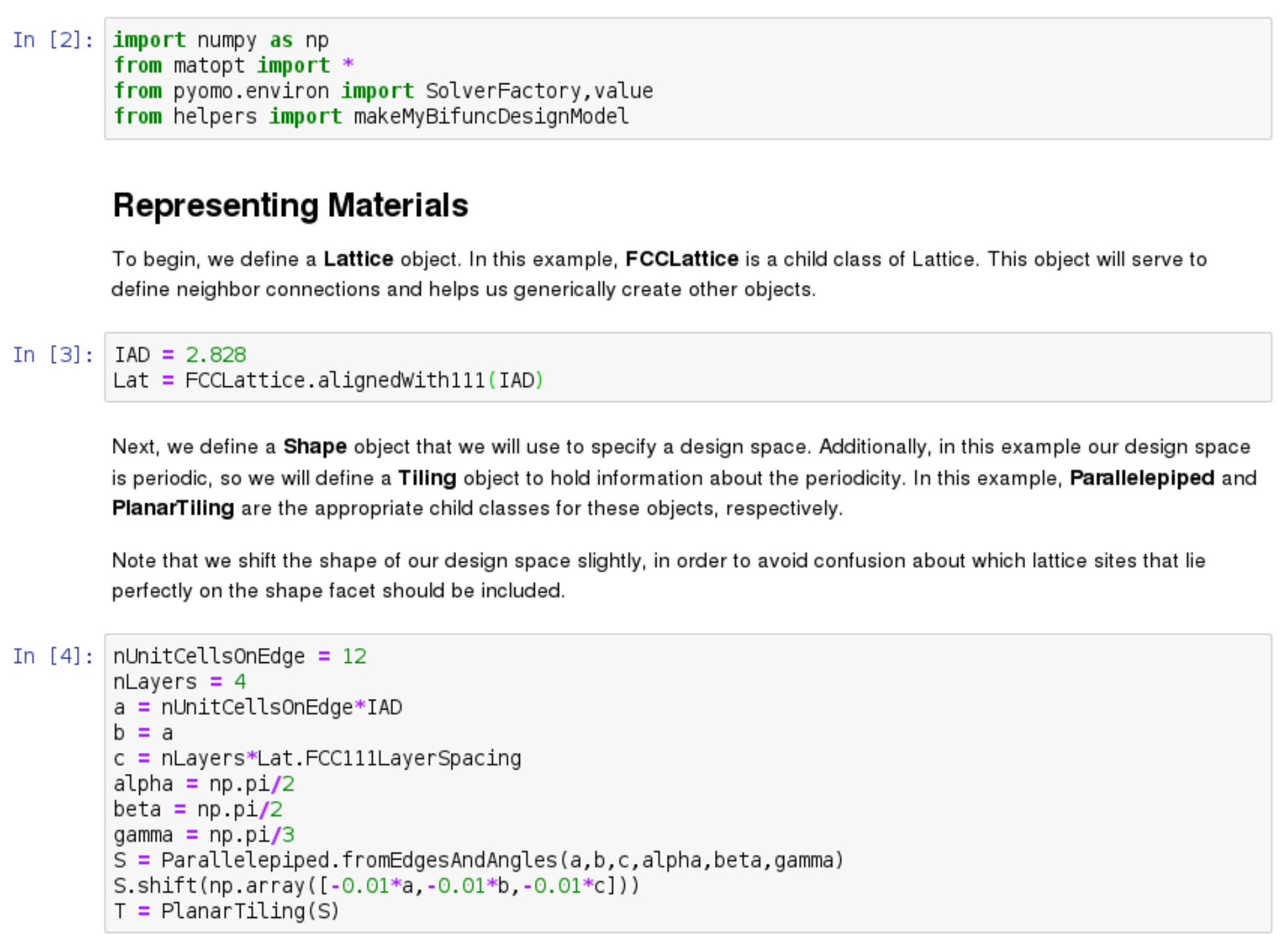
- Creating Lattice, Shape, Tiling, Canvas objects
- Using common functions to define optimization models
- Creating Design from solver results

Unique Features:

- Creating conformations programmatically
- Modifying Canvas neighborhood information
- User-defined descriptor variables and objective function

Observations:

- Simple model assumptions yields design identified in [5]
- Relatively large, shallow tile design spaces are tractable


In [2]:
```
import numpy as np
from matopt import *
from pyomo.environ import SolverFactory,value
from helpers import makeMyBifuncDesignModel
```

