



PROJET INTÉGRATEUR III

AER3900 - HIVER 2022

Guide d'utilisation des scripts calculant le taux de déformation au point d'extinction de flammes.

Par

Rani Naaman

Rani.naaman@polymtl.ca

29 mars 2022

Table des matières

1	Fonctionnement des scripts	1
1.1	Lecture du fichier <i>data.json</i>	1
1.2	Établissement des simulations en parallèle	3
1.3	Pre-simulation : Établissement d'une flamme libre	3
1.4	Simulation : Trouver le point de convergence et calculer le taux de défor- mation au point d'extinction	5

Liste des figures

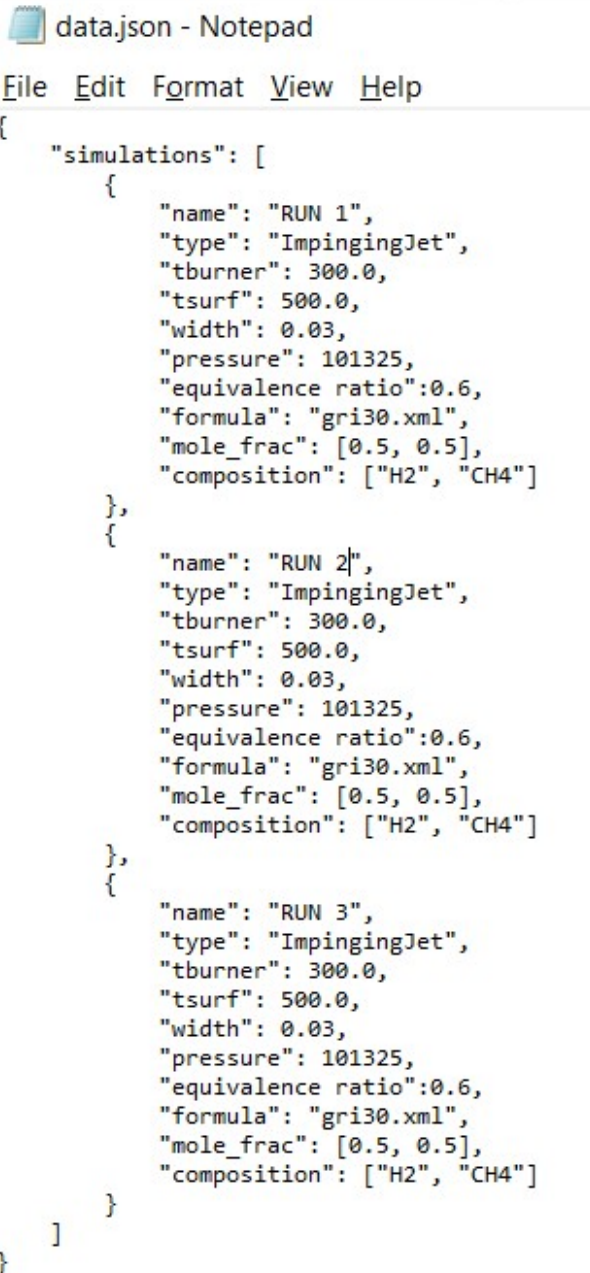
1	Exemple de plusieurs simulation dans <i>data.json</i>	1
2	Exemple de bar de progression du fichier <i>main_PI3.py</i>	3

1 Fonctionnement des scripts

Afin de calculer le taux de déformation au point d'extinction de flammes, les scripts fonctionnent de la manière suivante.

1.1 Lecture du fichier *data.json*

Dans le fichier zip se retrouve le premier fichier important, soit *data.json*, représenté dans l'illustration ci-dessous.



```
{
  "simulations": [
    {
      "name": "RUN 1",
      "type": "ImpingingJet",
      "tburner": 300.0,
      "tsurf": 500.0,
      "width": 0.03,
      "pressure": 101325,
      "equivalence ratio": 0.6,
      "formula": "gri30.xml",
      "mole_frac": [0.5, 0.5],
      "composition": ["H2", "CH4"]
    },
    {
      "name": "RUN 2",
      "type": "ImpingingJet",
      "tburner": 300.0,
      "tsurf": 500.0,
      "width": 0.03,
      "pressure": 101325,
      "equivalence ratio": 0.6,
      "formula": "gri30.xml",
      "mole_frac": [0.5, 0.5],
      "composition": ["H2", "CH4"]
    },
    {
      "name": "RUN 3",
      "type": "ImpingingJet",
      "tburner": 300.0,
      "tsurf": 500.0,
      "width": 0.03,
      "pressure": 101325,
      "equivalence ratio": 0.6,
      "formula": "gri30.xml",
      "mole_frac": [0.5, 0.5],
      "composition": ["H2", "CH4"]
    }
  ]
}
```

FIGURE 1 – Exemple de plusieurs simulation dans *data.json*.

