

전산유체역학

PJ#1: Grid Generation



항공우주 및 기계공학부

담당교수 : 김문상 교수님

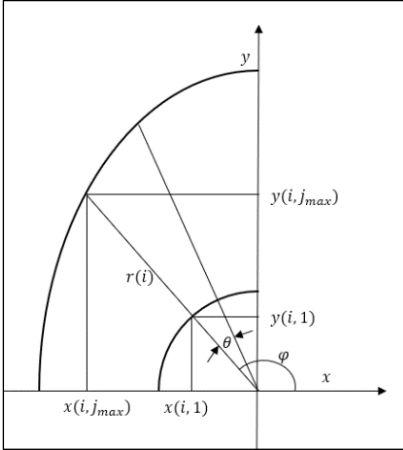
제출일 : 2021-04-20

2016121150 윤준영

1. 물체 표면과 외부 경계면에서의 격자점 생성

먼저 물체 표면에서의 격자점과 외부 경계면에서의 격자점을 구한 뒤 두 점을 이은 후 특정한 규칙에 따라 내부 영역을 나누어서 격자를 생성한다.

1-1) 물체 표면에서의 격자점 생성(등간격 호 길이)



반지름이 1인 사분원에서 사이각 θ 와 x 축과의 각도 φ 는 다음과 같다.

$$\theta = \frac{\frac{2}{\pi}}{f_{nx} - 1}, \quad \varphi = \pi - \theta(i - 1)$$

따라서 물체 표면에서($j = 1$) 격자점의 x, y 좌표는 다음과 같다.

$$x(i, 1) = radi1 \times \cos\varphi, \quad y(i, 1) = radi1 \times \sin\varphi$$

<그림 1 : 물체 표면, 외부 경계면>

위 식을 사용하여 물체 표면에서 격자점을 생성하는 <surfacegrid.m>함수를 작성하여 사용하였다.

```
function [x,y]=surfacegrid(fnx,radi1)
% grid generation - body surface
for i=1:1:fnx
    theta = (pi/2)/(fnx-1); % angle between surface grids (rad)
    phi=pi-theta*(i-1); % angle between x-axis and grid (>=90) (rad)
    x(i,1) = radi1*cos(phi); % x component of body surface grid
    y(i,1) = radi1*sin(phi); % y component of body surface grid
end
```

<MATLAB CODE 1 : 물체 표면에서의 격자점 생성 함수 surfacegrid.m>

1-2) 외부 경계면에서의 격자점 생성

외부 경계면에서의 격자점 (x,y) 는 원점까지의 거리 r 을 이용하면 다음과 같이 표현할 수 있다.

$$x(i, j_{max}) = r \cos \varphi, \quad y(i, j_{max}) = r \sin \varphi$$

여기서 장반경이 $brad$, 단반경이 $arad$ 인 타원 방정식을 이용하여 r 을 구하면

$$\frac{x^2}{arad^2} + \frac{y^2}{brad^2} = \frac{r^2 \cos^2 \varphi}{arad^2} + \frac{r^2 \sin^2 \varphi}{brad^2} = 1$$

$$r = \sqrt{\frac{1}{\frac{\cos^2 \varphi}{arad^2} + \frac{\sin^2 \varphi}{brad^2}}}$$

r 을 위 $x(i, j_{max}) = r \cos \varphi$, $y(i, j_{max}) = r \sin \varphi$ 에 대입하면 외부 경계면에서($j = j_{max}$) 격자점의 x , y 좌표는 다음과 같다

$$x(i, j_{max}) = \frac{\cos \varphi}{\sqrt{\frac{\cos^2 \varphi}{arad^2} + \frac{\sin^2 \varphi}{brad^2}}}, \quad y(i, j_{max}) = \frac{\sin \varphi}{\sqrt{\frac{\cos^2 \varphi}{arad^2} + \frac{\sin^2 \varphi}{brad^2}}}$$

위 식을 이용하여 외부 경계면의 격자점을 생성하는 <outergrid.m> 함수를 작성하여 사용하였다.

```
function [x,y]=outergrid(x,y,fnx,fny,arad,brad)
% grid generation - outer boundary
for i=1:1:fnx
    theta = (pi/2)/(fnx-1); % angle between surface grids (rad)
    phi=pi-theta*(i-1); % angle between x-axis and grid (>=90) (rad)
    r = 1/(sqrt(cos(phi)^2/arad^2+sin(phi)^2/brad^2)); % distance between outer boundary
    grid and origin
    x(i,fny) = r*cos(phi); % x component of outer boundary grid
    y(i,fny) = r*sin(phi); % y component of outer boundary grid
end
```

<MATLAB CODE 2 : 외부 경계면에서의 격자점 생성 함수 outergrid.m>

2. 등비수열의 합을 이용한 격자

2-1) 등비수열의 합을 이용한 격자 - 일반 격자

임의의 i 에서 $j = 1$ 과 $j = j_{max}$ 인 격자점 사이의 거리의 x 성분 $xleng$ 은 다음과 같이 구할 수 있다.

$$x(i, j_{max}) - x(i, 1) = xleng = \Delta x \frac{1 - strch^{j_{max}-1}}{1 - strch}$$

이를 정리하고 임의의 i , j 에서의 x 좌표를 구하면 다음과 같다.

$$x(i, j) = x(i, 1) + \Delta x \frac{1 - strch^{j-1}}{1 - strch} = x(i, 1) + \frac{1 - strch}{1 - strch^{j_{max}-1}} xleng \frac{1 - strch^{j-1}}{1 - strch}$$

$xleng$ 과 $coeff1$, $coeff2$ 을 다음과 같이 설정하면, $x(i, j)$ 은 다음과 같다.

