SU2 11주차 보고서

2019011579 김세형

Caradonna & Tung Rotor

- Caradonna & Tung rotor를 SU2 를 통해 해석한 결과와 NASA 에서 진행한 실험 결과를 비교.
- Mach 0.877, 회전RPM 2,500, Pitch angle 8 degree를 적용.
- Rotor blade span 의 50%, 68%, 80%, 89%, 96% 에 해당하는 지점에서 slice.
- 각 slice에서의 Pressure를 통해 NASA 의 실험값과 비교할 수 있도록 유의미한 값을 유도.
- 원활한 비교를 위해 NASA 에서 제공한 pressure coefficient 결과값 표를 엑셀로 옮겨 기입.

TABLE 25.- LOCAL PRESSURE COEFFICIENT

$\theta_c = 8°$ $\Omega = 2500$ rpm $M_{tip} = 0.877$

Upper surface ($-C_{p_U}$)

x/c	r/R = 0.5	x/c	r/R = 0.68	x/c	r/R = 0.80	x/c	r/R = 0.89	x/c	r/R = 0.96
0.00	-0.9138E+00	0.02	0.9416E+00	0.00	-0.1853E+00	0.01	0.4445E+00	0.00	-0.1092E+01
0.03	0.8691E+00	0.06	0.9399E+00	0.01	0.4832E+00	0.03	0.8531E+00	0.02	0.4984E+00
0.12	0.7173E+00	0.10	0.9083E+00	0.04	0.1152E+01	0.04	0.9171E+00	0.07	0.8911E+00
0.26	0.5317E+00	0.15	0.8169E+00	0.07	0.1091E+01	0.06	0.9325E+00	0.12	0.9721E+00
0.47	0.3517E+00	0.19	0.7256E+00	0.09	0.1095E+01	0.10	0.1046E+01	0.15	0.1062E+01
0.69	0.2055E+00	0.23	0.6448E+00	0.13	0.1166E+01	0.13	0.1108E+01	0.19	0.1090E+01
0.83	0.8736E-01	0.29	0.5834E+00	0.17	0.1170E+01	0.17	0.1166E+01	0.23	0.1028E+01
		0.33	0.5307E+00	0.21	0.1061E+01	0.21	0.1208E+01	0.29	0.9676E+00
		0.39	0.4183E+00	0.24	0.7839E+00	0.26	0.1172E+01	0.33	0.6605E+00
		0.44	0.3937E+00	0.30	0.4320E+00	0.30	0.1076E+01	0.39	0.2239E+00
		0.52	0.3235E+00	0.35	0.4990E+00	0.35	0.5771E+00	0.44	0.1947E+00
		0.61	0.2251E+00	0.42	0.4162E+00	0.42	0.2751E+00	0.50	0.1902E+00
		0.73	0.1338E+00	0.50	0.3230E+00	0.47	0.2111E+00	0.61	0.1474E+00
		0.80	0.6530E-01	0.56	0.2586E+00	0.52	0.1673E+00	0.69	0.1103E+00
				0.65	0.1746E+00	0.65	0.9980E-01	0.76	0.6527E-01
				0.76	0.7476E-01	0.73	0.5006E-01	0.90	-0.6074E-01
				0.90	-0.8021E-01	0.80	-0.6794E-02		
						0.87	-0.7194E-01		

Lower surface ($-C_{p_L}$)