Afin de commencer une simulation, l'utilisateur doit réaliser les étapes suivantes :

1. Définir le nom de la simulation dans la section "name".
2. Définir le type de la simulation dans "type". Pour l'instant, les scripts fonctionnent seulement pour les simulations «ImpingingJet». Soit une flamme qui se repend vers une surface non réactive.
3. Définir la température du brûleur (K) dans la section "tburner". À titre d'exemple, une plage entre [300 ; 1500] K serait attendue pour cette variable.
4. Définir la température de la surface non réactive (K) dans la section "tsurf". À titre d'exemple, une plage entre [300 ; 1500] K serait attendue pour cette variable.
5. Définir la distance entre le brûleur et la paroi (m) dans la section "width". À titre d'exemple, une plage entre [0.01 ; 0.1] m serait attendue pour cette variable.
6. Définir la pression de la simulation (pa) dans la section "pressure". À titre d'exemple, une plage entre [1 ; 10] atm (1 atm= 101325 pa) serait attendue pour cette variable.
7. Définir le ratio d'équivalence de la simulation dans la section "equivalence ratio". À titre d'exemple, une plage entre [0,4 ; 1;4] serait attendue pour cette variable.
8. Définir le mécanisme de calcul de la simulation dans la section "formula". À titre d'exemple, les mécanismes *Sd_mec.cti*, *gri30.xml* et *Reaction.cti*.
9. Définir les fractions molaires des composants de la flamme dans la section "mole_frac". Il est important à noter que la somme des fractions molaires doit être égal à 1. Sinon, la simulation sera erronée.
10. Définir les composants de la flamme dans la section "composition". Il est important à noter que l'ordre des composants est la même que l'ordre des fraction molaire. Pour l'instant, les scripts fonctionnent seulement pour les compositions d'ammoniac (NH_3), d'hydrogène (H_2) et de méthane (CH_4).
11. Si l'utilisateur le souhaite, celui-ci peut rajouter de nouvelles simulations dans la section "simulations" en rajoutant un nouvelle section après la fermeture des accolades et répétant les étapes ci-dessus.

1.2 Établissement des simulations en parallèle

Dans le fichier zip se retrouve le fichier *main_PI3.py* qui représente le code principal à exécuter. Tout d’abord, ce fichier va lire les conditions de simulations définies dans *data.json* et vérifier si leur contenu est existant. Par la suite, le script va lancer au maximum n simulations en parallèle, où n est le nombre de coeurs de l’ordinateur. Cette limite est introduite pour ne pas excéder les limites de l’ordinateur.

Lorsque les simulations sont lancés, une bar de progression est présentée dans le terminal pour suivre le déroulement des simulations. Soit la figure ci-dessous représente un exemple de bar de progression du fichier *main_PI3.py*.

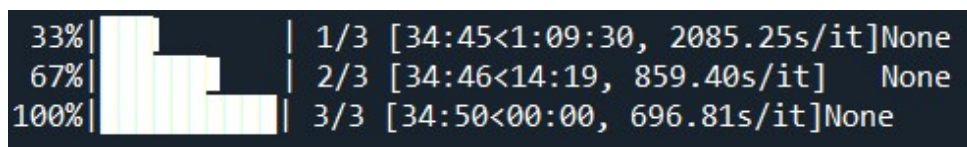


FIGURE 2 – Exemple de bar de progression du fichier *main_PI3.py*.

Sur la bar de progression, réalisée avec la librairie *tqdm*, il est possible de voir le nombre total de simulation, le pourcentage de complétion et le temps de compilation de chaque simulation.