$$xleng = x(i, j_{max}) - x(i, 1), \quad coeff1 = 1 - strch^{j-1}, \quad coeff2 = 1 - strch^{j_{max}-1}$$

$$x(i, j) = x(i, 1) + xleng \times \frac{coeff1}{coeff2}$$

마찬가지로 임의의 i , j 에서의 $yleng$ 과 $coeff1$, $coeff2$ 을 다음과 같이 설정하면, $y(i, j)$ 은 다음과 같다.

$$yleng = y(i, j_{max}) - y(i, 1), \quad coeff1 = 1 - strch^{j-1}, \quad coeff2 = 1 - strch^{j_{max}-1}$$

$$y(i, j) = y(i, 1) + yleng \times \frac{coeff1}{coeff2}$$

위 식을 이용하여 작성된 MATLAB 코드는 아래와 같다.

```
%% CFD HW#1_격자 : 등비수열의 합 이용
%% data input
clear; close all;
```

```

prompt = {'arad (grix x radius)', 'brad (grid y radius)', 'radil (body radius)', 'fnx (number of i grid(imax))', 'fny (number of j grid(jmax))', 'strch (common ratio r)'};
dlgtitle = 'Data'; dims = [1 35];
definput = {'3', '7', '1', '50', '100', '1.03'}; % set default value
inputdata = inputdlg(prompt,dlgtitle,dims,definput); % input data dialogue
arad = str2double(inputdata{1}); % grid x radius
brad = str2double(inputdata{2}); % grid y radius
radil = str2double(inputdata{3}); % body radius
fnx = str2double(inputdata{4}); % number of i grid(imax)
fny = str2double(inputdata{5}); % number of j grid(jmax)
strch = str2double(inputdata{6}); % common ration of sequence

%% grid generation - body surface and outer boundary
[x,y] = surfacegrid(fnx,radil); % body surface grid generation function
[x,y] = outergrid(x,y,fnx,fny,arad,brad); % outer boundary grid generation function

%% grid generation - inner flow region
for j = 2:1:fny
    coeff1 = 1 - strch^(j-1);
    coeff2 = 1 - strch^(fny-1);
    for i=1:1:fnx-1
        xleng = x(i,fny) - x(i,1); % x difference of outer grid and body surface
        yleng = y(i,fny) - y(i,1); % y difference of outer grid and body surface
        x(i,j) = x(i,1) + xleng * coeff1/coeff2; % x grid component of inner flow region(i,j)
        y(i,j) = y(i,1) + yleng * coeff1/coeff2; % y grid component of inner flow region(i,j)
    end
    x(fnx,j) = x(fnx,1); % align x grid component on y-axis
    y(fnx,j) = y(fnx,1) + yleng * coeff1/coeff2; % align y grid component on y-axis
end

%% 1/4 oval grid plot
figure(1); hold on; grid on;
for i = 1:1:fnx
    for j = 1:1:fny
        plot(x(i,:),y(i,:), 'k')
        plot(x(:,j),y(:,j), 'k')
    end
end
title('등비 수열의 합', 'FontSize',12, 'FontName', '맑은 고딕')
xlabel('x'); ylabel('y'); axis equal; grid on;
axis([-arad-0.5 0.5 -0.5 brad+0.5])
set(figure(1), 'OuterPosition', [1400, 300, 360, 630])

%% full oval grid plot
figure(2); hold on; grid on;
for i = 1:2:fnx
    for j = 1:2:fny
        plot(x(i,:),y(i,:), 'k'); plot(-x(i,:),y(i,:), 'k');
        plot(x(i,:),-y(i,:), 'k'); plot(-x(i,:),-y(i,:), 'k');
        plot(x(:,j),y(:,j), 'k'); plot(-x(:,j),y(:,j), 'k');
        plot(x(:,j),-y(:,j), 'k'); plot(-x(:,j),-y(:,j), 'k');
    end
end
title('등비 수열의 합 (full)', 'FontSize',12, 'FontName', '맑은 고딕')
xlabel('x'); ylabel('y'); axis equal; grid on;
axis([-arad-0.5 arad+0.5 -brad-0.5 brad+0.5])
set(figure(2), 'OuterPosition', [1400, 300, 360, 630])

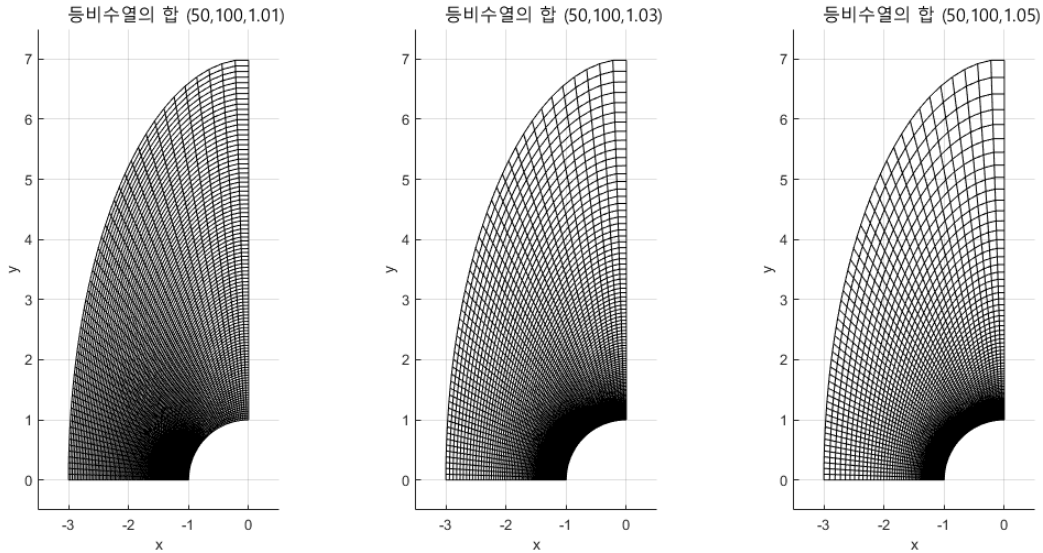
```