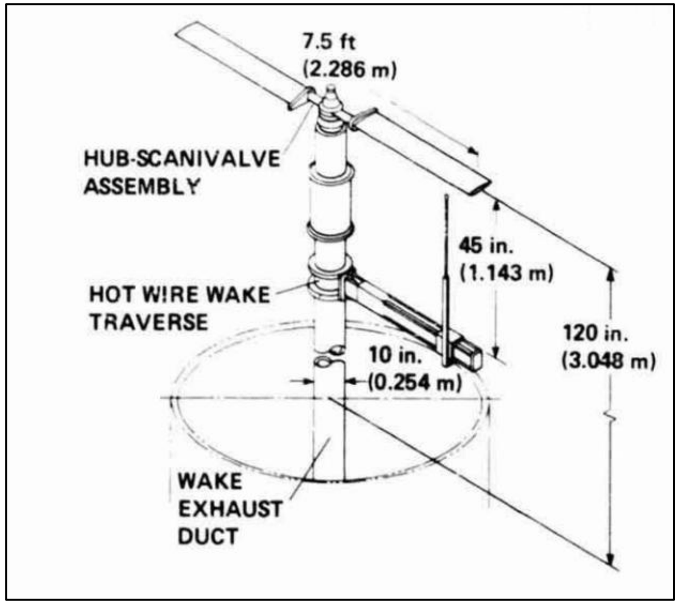
x/c	r/R = 0.5	x/c	r/R = 0.68	x/c	r/R = 0.80	x/c	r/R = 0.89	x/c	r/R = 0.96
0.04	-0.5606E-01	0.00	-0.9269E+00	0.01	-0.5175E+00	0.01	-0.7021E+00	0.00	-0.1141E+01
0.20	0.2758E+00	0.07	-0.2059E-01	0.02	-0.3691E+00	0.04	-0.2745E+00	0.07	-0.8774E-[illegible]
0.45	0.2195E+00	0.18	0.2523E+00	0.11	0.1601E+00	0.14	0.1886E+00	0.16	0.2487E+00
0.69	0.9861E-01	0.28	0.2804E+00	0.14	0.2534E+00	0.28	0.3035E+00	0.24	0.3263E+00
0.85	0.3674E-01	0.38	0.2335E+00	0.24	0.3020E+00	0.45	0.2514E+00	0.39	0.2588E+00
		0.51	0.1679E+00	0.34	0.2862E+00	0.57	0.1543E+00	0.51	0.1755E+00
		0.57	0.1364E+00	0.57	0.1601E+00	0.69	0.9980E-01	0.63	0.1227E+00
		0.79	0.2036E-01	0.74	0.6557E-01	0.79	0.2163E-01	0.74	0.6640E-01
				0.90	-0.6445E-01	0.90	-0.7312E-01	0.85	0.1127E-01
C_L	0.2298		0.2842		0.2736		0.2989		0.3175

upper

r/R=0.5 x/c	r/R=0.5 Cp	r/R=0.68 x/c	r/R=0.68 Cp	r/R=0.80 x/c	r/R=0.80 Cp	r/R=0.89 x/c	r/R=0.89 Cp	r/R=0.96 x/c	r/R=0.96 Cp
0	-0.9138	0.02	0.9416	0	-0.1853	0.01	0.4445	0	-1.092
0.03	0.8691	0.06	0.9399	0.01	0.4832	0.03	0.8531	0.02	0.4984
0.12	0.7173	0.1	0.9083	0.04	1.152	0.04	0.9171	0.07	0.8911
0.26	0.5317	0.15	0.8169	0.07	1.091	0.06	0.9325	0.12	0.9721
0.47	0.3517	0.19	0.7256	0.09	1.095	0.1	1.046	0.15	1.062
0.69	0.2055	0.23	0.6448	0.13	1.166	0.13	1.108	0.19	1.09
0.83	0.08736	0.29	0.5834	0.17	1.17	0.17	1.166	0.23	1.028
		0.33	0.5307	0.21	1.061	0.21	1.208	0.29	0.9676
		0.39	0.4183	0.24	0.7839	0.26	1.172	0.33	0.6605
		0.44	0.3937	0.3	0.432	0.3	1.076	0.39	0.2239
		0.52	0.3235	0.35	0.499	0.35	0.5771	0.44	0.1947
		0.61	0.2251	0.42	0.4162	0.42	0.2751	0.5	0.1902
		0.73	0.1338	0.5	0.323	0.47	0.2111	0.61	0.1474
		0.8	0.0653	0.56	0.2586	0.52	0.1673	0.69	0.1103
				0.65	0.1746	0.65	0.0998	0.76	0.06527
				0.76	0.07476	0.73	0.05006	0.9	-0.06074
				0.9	-0.08021	0.8	-0.06794		
						0.87	-0.07194		

lower

r/R=0.5 x/c	r/R=0.5 Cp	r/R=0.68 x/c	r/R=0.68 Cp	r/R=0.80 x/c	r/R=0.80 Cp	r/R=0.89 x/c	r/R=0.89 Cp	r/R=0.96 x/c	r/R=0.96 Cp
0.04	-0.0561	0	-0.9269	0.01	-0.5175	0.01	-0.7021	0	-1.141
0.2	0.2758	0.07	-0.02059	0.02	-0.3691	0.04	-0.2745	0.07	-0.8774
0.45	0.2195	0.18	0.2523	0.11	0.1601	0.14	0.1886	0.16	0.2487
0.69	0.09861	0.28	0.2804	0.14	0.2534	0.28	0.3035	0.24	0.3263
0.85	0.03674	0.38	0.2335	0.24	0.302	0.45	0.2514	0.39	0.2588
		0.51	0.1679	0.34	0.2862	0.57	0.1543	0.51	0.1755
		0.57	0.1364	0.57	0.1601	0.69	0.0998	0.63	0.1227
		0.79	0.02036	0.74	0.06557	0.79	0.02163	0.74	0.0664
				0.9	-0.06445	0.9	-0.07312	0.85	0.01127



The model and experimental setup

Pressure to Pressure Coefficient

- $Cp = \frac{P-P_\infty}{q} = \frac{P-P_\infty}{\frac{1}{2}\rho v^2}$
- $P = Pressure$
- $P_\infty = 101325\ Pa$
- $q = \frac{1}{2}\rho v^2$
- $\rho = 1.225\ kg/m^3$
- $v = r\omega$
- $r = 0.143 + (Section)$
- $\omega = \frac{2\pi n}{60} = \frac{2\pi \times 2500}{60}$

Simulation Condition

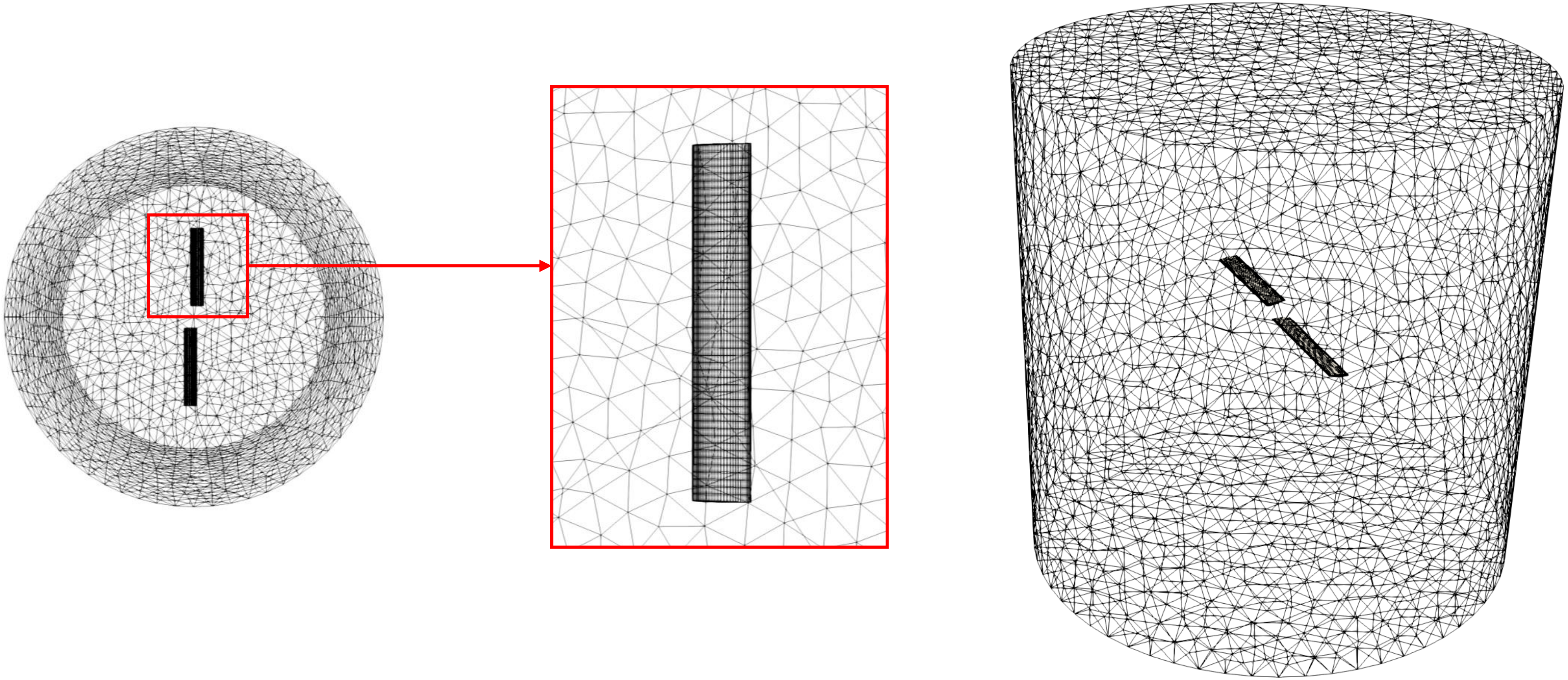
- FREESTREAM_PRESSURE= 101325.0
- FREESTREAM_TEMPERATURE= 288.15
- REF_LENGTH= 1.0 -> rotor blade span이 1m 라는 것을 알 수 있음.
- GRID_MOVEMENT= ROTATING_FRAME
- MACH_MOTION= 0.877 -> 공기를 돌리는 속도. 최대 tip MACH
- ROTATION_RATE = 261.79938779914943 0.0 0.0 -> x축을 기준으로 회전.

- Boundary Condition
  - MARKER_EULER= (blade_1, blade_2)
  - MARKER_FAR= (farfield)
  - MARKER_PLOTTING= (blade_1, blade_2)
  - MARKER_MONITORING= (blade_1, blade_2)

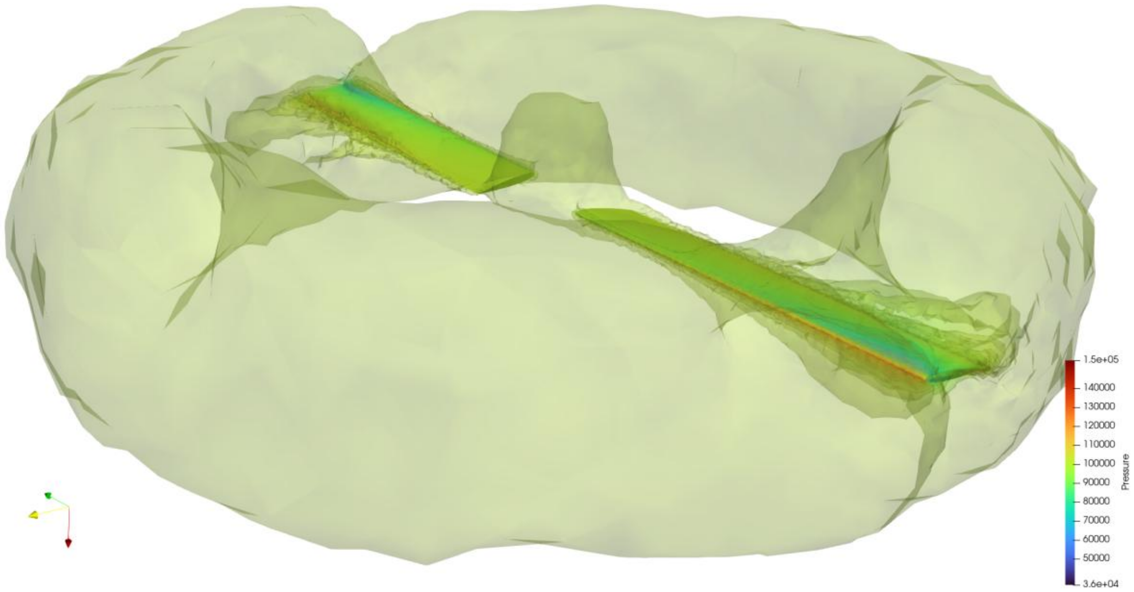
- COMMON PARAMETERS DEFINING THE NUMERICAL METHOD
  - NUM_METHOD_GRAD= GREEN_GAUSS
  - CFL_NUMBER= 1e3
  - CFL_ADAPT= NO
  - CFL_ADAPT_PARAM= (0.1, 2.0, 15, 1e6)
  - RK_ALPHA_COEFF= (0.66667, 0.66667, 1.000000)
  - ITER= 99999

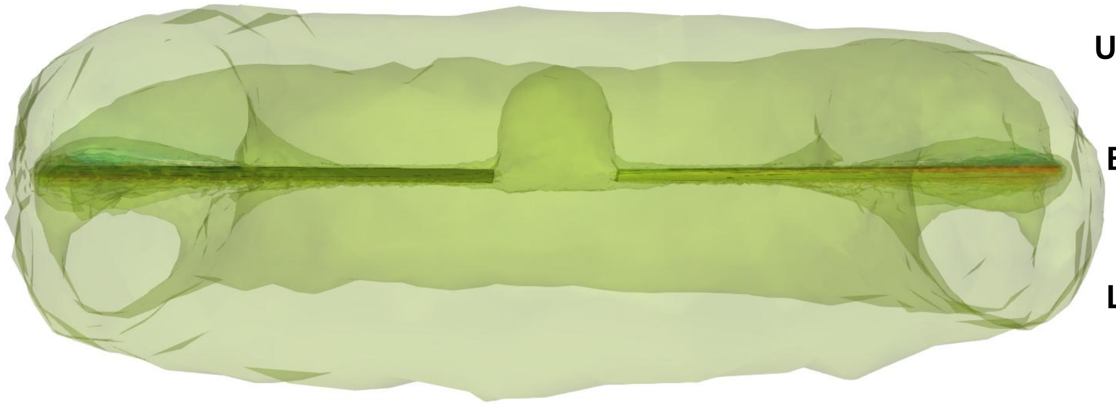
Mesh

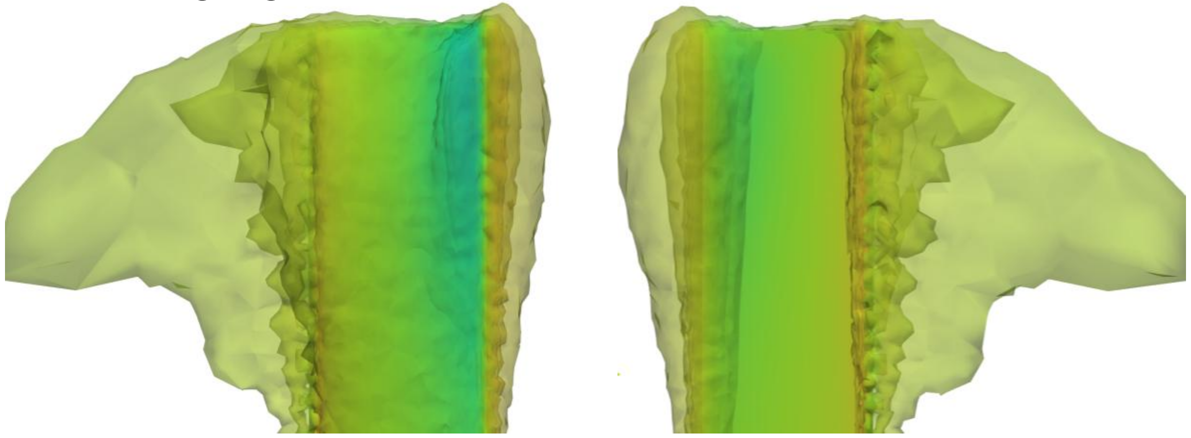
- **NDIME= 3**
- **NELEM= 326085**

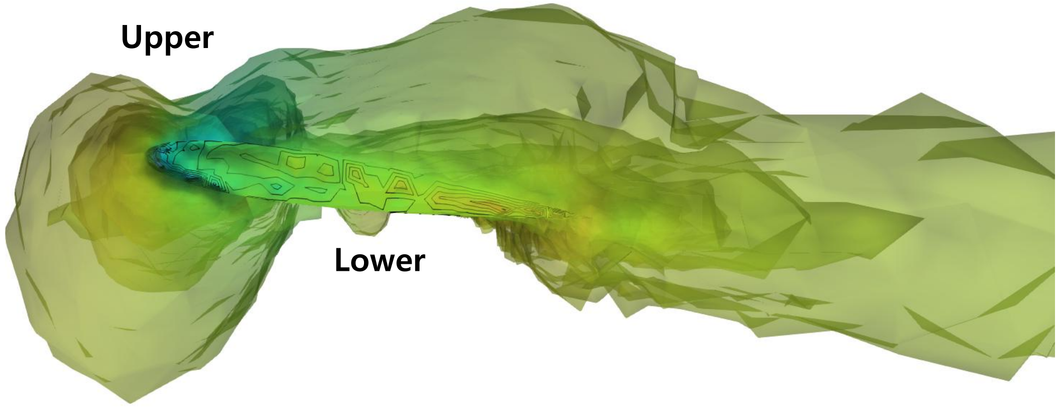


Pressure Contour_Toroidal “Vortex Ring”

