Autoignition of the acetylene and oxygen-nitrogen combinations mixture at 1200K

Ewelina Zaremba LiK Metody komputerowe w spalaniu

May 24, 2017

Scope

The scope of this examination was to observe the changes of four basic parameters describing autoignition, related to the ϕ ratio value. This parameters are:

- Autoignition time [s]
- Maximum ignition pressure [kPa]
- Explosiveness ratio – Kmax [MPa m s^{-1}]
- Final temperature [K]

Assumptions

The constant molar sum approach was conducted (it could also be the constant mass sum approach). Only overstechometric combustion was considered here (~0.2108 <= ϕ <= 1). Our two extreme cases was mixtures of acetylene and pure oxygen, and acetylene with air (simplified model – only oxygen and nitrogen included). Calculations were zero-dimensional.

Mathematical model

Build-in cantera functions used in the algorithm:

Solution('gri30.cti') - imputing properties typical for a gas mixture (here: 1200K – initial temperature, 1 at – initial pressure, and calculated molar sum) to the object (here: gas),

IdealGasReactor() - constant volumef, zero-dimensional reactor for ideal gas mixtures (used in find_auto_ign),

ReactorNet() - an integrator object with which the evolution of a reactor with time is performed mixtures (used in find_auto_ign).

Functions defined:

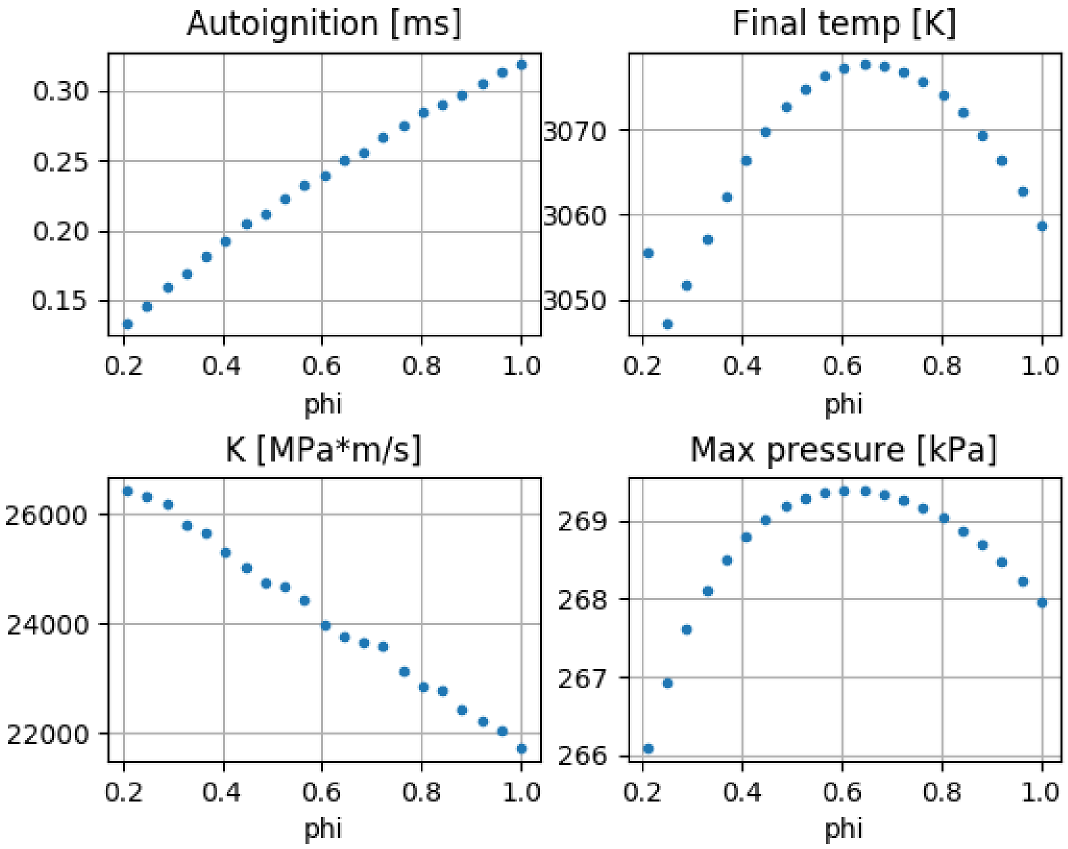
max_dt – returning maximum value of vector given, smooth – smoothing our plots, essential to eliminate errors in calculating derivatives,

find_auto_ign:

1. Autoignition returns autoignition time – the time for the value of max_dt in every state
2. K returns explosiveness ratio – a maximum pressure to time step ratio in every state
3. FinalTemp returns final temperature in every state
4. P_max returns maximum pressure in every state

Results

As it is shown on the plots below, the results were obtained in line with expectations.