<MATLAB CODE 3 : 등비수열의 합을 이용한 격자 생성>

2-2) 격자 Plot 결과

2-2-1) 등비수열의 공비 $strch$ 값에 따른 1/4 타원 격자

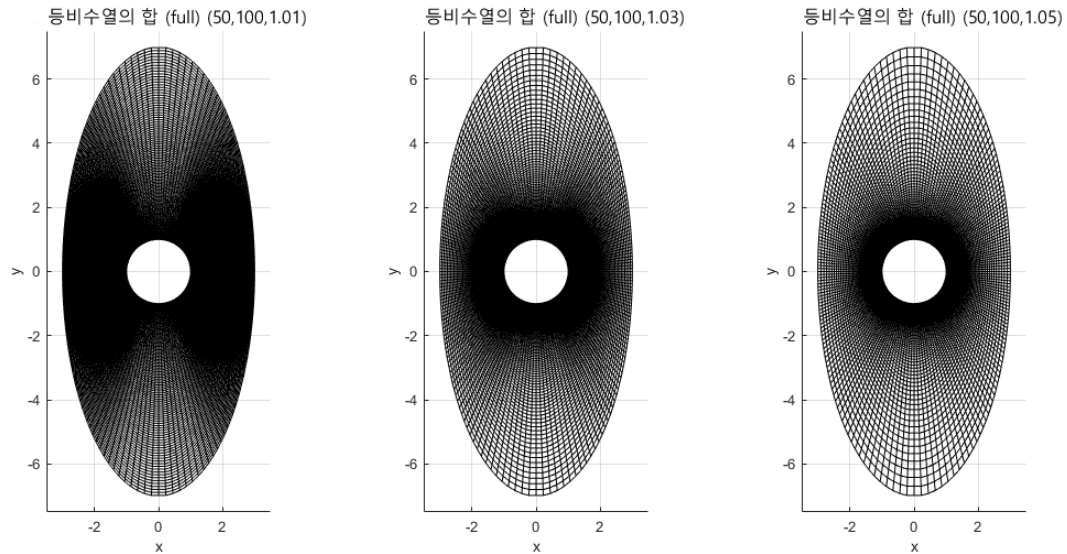
$fnx = 50$, $fny = 100$ 으로 고정한 후 $strch$ 값을 1.01, 1.03, 1.05로 증가시켜 1/4 타원형 격자를 생성하였다.



<그림 2, 3, 4 - $fnx = 50$, $fny = 100$, $strch = 1.01, 1.03, 1.05$ 일 때의 등비수열의 합을 이용한 1/4 타원형 grid>

2-2-2) 등비수열의 공비 $strch$ 값에 따른 full 타원 격자

$fnx = 50$, $fny = 100$ 으로 고정한 후 $strch$ 값을 1.01, 1.03, 1.05로 증가시켜 full 타원형 격자를 생성하였다.



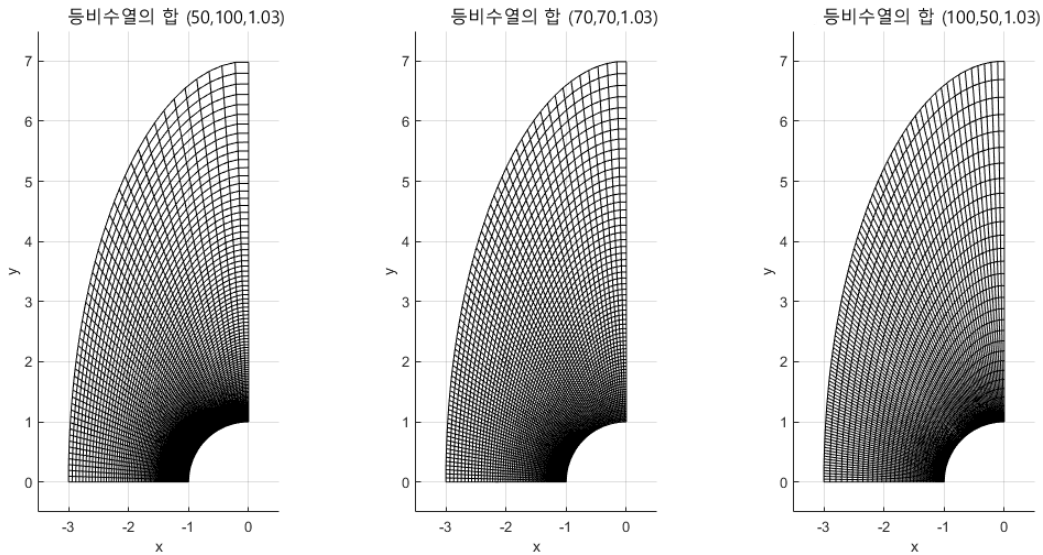
<그림 5, 6, 7 - $fnx = 50$, $fny = 100$, $strch = 1.01, 1.03, 1.05$ 일 때의 등비수열의 합을 이용한 full 타원형 grid>

그림 2, 3, 4, 그림 5, 6, 7을 비교하면 같은 fnx , fny 값을 가지는 격자에서 등비수열의 공비 $strch$ 값에 따른 격자의 특성을 확인할 수 있다. $strch$ 값이 증가하면 격자가 물체 표면에 더 집중적으로 분포되는 것

을 확인할 수 있다. 항공기나 자동차와 같은 물체 주위의 유동을 해석할 때 물체 표면 부근에서의 유동의 변화가 크므로 $strch$ 값을 적당히 증가시켜 물체 표면에 격자가 밀집되게 격자 생성을 하여야 한다.

2-2-3) f_{nx} , f_{ny} 비율에 따른 1/4 타원 격자

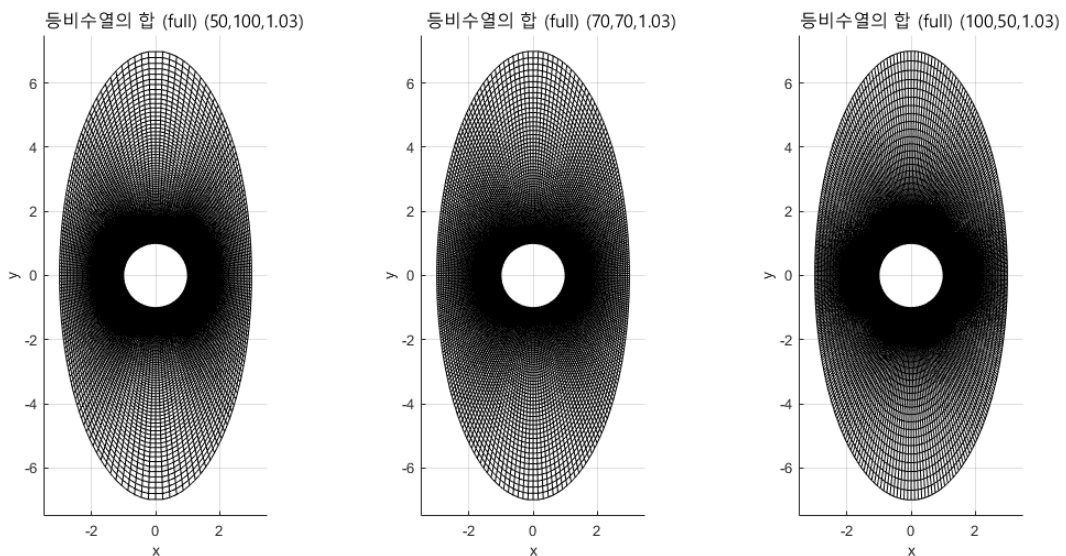
$strch = 1.03$ 으로 고정한 후 값을 f_{nx} 를 각각 50, 70, 100, f_{ny} 를 각각 100, 70, 50으로 설정하여 1/4 타원 격자를 생성하였다. ($f_{nx}:f_{ny}$ 는 각각 1:2, 1:1, 2:1, 격자점의 개수는 각각 5000, 4900, 5000이다.)



<그림 8, 9, 10 – $f_{nx}:f_{ny} = 1:2, 1:1, 2:1$ 일 때의 등비수열의 합을 이용한 1/4 타원형 grid>

2-2-4) f_{nx} , f_{ny} 비율에 따른 full 타원 격자

$strch = 1.03$ 으로 고정한 후 값을 f_{nx} 를 각각 50, 70, 100, f_{ny} 를 각각 100, 70, 50으로 설정하여 full 타원 격자를 생성하였다. ($f_{nx}:f_{ny}$ 는 각각 1:2, 1:1, 2:1, 격자점의 개수는 각각 5000, 4900, 5000이다.)



<그림 11, 12, 13 – $f_{nx}:f_{ny} = 1:2, 1:1, 2:1$ 일 때의 등비수열의 합을 이용한 1/4 타원형 grid>

위 그림 2~13을 보면 *strch*를 변화시키거나 *fnx:fn_y* 비율을 변화시키면 *i*방향과 *j*방향의 격자의 수가 달라져 수가 더 많아지는 방향으로 격자가 밀해진다. 또한, aspect ratio와 skewness가 변화하는 것을 확인할 수 있었다. 양질의 격자를 생성하기 위하여 aspect ratio는 1에 가까워야 하고 skewness는 0에 가까워야 한다. 따라서 먼저 *strch*값을 조절하여 관심있는 영역(ex. 유동의 변화가 큰 영역, 물체 표면 주위)에 격자점이 많이 분포되도록 설정한 뒤 *fnx:fn_y* 비율을 aspect ratio와 skewness를 최대한 1과 0을 가지는 비율로 설정하여 양질의 격자를 생성할 수 있다. 위 그림 8, 9, 10을 보면 *strch* = 1.03의 경우 그림 9의 격자(*fnx:fn_y* = 1:1)가 가장 정사각형에 가깝고 양질의 격자라고 할 수 있다.

2-3) 등비수열의 합을 이용한 격자 - 적응 격자

초음속으로 비행하는 물체 전방에 충격파가 발생하면 충격파 부근의 유동은 급격하게 변화하게 된다. 따라서 위와 같은 일반 격자에서는 물체 표면과 떨어진 충격파(detached shock)가 발생할 경우 충격파 부근의 유동은 정확하게 해석하기 힘들다. 따라서 충격파 부근에서는 격자점들이 다시 밀해지는 적응 격자를 생성해야 한다.

2-3-1) 각 구역을 나누는 경계면 생성

적응 격자는 3가지 구역으로 나뉘서 격자를 생성하여야 한다. 충격파가 발생한 지점에 가까운 *j*-index를 *j_{shock}*이라고 하고, *j_{shock}*의 절반을 *j_{half}*라고 설정한다. 먼저 각 구역을 구분하는 경계(*j_{half}, j_{shock}*)를 먼저 생성하여야 한다. 여기서 *sh*는 충격파가 일어나는 위치를 물체 표면에서 외부 경계면까지의 거리의 배수로 나타낸 것이다. ($0 \leq sh \leq 1$) (0일 경우 물체 표면에, 1일 경우 외부 경계면에 충격파 발생)

$$x(i, j_{half}) = \{x(i, j_{max}) - x(i, 1)\} \times \frac{sh}{2} + x(i + 1)$$

$$y(i, j_{half}) = \{y(i, j_{max}) - y(i, 1)\} \times \frac{sh}{2} + y(i + 1)$$

$$x(i, j_{max}) = \{x(i, j_{max}) - x(i, 1)\} \times sh + x(i + 1)$$

$$y(i, j_{max}) = \{y(i, j_{max}) - y(i, 1)\} \times sh + y(i + 1)$$

위 식을 이용하여 *j_{half}, j_{shock}*에서 경계면의 격자점을 생성하는 <shjgrid.m>함수를 작성하여 사용하였다.

```
function [x,y]=shjgrid(x,y,fnx,fny,sh,shj)
% grid generation - shock boundary
for i=1:1:fnx
    x(i,shj) = (x(i,fny)-x(i,1))*sh + x(i,1); % x grid component of j_shock
    y(i,shj) = (y(i,fny)-y(i,1))*sh + y(i,1); % y grid component of j_shock
end
```

<MATLAB CODE 4 : *j_{half}, j_{shock}*에서 경계면의 격자점 생성 함수 shjgrid.m>

위 경계면으로 3개의 구역으로 나누어 적응격자를 생성할 수 있다. 구역 1은 $j = j_1 \sim j_{half}$ 구간으로 격자의 간격이 늘어나는 구간, 구역 2는 $j = j_{half+1} \sim j_{shock}$ 구간으로 충격파가 생성되는 위치로 가까워질수록 격자의 간격이 줄어드는 구간, 그리고 구역 3은 $j = j_{shock+1} \sim j_{max}$ 구간으로 충격파에서 멀어질수록 다시 격자의 간격이 줄어드는 구간이다.

2-3-2) 구역 1 ($j_1 \sim j_{half}$)

구역 1에서 i, j 에서 $x_1(i, j), y_1(i, j)$ 는 2-1)과 같이 구하면 다음과 같다.

$$\begin{aligned}xleng_1 &= x(i, j_{half}) - x(i, 1), & ylen_1 &= y(i, j_{half}) - y(i, 1) \\coeff1_1 &= 1 - strch^{j-1}, & coeff2_1 &= 1 - strch^{j_{half}-1} \\x_1(i, j) &= x(i, 1) + xleng_1 \times \frac{coeff1_1}{coeff2_1} \\y_1(i, j) &= y(i, 1) + ylen_1 \times \frac{coeff1_1}{coeff2_1}\end{aligned}$$

2-3-3) 구역 2 ($j_{half+1} \sim j_{shock}$)

구역 2에서는 공비 $strch_2 = \frac{1}{strch}$ 를 가지는 등비수열의 합으로 나타내어야 한다. i, j 에서 $x_2(i, j), y_2(i, j)$ 는 다음과 같다.

$$\begin{aligned}xleng_2 &= x(i, j_{shock}) - x(i, j_{half}), & ylen_2 &= y(i, j_{shock}) - y(i, j_{half}) \\coeff1_2 &= 1 - strch_2^{j-j_{half}}, & coeff2_2 &= 1 - strch_2^{j_{shock}-j_{half}} \\x_2(i, j) &= x(i, j_{half}) + xleng_2 \times \frac{coeff1_2}{coeff2_2} \\y_2(i, j) &= y(i, j_{half}) + ylen_2 \times \frac{coeff1_2}{coeff2_2}\end{aligned}$$

2-3-4) 구역 3 ($j_{shock+1} \sim j_{max}$)

구역 3에서는 다시 구역 1과 마찬가지로 공비 $strch$ 를 가지는 등비수열의 합으로 나타낼 수 있다. i, j 에서 $x_3(i, j), y_3(i, j)$ 는 다음과 같다.

$$\begin{aligned}xleng_3 &= x(i, j_{max}) - x(i, j_{shock}), & ylen_3 &= y(i, j_{max}) - y(i, j_{shock}) \\coeff1_3 &= 1 - strch^{j-j_{shock}}, & coeff2_3 &= 1 - strch^{j_{max}-j_{shock}} \\x_3(i, j) &= x(i, j_{shock}) + xleng_3 \times \frac{coeff1_3}{coeff2_3} \\y_3(i, j) &= y(i, j_{shock}) + ylen_3 \times \frac{coeff1_3}{coeff2_3}\end{aligned}$$

위 식을 이용하여 작성된 MATLAB 코드는 다음과 같다.

```
%% CFD HW#1_격자 : 등비수열의 합 이용 (적응격자, shock)
%% data input
clear; close all;
prompt = {'arad (grid x radius)', 'brad (grid y radius)', 'radil (body radius)', 'fnx (number of i grid(imax))', 'fny (number of j grid(jmax))', 'strch (common ratio r)', 'shock position (%)'};
dlgtitle = 'Data'; dims = [1 35];
definput = {'3', '7', '1', '50', '100', '1.03', '50'}; % set default value
inputdata = inputdlg(prompt, dlgtitle, dims, definput); % input data dialogue
arad = str2double(inputdata{1}); % grid x radius
brad = str2double(inputdata{2}); % grid y radius
```



```

radil = str2double(inputdata{3}); % body radius
fnx = str2double(inputdata{4}); % number of i grid(imax)
fny = str2double(inputdata{5}); % number of j grid(jmax)
strch = str2double(inputdata{6}); % common ration of sequence
sh = str2double(inputdata{7})/100; % shock position(0~1)
j_shock = round(fny*sh); % j-index of shock position (rounding)
j_half = round(fny*sh/2); % j-index of grid start to dense (shock/2 position, rounding)

%% grid generation - body surface, outer boundary, shock boundaries
[x,y] = surfacegrid(fnx,radil); % body surface grid generation function
[x,y] = outergrid(x,y,fnx,fny,arad,brad); % outer boundary grid generation function
[x,y] = shjgrid(x,y,fnx,fny,sh,j_shock); % shock boundary grid generation function
[x,y] = shjgrid(x,y,fnx,fny,sh/2,j_half); % j_shock/2 grid generation function

%% grid generation - inner flow region 1 [j = 1 ~ j_shock/2]
for j = 2:1:j_half
    coeff1 = 1 - strch^(j-1);
    coeff2 = 1 - strch^(j_half-1);
    for i = 1:1:fnx-1
        xleng = x(i,j_half) - x(i,1); % x difference of j_shock/2 and body surface
        yleng = y(i,j_half) - y(i,1); % y difference of j_shock/2 and body surface
        x(i,j) = x(i,1) + xleng * coeff1/coeff2; % x gridcomponent of inner flow region 1
        y(i,j) = y(i,1) + yleng * coeff1/coeff2; % y grid component of inner flow region 1
    end
    x(fnx,j) = x(fnx,1); % align x grid component on y-axis
    y(fnx,j) = y(fnx,1) + yleng * coeff1/coeff2; % align y grid component on y-axis
end

%% grid generation - inner flow region 2 [j = j_shock/2+1 ~ j_shock]
strch2 = 1/strch; % strch of region 2 = 1/strch
for j = j_half+1:1:j_shock
    coeff1 = 1 - strch2^(j-j_half);
    coeff2 = 1 - strch2^(j_shock-j_half);
    for i = 1:1:fnx-1
        xleng = x(i,j_shock) - x(i,j_half); % x difference of j_shock/2 and j_shock
        yleng = y(i,j_shock) - y(i,j_half); % y difference of j_shock/2 and j_shock
        x(i,j) = x(i,j_half) + xleng * coeff1/coeff2; % x grid component of inner flow region 2
        y(i,j) = y(i,j_half) + yleng * coeff1/coeff2; % y grid component of inner flow region 2
    end
    x(fnx,j) = x(fnx,1); % align x grid component on y-axis
    y(fnx,j) = y(fnx,j_half) + yleng * coeff1/coeff2; % align y grid component on y-axis
end

%% grid generation - inner flow region 3 [j = j_shock+1 ~ max]
for j = j_shock+1:1:fny
    coeff1 = 1 - strch^(j-j_shock);
    coeff2 = 1 - strch^(fny-j_shock);
    for i = 1:1:fnx-1
        xleng = x(i,fny) - x(i,j_shock); % x difference of j_shock and outer boundary
        yleng = y(i,fny) - y(i,j_shock); % y difference of j_shock and outer boundary
        x(i,j) = x(i,j_shock) + xleng * coeff1/coeff2; % x grid component of inner flow region 3
        y(i,j) = y(i,j_shock) + yleng * coeff1/coeff2; % y grid component of inner flow region 3
    end
    x(fnx,j) = x(fnx,1); % align x grid component on y-axis
    y(fnx,j) = y(fnx,j_shock) + yleng * coeff1/coeff2; % align y grid component on y-axis
end

%% 1/4 oval grid plot
figure(1); hold on; grid on;
for i = 1:1:fnx
    for j = 1:1:fny
        plot(x(i,:),y(i,:), 'k')
        plot(x(:,j),y(:,j), 'k')
    end
end
title('등비수열의 합 (적응)', 'FontSize',12, 'FontName', '맑은 고딕')
xlabel('x'); ylabel('y'); axis equal; grid on;

```

```

axis([-arad-0.5 0.5 -0.5 brad+0.5])
set(gcf(1), 'OuterPosition', [1400, 300, 400, 700])

%% full oval grid plot
figure(2); hold on; grid on;
for i = 1:1:fnx
    for j = 1:1:fny
        plot(x(i,:),y(i,:), 'k'); plot(-x(i,:),y(i,:), 'k');
        plot(x(i,:),-y(i,:), 'k'); plot(-x(i,:),-y(i,:), 'k');
        plot(x(:,j),y(:,j), 'k'); plot(-x(:,j),y(:,j), 'k');
        plot(x(:,j),-y(:,j), 'k'); plot(-x(:,j),-y(:,j), 'k');
    end
end
title('등비수열의 합, (적응, full)', 'FontSize',12, 'FontName', '맑은 고딕')
xlabel('x'); ylabel('y'); axis equal; grid on;
axis([-arad-0.5 arad+0.5 -brad-0.5 brad+0.5])
set(gcf(2), 'OuterPosition', [1400, 300, 400, 700])

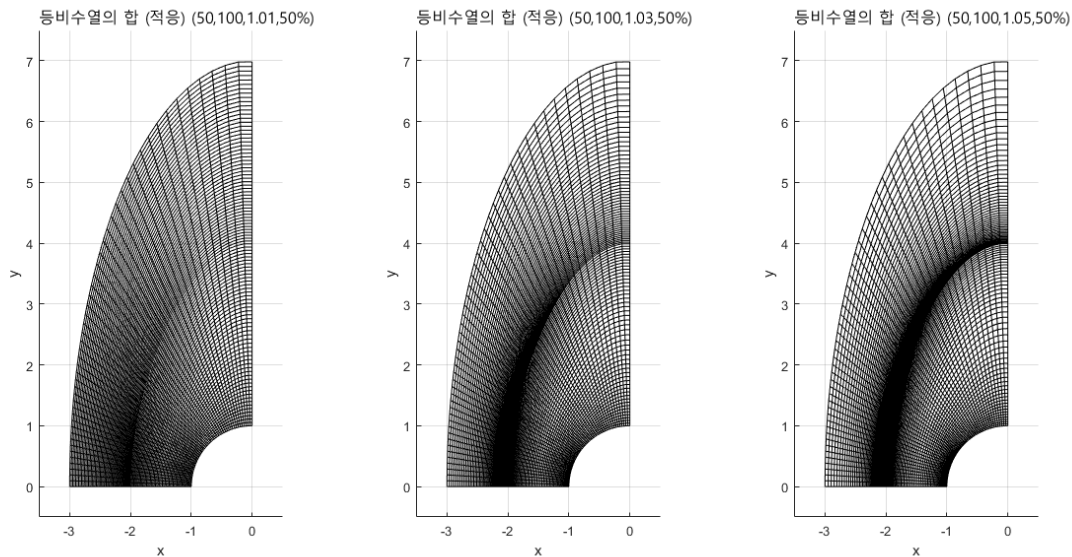
```

<MATLAB CODE 5 : 등비수열의 합을 이용한 적응 격자 생성>

2-4) 격자 Plot 결과

2-4-1) 등비수열의 공비 *strch*값에 따른 1/4 타원 적응 격자

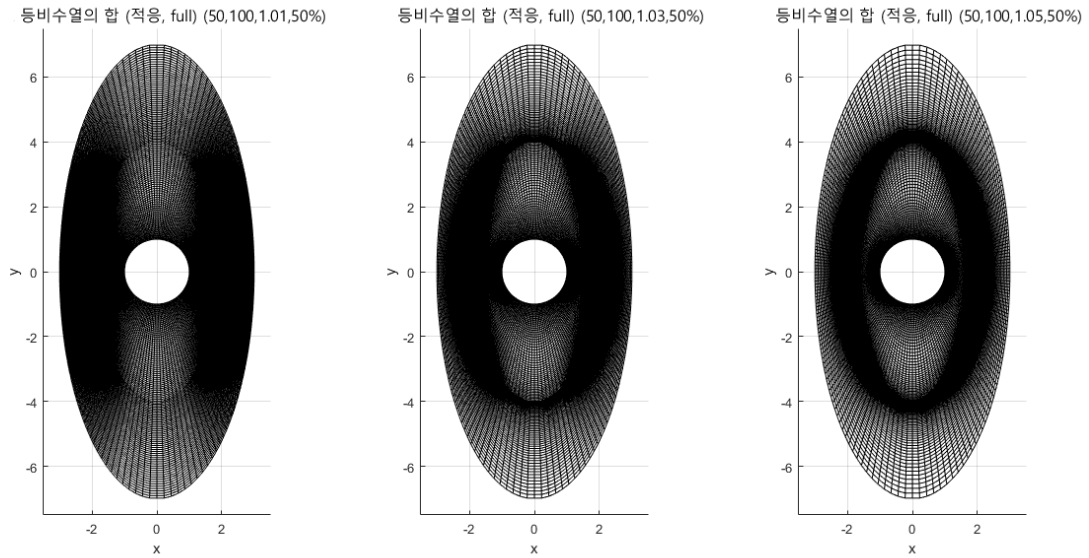
$fnx = 50$, $fny = 100$ 으로 고정한 후 *strch*값을 1.01, 1.03, 1.05로 증가시켜 1/4 타원형 적응 격자를 생성하였다. 충격파 위치는 물체 표면과 외부 경계면의 50% 지점으로 설정하였다.



<그림 14, 15, 16 – $fnx = 50$, $fny = 100$, $strch = 1.01, 1.03, 1.05$, $sh = 50$ 일 때의 등비수열의 합을 이용한 1/4 타원형 적응 grid>

2-4-1) 등비수열의 공비 *strch*값에 따른 1/4 타원 적응 격자

$fnx = 50$, $fny = 100$ 으로 고정한 후 *strch*값을 1.01, 1.03, 1.05로 증가시켜 full 타원형 적응 격자를 생성하였다. 충격파 위치는 물체 표면과 외부 경계면의 50% 지점으로 설정하였다.

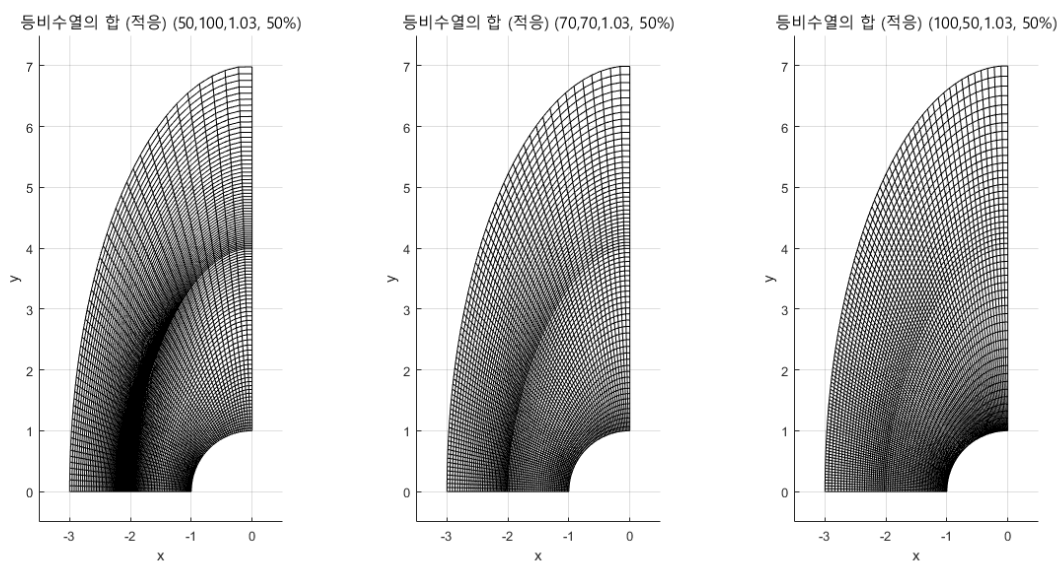


<그림 17, 18, 19 – $fnx = 50$, $fny = 100$, $strch = 1.01, 1.03, 1.05$, $sh = 50$ 일 때의 등비수열의 합을 이용한 full 타원형 적응 grid>

일반 격자에서의 결과(그림 2~7)와 같이 적응형 격자에서도(그림 14~19) 같은 fnx , fny 값에서 등비수열의 공비 $strch$ 값이 증가하면 격자가 물체 표면에 더 집중적으로 분포하는 것을 확인할 수 있다. 또한 충격파 위치에서도 $strch$ 값이 증가할수록 충격파에 더 집중적으로 격자가 분포한다 따라서 물체 표면 주위, 충격파 주위의 유동을 정확하게 확인하기 위해서 $strch$ 값을 적당히 증가시켜 물체 표면과 충격파 생성 지점에 격자가 밀집되게 격자 생성을 하여야 한다.

2-4-3) fnx , fny 비율에 따른 1/4 타원 격자

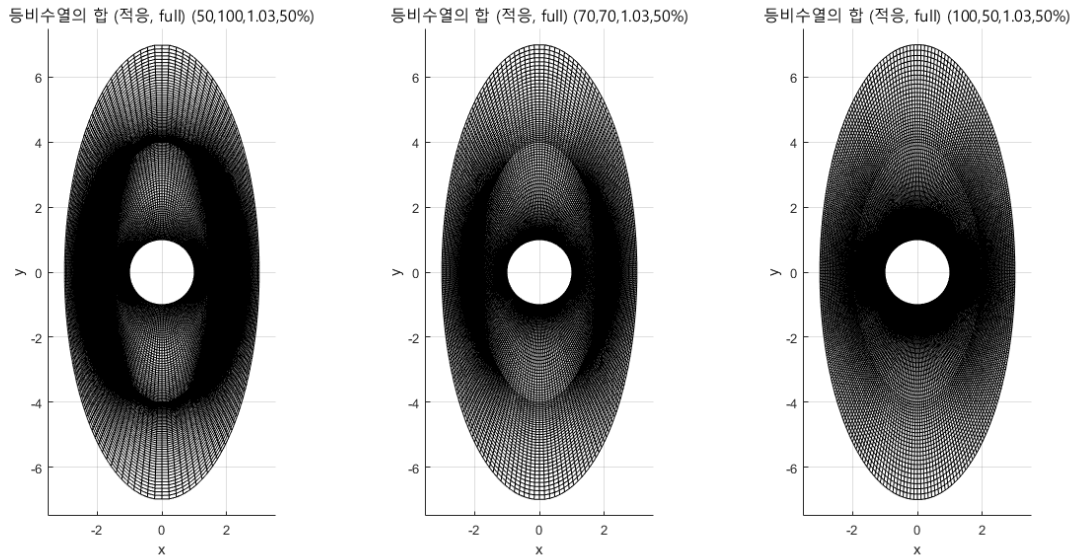
$strch = 1.03$ 으로 고정한 후 값을 fnx 를 각각 50, 70, 100, fny 를 각각 100, 70, 50으로 설정하여 1/4 타원 격자를 생성하였다. 충격파 위치는 물체 표면과 외부 경계면의 50% 지점으로 설정하였다. ($fnx:fny$ 는 각각 1:2, 1:1, 2:1, 격자점의 개수는 각각 5000, 4900, 5000이다.)



<그림 20, 21, 22 – $fnx:fny = 1:2, 1:1, 2:1$ 일 때의 등비수열의 합을 이용한 1/4 타원형 적응 grid>

2-4-4) fnx , fny 비율에 따른 full 타원 격자

$strch = 1.03$ 으로 고정한 후 값을 fnx 를 각각 50, 70, 100, fny 를 각각 100, 70, 50으로 설정하여 full 타원 격자를 생성하였다. 충격파 위치는 물체 표면과 외부 경계면의 50% 지점으로 설정하였다. ($fnx:fny$ 는 각각 1:2, 1:1, 2:1, 격자점의 개수는 각각 5000, 4900, 5000이다.)



<그림 23, 24, 25 - $fnx:fny = 1:2, 1:1, 2:1$ 일 때의 등비수열의 합을 이용한 full 타원형 적응 grid>

그림 8~13과 그림 20~25를 비교해보면 적응형 격자가 일반 격자보다 $fnx:fny$ 비율이 물체 표면 또는 충격파 주위 격자의 밀집도에 미치는 영향을 더 확실하게 확인할 수 있다. 또한, 적응형 격자에서도 일반 격자에서 확인 가능한 aspect ratio와 skewness의 변화도 확인할 수 있었다. 위 결과를 보면 $fnx:fny = 1:2$ 에서 가장 aspect ratio와 skewness 특성이 좋지 않았지만 물체 표면과 충격파 주변에 격자가 더 많이 생성된 것을 확인할 수 있었다. 위 결과를 조합해보면 그림 14~25 중 그림 16($fnx = 50$, $fny = 100$, $strch = 1.05$, $sh = 50$)이 가장 적합한 격자라는 것을 확인할 수 있었다.