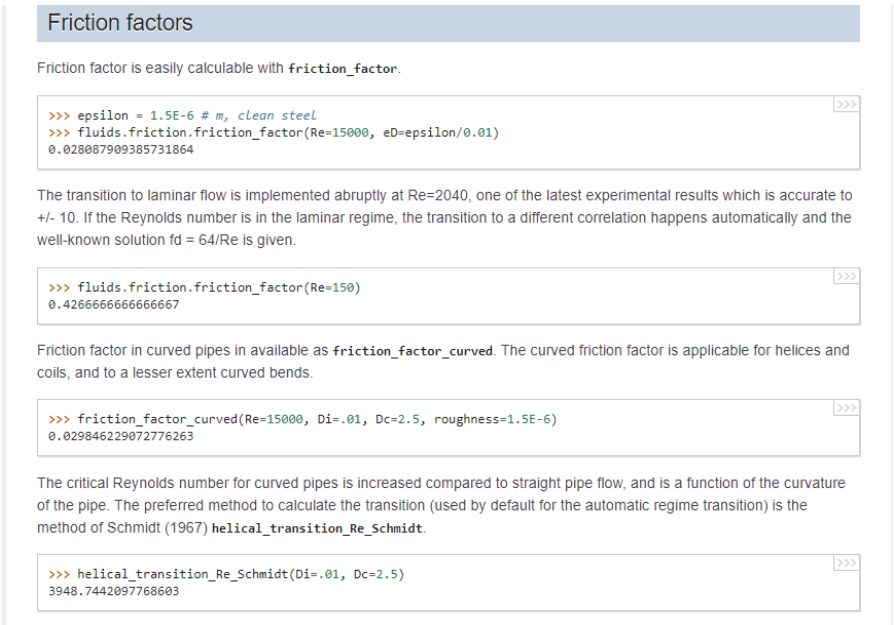
PYTHON IN CHEMICAL ENGINEERING

An article by Vicinisvarri Inderan

In current years, analysing a vast number of scientific information becomes highly promising for chemical informatics professionals. The limitation of common platform for informatics integration causes extensive imitation of work and impedes the chemistry related researches, specifically biopharmaceutical research.

Recently, Python is introduced as a common language platform for integrate in computational chemistry. It is an open-source scripting language that mainly support the programming interfaces with both open source and copy-righted. Dealing with a huge number of data than ever before, Python plays an important role in machine learning and data science.

Today, Python is the most versatile coding language not only for pharmaceutical field but also for chemical engineering applications. In chemical engineering it is mainly used for design a process, simulation and perform automated calculation using the data, and later used to obtain some variation. With over 185 000 packages freely available, there is almost library for each topic. For instance, fluid library for fluid dynamics, thermo for thermodynamics and Ht library for heat transfer. Some of the other packages that highly recommended for chemical engineers are Pint, uncentainties and SymPy.



Friction factors

Friction factor is easily calculable with **friction_factor**.

```
>>> epsilon = 1.5E-6 # m, clean steel
>>> fluids.friction.friction_factor(Re=15000, eD=epsilon/0.01)
0.028087909385731864
```

The transition to laminar flow is implemented abruptly at Re=2040, one of the latest experimental results which is accurate to +/- 10. If the Reynolds number is in the laminar regime, the transition to a different correlation happens automatically and the well-known solution fd = 64/Re is given.

```
>>> fluids.friction.friction_factor(Re=150)
0.4266666666666667
```

Friction factor in curved pipes in available as **friction_factor_curved**. The curved friction factor is applicable for helices and coils, and to a lesser extent curved bends.

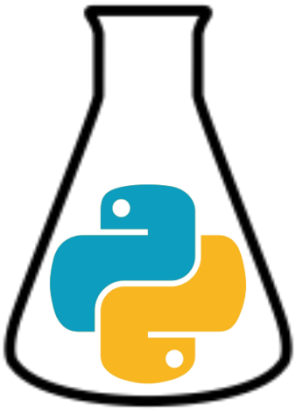
```
>>> friction_factor_curved(Re=15000, Di=.01, Dc=2.5, roughness=1.5E-6)
0.029846229072776263
```

The critical Reynolds number for curved pipes is increased compared to straight pipe flow, and is a function of the curvature of the pipe. The preferred method to calculate the transition (used by default for the automatic regime transition) is the method of Schmidt (1967) **helical_transition_Re_Schmidt**.

```
>>> helical_transition_Re_Schmidt(Di=.01, Dc=2.5)
3948.7442097768603
```

In comparison with other coding language Python has many advantages. Firstly, Python clean syntax is easy to learn and maintain. It also supports all common integration strategies from one language such as CORBA, COM, XML/RPC, SOAP, Java and HTTP. Besides, Python provide excellent cross- platform efficiency. Most importantly Python application programming interfaces (APIs) are much more flexible than other programming interface such as VBA and require less end user effort to access.

Therefore, learning phyton very useful in chemical engineering field. However, it is not compulsory for a chemical engineer to know the Python coding, but it is certainly an advantage, as the application of this language has skyrocketed in almost all the science fields.



Reference: 1. http://www.chempython.org/

2. https://fluids.readthedocs.io/tutorial.html#friction-factors