Summary

Our x axis on every plot represents fi ratio - ~0.2108 for pure oxygen and 1 for air.

Time is increasing, because the less oxygen we have, the harder it is to obtain autoignition - mixture need more time to get to its autoignition temperature.

K is decreasing, what can be explained same as insceasing time - we have less oxygen what lowers the chances of autoignition.

Characteristics of final temperature and maximum pressure results from the different molar specific heat of oxygen and nitrogen.

References

[1] CANTERA Tutorials - A series of tutorials to get started witlyxh the python interface of Cantera version 2.1.1, Anne Felden

[2] Division of Engineering and Applied Science California Institute of Technology, Cantera Workshop, July 25, 2004, D. G. Goodwin

[3] Spalanie, 1969, Stanisław Wójcicki

Listing 1: autoign.py

```
import sys
import numpy as np
from cantera import *
import csv

def max_dt(vec, nt):
        nmax = 0
        dmax = 0
        for n in range(nt-2):
                dvec = (vec[n+1] - vec[n])
                if (dvec > dmax):
                        nmax = n
                        dmax = dvec
        return nmax

def smooth(y, box_pts):
    box = np.ones(box_pts)/box_pts
    y_smooth = np.convolve(y, box, mode='same')
    return y_smooth

def find_auto_ign(gas):
        #steps
        nt = 3000
        dt = 1.e-3 / nt#s
        #Storage space
        #mfrac = {} #tu beda sklady masowe gazow
        #for i in range(0, nt):
        #         mfrac[i] = np.zeros(gas.n_species, 'd') #wektor wektorow
        tim = np.zeros(nt,'d') #wektor o wymiarze nt
        temp = np.zeros(nt,'d')
        dtemp = np.zeros(nt-1,'d')
        pressure = np.zeros(nt, 'd')
        #################################################################
        #Create the batch reactor
        r = IdealGasReactor(gas)
        # Now create a reactor network consisting of the single batch reactor
        sim = ReactorNet([r])
        #Run the simulation
        # Initial simulation time
        time = 0.0
        #Loop for nt time steps of dt seconds.

        for n in range(nt):
                time += dt
                sim.advance(time)
                tim[n] = time
                temp[n] = r.T
                #mfrac[n][:] = r.thermo.Y
                pressure[n] = r.thermo.P #Pa
                #r.thermo.Y - sklad masowy, .X - sklad molowy
                #http://www.cantera.org/docs/sphinx/html/cython/thermo.html
        ####################################################################
        # Catch the autoignition timing
        ####################################################################

        Dtmax = max_dt(temp, nt)
        Autoignition = tim[Dtmax + 1]

        pressure_smooth = smooth(pressure, 15)
        Dpmax = max_dt(pressure, nt)
        dpdt_max = (pressure_smooth[Dpmax + 1] - pressure_smooth[Dpmax])/(dt)
        K = dpdt_max / 1e6 * 1#m3; K = MPa*m/s

        Autoignition = Autoignition*1000 #ms
        FinalTemp = temp[-1]

        p_max = max(pressure)


        #print "species_names: " + ', '.join(r.thermo.species_names)
        #index = r.thermo.species_index('C2H2') #indeks w wektorze mfrac
        #to_plot = np.zeros(nt,'d')
        #for i in range(0, nt):
        #       to_plot[i] = mfrac[i][index]
        #plot(tim, to_plot)
        #show()

        #print(r.thermo.report())

        return [Autoignition, FinalTemp, K, p_max]

#main

print "Computing:"

data = []
rg = 20
for i in range(rg + 1):
        gas = Solution('gri30.cti')

        theta = 1.*i / rg

        #N2_mole == 0: 7.52 * O2pzero - O2_mole + C2H2_mole * 2.5 == 0
        # 7.52 *  C2H2_mole * 2.5 / 2 - C2H2_mole * 2.5 / min_phi  + C2H2_mole *
            2.5 == 0
        # 7.52 *  2.5 / 2 - 2.5 / min_phi  + 2.5 == 0
        # 2.5/ min_phi == 7.52 * 2.5/2 + 2.5
        # min_phi = 2.5/(7.52 * 2.5/2 + 2.5)
        min_phi = 1 / (7.52 / 2 + 1)

        phi = 1 - theta * ( 1 - min_phi)
        C2H2_mole = 2

        #yCmHn + y(m + n/4)O2 -> ...
        #m = 2; n = 2; y = C2H2_mole;
        #O2_mole = C2H2_mole * (m +n /4) / phi

        O2_mole = C2H2_mole * 2.5 / phi

        O2pzero = C2H2_mole * 2.5 / 2

        N2_mole = 7.52 * O2pzero - O2_mole + C2H2_mole * 2.5
        H2O_mole = 0



        s = 'C2H2:' + str(C2H2_mole) + ',O2:' + str(O2_mole) + ',N2:' + str(
           N2_mole) + ',H2O:' + str(H2O_mole)
        print(s)

        gas.TPX = 1200, one_atm, s

        data.append([phi, C2H2_mole, O2_mole, N2_mole, H2O_mole] + find_auto_ign(
           gas))

print "done"


csv_file = 'data.csv'
with open(csv_file, 'w') as outfile:
        writer = csv.writer(outfile)
        writer.writerow(['phi', 'C2H2_mole', 'O2_mole', 'N2_mole', 'H2O_mole', '
           Auto␣ignition␣time␣[ms]','Final␣Temperature␣[K]','K␣[MPa*m/s]', 'Max␣
           pressure␣Pa'])
        for d in data:
                writer.writerow(d)
        print 'output␣written␣to␣'+csv_file
```