Representing Materials

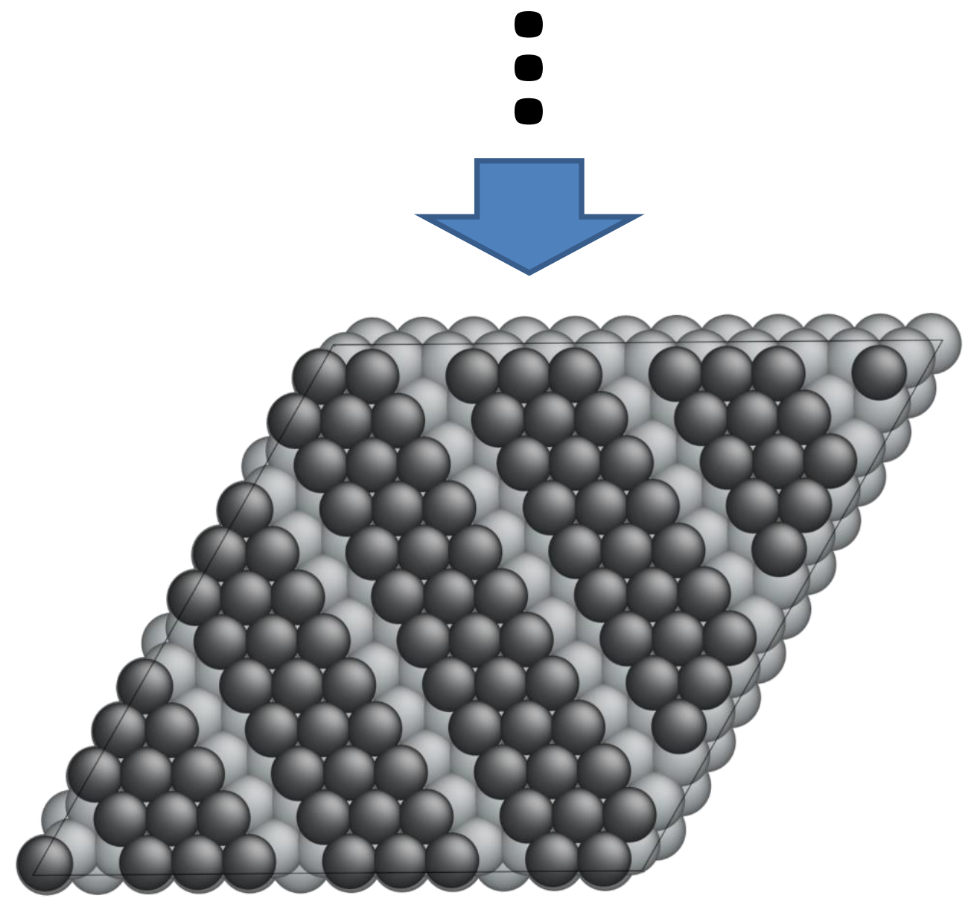
To begin, we define a **Lattice** object. In this example, **FCCLattice** is a child class of Lattice. This object will serve to define neighbor connections and helps us generically create other objects.

In [3]:
```
IAD = 2.828
Lat = FCCLattice.alignedWith111(IAD)
```

Next, we define a **Shape** object that we will use to specify a design space. Additionally, in this example our design space is periodic, so we will define a **Tiling** object to hold information about the periodicity. In this example, **Parallelepiped** and **PlanarTiling** are the appropriate child classes for these objects, respectively.

Note that we shift the shape of our design space slightly, in order to avoid confusion about which lattice sites that lie perfectly on the shape facet should be included.

In [4]:
```
nUnitCellsOnEdge = 12
nLayers = 4
a = nUnitCellsOnEdge*IAD
b = a
c = nLayers*Lat.FCC111LayerSpacing
alpha = np.pi/2
beta = np.pi/2
gamma = np.pi/3
S = Parallelepiped.fromEdgesAndAngles(a,b,c,alpha,beta,gamma)
S.shift(np.array([-0.01*a,-0.01*b,-0.01*c]))
T = PlanarTiling(S)
```



Future Steps:

- Develop simplified interfaces for common material systems
- Finish make documentation, automatic test, and examples

Conclusions

➢ An open-source toolkit for modeling and optimizing atomic-scale nanostructured materials was proposed
➢ Toolkit interface takes advantage of common modeling elements, simplifying user interaction and allowing rapid model development
➢ Developed framework can quickly be deployed to model new examples of bifunctional catalysts from recent literature.

References

[1] Calle-Vallejo, F., et al. (2015). Science, 350(6257), 185–189.
[2] Hanselman, C.L., Gounaris, C.E. (2016). AIChE Journal, 62(9), 3250-3263.
[3] Hanselman, C.L., et al. (2019) Computers & Chemical Engineering, 126, 168-177.
[4] Isenberg, N.M., et al. Manuscript in preparation.
[5] Núñez, M., Vlachos, D. G. (2019). Industrial & Engineering Chemistry Research, 58, 6146-6154.

Contact

Christopher L. Hanselman	chanselm@cmu.edu
David C. Miller	David.miller@netl.doe.gov
Chrysanthos E. Gounaris	gouaris@cmu.edu

Disclaimer This presentation was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof