## 5 - Métodos computacionais

Os métodos utilizados foram a linguagem de programação python, utilizando o pacote de funções de análise química cantera e o mecanismo químico de gases GRI mech 3.0, que é uma base de dados com informações experimentais de propriedades de elementos químicos em forma gasosa.

## 6 - Valores de entrada do problema

As propriedades dos reagentes dados pelos problema eram as de composição química por C2H6 (etano) misturado com AR atmosférico padrão na condição estequiométrica em forma gasosa, temperatura inicial igual a 1000 K e pressão inicial igual a 0,2 atm, as propriedades do reator eram a de raio igual a 5cm e comprimento igual a 10cm.

Com os valores de entrada pode-se calcular a área da seção transversal do reator e utilizando o GRI mech 3.0, obtém-se a relação molar estequiométrica e densidade da mistura, para então calcular-se a taxa de fluxo mássico, que se conserva na entrada e saída do reator. O valor da velocidade inicial, que o que precisava ser encontrado no problema foi arredondado por tentativa e erro nas primeiras execuções do algoritmo e 4.18mm/s foi avaliado como um resultado aceitavel.

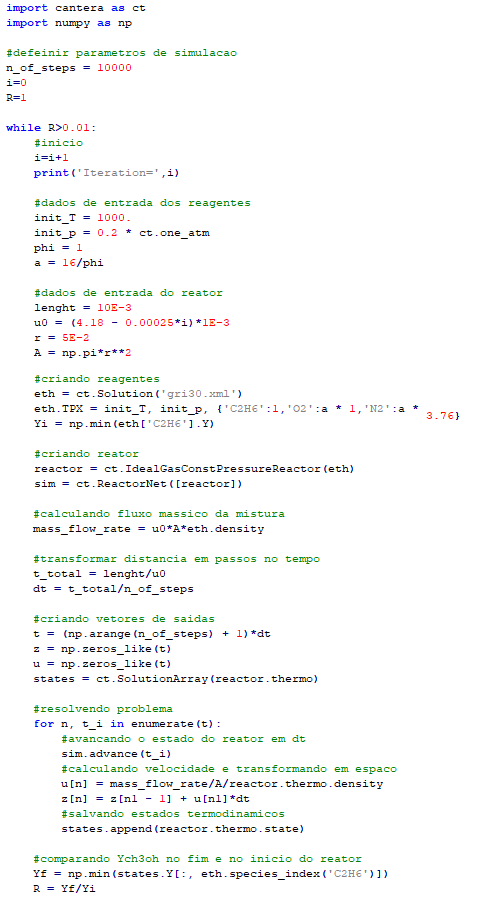
## 7 - Implementação computacional

Uma das funções do cantera utilizadas para solucionar o problema são de criação de modelo matemático de reator chamado de *ct.IdealGasReactor()* que modela um objeto que representa os estados das propriedades de um reator zero-dimensional, onde o volume do reator muda em função do tempo, para assim manter a pressão constante.

O equacionamento dessa função é baseada em uma partícula lagrangiana com uma velocidade constante no tempo, conforme o tempo avança, a partícula muda de posição, e de estados termoquímicos, essas propriedades então são armazenadas em vetores para representar as variáveis de saída desejadas.A função *ct.ReactorNet()* serve para criar uma cadeia de reatores a serem integrados no tempo, e a função *advance* avança o estado do reator de zero até o instante de tempo desejado.

Após isso, calcula-se a velocidade da partícula no instante de tempo e transforma-se a coordenada de tempo em uma coordenada de espaço. a função *states.append()* armazena os dados termodinâmicos do estado no instante em um objeto, para se fazer uma análise posterior e gerar os gráficos.

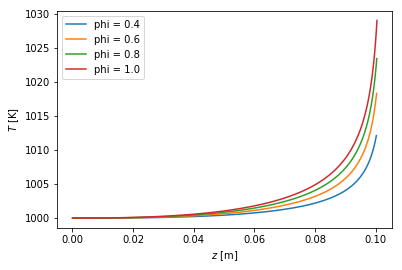
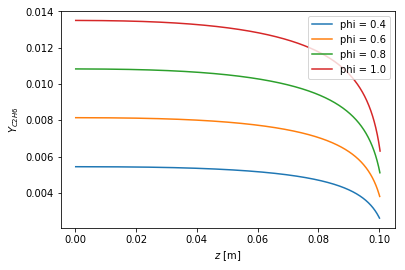
A Primeira parte do problema pedia para encontrar a velocidade inicial do fluido para se ter 99% de reação completa, para fazer isso, foi desenvolvido um laco de repeticao *while*, que a cada iteração ajustava a velocidade inicial entrada, executava todo o método de resolução do reator e média o total de Y{C2H6} na saída do comparava com o seu valor de entrada. quando a fração mássica da saída representasse 1% ou menos da fração de entrada, o algoritmo era interrompido.



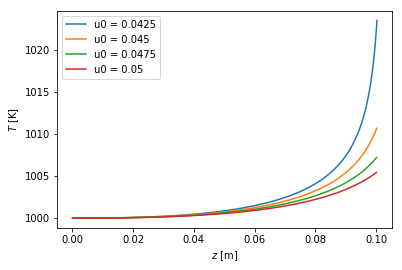
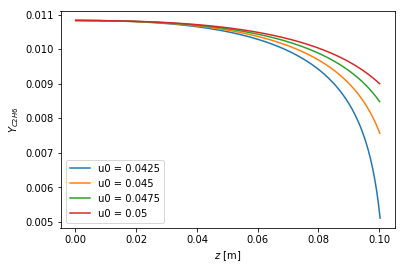
## 8 - Resultados e discussões

O resultado de velocidade inicial aproximado para 99% de reação foi de u0=0.041735m/s, que foi obtido com a execução do laço while após 26 iterações. Com o auxílio do pacote de gráficos *matplotlib* o algoritmo também pode gerar gráficos de temperatura e fração mássica de C2H6 ao longo do comprimento do reator, então, esses dois parâmetros foram avaliados para diversos valores de pressão, velocidade inicial e phi.

Observa-se que com o aumento da relação ar-combustível, a quantidade de C2H6, que permanece sem grandes variações no inicio até a metade do reator, decai a uma taxa de variação maior no final do reator, o que influi em uma maior variação de temperatura da reação.



Com a diminuição da velocidade inicial, tem-se o resultado esperado de que o total de YC2H6 e menor no final do reator, logo, com a reação quase completa, observa que a temperatura final e aumenta consideravelmente em relação aos outros estados.



Mesmo com um pequeno aumento da pressão dos reagentes (na faixa de 0,1atm), observa-se uma grande aumento da taxa de variação de consumo da fração mássica de C2H6 ao longo do reator, o que influi no parâmetro estudado mais eficiente para acelerar a reação.