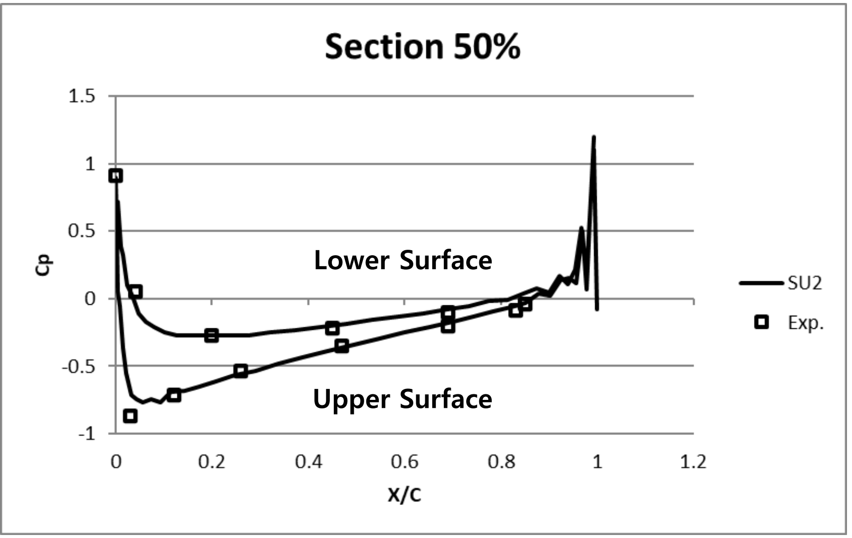


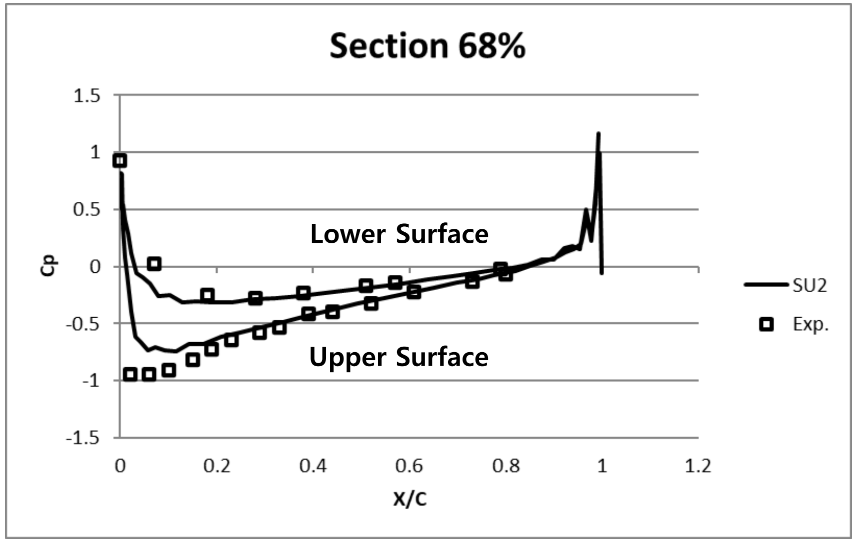


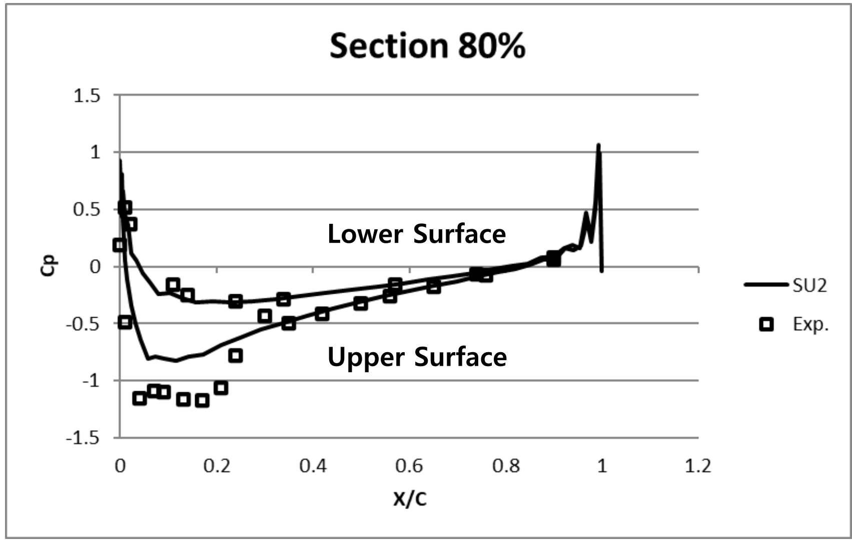


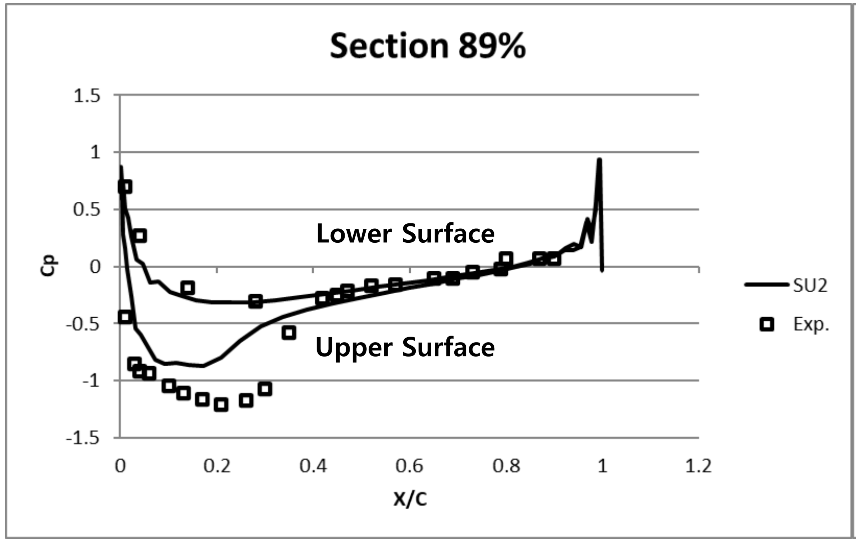


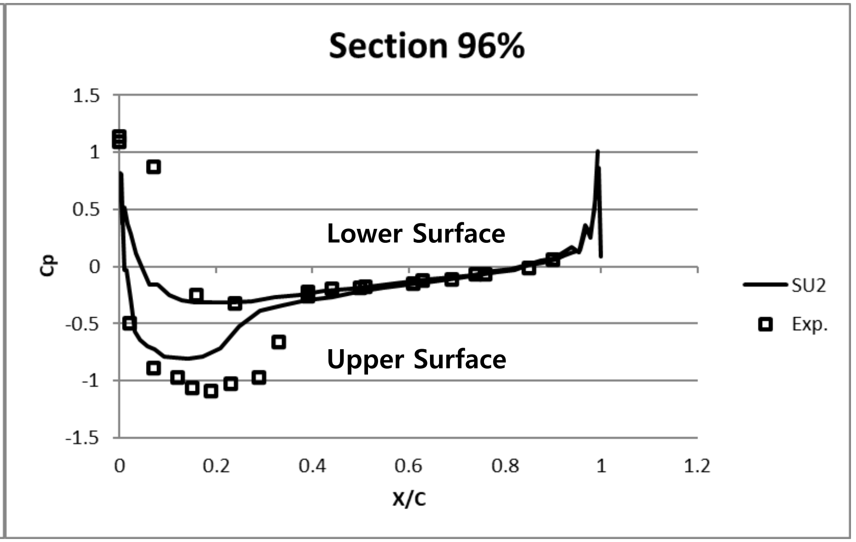
Result


Section 50%
Cp
x/c
Lower Surface
Upper Surface
SU2
Exp.


Section 68%
Cp
x/c
Lower Surface
Upper Surface
SU2
Exp.


Section 80%
Cp
x/c
Lower Surface
Upper Surface
SU2
Exp.


Section 89%
Cp
x/c
Lower Surface
Upper Surface
SU2
Exp.


Section 96%
Cp
x/c
Lower Surface
Upper Surface
SU2
Exp.

Analysis

- Section 50% 에서는 해석값과 실험값이 거의 유사.
- Section 68%, 80%, 89%, 96% 에서는 뒷전 부근에서는 해석값과 실험값이 유사하지만 앞전부터 압력의 최저점부근에서 해석값과 실험값의 차이가 확인됨.
- 해석 결과와 실험결과에서 차이가 발생하는 이유는 해석 툴인 SU2가 세밀하게 압력을 계산해내지 못함을 생각할 수 있다.
- 격자가 충분히 fine 하지 않은 것도 원인이 될 수 있다고 생각.
- 실험값에 더 근사하기 위해서는 어떤 수정을 해야 할지 더 고민이 필요.
- 격자를 더 fine 하게 생성하는 것이 첫번째 대안.
- 전체 farfield 격자를 360 degree 모두 생성하지 않고 180 degree 만 생성, rotor blade 역시 한쪽만 생성한 후 해석을 돌린다면 같은 시간 대비 보다 더 정밀한 해석 결과를 얻을 수 있을 것으로 예상됨.