1.3 Pre-simulation : Établissement d’une flamme libre

Dans le but de converger les simulations vers le point d’extinction, il faut tout d’abord trouver le point d’extinction. Pour se faire, il faut préalablement trouver une vitesse en $[\text{kg}/(\text{m}^2 * \text{s})]$ qui sera itérer vers la vitesse maximal, soit celle au point d’extinction.

Le fichier *freeflame.py* permet de simuler une flamme libre avec les conditions établies par l’utilisateur dans *data.json* pour extraire une vitesse et un taux de déformation servant de références pour les simulations ayant des parois limitant les flammes.

D'abord, le script calcule la masse molaire totale de la composition avec l'équation suivante :

$$M_{tot} = \left(\sum_{i=1}^n (M_i * x_i / eq) + M_{air} \right) / \left(\sum_{i=1}^n (x_i / eq) + n_{air} \right) \quad (1)$$

Où M_{tot} est la masse molaire total en [kg/mol], M_i est la masse molaire de chaque composant en [kg/mol], x_i est la fraction molaire de chaque composant en [mol], eq est le ratio d'équivalence, M_{air} est la masse molaire de l'air en [kg/mol], soit $M_{O_2} + M_{N_2}$, et n_{air} est la fraction molaire de l'air en [mol], soit $n_{O_2} + n_{N_2}$.

Ensuite, le script calcule la densité de la flamme libre avec l'équation suivante :

$$\rho = P * M_{tot} / (R * T) \quad (2)$$

Où P est la pression en [pa], R est la constante universelle des gaz parfaits $\approx 8,314 J / (mol * K)$ et T est la température du brûleur en [K].

Après, le script calcul l'épaisseur de la flamme avec l'équation suivante :

$$e = (T_{max} - T_{min}) / \nabla T \quad (3)$$

Où e est l'épaisseur en [m], T_{max} et T_{min} sont respectivement les températures maximales et minimales de la flamme en [K] et ∇T est le gradient de la température en [K/m].

Ensuite, le taux de déformation de référence de la flamme est calculé avec l'équation suivante :

$$a = v_{norm} / e \quad (4)$$

Où a est le taux de déformation en [1/s] et v et la vitesse normal sortant du brûleur en [m/s].

Par la suite, la vitesse de la flamme libre en [m/s] est calculé avec l'équation suivante :

$$v = a / d \quad (5)$$

Où v est la vitesse en [m/s] et d est la distance entre la paroi et le brûleur en [m].

Finalement, la vitesse en $[\text{kg}/(\text{m}^2 * \text{s})]$ de la flamme libre est calculé avec l'équation suivante :

$$\dot{m} = \rho \cdot v \quad (6)$$

Où \dot{m} est la vitesse en $[\text{kg}/(\text{m}^2 * \text{s})]$ et ρ est la densité en $[\text{kg}/\text{m}^3]$.

La valeur de la vitesse en $[\text{kg}/(\text{m}^2 * \text{s})]$ (\dot{m}) et celle du taux de déformation (a) sont retenues et transférer au fichier `main_PI3.py`.

1.4 Simulation : Trouver le point de convergence et calculer le taux de déformation au point d'extinction

Le fichier `ImpingingJet.py` permet de simuler une flamme axisymétrique qui se propage vers une surface non réactive avec les conditions établies par l'utilisateur dans `data.json` pour extraire le taux de déformation au point d'extinction.

Afin de converger la simulation vers la vitesse maximale, soit celle au point d'extinction, le script suit la logique suivante :

1. Un vecteur de vitesse est défini en divisant la vitesse de la flamme libre (\dot{m}) en 10 points.
2. La simulation est lancée pour la première vitesse et itère pour la vitesse suivante jusqu'à la convergence de la flamme, soit lorsque la température maximal de la simulation est égal à la température de la paroi.
3. Lorsque la simulation a convergé pour la première fois, un nouveau vecteur de vitesse est défini entre la vitesse avant la convergence et la vitesse de convergence pour raffiner les points d'une grandeur 10 fois plus petite.
4. La simulation est lancée une seconde fois en prenant la première vitesse de ce second vecteur et itérant pour la vitesse suivante jusqu'à la seconde convergence de la flamme. Ceci nous permet d'avoir une précision de +/- 1% la vitesse de la flamme libre (\dot{m}).
5. Lorsque la vitesse de convergence est trouvé, le taux de déformation est calculé à ce point en utilisant l'équation (4) pour obtenir a_{max} .