
hydrosolver

Dmytro Strelnikov

Sep 21, 2021

CONTENTS:

1	Working with compositions	1
1.1	Defining a composition	1
1.2	Loading and dumping compositions	2
1.3	Operations on compositions	2
2	Working with solutions	3
2.1	Defining a solution	3

WORKING WITH COMPOSITIONS

The simplest entity in hydrosolver is `Composition`. Compositions can be defined on the go or loaded from a file, added and scaled.

1.1 Defining a composition

The most straightforward way to define a composition is using its constructor `Composition(name, vector)`. The simplest composition which does not contain any of the nutrient elements of our interest would be `Composition(name='Pure water')`.

The monopotassium phosphate can be defined as follows:

```
>>> from hydrosolver.composition import Composition
>>> MKP = Composition(
...     name='Monopotassium phosphate',
...     vector=[0, 0, 0.2276, 0.2873, 0, 0, 0, 0, 0, 0, 0, 0, 0],
... )
>>> MKP
Composition: Monopotassium phosphate
```

Nutrient	Ratio	Amount mg/kg
P	0.2276	227600
K	0.2873	287300

Here `vector` follows the structure of `composition.nutrients_stencil`. Let us check the result.

It is hard to not notice that this kind of definition is cumbersome and can be barely used by humans. Therefore class `Composition` contains an alternative constructor `from_dict`, so the same result could be achieved in the following way:

```
>>> MKP = Composition.from_dict(
...     {'Monopotassium phosphate': {'P': 0.2276, 'K': 0.2873}}
... )
>>> MKP
Composition: Monopotassium phosphate
```

Nutrient	Ratio	Amount mg/kg
P	0.2276	227600
K	0.2873	287300

1.2 Loading and dumping compositions

It makes sense to save frequently used composition into a database and further load it from there. Here is an example:

```
import yaml

with open('database.yaml', 'w') as database:
    database.write(yaml.dump(MKP.as_dict()))
```

Multiple compositions can be loaded at once from a file:

```
from hydrosolver.utils import load_file

compositions = load_file('compositions/pure.yaml')
```

1.3 Operations on compositions

Compositions can be added and scaled, i.e. multiplied by scalars. You will typically not need to add or subtract compositions, but consider the following use case for scaling:

```
>>> KOH = Composition.from_dict(
...     {'Potassium hydroxide': {'K': 0.69687}}
...     )
>>> KOH_94 = 0.94 * KOH
>>> KOH_94
Composition: 0.94 * (Potassium hydroxide)
```

Nutrient	Ratio	Amount mg/kg
K	0.655058	655058

WORKING WITH SOLUTIONS

A more advanced entity in hydrosolver is `Solution`. Solutions consist of a few compositions and can be constructed in different ways. Solutions added and scaled, extended and merged.

2.1 Defining a solution

To define a solution we must first define the compositions constituting it. Let us consider a simple example:

```
from hydrosolver.composition import Composition
from hydrosolver.solution import Solution

water = Composition('Pure water')
CN = Composition.from_dict(
    {'Calcium nitrate tetrahydrate': {'N (NO3-)': 0.1186, 'Ca': 0.1697}}
)

solution_CN_10 = Solution(
    [CN, water],
    [0.1, 0.9],
)
```

Here we just defined a 10% (by mass) aqueous solution of calcium nitrate tetrahydrate.