Cp_Visualizer_python code

```python
import pandas as pd
import os
import xlsxwriter
import sys

try:
    file_input = input("Enter the file number (e.g., 0.5): ").strip()
except KeyboardInterrupt:
    sys.exit()

csv_filename = f"{file_input}.csv"

if not os.path.exists(csv_filename):
    print(f"Error: File '{csv_filename}' not found.")
    sys.exit()

try:
    file_value = float(file_input)
except ValueError:
    file_value = 0.5

df = pd.read_csv(csv_filename)

if 'Points_1' not in df.columns or 'Pressure' not in df.columns:
    print("Error: 'Points_1' or 'Pressure' columns are missing in the CSV file.")
    sys.exit()

P_at = 101325
r = 0.143 + file_value
omega = 261.7993878
v = r * omega
q = 0.5 * 1.225 * (v**2)

section_percent = int(round(file_value * 100))
chart_title_text = f"Section {section_percent}%"

df['P_minus_Pat'] = df['Pressure'] - P_at
df['Cp'] = df['P_minus_Pat'] / q

sheet3_data = df[['Points_1', 'Cp']].copy()
sheet3_data = sheet3_data.sort_values(by='Points_1', ascending=False).reset_index(drop=True)

tags = []
n_rows = len(sheet3_data)
```

```python
for i in range(n_rows):
    if i == 0:
        val = 1
    elif i == 1:
        val = 1
    elif i == 2:
        val = 2
    elif i == n_rows - 1:
        val = 2
    else:
        if i % 2 != 0:
            val = 1
        else:
            val = 2
    tags.append(val)

sheet3_data['SortKey'] = tags
sheet3_data = sheet3_data.sort_values(by=['SortKey', 'Points_1'], ascending=[False, False])

output_filename = f"{file_input}.xlsx"
writer = pd.ExcelWriter(output_filename, engine='xlsxwriter')
workbook = writer.book

sheet1 = workbook.add_worksheet('Sheet1')

sheet1.write('A1', 'Points_1')
sheet1.write('B1', 'Pressure')
sheet1.write('C1', 'P-P_at')
sheet1.write('E1', 'Parameters')
sheet1.write('F1', 'Values')
sheet1.write('G1', 'Cp')

params = [('r', r), ('omega', omega), ('v', v), ('q', q)]

for i, row in df.iterrows():
    sheet1.write(i+1, 0, row['Points_1'])
    sheet1.write(i+1, 1, row['Pressure'])
    sheet1.write(i+1, 2, row['P_minus_Pat'])
    sheet1.write(i+1, 6, row['Cp'])

    if i < len(params):
        sheet1.write(i+1, 4, params[i][0])
        sheet1.write(i+1, 5, params[i][1])
```

Cp_Visualizer_python code

```python
sheet2 = workbook.add_worksheet('Sheet2')
sheet2.write(0, 0, 'Points_1')
sheet2.write(0, 1, 'Pressure')
for i, row in df.iterrows():
    sheet2.write(i+1, 0, row['Points_1'])
    sheet2.write(i+1, 1, row['Pressure'])

sheet3 = workbook.add_worksheet('Sheet3')
sheet3.write('A1', 'SortKey')
sheet3.write('B1', 'Points_1')
sheet3.write('C1', 'Cp')

data_rows = sheet3_data.to_dict('records')
current_excel_row = 1
inserted_row_index = -1
previous_tag = None

for i, row_data in enumerate(data_rows):
    current_tag = row_data['SortKey']

    if previous_tag == 2 and current_tag == 1:
        inserted_row_index = current_excel_row
        current_excel_row += 1

    sheet3.write(current_excel_row, 0, current_tag)
    sheet3.write(current_excel_row, 1, row_data['Points_1'])
    sheet3.write(current_excel_row, 2, row_data['Cp'])

    previous_tag = current_tag
    current_excel_row += 1

last_data_row = current_excel_row

range_str = f"B2:B{last_data_row}"
sheet3.write_formula('E2', f'=MAX({range_str})')
sheet3.write_formula('E3', f'=MIN({range_str})')
sheet3.write_formula('E4', '=E2-E3')
```

```python
sheet3.write('G1', 'Points_1')
sheet3.write('H1', 'Cp')

for r_idx in range(1, last_data_row):
    if r_idx == inserted_row_index:
        continue

    excel_row = r_idx + 1
    g_formula = f'=(B{excel_row}-$E$3)/$E$4'
    h_formula = f'=C{excel_row}'

    sheet3.write_formula(r_idx, 6, g_formula)
    sheet3.write_formula(r_idx, 7, h_formula)

chart = workbook.add_chart({'type': 'scatter', 'subtype': 'straight'})
chart.add_series({
    'name':       'SU2',
    'categories': ['Sheet3', 1, 6, last_data_row - 1, 6],
    'values':     ['Sheet3', 1, 7, last_data_row - 1, 7],
    'line':       {'color': 'black', 'width': 2},
})

chart.set_title ({'name': chart_title_text})
chart.set_x_axis({
    'name': 'X/C',
    'label_position': 'low',
})
chart.set_y_axis({
    'name': 'Cp',
})

sheet3.insert_chart('J2', chart)

writer.close()
print("Done.")
```