Listing 2: show.py

```
import csv
from matplotlib.pylab import *

csv_file = 'data.csv'

#data
data = [ [] ]
frac = []
autoign = []
temp = []
K = []
press = []

#read and parse
with open(csv_file, 'r') as infile:
        reader = csv.reader(infile)
        i = 0
        for row in reader:
```

```
        i = i + 1
        if i == 1:
            for n in range(len(data), len(row)):
                data.append([])
            continue

        for j, elem in enumerate(row):
            val = float(elem)
            data[j].append(val)

frac = data[0]
autoign = data[5]
temp = data[6]
K = data[7]
press = data[8]
for i in range(0, len(press)):
    press[i] /= 1000


f, axarr = plt.subplots(2, 2)
axarr[0][0].plot(frac, autoign, '.')
axarr[0][0].set_title("Autoignition␣[ms]")
axarr[0][0].set_xlabel("phi")
axarr[0][1].plot(frac, temp, '.')
axarr[0][1].set_title("Final␣temp␣[K]")
axarr[0][1].set_xlabel("phi")
axarr[1][0].plot(frac, K, '.')
axarr[1][0].set_title("K␣[MPa*m/s]")
axarr[1][0].set_xlabel("phi")
axarr[1][1].plot(frac, press, '.')
axarr[1][1].set_title("Max␣pressure␣[kPa]")
axarr[1][1].set_xlabel("phi")

axarr[0][0].grid()
axarr[0][1].grid()
axarr[1][0].grid()
axarr[1][1].grid()

f.subplots_adjust(hspace=0.5)

#plot(Ti2,Autoignition_cas, '^', color = 'orange')
#xlabel(r'Temp [1000/K]', fontsize=20)
#ylabel("Autoignition [ms]")
#title(r'Autoignition of $CH_{4}$ + Air mixture at $\Phi$ = 1, and P = 1 bar',
#fontsize=22,horizontalalignment='center')
    #


#       to_plot = np.zeros(nt,'d')
#       for i in range(0, nt):
#           to_plot[i] = mfrac[i][index]
#             plot(tim, to_plot)
            #plot(tim, pressure)

            #axis([0, nt*dt, 0, 1])

show()
```